SOFTLY SEMI-SUPERVISED LEARNING FOR BIOINFORMATICS APPLICATIONS

by

Melis Özgür Çetinkaya Demir

B.S., Computer Engineering, Ege University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2014

SOFTLY SEMI-SUPERVISED LEARNING FOR BIOINFORMATICS APPLICATIONS

APPROVED BY:

Assist. Prof. Arzucan Özgür (Thesis Supervisor)	.....
Prof. Ethem Alpaydın	.....
Assoc. Prof. Olcay Taner Yıldız	.....

DATE OF APPROVAL: 04.02.2014

ACKNOWLEDGEMENTS

I would like to thank Assist. Prof. Arzucan Ozgur and Prof. Ethem Alpaydın for showing me the directions and motivating to follow this study. I also thank them for being such a nice persons.

I thank to Assoc. Prof. Olcay Taner Yildiz for participating in my thesis jury and giving me feedbacks.

I thank to my lovely husband Emre Demir for his consistent patient, help and understanding, delicious meals and love.

Finally, I want to express my gratefulness to my mother Zumrut Binici Cetinkaya and my father Ekrem Cetinkaya for their complimentary love, encouragement, and self-sacrifice and believe in me. I am also thankful to my siblings Ugur Cetinkaya, Yesim Cetinkaya, Aysin Tonaz and Zeynep Cetinkaya Edizer for never leaving me alone or lonely.

I would like to devote this study to the youth of my mother Zumrut and father Ekrem.

ABSTRACT

SOFTLY SEMI-SUPERVISED LEARNING FOR BIOINFORMATICS APPLICATIONS

Binary classification of biological data is an important research problem both in the Bioinformatics and Machine Learning fields. This problem is particularly challenging when the number of labeled instances is very few. There are three main machine learning approaches for classification: supervised methods, which only use labeled data, unsupervised methods, which only use unlabeled data, and semi-supervised methods, which use both labeled and unlabeled data. In this study, we compare the supervised and various developed semi-supervised methods which are based on k-NN (k Nearest Neighbor), SVM (Support Vector Machine) with linear kernel, and SVM with RBF (Radial Basis Function) kernel for two different Bioinformatics problems: predicting reccurrence in colorectal cancer from microarray data and predicting HIV-1-Human protein-protein interactions. As distinct from traditional semi-supervised learning approaches, we introduce the definition of 'softly labeled' data that defines unlabeled data with additional information about their highly expected labels. We also evaluate our algorithms on a well-known optical digit dataset to classify the numbers '5' and '6' by generating synthetic noise and use as softly labeled data to better understand the behaviors of our algorithms. For all datasets, we concluded that softly labeled data are informative and enhances the evaluation results. Our semi-supervised methods SS-kNN (Semi-supervised kNN) and SS-SVM (Semi-supervised SVM) perform better than other algorithms in terms of accuracy for colorectal cancer and optical digit data, and area under the precision-recall curve for HIV-1-human protein-protein interaction data. Furthermore, in general, our semi-supervised methods achieve better performances than the supervised ones.

ÖZET

BİYOENFORMATİK UYGULAMALARI İÇİN KESİN ETİKETLENMEMİŞ VERİLER İLE YARI-GÖZETİMLİ ÖĞRENME

Biyolojik verilerin ikili sınıflandırması hem Biyoenformatik hem de Makine Öğrenmesi alanları için önemli bir araştırma problemidir. Özellikle etiketlenmiş verilerin sayısı çok azsa, bu problem daha da zorlaşmaktadır. İkili sınıflandırma için kullanılan üç ana makine öğrenmesi yontemi bulunmaktadır: sadece etiketlenmiş verileri kullanan gözetimli öğrenme, sadece etiketlenmemiş verileri kullanan gözetimsiz öğrenme ve hem etiketlenmiş hem de etiketlenmemiş verileri kullanan yarı-gözetimli öğrenme. Bu çalışmada, k-NN (En Yakın k Komşu), Liner ve RBF (Radyal Temelli Fonksiyon) çekirdek fonksiyonları ile SVM (Destek Vektör Makinesi) algoritmalarını temel alan gözetimli öğrenme ve bizim geliştirdiğimiz çeşitli yarı-gözetimli öğrenme algoritmaları, iki farkli biyoenformatik verisi olan insan HIV-1 virüsü protein-protein etkileşimlerini öngörmek ve kolon kanseri tekrarlamasını öngörmek için karşılaştırılmıştır. Geleneksel yarı-gözetimli öğrenme yaklaşımlarından farklı olarak, belirli bir sınıfa ait olduğuna dair bulgular bulunan etiketlenmemiş verileri anlatan 'kesin olarak etiketlenmemiş' veri kavramı ortaya atılmıştır. Geliştirdiğimiz algoritmaların davranışsal özelliklerini gözlemlemek adına bilindik bir optik sayı verisi üzerinde '5' ve '6' sayıları sınıflandırılmıştır. Bu veri kümesinde 'kesin etiketlenmemiş veri' oluşturmak için etiketlenmiş verilere yapay gürültü eklenmiştir. Tüm veri kümelerinde kesin etiketlenmemiş veri kullanımının performans göstergelerini arttırdığı görülmüştür. Bu çalışmada geliştirilen yarı-gözetimli yöntemlerden SS-kNN (Yarı-gözetimli k-NN)'nin ve SS-SVM (Yarı-gözetimli SVM)'in diğer algoritmalara göre kolon kanseri verisi ve optik sayılar verisi için doğruluk ve insan HIV-1 protein-protein etkileşimi verisi için kesinlik-geri getirme eğrisinin altında kalan alan açısından daha iyi sonuç verdiği gözlemlenmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES ix
LIST OF SYMBOLS xiii
LIST OF ACRONYMS/ABBREVIATIONS xiv
1. INTRODUCTION 1
1.1. Colorectal Cancer Reccurence Prediction 2
1.2. Human and HIV-1 Protein Protein Interaction Prediction 4
1.3. Contributions 5
1.4. Thesis Organization 6
2. PROPERTIES OF THE DATASETS 7
2.1. Colorectal Cancer Dataset 7
2.2. HIV-1 and Human PPI Dataset 8
2.3. Optical Digit Dataset 10
3. METHODOLOGY 13
3.1. k Nearest Neighbor Based Algorithms 13
3.1.1. Semi-Supervised k-NN Algorithm (SS-kNN) 14
3.1.2. Semi-Supervised Self Trained k-NN Algorithm (ST-kNN) 15
3.1.3. Semi-Supervised Instance Weighted k-NN Algorithm (IW-kNN) 16
3.2. Support Vector Machine Based Algorithms 18
3.2.1. Semi Supervised SVM Algorithm (SS-SVM) 21
3.2.2. Semi-Supervised Self Trained SVM Algorithm (ST-SVM) 22
3.2.3. Semi-Supervised Instance Weighted SVM Algorithm (IW-SVM) 23
4. EXPERIMENTS AND RESULTS 26
4.1. Evaluation Metrics 26
4.2. Results for the Colorectal Cancer Data Set 30
4.3. Results for the HIV-1 and Human PPI Data Set 34

4.4. Results for the Optical Digit Data Set 36
5. CONCLUSION 41
APPENDIX A: Colorectal Cancer Data Results 45
APPENDIX B: HIV-1 and Human PPI Data Results 54
APPENDIX C: Optical Digit Data Results 59
REFERENCES 94

LIST OF FIGURES

Figure 1.1. Stages of colorectal cancer [1]. 3

Figure 1.2. HIV-1 and human protein-protein interactions [2]. 5

Figure 2.1. A set of handwritten digit preprinted forms of MNIST database [3]. 11

Figure 3.1. Training in the IW-kNN algorithm. 19

Figure 3.2. Testing in the IW-kNN algorithm. 20

Figure 4.1. 5 fold cross-validation accuracy for colorectal cancer recurrence data set. 30

Figure 4.2. PR-AUC on validation set for HIV-1 and human PPI data set. . . 34

Figure 4.3. Break even point calues on the validation set for HIV-1 and human PPI data set. 35

Figure 4.4. ROC-AUC on the validation set for HIV-1 and human PPI data set. 36

Figure 4.5. Optical digit: accuracy of k-NN based algorithms on the test set for 10 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft. . 37

Figure 4.6. Optical digit: accuracy of SVM based algorithms with linear kernel on the test set for 10 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft. 38

Figure 4.7. Optical digit: accuracy of SVM based algorithms with RBF kernel on the test set for 10 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft. 38

Figure 4.8. Optical digit: accuracy of k-NN based algorithms on the test set for 30 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft. . 39

Figure 4.9. Optical digit: accuracy of SVM based algorithms with linear kernel on the test set for 30 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft. 39

Figure 4.10. Optical digit: accuracy of SVM based algorithms with RBF kernel on the test set for 30 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft. 40

LIST OF TABLES

Table 2.1. CRC datasets (+:'recurrence', -:'non-recurrence'). 7
Table 2.2. HIV-1 and human protein interaction data. 9
Table 2.3. MNIST handwritten digits database. 11
Table 4.1. Confusion matrix. 27
Table 4.2. Colorectal cancer: accuracy values on test data with various parameters of all algorithms. 32
Table 4.3. Accuracy in the colorectal cancer data set and the parameters learned with cross-validation for the semi-supervised methods. 32
Table 4.4. ROC-AUC and break even values on colorectal cancer test set. . . 33
Table A.1. Colorectal cancer: accuracy values on test data with various parameters of all algorithms. 45
Table A.2. Colorectal cancer: accuracy values on validation data with various parameters of all algorithms. 46
Table A.3. Colocteral cancer: k-NN results on 23 test instances with various decision thresholds. 47
Table A.4. Colocteral cancer: SVM with linear kernel results on 23 test instances with various decision thresholds. 48
Table A.5. Colocteral cancer: SVM with RBF kernel results on 23 test instances with various decision thresholds. 49
Table A.6. Colocteral cancer: k-NN results on validation set. 50
Table A.7. Colocteral cancer: SVM with linear kernel results on validation set. 51
Table A.8. Colocteral cancer: SVM with RBF kernel results on validation set. 52
Table A.9. The parameters of colorectal cancer k-NN results. 52
Table A.10. The parameters of colorectal cancer SVM with linear kernel results. 53
Table A.11. The parameters of colorectal cancer SVM with RBF kernel results. 53
Table B.1. HIV-1-human PPI: PR-AUC values on test data with various parameters of all algorithms. 54

Table B.2. HIV-1-human PPI: k-NN results on test set with various decision thresholds. 55
Table B.3. HIV-1-human PPI: SVM with linear kernel results on test set with various decision thresholds. 56
Table B.4. HIV-1-human PPI: SVM with RBF kernel results on test set with various decision thresholds. 57
Table B.5. The parameters of HIV-1-human PPI results. 58
Table C.1. The parameters of optical digit results on test set: 10 hard no soft. 59
Table C.2. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled. 59
Table C.3. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled. 60
Table C.4. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled. 61
Table C.5. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled. 62
Table C.6. Optical digit results on test set: 10 hard no soft. 63
Table C.7. Optical digit: k-NN results on test set: 10 hard 24 soft. 64
Table C.8. Optical digit: SVM with linear kernel results on test set: 10 hard 24 soft. 65
Table C.9. Optical digit: SVM with RBF kernel results on test set: 10 hard 24 soft. 66
Table C.10. The parameters of optical digit results: 10 hard 24 soft. 67
Table C.11. Optical digit: k-NN results on test set: 10 hard 50 soft. 68
Table C.12. Optical digit: SVM with linear kernel results on test Set: 10 hard 50 soft. 69
Table C.13. Optical digit: SVM with RBF kernel results on test set: 10 hard 50 soft. 70
Table C.14. The parameters of optical digit results: 10 hard 50 soft. 71
Table C.15. Optical digit: k-NN results on test set: 10 hard 100 soft. 72

Table C.16. Optical digit: SVM with linear kernel tesults on test set: 10 hard 100 soft. 73
Table C.17. Optical digit: SVM with RBF kernel results on test set: 10 hard 100 soft. 74
Table C.18. The parameters of optical digit results: 10 hard 100 soft. 75
Table C.19. Optical digit: accuracy values on test data with various parameters of all algorithms for 30 hardly labeled. 76
Table C.20. Optical digit: accuracy values on test data with various parameters of all algorithms for 30 hardly labeled. 77
Table C.21. Optical digit: accuracy values on test data with various parameters of all hlgorithms for 30 hardly labeled. 78
Table C.22. Optical digit: accuracy values on test data with various parameters of all algorithms for 30 hardly labeled. 79
Table C.23. Optical digit results on test set: 30 hard no soft. 80
Table C.24. The parameters of optical digit results on test set: 30 hard no soft. 81
Table C.25. Optical digit: k-NN results on test set: 30 hard 24 soft. 82
Table C.26. Optical digit: SVM with linear kernel results on test set: 30 hard 24 soft. 83
Table C.27. Optical digit: SVM with RBF kernel results on test set: 30 hard 24 soft. 84
Table C.28. The parameters of optical digit results: 30 hard 24 soft. 85
Table C.29. Optical digit: k-NN results on test Set: 30 hard 50 soft. 86
Table C.30. Optical digit: SVM with linear kernel results on test set: 30 hard 50 soft. 87
Table C.31. Optical digit: SVM with RBF kernel results on test set: 30 hard 50 soft. 88
Table C.32. The parameters of optical digit results: 30 hard 50 soft. 89
Table C.33. Optical digit: k-NN results on test set: 30 hard 100 soft. 90
Table C.34. Optical digit: SVM with linear kernel results on test set: 30 hard 100 soft. 91

Table C.35. Optical digit: SVM with RBF kernel results on test set: 30 hard 100 soft. 92
Table C.36. The parameters of optical digit results: 30 hard 100 soft. 93

LIST OF SYMBOLS

C	The penalty factor for the slack variable ϵ
d	A feature value
e_i	The expected label for the ith instance
$f(x_i)$	The prediction result for the ith instance
k	Number of nearest neighbors
L	The set of labeled instances
$L(x)$	The loss function
l_i	The label of the ith instance in the set of L
$NN_{i,k}$	The coefficients vector of k nearest neighbors of the ith datum
p_{value}	The post probability value for a prediction
S	The set of softly labeled instances
SV	The vector of support vectors
U	The set of unlabeled instances
T	The set of test instances
w	The normal vector of the hyperplane in the primal form of SVM
X	The feature vector of all instances
x_i	The feature vector of the ith instance
y_i	The label of the ith instance
μ	Similarity value
θ	The reliability coefficient for k-NN based algorithms and coefficient to the parameter C for SVM based algorithms
ϕ	The feature map
ϵ_i	The slack variable ϵ of the ith instance

LIST OF ACRONYMS/ABBREVIATIONS

AUC	Area Under the Curve
BEP	Break Even Point
CRC	Colorectal Cancer
FN	False Negative
FP	False Positive
HIV-1	Human Immunodeficiency Virus
k-NN	k Nearest Neighbor
IW-kNN	Semi-Supervised Instance Weighted k Nearest Neighbor
IW-SVM	Semi-Supervised Instance Weighted Support Vector Machine
MAP	Mean Average Precision
PPI	Protein-Protein Interaction
PR-AUC	Area Under the Precision-Recall Curve (aka. MAP)
PR Curve	Precision-Recall Curve
RBF	Radial Basis Function
ROC	Receiver Operating Curve
ROC-AUC	Area Under the Receiver Operating Curve
SS-kNN	Semi-Supervised k Nearest Neighbor
SS-SVM	Semi-Supervised Support Vector Machine
ST-kNN	Semi-Supervised k Nearest Neighbor
ST-SVM	Semi-Supervised Self-Trained Support Vector Machine
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
WHO	World Health Organization

1. INTRODUCTION

Interpreting biological data is a very crucial and rapidly expanding need in the field of molecular biology. Biological data are often complex structures, which may include many known or unknown correlations. Moreover, most of the time biological data are noisy and missing. Experts may use their know-how to interpret biological data. Unfortunately, it is often difficult to see many direct or indirect correlations between data for a human being. Computational techniques such as Machine Learning methods provide solutions to these challenges.

Supervised learning based classifiers learn only from labeled examples. However, most of the time it is expensive and time-consuming to obtain labeled data, especially in the field of molecular biology, since it needs annotation of subject experts. On the other hand, gathering unlabeled data may be easier than labeled ones. So another learning approach, semi-supervised learning that uses both labeled and unlabeled data, comes into play to create better classifiers. However, it is important to note that, it is not always possible to get better results by using unlabeled data. For example, the study in [4] showed that training a Hidden Markov Model with unlabeled data decrease the accuracy for certain initial conditions.

Various semi-supervised approaches exist including self-training, co-training, transductive learning and graph-based methods. Early works on semi-supervised learning methods focused on Generative Models that are based on the assumption that each class has an identifiable mixture distribution such as a normal distribution for binary classification problems [5]. Self-training is another method which labels each unlabeled instance with the help of labeled instances and adds the most confident labeling to the training set. Although these two methods are not recent, they are still in use because of the simplicities of their structures [5]. The third method is co-training, which separates the features into two subsets such that these two sets are conditionally independent. Two classifiers are trained with the instances of the two-feature subsets initially. After that, the classifiers label unlabeled data separately and teache each other their predic-

tions iteratively. Another semi-supervised approach is transductive methods, which are reasoning from training set to test set by minimizing the exptected loss for that particular test set. Nowadays, graph-based semi supervised methods are widely used by researchers. In graph-based solutions, nodes represent labeled and unlabeled instances, while edges represent various similarity relations among them [5].

In this study, we take the semi-supervised concept a step further by introducing softly labeled data, which which can be defined as unlabeled data incorporated with additional information about their highly expected labels. k-NN (k Nearest Neighbor) based and SVM (Support Vector Machine) based supervised methods and various developed semi-supervised methods are applied on two bioinformatics problems: predicting colorectal cancer (CRC) recurrence within five years and predicting new HIV-1-Human Protein-Protein Interactions (PPIs).

1.1. Colorectal Cancer Reccurence Prediction

Cancer is a disease, which is caused by unregulated cell growth and proliferation. According to the Globocan 2012 report [6] by World Health Organization (WHO), cancer causes 8.2 million of deaths worldwide in 2012. The most cancer deaths happen because of lung, stomach, liver, colon and breast cancer. 19.3 million new cancer cases are expected per year by 2025 with these increasing counts. Colorectal cancer occurs as a tumor in colon, rectum or appendix. According to the same report 694,000 people died among 1.4 million cases in 2012 because of colorectal cancer.

For cancer diagnosis, gene expression profiling has been used to help classifying a tumor and form more accurate clinical outcome. The prognosis is very vital prediction when a tumor is discovered, especially for selecting a proper treatment. For example, if a colorectal cancer patient with high risk of recurrence within five years could be accurately classified, chemotherapy would be applied to this patient to cure, prolong life or palliate the symptoms. On the other hand, a right classification can prevent toxic chemotherapy treatment for a patient with low risk of five years recurrence.

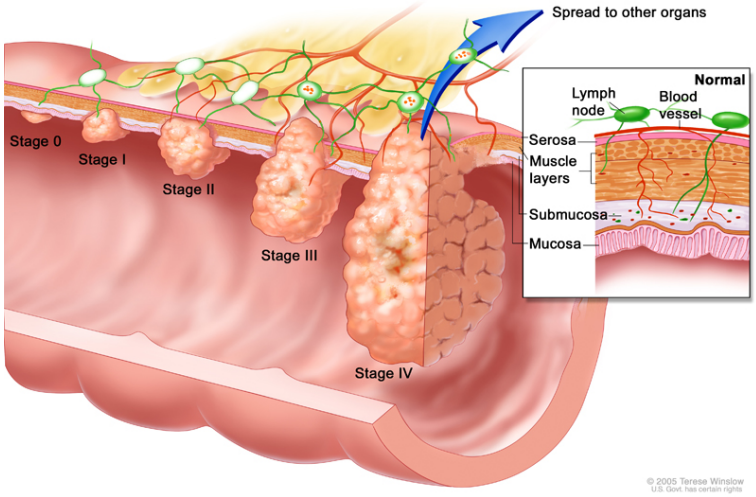

Figure 1.1. Stages of colorectal cancer [1].

There are several studies in the field of microarray-based gene expression profiling [7]. Various well-known supervised approaches are used for predicting different cancer types [8]. However, these algorithms suffer from insufficient number of samples in microarray-based cancer profiling because of high expenses and time consumption. Even if there exist microarray based profiling experiments, the availability of follow-up data for outcome prediction is very low. Supervised approaches use only labeled instances by discarding unlabeled instances, which increases misclassifications. On the other hand, semi-supervised approaches which, use both labeled and unlabeled data, have been applied on various biological problems such as protein classification [9] and, gene expression based cancer subtype discovery [10]. However to the best of our knowledge, there are no any prior studies on semi-supervised approaches for colorectal cancer recurrence prediction except the work in [7], which proposes using a semi-supervised low density separation (LDS) method [11] for gene-based prediction of recurrence in colorectal cancer.

Figure 1.1 illustrates the colorectal cancer stages from 0 to IV. While the stage number is increasing, the colorectal cancer is getting worse. Surgery can be applied on stage I, II and III patients, since the colorectal cancer spreads to other organs in stage IV patients. It is proved that 93.2% of stage I, 77% of stage II, 47.7% of stage III and 6.6% of stage IV patients survive within five years [12]. These numbers emphasize the importance to find the accurate individual therapy by making more

accurate predictions. To provide more accurate predictions, microarray data of patients that have no follow-up information can be utilized as unlabeled data in semi-supervised context. Moreover, instead of using unlabeled data like it has no information on it, we use the cancer stages information from I to III by discarding stage IV. Notably, we expect stage I unlabeled instances to be non-recurrence within 5 years. On the other hand, stage III instances are highly expected to be recurrence within 5 years. Hence, we called the unlabeled data with stages information as 'softly labeled' data. By considering different label expectations on stages, we propose three different semi-supervised algorithms which are built on k-NN and SVM approaches by introducing softly labeled data.

1.2. Human and HIV-1 Protein Protein Interaction Prediction

The Human Immunodeficiency Virus (HIV) makes the immune system down by destroying the immune system cells' functions. In 2012, approximately 35.2 million people were HIV-infected and 6300 people died from AIDS as the HIV/AIDS No 360 fact sheet states [13]. Although there is no cure for HIV infection, antiretroviral drugs can help to control the HIV virus. Many of the critical biological events (e.g. cell cycle control, apoptosis, and metabolic and signaling pathways) occur as a result of protein-protein interactions (PPIs) such as binding or unbinding of a protein to/from another protein [14]. Besides intra-species interactions (e.g. interactions among human proteins), inter-species interactions can occur as well. Interactions among pathogen and human proteins play crucial roles in several diseases including AIDS. Therefore, it is necessary to delve into PPIs between human and HIV-1 to develop drugs for AIDS.

In order to identify PPIs, many experimental methods have been proposed in the literature [15]. However, the produced data of PPI identifications are incomplete and inconsistent. Also, the process is very time consuming and expensive [16]. Therefore, various computational techniques [16] have been proposed for predicting PPIs as complementary to wet lab experiments, thereby narrowing the target space or validating experiment results. Many of the computational techniques focus on intra-species predictions and use only labeled data. Some of these studies are based on random forest

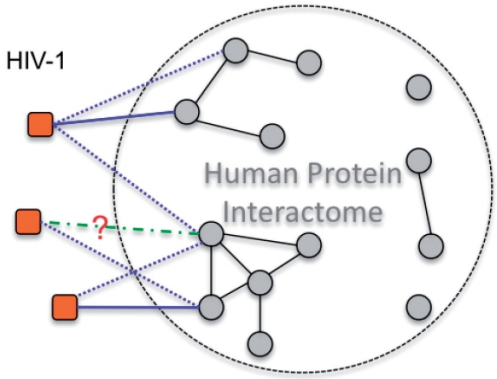

Figure 1.2. HIV-1 and human protein-protein interactions [2].

similarity [17], Bayesian Networks [18], genomic, and proteomic data integration [19] and support vector machines [20].

A random forest based supervised classifier for predicting inter-species PPIs between human and HIV-1 proteins was proposed in [21]. However, supervised learning classifiers need sufficient fully labeled training data to get an acceptable accuracy. Unfortunately, many of the inter-species PPI datasets do not have sufficient positive (interact) and negative (not interact) labeled data as in the example of human and HIV-1 interactions. Recently, various semi-supervised methods are proposed to overcome the insufficient data problem. A Multi-Layer perceptron network based semi-supervised classifier was proposed in [2]. The data used was literature curated from NIAID database [21]. It is proposed that literature-curated PPIs can be erroneous [22]. Therefore, Qi et al. call this data as 'partially labeled' and use as auxiliary training data for labeled data which are expert annotated. In Figure 1.2 solid blue edges indicate expert annotated labeled interactions, dashed blue edges express NIAID curated softly labeled interactions and dashed green edge shows the to be labeled interaction. In this paper, we will use the 'softly labeled' term as synonym to 'partially labeled'. We use the same human HIV-1 PPI dataset that was used in [2].

1.3. Contributions

In this study, we introduce the notion of 'softly labeled' data in Bioinformatics context. We propose three semi-supervised classifiers by introducing softly labeled data

for both k Nearest Neighbor (k-NN) and Support Vector Machines (SVM) approaches. The algorithms that we propose can be summarized as follows;

- k-NN based semi-supervised classifier (aka. SS-kNN) adds the whole softly labeled data into the training phase with different reliability values θ between (0,1].
- Self Trained k-NN (aka. ST-kNN) uses self training to label the softly labeled data and adds the instances, which are over a certain confidence threshold, into the training set.
- Instance weighted k-NN (aka. IW-KNN) gives random weights to each labeled training instances. First, it tries to arrange the weights with a loss function by labeling the labeled instances as in self-training. Then, it uses the softly labeled ones to arrange the weights while adding them into the training set in a self-training manner.
- SS-SVM takes softly labeled data into consideration by giving different reliability coefficient values θ to the parameter C in the SVM objective function. Thus, we define different penalty factors for each instance.
- ST-SVM combines SVM and self training together as in the ST-kNN algorithm.
- IW-SVM is similar to IW-kNN and gives random weights for each instance for the SVM parameter C. We also propose a majority voting based approach to give new weights to softly labeled data.

  We apply the proposed algorithms on the colorectal cancer, and HIV-1-Human PPI datasets. In addition, we use the optical digit dataset by creating synthetic softly labeled data by applying various proportions of noise on the labeled instances to better understand the behaviors of our algorithms.

1.4. Thesis Organization

The rest of the thesis is organized as follows: we begin by giving a background of the datasets that we used in Section 2. Section 3 gives an overview of the methods. Section 4 gives the experiments, and Section 5 presents the conclusions.

2. PROPERTIES OF THE DATASETS

2.1. Colorectal Cancer Dataset

We gather the colorectal cancer (CRC) gene expression dataset from GEO Database [23] which includes 66 labeled and 72 unlabeled instances from the GSE17536 dataset and 23 labeled data from the GSE14333 dataset. The labels indicates 'recurrence' or 'non-recurrence' of colorectal cancer within 5 years of follow-up. GSE17536 dataset is used for training purposes, while GSE14333 is used for tests. Colorectal Cancer is numerated from stage I to IV according to the spreading of cancer anatomically and it goes worse from stage I to IV. Stage IV instances were removed from both of the datasets because this level of cancer could not be cured by surgery. Therefore, it cannot be a case to search for recurrence in 5 years. We use the stage I, II and III information on unlabeled data in a semi-supervised manner and named them as 'softly labeled'.

Table 2.1. CRC datasets (+:'recurrence', -:'non-recurrence').

		Number of	
		Labeled Data	**Unlabeled Data**
GEO ID	**GSE17536**	66 (30+,36-)	72
	GSE14333	23 (12+,11-)	-

Various pre-processing operations were applied on these gene expression datasets [7]. First, the two CRC datasets were generated by using the Affymetrix U133 plus2.0 array separately to keep the validation dataset independent.Principle of separating the data that build a classifier and the data that validate it is introduced in [8]. Second, the Cel files of both datasets are processed using MAS5 and then quantile normalization was applied on them. The probe sets are ranked based on the average gene expression value. The lowest %30 value probe sets were eliminated.The probe sets correspond to

the features in our data set. After the elimination step, 26790 features remain. At the end, Z-score transformation was applied to make the expression levels comparable across genes and the features that hold the condition $p < 0.05$ were selected. The pre-processed form of the CRC data is received from [7] directly. We applied 5-fold stratified cross-validation for the CRC17536 dataset to optimize the parameters of the algorithms. We use the CRC14333 dataset for test purposes.

Literally, each instance of the dataset $X = x_1, x_2, \ldots, x_n$ represents a patient with colorectal cancer. An instance x_i has 26790 dimensional feature vector $x_i, = (x_i^1, x_i^2, \ldots, x_i^{26790})^T$. The binary classification problem is defined as the set of X_i, y_i pairs where the training set $L \in X$ consists from the data in GSE17536, the validation set $S \in X$ consists from the data in GSE17536, the test set $V \in X$ consists from the data in GSE14333, and $y \in \{1(recurrence), 0(non\text{-}recurrence)\}$ within 5 years. This dataset contains positive, negative and softly labeled data for the subset L and positive and negative labeled data for the subsets S and V. Softly labeled data have stages $st \in \{1, 2, 3)\}$.

2.2. HIV-1 and Human PPI Dataset

The HIV-1-Human protein protein interaction dataset is obtained from the literature-curated NIAID database [24]. Unfortunately, literature curated interactions tend to be incorrect [22]. Therefore, Qi et al. name these manually curated interactions as 'partially labeled' (aka. 'softly labeled'). Additionally, truly positive interactions (labeled) were generated by conferring 16 HIV-1 experts and letting them to annotate the known interactions in the softly labeled dataset [2]. All of the remaining PPI combinations are assumed as negative samples, since it is hard to prove that an interaction does not exist.It is assumed that the ratio positive:negative is 1:100 [2]. We use the dataset that was used in [2]. The number of instances in the dataset is given in the Table 2.2.

Each instance of the dataset $X = x_1, x_2, \ldots, x_n$ represents a PPI between HIV-1 and human protein pair. An instance x_i is represented with an 18 dimensional fea-

Table 2.2. HIV-1 and human protein interaction data.

# Features	# Softly Labeled	# Positive Labeled	# Remaining Pairs
18	2119	158	352338

ture vector $x_i = (x_i^1, x_i^2, \ldots, x_i^{18})^T$. The names and the numbers of these features are GO function similarity (1), GO process similarity (1), GO component similarity (1),GO neighbor function similarity (1), GO neighbor process similarity (1), GO neighbor similarity (1), GO neighbor component similarity (1), Post translational modification (1), Degree (1), Clustering coefficients (1), Betweenness centrality (1), Sequence similarity to human protein‘s neighbors (1), Pairwise sequence similarity (1), ELM ligand feature (1), Gene expression features (4) and Tissue distribution (1). These features are described in detail in [21]. The binary classification problem is defined as the set of X_i, y_i pairs where the training set $L \in X$, the test set $S \in X$ and $y \in \{1(interact), 0(notinteract)\}$. This dataset contains positive, negative and softly labeled data for the subset L and positive and negative labeled data for the subset S.

In some datasets, the majority class constitutes a large proportion in the dataset and the minority class constitutes a very small proportion relative to the majority class. These datasets are called ‘imbalanced’ and the minority class suffers from the classifiers’ prediction tendency towards the majority class data. As it is mentioned earlier, imbalanced data causes misclassifications for many algorithms [25]. The methods used to solve this problem are ‘learning from minority class’, ‘adjusting misclassification error’, ‘oversampling the minority class labeled training data’ and ‘undersampling the majority class labeled training data’. Therefore, under sampling is applied to the majority class (‘not interact’) in this study.

The instances of a fold are the same for all algorithms that we used. To prepare the instances of folds, we apply under sampling by choosing 10 different negative labeled subsets among 352338 pairs randomly and the size of each is equal to sum of positive

labeled and softly positive labeled data (2277). Moreover, we do not choose a negative instance more than once for all folds during random selection. We do 10-fold cross-validation. A fold includes a negative subset and the whole positive and softly labeled dataset. Notice that the negative subset changes in each fold but the softly labeled and positive labeled data remain the same. For each fold, the training data includes 142 positive, 2119 softly positive, 2273 negative instances and the validation data includes 16 positive, 228 negative instances. The validation data do not have any softly labeled instances.

2.3. Optical Digit Dataset

The original handwritten digits dataset of NIST (MNIST) was widely studied by many researchers in the field of machine learning. This dataset contains handwritten integer examples from 0 to 9 (See Figure 2.1). $60,000$ instances were taken from American Census Bureau employees as training set and $10,000$ instances were taken from American high school students as test set [26] . The optical digit data that we used was constructed from MNIST by using preprocessing programs of NIST [27]. This dataset includes instances from 30 people for the training set and 13 people for the test set. All digit instances are normalized to 32x32 binary pixel boxes and anti-aliased [27]. Each 32x32 bitmap was divided into non-overlapping 4x4 bitmap blocks and pixels in each 4x4 block were counted. Since it is a 4x4 block, these counts are integers between 0 and 16 and was an element of a 8x8 input matrix. Thus, the dimensionality is reduced by generating 64 features for each instance [27] . The number of instances is given in the Table 2.3.

Instead of classifying the optical digit data into 10 different classes, we formulated a binary problem, which classifies digits as 'even' and 'odd'. By this way, the problem becomes non-linear, more complex and more challenging. Moreover, we also created softly labeled data by selecting certain number of labeled data and adding uniform noise between $-n/2$ and $n/2$, where n is the maximum noise, to distort them. Literally speaking, each instance of the dataset $X = x_1, x_2, \ldots, x_n$ represents an optical digit. An instance x_i is represented with an 64 dimensional feature vector $x_i = (x_i^1, x_i^2, \ldots, x_i^{64})^T$.



Figure 2.1. A set of handwritten digit preprinted forms of MNIST database [3].

The binary classification problem is defined as the set of X_i, y_i pairs where the training set $L \in X$, the validation set $V \in X$, the test set $S \in X$ and $y \in \{1(even), 0(odd)\}$. This dataset contains positive, negative and softly labeled data for the subset L and positive and negative labeled data for the subsets S and V.

Table 2.3. MNIST handwritten digits database.

		Number of	
		Training Data	**Test Data**
Class	**0**	376	178
	1	389	182
	2	380	177
	3	389	183
	4	387	181
	5	376	182
	6	377	181
	7	387	179
	8	380	174
	9	382	180
	Total	3823	1797

As we indicated in the earlier sections, the classifier has the tendency to ignore the minority class for imbalanced datasets. Therefore, we select the same number of instances from each class from 0 to 9 for both hardly and softly labeled training data and hardly labeled test data. Here, hardly expresses the truly labeled instances. We applied 10-fold stratified cross-validation for the datasets '100 hardly labeled, 50 softly labeled', '100 hardly labeled, 100 softly labeled', '100 hardly labeled, 200 softly labeled', '200 hardly labeled, 50 softly labeled', '200 hardly labeled, 100 softly labeled', '200 hardly labeled, 200 softly labeled', '300 hardly labeled, 50 softly labeled', '300 hardly labeled, 100 softly labeled' and '100 hardly labeled, 200 softly labeled'. We applied different amounts of noise from 0 to 50 by 10 on softly labeled data on all datasets. For instance, for '100 hardly labeled, 50 softly labeled' dataset, training data includes 5 even hardly labeled instances by selecting one each from class 0, 2, 4, 6, 8 and 5 odd hardly labeled instances one each from class 1, 3, 5, 7, 9 for each fold. We use complete 50 softly labeled data occurring 5 softly labeled instances from each class 0 through 9 in each fold. After all parameters are optimized via cross-validation, each algorithm runs on a test set which includes 200 hardly labeled instances from each class 0 to 9. The instances of a fold are the same for all of the algorithms that we use.

The even-odd classification problem we created could not be so successful to show the performance of our algorithms since even training with 100 hardly labeled instances on its own achieve 98% accuracy. Therefore, we change the problem as classifying the numbers as '5' and '6'. In this way, we are able to try training with 10, 30 and 50 hardly labeled and see the change by adding softly labeled data. Using only 50 and 30 hardly labeled data led to nearly 98% accuracy as well. Fortunately, lower accuracies were obtained by using only 10 hardly labeled data, which enabled us to observe the effect of including softly labeled data.

3. METHODOLOGY

3.1. k Nearest Neighbor Based Algorithms

k-NN is a lazy learning algorithm, which means the training phase does not exist. It uses the training data directly during the classification phase of the test data. To classify a test datum, the distance (e.g. Euclidean, Mahalanobis, Hamming) between the datum and the training data is calculated. The k nearest neighbor vector $NN_{i,k}$ of the ith datum is used to decide the class of the datum by majority voting. The main motivation behind the k-NN algorithm is the expectation that data instances close to each other are likely to have the same label. The loss function of the instance weighted k-NN method that we propose includes a similarity function, which is based on Euclidean distance. In order to use the same metric for all algorithms, we use the similarity function that is described in [28]. The similarity function is the opposite of Euclidean distance function. Moreover, these similarities are normalized into interval $[0, 1]$. First, the Euclidean distance between two data instances x_i and x_j is calculated as shown in Equation 3.1. Then, Euclidean distance is converted to a similarity function as shown in the Equation 3.2. Here, $dist_{max}$ is the Euclidean distance between the vector that holds the maximum value of each feature and the vector that holds the minimum value of each feature d from 1 to N.

$$\mathbf{dist}(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\sum_{d=1}^{N} (\mathbf{x}_i^d - \mathbf{x}_j^d)^2} \tag{3.1}$$

$$\mu(\mathbf{x_i}, \mathbf{x_j}) = 1 - (dist(x_i, x_j)/dist_{max}) \tag{3.2}$$

The choice of parameter k can be determined by various heuristic or computational techniques. The smaller values of k may make the classifier to be more easily affected from noisebut, might also lead to a better distinction between classes. On the

other hand, the larger values of k give relatively worse distinction between classes, but may be more resistant against noisy data. In this study, a greedy approach is used to choose k. The k-NN based algorithms given below are executed with odd k values from 1 to 9 ($k = 1, 3, 5, 7, 9$). Only odd numbers are used to avoid tied votes. For each value of k, the same set of folds are used for cross-validation. The objective function of this hyper parameter optimization is maximizing the accuracy for optical digit and CRC data and area under the Precision-Recall curve (PR AUC) for HIV-1-Human PPI data, which are the performance indicators of these data. Therefore, the best value of k is selected in terms of the given metrics.

Semi supervised k-NN includes both labeled and unlabeled data. Assume that L denotes the set of labeled instances and U denotes the set of unlabeled instances. For each instance $x_i \in U$ is labeled by the k nearest neighbors which are the instances in the labeled set L. Then, these newly labeled unlabeled data may be utilized by the semi-supervised learning techniques. However, we have unlabeled data with some information, which we called as 'softly labeled' data. Since we created softly labeled data by adding certain amount of artificial noise, we know the true even-odd label of the softly labeled optical digit data. We have colorectal cancer stage information for CRC data. For HIV-1-Human PPI, the softly labeled data are not surely positive but thought to be highly positive. Therefore, we adapted semi-supervised k-NN algorithms by using softly labeled data in three different ways as described in the following sections.

3.1.1. Semi-Supervised k-NN Algorithm (SS-kNN)

In this scheme, we give a reliability coefficient θ=1 to each hardly labeled training datum. The same value for the coefficient θ is assigned to each softly labeled datum. We experimented with various θ values for the softly labeled data in the range $(0, 1]$. The test set and the validation set include only positive and negative instances. To label an instance, majority voting is applied by summing the weights of the k nearest neighbors $NN_{i,k}$ with the same label y. If $f(x_i) > 0$ then the predicted label of the ith instance is positive, else it is negative (See Equation 3.3). If $f(x_i) = 0$ then we decide the label randomly which is expected to be a very rare case. To see whether softly

labeled data have an effect on the results, we also apply the same algorithm by using only the labeled set with the coefficient θ=1 for all instances. The results are given in the experiments section.

$$\mathbf{f}(\mathbf{x_i}) = \sum_{n=1,k\in odd}^{k} \theta(NN_{i,n}) \times y_n \tag{3.3}$$

3.1.2. Semi-Supervised Self Trained k-NN Algorithm (ST-kNN)

Self-training (self-teaching, bootstrapping) is a commonly used technique in Machine Learning. In self-training, a classifier is first trained with labeled data at first. Then the classifier labels the unlabeled data. The most confidently labeled unlabeled data is added to the training set with its predicted label. Then, the classifier is trained again and these steps are repeated iteratively. In this study, one can think of unlabeled data in the self-training set up as softly labeled data. All instances have weight θ=1 in this algorithm. In this scheme, the softly labeled data are labeled with the help of the labeled training data and the ones with post probability p_{value} over a certain threshold are added to the training set and excluded from the softly labeled data. This process continues iteratively until there is no remaining softly labeled data or no softly labeled data can be labeled with a confidence value above the threshold at that iteration. The k nearest neighbors of an instance decide on the label of the new data as in Equation 3.3 where all the weights are equal to 1.

The post probability of an instance is calculated as in Equation 3.4 which is valid for all k-NN based algorithms. We experiment with three different thresholds 0.7, 0.8, and 0.9 for this algorithm. If p_{value} is equal to 0.5, it means the algorithm decides the label of a data randomly and this is expected to be very rare.

$$p_{value_i} = \begin{cases} \dfrac{\sum_{n=1}^{k} \theta(NN_{i,n})^{+}}{\sum_{n=1}^{k} \theta(NN_{i,n})^{+} + \sum_{n=1}^{k} \theta(NN_{i,n})^{-}} & \text{, if } f(x_i) > 0.5 \\ \\ \dfrac{\sum_{n=1}^{k} \theta(NN_{i,n})^{-}}{\sum_{n=1}^{k} \theta(NN_{i,n})^{+} + \sum_{n=1}^{k} \theta(NN_{i,n})^{-}} & \text{, if } f(x_i) < 0.5 \end{cases} \tag{3.4}$$

3.1.3. Semi-Supervised Instance Weighted k-NN Algorithm (IW-kNN)

In this modified version of k-NN, each instance is associated with a weight θ in the interval (0,1]. Unlike the SS-kNN approach where all instances are associated with the same θ value, the instances in the IW-kNN approach might have different θ values. The training phase includes two main steps; labeling the labeled training set once and joining the softly labeled data into the training set after labeling. The k nearest neighbors of an instance decide on its label as in Equation 3.3 and its corresponding weight θ_i as in Equation 3.5, where θ denotes the weights.

$$\theta_i = \begin{cases} \dfrac{\sum_{n=1,k\in odd}^{k} \theta(NN_{i,n})^{+}}{\left|NN_{i,k}^{+}\right|} & \text{, if } f(x_i) = 1 \\ \dfrac{\sum_{n=1,k\in odd}^{k} \theta(NN_{i,n})^{-}}{\left|NN_{i,k}^{-}\right|} & \text{, if } f(x_i) = 0 \end{cases} \tag{3.5}$$

First, we give random weights θ to the labeled training set $L \in X$, which includes only positive and negative instances. Then, we let the k-NN algorithm to label each of the training set instances. When the predicted label and the actual label are not the same for an instance, the weights of that instance's neighbors, which have the same label as the expected label are encouraged according to their weights and similarity to the current instance. Likewise, the neighbors who have the opposite label of the

expected label are punished by the loss function given in Equation 3.7.

Second, instead of putting all the softly labeled data $S \in X$ into the training set directly, we labeled them with the current training set one by one in a self-training manner. In each iteration of labeling, if the predicted label is not the same as the expected label, we punish the weights of that instance's k neighbors that have the opposite label to the expected label, and incrase the weights of the neighbors with the same label as the expected label. The loss functions are the same for both the training data and the softly labeled data labeling. After all the softly labeled data are labeled and added into the training set, the test set is labeled by using the found weights using the k-NN algorithm with majority voting. At the end, the final training set classifies the test set. The pseudo-code of the training and test phases of the algorithm are given in the Figure 3.1 and Figure 3.2 respectively.

In the loss function, the selected k nearest neighbors are punished/encouraged inversely proportionally to their distances, in other words proportionally to their similarity μ. That is, the closer (or more similar) neighbor will have higher punishment/encouragement than the further ones. After labeling of a training instance, the loss function is calculated for that labeling. For an expected label $t = \pm 1$ and a predicted label $y = \pm 1$, the new weight of the nth nearest neighbor $\theta(NN_{i,k,n})'$ is computed as shown in Equation 3.6 where m is the label of the nth neighbor, t is the expected label of the ith instance and the loss function $L(NN_{k,n})$ is as shown in Equation 3.7.

$$\theta(\mathbf{NN_{i,k,n}})' = \begin{cases} max(0, \theta(NN^m_{i,k,n}) - L(NN_{k,n})) & \text{, if } t \neq m \\ min(1, \theta(NN^m_{i,k,n}) + L(NN_{k,n})) & \text{, if } t = m \end{cases} \tag{3.6}$$

where

$$\mathbf{L}(\mathbf{NN_{k,n}}) = \begin{cases} \theta(NN_{i,k,n}) \times \mu(x_i, NN_{i,k,n}) & \text{, if } t \times y < 1 \\ 0 & \text{,otherwise} \end{cases} \tag{3.7}$$

$L(NN_{k,n}) = 0$ means the classifier predicts an instance correctly, otherwise the loss function is applied on the k nearest neighbors' weight vector $NN_{i,k} = [\theta_1, \theta_2, \ldots, \theta_{|NN|}]^T$. The k nearest neighbors' weight vector is updated to new weights' vector $NN'_{i,k} = [\theta'_1, \theta'_2, \ldots, \theta'_{|NN|}]^T$.

3.2. Support Vector Machine Based Algorithms

Support vectors are the training instances of each class that are closest to the decision boundary [29]. The SVM algorithm determines a decision boundary that maximizes the margin between the support vectors of two classes. In linear classification with SVM, the training data is separated with a hyperplane with maximum margin. The test data are labeled by the support vectors SV according to the side of the margin that they falls. To be able to do non-linear classification with SVM, the kernel trick is used. In this scheme, a selected kernel maps the training instances into a higher dimensional feature space. Assume that each instance is represented by a (x_i, y_i) pair, where $i = 1, ..., m$, m is the number of instances, $x_i \in \mathbb{R}$ is a feature vector with d dimensions and $y_i \in \{-1, 1\}$. The primal form of the SVM optimization problem is given in Equation 3.8. The function ϕ, which is called as feature map, maps each feature vector x_i into a feature space in which SVM finds a hyperplane that separates the instances of the two classes with maximal margin. The vector w is the normal vector of the hyperplane. The parameter C is the penalty factor for slack variable ξ_i which is the bias value for the feature vector x_i. The kernel trick $K(x_i, x_i) = \phi(x_i)^T \phi(x_i)$, is used so that $\phi(x_i)$ does not have to be computed explicitly. The support vectors array SV includes the instance pairs from the training set T which are on the margin and satisfy $y_i(w^T \phi(x_i) + b) = 1$. A test instance can be labeled by $f(x) = y_i(w^T \phi(x_i) + b)$. If $f(x) > 0$) the instance is labeled as 0, otherwise it is labeled as 1. The kernels used

$\theta \Leftarrow$ random weights between $(\mathbf{0}, \mathbf{1}]$

for each $\mathbf{l}_i \in \mathbf{L}$ **do**

current datum$\Leftarrow \mathbf{L}_i$

Find k nearest neighbors $\mathbf{NN}_{i,k}$ using Eq.3.1 and Eq.3.2

Get weight θ_i of current datum using Eq. 3.5

if $\theta_i < 0$ **then**

predicted label $\Leftarrow \mathbf{'0'}$

else

predicted label $\Leftarrow \mathbf{'1'}$

end if

if predicted label $\neq$ actual label **then**

Calculate loss values of vector $\mathbf{NN}_{i,k}$ using Eq. 3.6 and Eq.3.7

Update new weights of vector $\mathbf{NN}_{i,k}$.

end if

end for

for each $\mathbf{s}_j \in \mathbf{S}$ **do**

Find k nearest neighbors $\mathbf{NN}_{j,k}$ of $\mathbf{s}_i$ using Eq.3.1 and Eq.3.2

Get weight θ_j of $\mathbf{s}_i$ using Eq. 3.5

if $\theta_j < 0$ **then**

predicted label $\Leftarrow \mathbf{'0'}$

else

predicted label $\Leftarrow \mathbf{'1'}$

end if

if predicted label $\neq$ expected label **then**

Calculate loss values of vector $\mathbf{NN}_{j,k}$ using Eq. 3.6 and Eq.3.7

Update new weights of vector $\mathbf{NN}_{j,k}$.

end if

end for

Figure 3.1. Training in IW-kNN Algorithm.

for each $\mathbf{t}_i \in \mathbf{T}$ **do**
  Find k nearest neighbors $\mathbf{NN}_{i,k}$ of $\mathbf{t}_i$ using Eq.3.1 and Eq.3.2
  if $\theta_i < 0$ **then**
    predicted label $\Leftarrow$ **'0'**
  else
    predicted label $\Leftarrow$ **'1'**
  end if
end for

Figure 3.2. Testing in IW-kNN Algorithm.

in this study are given in Equation 3.9.

$$\begin{aligned} \underset{w,\xi,b}{\text{minimize}} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(w^T\phi(x_i) + b) \geq 1 \\ & \xi \geq 0. \end{aligned} \tag{3.8}$$

Various kernels such as RBF (Gaussian), and exponential kernels have been proposed in the literature. Choosing the most suitable kernel is an important problem and depends on the properties of the used data directly. We use linear kernel and RBF kernel in our experiments. The most common way to choose the parameter C is trying various values in the interval $[2^{-3}, 2^3]$ and get the best result in terms of the performance metric of a problem. This interval is not restricted and generally it is changed into smaller intervals to get better results. C is actually a trade-off parameter between how much the training set will be fitted and how much the hyper-parameters will affect the discriminant function. In other words, large C values cause lower bias,

high variance and fit more to the training data, while small C values cause higher bias, low variance and fit less to the training data. We use stratified cross-validation to tune C and γ. In our case, the objective function of this hyper parameter optimization is maximizing our performance indicators 'accuracy' for optical digit and CRC data and 'area under the Precision-Recall curve (PR AUC)' for HIV-1 and Human PPI data.

$$\begin{aligned} K_{Linear}(x_i, x_j) &= x_i^T x_j \\ K_{RBF}(x_i, x_j) &= \exp(-\gamma \|x_i, x_j\|^2), \gamma > 0 \end{aligned} \tag{3.9}$$

Semi Supervised SVM uses both labeled and unlabeled data. Assume that L denotes the set of labeled instances and S denotes the set of softly labeled instances. Each instance $x_i \in S$ is labeled by the support vectors SV which are instances in the labeled set L. Since we have unlabeled data with some information aka. 'Softly Labeled' data, we adapted semi-supervised SVM based algorithms by using softly labeled data in three different ways as described in the following sections. The SVM based algorithms are similar to the corresponding k-NN based algorithms except;

- The similarity function of k-NN changes to a kernel function (linear and RBF) in SVM,
- θ_i is no longer a weight coefficient for the instances as in the k-NN algorithm but regulator coefficient for the C parameter as θ_ixC for SVM
- The loss function changes as explained in the following sections.

3.2.1. Semi Supervised SVM Algorithm (SS-SVM)

In the problems that we tackle there can both labeled data and softly labeled data. We are certain about the labels of the labeled data. On the other hand, we are uncertain about the labels of the softly labeled data. We assign different reliability coefficients θ for the C parameters of each training datum. This enables each training instance to

penalize the objective function of SVM in proportion to its reliability coefficient θ. In other words, if θ_i gets closer to 1 for an instance i, the model fits more to this instance i. On the other hand, if θ_i gets closer to 0 for an instance i, this instance effects the model less. By introducing reliability coefficients θ, where $0 < \theta_i \leq 1$ and $i = 1, 2, ..., n$ where n is the size of the training data, the SVM objective function in the primal form is;

$$\begin{aligned} \underset{w,\xi,b}{\text{minimize}} \quad & \frac{1}{2}\|w\|^2 + \theta_i \times C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(w \times x_i - b) \geq 1 \\ & \xi \geq 0 \\ & 0 < \theta_i \leq 1. \end{aligned} \tag{3.10}$$

We set the reliability coefficient θ=1 for the labeled data. For the softly labeled data, we experiment with various coefficients in the range $(0, 1]$.

3.2.2. Semi-Supervised Self Trained SVM Algorithm (ST-SVM)

As we indicated in the previous sections, self-training is one of the earliest forms of semi-supervised approaches in which a classifier is trained with labeled data and then label the unlabeled data. Iteratively, the most confident predictions on the softly labeled data are added to the training data with their predicted labels. All instances have coefficient θ=1 for the parameter C in this algorithm. In this scheme, the softly labeled data are labeled by the help of the training data and the ones with post probability p_{value} over a certain threshold are added to the training set and excluded from the softly labeled set. This process continues iteratively until there are no remaining softly labeled data or no softly labeled data are labeled with the confidence above the threshold at that iteration.

In this study, we estimate the post probability of an instance in SVM algorithms using Platt's method [30]. Platt's method uses the sigmoid function to approximate the scores [30]. We experimented with three different thresholds 0.7, 0.8, and 0.9 as well as with no threshold for this algorithm. If p_{value} is equal to 0.5, it means the algorithm decides the label of a data instance randomly and this is expected to be very rare.

3.2.3. Semi-Supervised Instance Weighted SVM Algorithm (IW-SVM)

In this modified version of SVM, each instance is associated with a coefficient θ for the parameter C in the interval (0,1]. The training phase includes two main steps; labeling the labeled training set once and joining the softly labeled data into the training set after labeling as in the IW-kNN algorithm. The support vectors SV_i decide on the labels of the new data and the corresponding coefficients θ_i as in the Equation 3.11, where θ denotes the coefficient of the parameter C.

$$\theta_i = \begin{cases} \dfrac{\sum_{n=1}^{|SV|} \theta(SV_n^+)}{\left|SV^+\right|} & \text{, if } f(x_i) > 0 \\ \\ \dfrac{\sum_{n=1}^{|SV|} \theta(SV_n^-)}{\left|SV^-\right|} & \text{, if } f(x_i) < 0 \end{cases} \tag{3.11}$$

First, we give random weights θ to the labeled training set $L \in X$, which includes only positive and negative instances. Then, we let the SVM algorithm label each of the training set instances. When the predicted label and actual label are not the same for an instance i, the coefficients of the support vectors SV that have the same sign with the expected label of instance i are increased according to their coefficients and similarities to the current instance. Likewise, the support vectors which have the opposite sign of the expected label of instance i are punished by the given loss function.

In each iteration, we run the SVM training phase with the updated coefficients θ.

Second, instead of putting all the softly labeled data $S \in X$ into the training set directly, we labeled them using the current training set one by one in a self-training manner. In each iteration of labeling an instance $j \in S$, if the predicted label is not the same as the expected label of instance j, we punish the coefficient of the support vectors that have opposite sign to the expected label of instance j and increase the support vectors with the coefficients of the same label as the expected label of instance j. The loss functions are the same for both the training data and the softly labeled data labeling. After all the softly labeled data are labeled and added into the training set, the test set is labeled by using these tuned and updated coefficients.

In the loss function, the support vectors SV are punished/encouraged proportionally to their similarities μ, which are computed by using the RBF kernel or the Linear kernel as in Equation 3.9. That is, the closer (or more similar) support vectors will have greater punishment/encouragement than the further ones. After labeling of a training instance, the loss function is calculated for that labeling. For an expected label $t = \pm 1$ and a predicted label $f(x) = \pm 1$, the new weight of the nth support vector $\theta(SV_n)'$ is calculated using Equation 3.12 where y is the label of the nth support vector, t is the expected label of the ith instance and the loss function $L(SV_n)$is as shown in Equation 3.13.

$$\theta(\mathbf{SV_n})' = \begin{cases} max(0, \theta(SV_n^m) - L(SV_{i,n}) & \text{, if } t \neq y \\ min(1, \theta(SV_n^m) + L(SV_{i,n}) & \text{, if } t = y \end{cases} \tag{3.12}$$

where

$$\mathbf{L}(\mathbf{SV_{i,n}}) = \begin{cases} \theta(SV_{i,n}) \times \mu(x_i, SV_{i,n}) & \text{, if } t \times f(x_i) < 1 \\ 0 & \text{,otherwise} \end{cases} \tag{3.13}$$

$L(SV_{i,n}) = 0$ means the classifier predicts an instance correctly, otherwise the loss function is applied on the coefficients of the support vectors' vector $SV_n = [\theta_1, \theta_2, \ldots, \theta_{|SV|}]^T$. The support vectors' coefficient vector is updated to the new co-efficeints' vector $SV'_{i,n} = [\theta'_1, \theta'_2, \ldots, \theta'_{|SV|}]^T$.

4. EXPERIMENTS AND RESULTS

We evaluate the performances of the proposed algorithms on the optical digit and colorectal cancer datasets according to their accuracies both by using cross-validation on the training set, and on a separate test set. On the other hand, since we do not have any hardly labeled negative instances in the HIV-1-Human PPI dataset, we use area under the precision-recall curve as the primary evaluation metric on the training set with cross-validation. We use 10-fold cross-validation for the optical digit and HIV-1-Human PPI datasets, and 5-fold cross-validation for the colorectal cancer data set. It is stated that the sample to build a classifier should not be used for testing it [8]. For this reason, cross-validation and test sets should be completely different. Therefore, we used two different datasets, which are collected in different studies for colorectal cancer. The pre-processesing steps are also applied separately to the data sets. The following subsections describe the evaluation metrics and results.

4.1. Evaluation Metrics

The confusion matrix of a classifier (X_i, C_i), where X denotes the feature set and C denotes the class set consists of four outcomes. One of the possible outcomes is 'True Positive (TP)' state in which both the predicted class and the actual class are 'positive'. Another one is the outcome 'False Negative (FN)', which occurs when the actual class is 'positive' but the predicted class is 'negative'. Conversely, the 'False Positive (FP) ' outcome occurs when the actual class is 'negative' but the predicted class is 'positive'. The last one is the 'True Negative (TN)' outcome where both the predicted and actual classes are 'negative'. The confusion matrix is given in Table 4.1. In this study, the 'positive' class refers to digit '6' for the optical digit dataset, 'recurrence' within 5 years for the colorectal cancer recurrence dataset, and 'interact' for the HIV-1-Human PPI dataset. On the other hand, 'negative' indicates digit '5' for the optical digit dataset, 'non-recurrence' within 5 years for the colorectal cancer recurrence dataset, and 'not-interact' for the HIV-1-Human PPI dataset.

Table 4.1. Confusion matrix.

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	TP	FN
	Negative	FP	TN

Accuracy, Precision, Recall, F-Measure, Receiver Operating (ROC) Curve, Precision Recall (PR) Curve, and Area Under the Curve (AUC) are commonly used evaluation measures for binary classification problems. Since we have correctly labeled positive and negative instances in the optical digit and colorectal cancer data, we use accuracy as the main metric. Accuracy is calculated with the formula in Equation 4.1. We also compute area under the ROC and PR curve, break-even value and confusion matrix. For these two datasets, parameter tuning was handled with cross-validation with the objective function that maximizes accuracy. Then the algorithms are trained with these found optimal parameters and tested on a separate dataset. The cross-validation results, and the results on the test sets are given in Appendix C.

$$\textbf{Accuracy} = \frac{TP + TN}{AllPredictions} \tag{4.1}$$

For the PPI dataset, it is hard to determine a 'not-interact' pair. We may only reveal an 'interacting' pair. A pair of proteins labeled as 'not interact' can correspond to one of the following two facts: (i) The protein pair does not interact, (ii) The protein pair interacts, however, we do not know that yet. Since we do not have any hardly labeled negative samples, most of the binary classification evaluation metrics are not valid and meaningful. Therefore, we choose area under the PR curve, which is also called as MAP value, to evaluate the HIV-1 and human PPI dataset.

PR curve shows the success of labeling true positives. It consists of various Precision values on the y-axis and Recall values on the x-axis. Precision is the value of true positive fraction among all retrieved positive instances (Equation 4.2). Recall is the value of true positive fraction among all positive instances (Equation4.3). AUC value of PR curve actually corresponds to Mean Average Precision (MAP) value. In PPI domain, we are more interested to predict positive interactions than negative ones for narrowing the possibilities and reducing the number of wet lab experiments. Therefore, we calculate the area under the curve (AUC) value of this curve to evaluate our algorithms on this dataset.

ROC Curve shows good performance for low false positive rate and high true negative rates. It is a pre-condition to have truly labeled positive and negative instances to consider ROC curve as a metric. Since, we are not able to label an instance as negative for PPI dataset, we think ROC is just meaningful for optical digit and colorectal cancer datasets. ROC Curve consists of various True Positive Rate (Recall) values on the y-axis and False Positive Rate (Equation 4.4) values on the x-axis. The faster the curve gets to the point $[1, 1]$, the better performs the algorithm. PR Curve is sensitive to highly skewed datasets [31]. Fortunately, imbalance situation is removed by under-sampling.

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

$$\textbf{Recall} \approx \frac{TP}{P} \tag{4.3}$$

$$\textbf{FalsePositiveRate} \approx \frac{FP}{N} \tag{4.4}$$

Labeling each instance and calculating a single point with precision, recall and false positive values do not give the full curve. This single point just gives the value where decision threshold between two classes is 0.5. To be able to get full curve,

precision, recall and false positive rate is calculated with different threshold values. Therefore, instead of just labeling each test instances, class membership scores of each labeling is estimated for the classifier. A class membership score gives the labeling probability that a test instance i belongs to the predicted class. For k-NN based algorithms, the score p_{value_i} of a test instance i is estimated roughly by proportioning the number of positive labeled nearest neighbors to number of the nearest neighbors considering their weights too as shown in Equation 3.4. The classification score for SVM based algorithms is estimated by using LibSVM library [32] which uses a different form of Platt's method [30] including sigmoid function to approximate the scores. After sorting classification scores from highest to lowest, a method to approximate precision, recall and false positive values for various thresholds as proposed in [33], and [34]. We also calculate the Break-Even Point (BEP), which is the point where precision and recall is equal in the precision-recall curve.

In some domains, instead of labeling all test instances, it is denied to label some instances having certain amount of uncertainty on labeling since it is too risky. As an illustration, if the probability of colorectal cancer recurrence within 5 years for a patient who had been colorectal cancer before was 0.6, it would have been hard to classify this patient as a 'recurrence' instance. If the 'recurrence' within 5 years is not observed, the patient would have been applied chemotherapy treatment unnecessarily. On the other hand, if this patient is classified as 'non-recurrence' because of the previous risk and the cancer showed up within 5 years, this might be fatal for the patient since the treatment would not have been applied. To prevent this kind of highly risky decisions, algorithms could simply deny to give response for that instance. Therefore, while classifying the validation instances for PPI dataset and test set instances for optical digit and colorectal cancer datasets, we applied different thresholds from 0.1 to 0.5 between two classes. In this scheme, if p_{value_i} is greater than or equal to 0 and p_{value_i} is less than or equal to $currentthreshold$, the instance i is labeled as 'negative'. If p_{value_i} is less than or equal to 1 and p_{value_i} is greater than or equal to $(1 - currentthreshold)$, the instance i is labeled as 'positive'. Otherwise, the instance is rejected, that is a class label is not assigned. In addition, if p_{value_i} is equal to 0.5 and $currentthreshold$ is equal to 0.5 the label of the instance i is determined randomly, which is expected to

happen rarely.

4.2. Results for the Colorectal Cancer Data Set

The average accuracy values on the test data set in each fold of cross-validation, where p_{value} cutoff=0.05, are as shown in Figure 4.1. Traditional training by using only labeled data produces relatively lower accuracies in the given figure below. For example Supervised k-NN, SVM with Linear Kernel, and SVM with RBF kernel achieve 50.43±6.59, 21.74±3.07 and 87.83±27.22 respectively. All supervised methods fail except the SVM R (SVM with RBF kernel) which has very high variance. On the other hand, the semi-supervised k-NN (65.22±3.07), SS-SVM L (Semi-supervised SVM with Linear kernel) (60.87±4.35) and SS-SVM R (Semi-supervised SVM with RBF kernel) (100±0) achieve the best accuracies in all cases. Especially, SS-SVM R hits the 100% accuracy. Thus, it proves that softly labeled data are informative and can increase the accuracy.

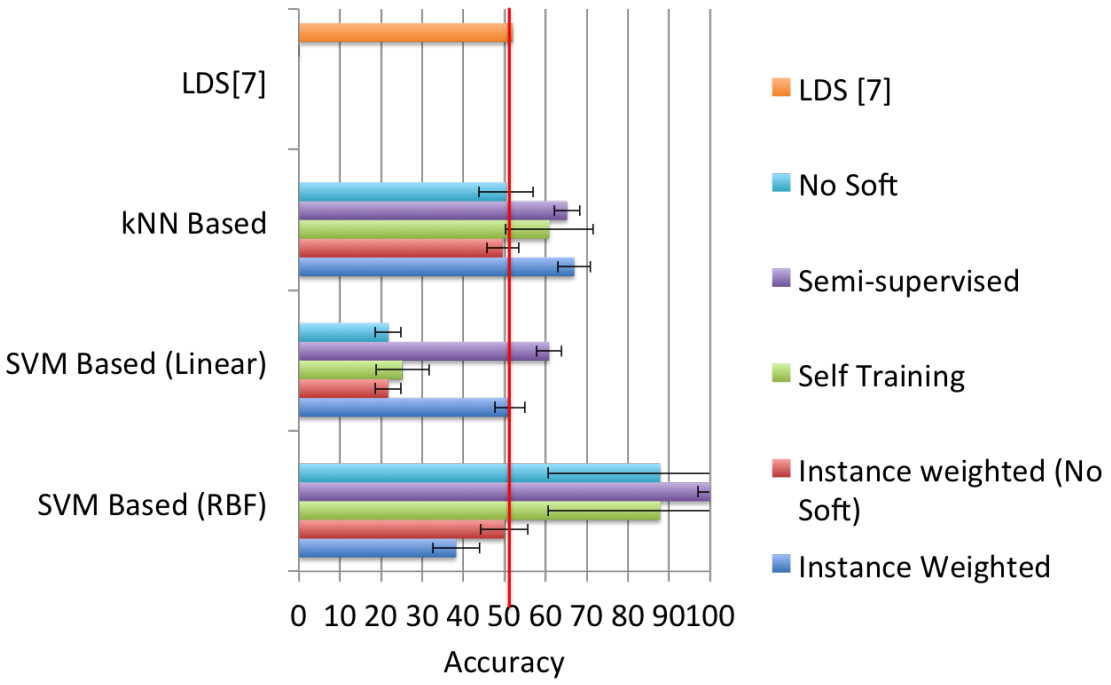

Figure 4.1. 5 fold cross-validation accuracy for colorectal cancer recurrence data set.

The instance weighted k-NN (IW-kNN) algorithm achieves higher accuracy (66.96 ±10.65) than the instance weighted SVM with Linear kernel (IW-SVM L) (51.3±3.64), and the instance weighted SVM with RBF kernel (IW-SVM R) (38.26±5.67) methods. So we may infere that the instance weighted methods need more data to regulate the parameter C of the SVM and θ values since the hyper parameters make the model more complex. If we compare the RBF and linear kernel results, one can see that the results with linear kernel perform worse than RBF kernel in terms of accuracy which indicates that the data are not linearly separable. Comparing the results with the 52% accuracy obtained by Low Density Separation (LDS) algorithm proposed in [7] we gained higher accuracy by using the IW-kNN, SS-kNN, SS-SVM L and SS-SVM R methods. SVM R and ST-SVM R methods also work better than the LDS algorithm of [7], but their standard deviation is quite high. SVM based methods with linear kernel perform worse than k-NN and SVM RBF based methods which shows that data is not linearly separable.

As we indicated in the previous sections, we expect that a stage I patients are more likely to belong to the 'non-recurrence' class, stage III patients are more likely to belong to 'recurrence' class, whereas it is hard to tell the same thing for stage II patients. Since the semi-supervised methods achieved the best accuracies, it would be better to investigate the parameters for those methods. Table 4.3 shows the parameters for SS-kNN, SS-SVM L and SS-SVM R methods. In the SS-SVM R we achieve 100% accuracy with the assumption that stage1 is negative (aka. 'non-recurrence') and stage3 is positive (aka. 'recurrence') as we expected.

Table 4.4 shows the other evaluation metrics: area under the ROC curve (ROC-AUC) and Break Even Point values. The bolded ROC-AUC values indicate meaningful ones which are higher than 0.5, which is the threshold of randomness for ROC curve. We also achieve higher ROC-AUC values IW-kNN(0.58±0.09), ST-kNN(0.59±0.1), SS-kNN(0.65±0.03), SS-SVM L(0.73±0.12), SVM R(0.92±0.18), IW-SVM R No Soft (0.62±0.08), IW-SVM R(0.62±0.08), ST-SVM R(0.92±0.18) and SS-SVM R(1±0) than the 0.55 value obtained in [7].

Table 4.2. Colorectal cancer: accuracy values on test data with various parameters of all algorithms.

	Accuracy
k-NN	50.43±6.59
IW-kNN (No Soft)	49.57±3.89
SS-kNN	65.22±3.07
ST-kNN	60.87±10.65
IW-kNN	66.96±3.89
SVM L	21.74±3.07
IW-SVM L (No Soft)	21.74±3.07
SS-SVM L	60.87±4.35
ST-SVM L	25.22±6.45
IW-SVM L	51.3±3.64
SVM R	87.83±27.22
IW-SVM R (No Soft)	50±5.67
SS-SVM R	100±0
ST-SVM R	87.83±27.22
IW-SVM R	38.26±5.67

Table 4.3. Accuracy in the colorectal cancer data set and the parameters learned with cross-validation for the semi-supervised methods.

	Accuracy	**Parameters**
SS-kNN	65.22±3.07	k=1, StageI(+,θ=1), StageII(+, θ=1),StageIII(+, θ=1)
SS-SVM L	60.87±4.35	Stage I(+, θ=1),StageIII(+, θ=1),
SS-SVM R	100±0	StageI(-, θ=1),StageIII(+, θ=0.6)

Table 4.4. ROC-AUC and break even values on colorectal cancer test set.

	ROC-AUC	**Break-Even**
LDS [7]	0.55	-
k-NN	0.49±0.05	0.58±0.12
IW-kNN (No Soft)	**0.51±0.02**	0.53±0.05
iW-kNN	**0.58±0.09**	0.6±0.07
ST-kNN	**0.59±0.1**	0.63±0.1
SS-kNN	**0.65±0.03**	0.68±0.04
SVM L	0.12±0.03	0.25±0
IW-SVM L (No Soft)	0.12±0.03	0.25±0
iW-SVM L	0.43±0.05	0.5±0.06
ST-SVM L	0.14±0.05	0.3±0.07
SS-SVM L	**0.73±0.12**	0.65±0.04
SVM R	**0.92±0.18**	0.9±0.22
IW-SVM R (No Soft)	**0.62±0.08**	0.55±0.07
IW-SVM R	**0.62±0.08**	0.55±0.07
ST-SVM R	**0.92±0.18**	0.9±0.22
SS-SVM R	**1.00±0**	1.00±0

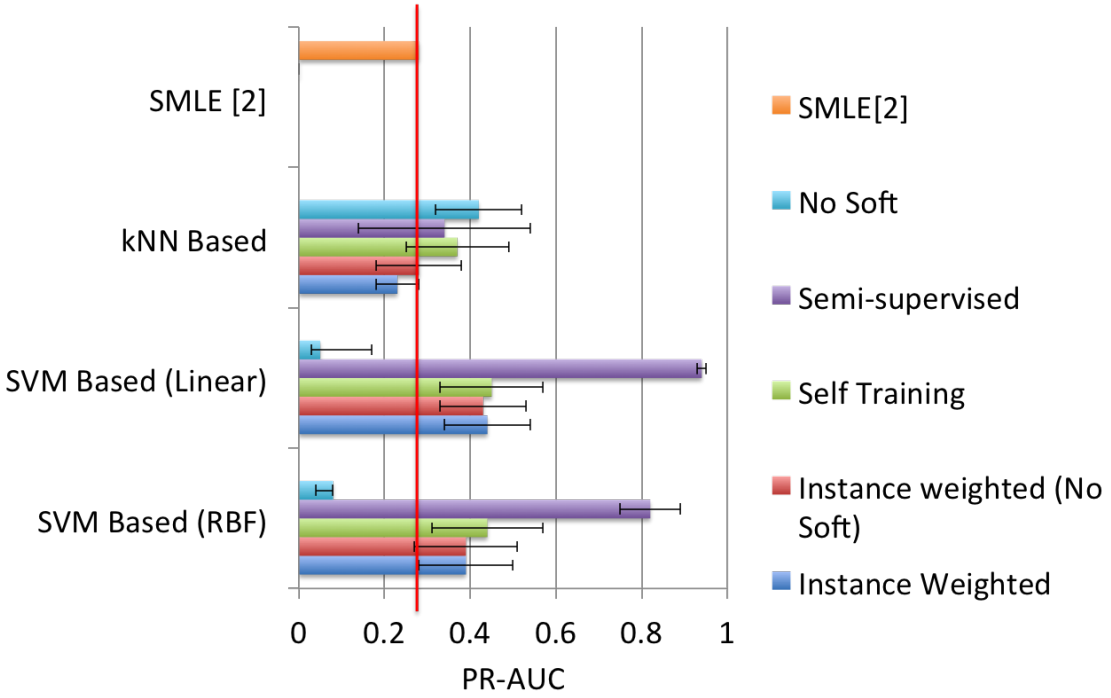

Figure 4.2. PR-AUC on validation set for HIV-1 and human PPI data set.

4.3. Results for the HIV-1 and Human PPI Data Set

As we explained in section 4.1, since the HIV-1-Human PPI data does not have truly labeled (aka. hardly) negative data, it is a more accurate approach to compare area under the precision-recall curve (PR-AUC) and break even (BE) point rather than accuracy. The average PR-AUC values on the validation data in each fold of cross-validation is shown in Figure 4.2. The supervised method performs well for k-NN with the value 0.42±0.1, whereas SVM L gets the value 0.05±0.03 and SVM R gets the value 0.08±0.04 which are very poor. This situation probably occurs because of having very few positive labeled instances compared to the negative labeled instances, therefore the methods could not model the true structure of the data distribution. On the contrary, SS-SVM L and SS-SVM R methods achieve the highest scores 0.94±0 and 0.82±0.07 respectively. Overall, we achieve the values 0.42±0.1 for k-NN, 0.94±0 for SS-SVM L, 0.45±0.11 for ST-SVM L, 0.44±0.1 for IW-SVM L, 0.82±0.07 for SS-SVM R and 0.44±0.13 for ST-SVM R which are higher than the Scalable Machine Learning with Embedding On Output Space (SMLE) algorithm‘s value 0.28 reported in [2].

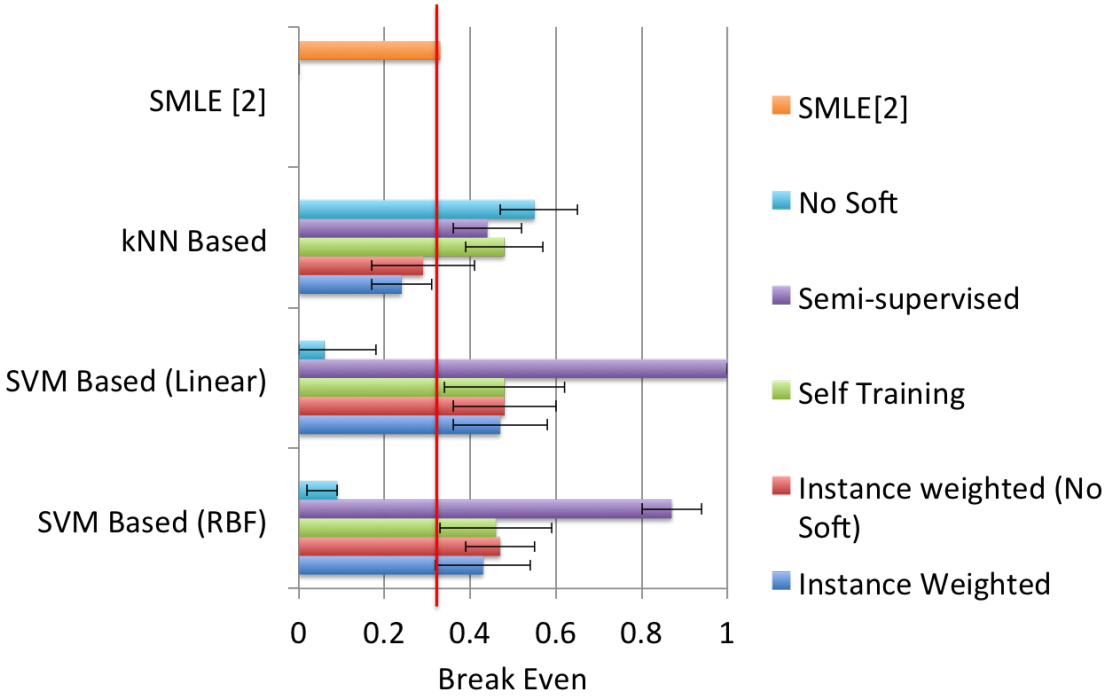

Figure 4.3. Break even point calues on the validation set for HIV-1 and human PPI data set.

Figure 4.3 shows the average break even (BE) values on the validation data in each fold of cross-validation. We obtain the values 0.55±0.08 for k-NN, 0.44±0.08 for SS-kNN, 0.48±0.09 for ST-kNN, 1.0±0 for SS-SVM L, 0.48±0.14 for ST-SVM L, 0.48±0.12 for IW-SVM L (no soft), 0.48±0.07 for IW-SVM L, 0.87±0.07 for SS-SVM R and 0.47±0.08 for IW-SVM R (no soft) which are higher than the SMLE algorithm 0.33 value reported in [2] in terms of break even values. The values 0.47±0.13 for ST-SVM R, 0.43±0.11 for IW-SVM R give similar values to the SMLE algorithm [2] in the worst case. Thus we may deduce that our methods are much stronger to label positive labeled instances correctly. We also calculate the area under the ROC curve to compare as given in Figure 4.4. SS-SVM L and SS-SVM R achieve ROC-AUC values 1.0±0 and 0.98±0.02 which are higher than the Scalable Machine Learning with Auxiliary Ranking Task (SMLR) algorithm's value 0.92 obtained in [2].

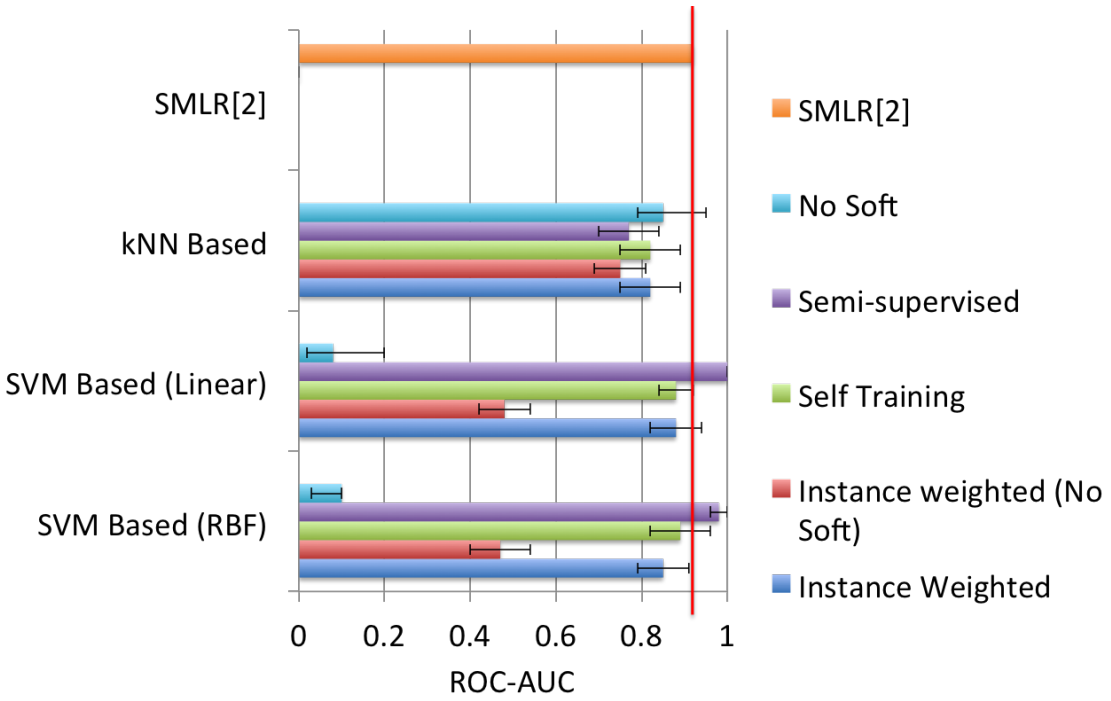

Figure 4.4. ROC-AUC on the validation set for HIV-1 and human PPI data set.

4.4. Results for the Optical Digit Data Set

The average accuracy values for different propostions of noise and number of softly labeled data for 10 hardly labeled data are given in Figure 4.5, Figure 4.6, and Figure 4.7. The supervised methods k-NN, and SVM L achieve 79.39±10.54 and 60.39±8.64 accuracies respectively. On the other hand, SS-kNN, ST-kNN, SS-SVM L, SS-SVM R achieves higher accuracy (See Table. C.5). SS-kNN achieves 87.88±0 accuracy even when the noise is very high, i.e. 50% for 24 softly labeled data. When the noise decreases, its accuracy values increase and reach to 99.45±0 when there is no noise on the softly labeled data. ST-kNN obtains 87.88±0 accuracy when the noise is 50%. In contrast with SS-kNN, ST-kNN achieves its highest accuracy when the noise is 30%. Generally, adding more softly labeled data helps to increase accuracy to some degree. SS-SVM L and SS-SVM R achieve 100±0 accuracy for all number of softly labeled data. Unfortunately, the instance weighted approaches of SVM could not produce any results since the number of data instances is too few. So, the SVM based instance-weighted approaches need more data to be able to be trained. These

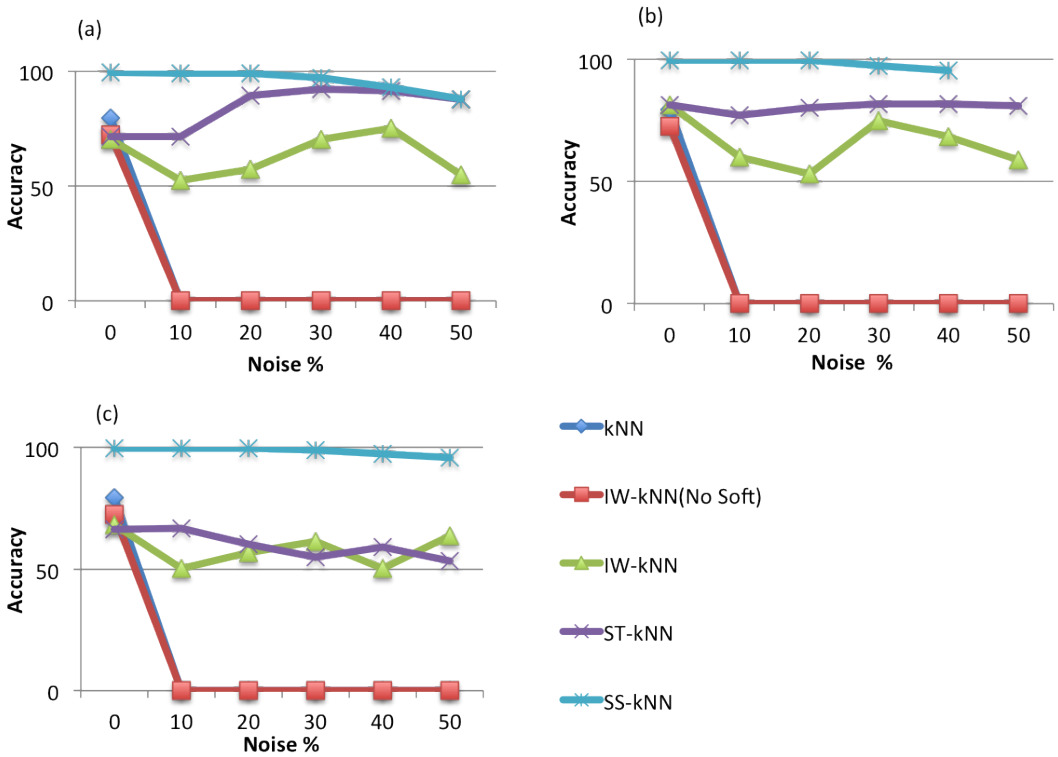

Figure 4.5. Optical digit: accuracy of k-NN based algorithms on the test set for 10 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft.

values indicate that adding softly labeled data increases accuracy. When the noise on softly labeled data increases, generally accuracy decreases.

We also evaluate the algorithms when 30 hardly labeled data are used. However, 30 hardly labeled data was sufficient to train the supervised algorithms individually (See Table C.22, Figures 4.8, 4.9, and 4.10). For example, k-NN achieves 96.68±1.47 accuracy and SS-kNN achieves 99.28±0.42 when the noise is 20%. It looks like SS-kNN performs better than the supervised k-NN in terms of accuracy, but these values are so close and it is not possible to infer the real effect of softly labeled data. Among all algorithms the instance weighted ones achieve 100% accuracy.

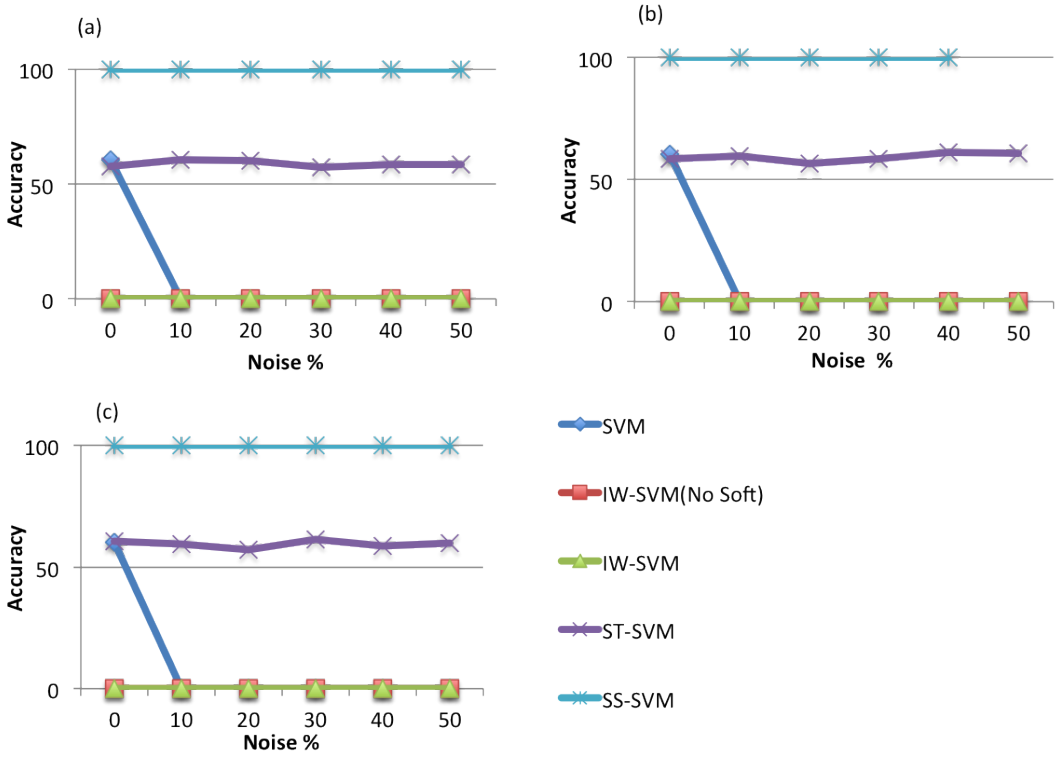

Figure 4.6. Optical digit: accuracy of SVM based algorithms with linear kernel on the test set for 10 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft.

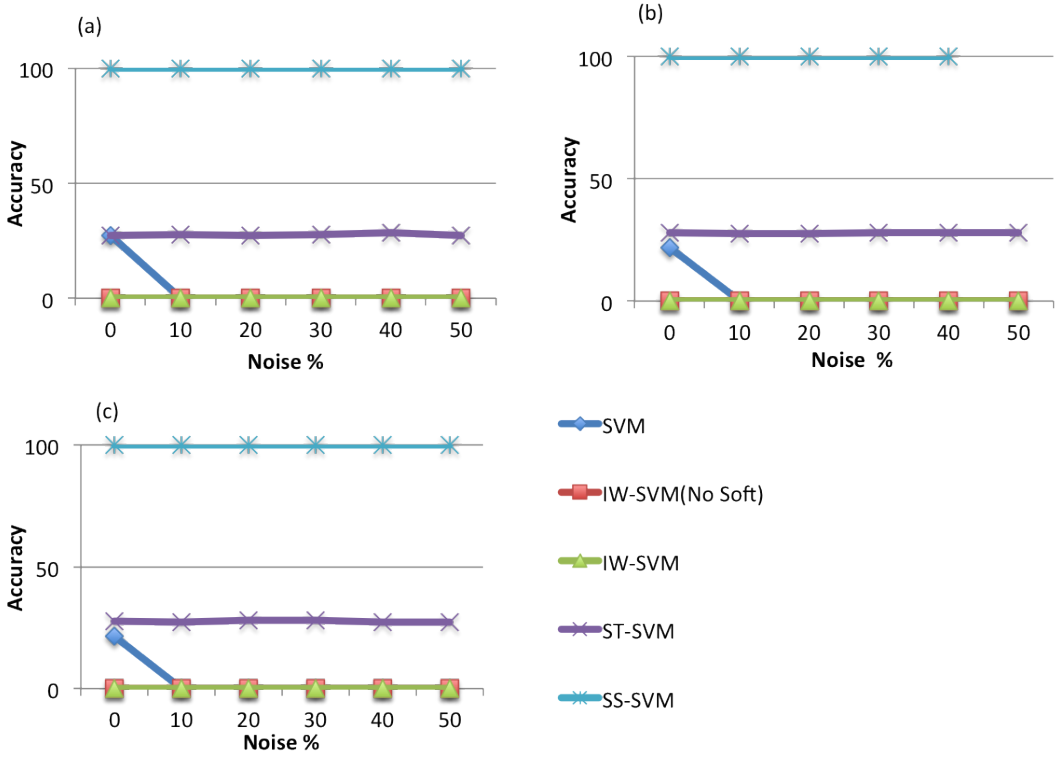

Figure 4.7. Optical digit: accuracy of SVM based algorithms with RBF kernel on the test set for 10 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft.

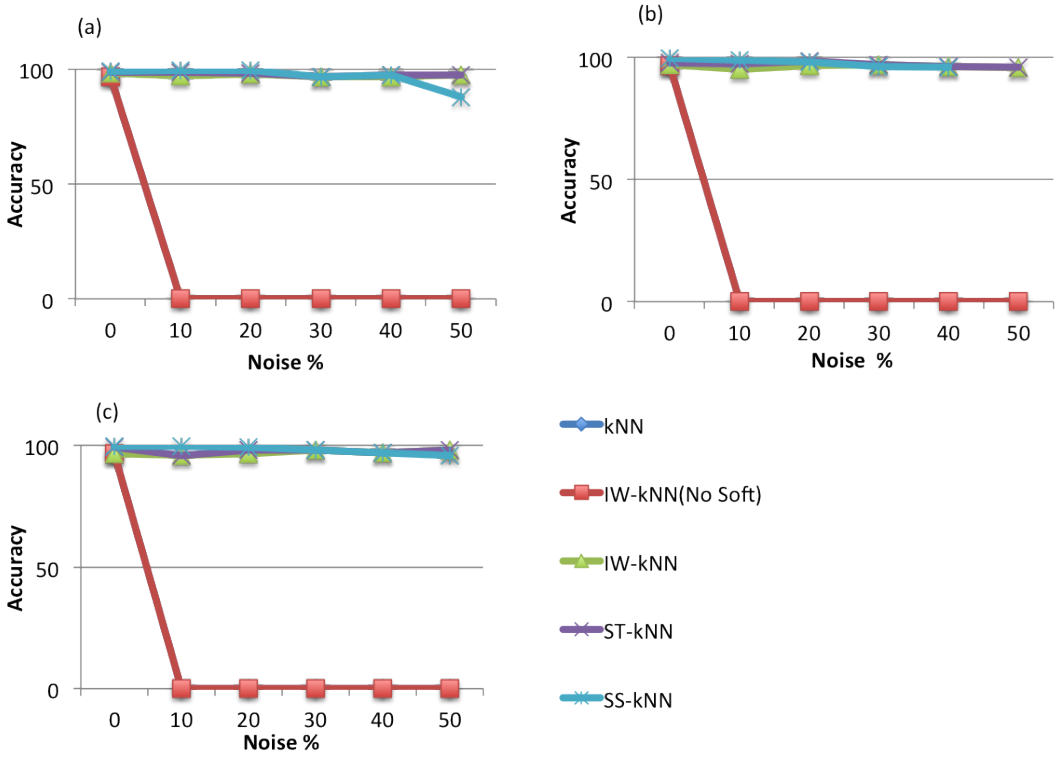

Figure 4.8. Optical digit: accuracy of k-NN based algorithms on the test set for 30 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft.

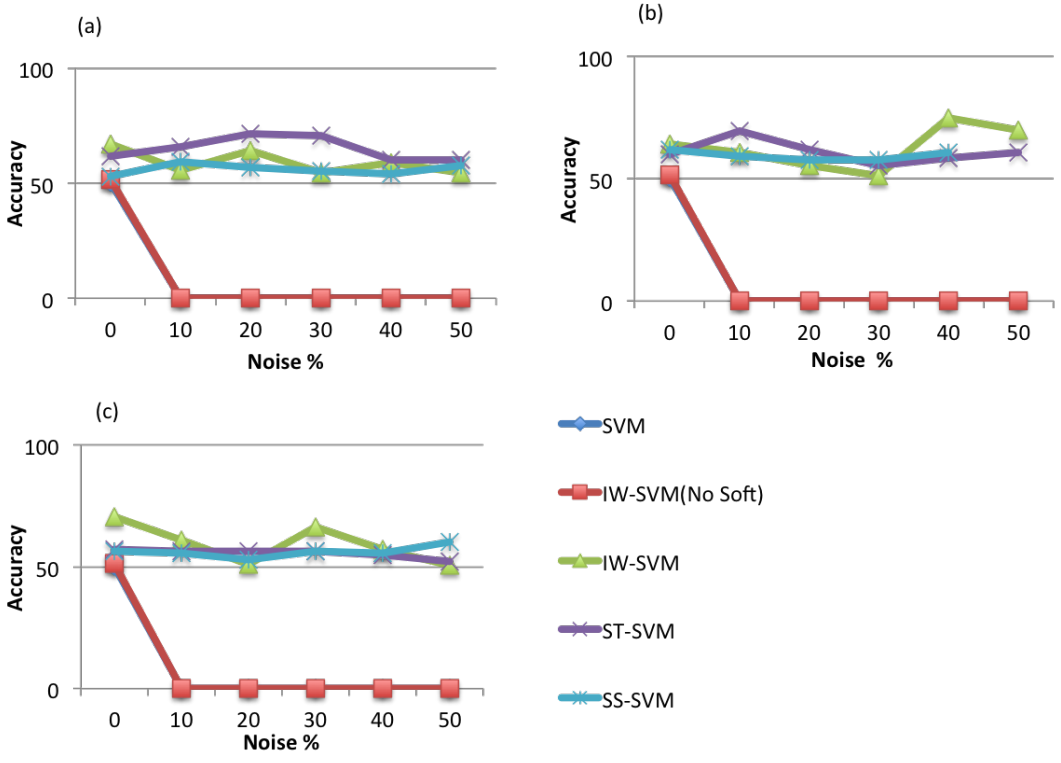

Figure 4.9. Optical digit: accuracy of SVM based algorithms with linear kernel on the test set for 30 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft.

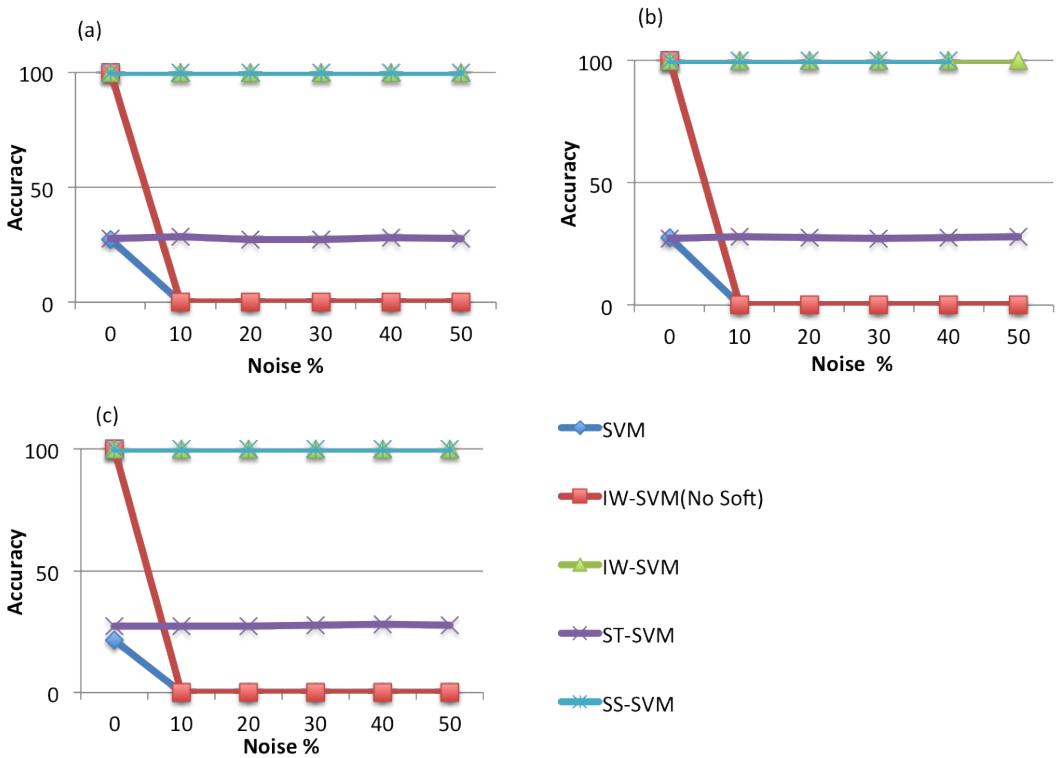

Figure 4.10. Optical digit: accuracy of SVM based algorithms with RBF kernel on the test set for 30 hardly labeled data: (a) 24 Soft, (b) 50 Soft, (c) 100 Soft.

5. CONCLUSION

Biological data involve many direct or indirect correlations, as well as noisy and missing values. Molecular biology experts interpret the data according to their knowledge. However, in addition to the challenge of interpreting noisy and missing data, also it may not be possible to manually detect some of the correlations. Therefore, various computational methods are being used to delve into the complex structures of biological data. One of the widely used computational approaches is Machine Learning, which includes three main classes of methods, namely, supervised, unsupervised, and semi-supervised. Supervised methods require labeled data, whereas unsupervised approaches only make use of unlabeled data. On the other hand, the goal of semi-supervised methods is to effectively utilize both labeled and unlabeled data. Obtaining labeled data in the biomedical domain is challenging in general, since it usually requires performing wet lab experiments or observations that span a long time. In addition, it is not often possible to label instances with certainty. For example, some types of experimental methods have high error rates, rendering the results highly uncertain. In this study, we introduce semi-supervised methods that make use of labeled and uncertainly labeled data, which we name as 'softly labeled' data. We apply these methods to two different problems in the biomedical domain: (i) predicting interactions between human and HIV-1 proteins, and (ii) predicting recurrence of colorectal cancer from gene expression data. In order to better understand the characteristics of the semi-supervised algorithms, we also apply them on a well-known data set in machine learning, namely the optical digit data set by synthetically introducing softly labeled data with varying amounts of noise. We compare the performances of the proposed semi-supervised algorithms with their supervised counterparts.

We describe the colorectal cancer recurrence prediction, HIV-1 and human protein protein interaction detection, and optical digit identification problems, as well as the datasets that we used in Chapter 2. In the colorectal cancer data, each instance represents a patient with a label 'recurrence within 5 years' or 'non-recurrence within 5 years'. In the HIV-1-Human PPI data, each instance indicates a protein pair with

a label 'interact' or 'not interact'. We use the optical digit data to understand the algorithmic behavior of our methods. In this dataset, we try to classify the digits '5' and '6'. We also give the definition of 'softly labeled' data, which means uncertainly labeled data that is highly likely to belong to the class that it is labeled with. We use cancer stage information from 1 to 3 of colorectal cancer patients as additional information on the unlabeled data that do not have sufficient follow-up information. The cancer recurrence risk increases from stage 1 to stage 3. Therefore, we expect most of the stage 1 patients to belong to the 'non-recurrence' class, and stage 3 patients to the 'recurrence' class. For the HIV-1-Human PPI dataset, we use the PPIs that are not experimentally proven, but have been curated from the literature and included in the NIAID database as softly labeled positive data, since they are highly likely to be positive interactions. Distinctively in this dataset, although there are hardly-labeled positive instances, which have been labeled by HIV-1 experts, there are no hardly labeled negative instances, since it is difficult to prove a non-interaction experimentally. The interactions that are not included in the NIAID database, or have not been shown experimentally by experts to exist are assumed to be negative, i.e. are labeled as 'not interact'. In the optical digit dataset, we gather softly labeled data by adding uniform noise to some of the labeled data.

We describe the implemented k-NN and SVM based supervised and semi-supervised methods in Chapter 3. The k-NN based semi-supervised method SS-kNN adds the entire softly labeled data into the training data by associating the instances with different reliability coefficient values between 0 and 1. The second method, self-training k-NN (ST-kNN), labels softly labeled data and adds the more confidently labeled predictions into the training set in a self-training manner. The third method, instance weighted k-NN (IW-kNN), assigns random weights (aka. reliability coefficient) for each labeled data and arranges these weights with a loss function iteratively. Then, it labels softly labeled data with the new weights in a self-training manner and continues to arrange the weights. The SVM based methods also have the same variations as SS-SVM, ST-SVM and IW-SVM. However, differently from the k-NN based methods, in the SVM based approaches, instead of assigning the reliability coefficients to the instances, we assign them to the hyper parameter C of the SVM algorithm. We use

Linear and RBF kernels for the SVM based methods.

We evaluate the performances of the proposed methods and give the results in Chapter 4. Our results reveal that, softly labeled data are really informative and increase the performance metrics which are 'accuracy' for colorectal cancer and optical digit datasets and 'area under the precision-recall curve' and 'break even value' for HIV-1-Human PPI dataset. Since there are no hardly labeled negative instances in HIV-Human PPI dataset, it is not reasonable to evaluate this dataset in terms of accuracy. Instead we use the area under the precision-recall curve (aka. MAP) value, since precision-recall curve better represents the success of the algorithm in terms of labeling true positives [35].

Among our semi-supervised methods SS-kNN and SS-SVM with linear and RBF kernels perform better than the supervised methods and the other semi-supervised methods for the colorectal cancer and optical digit data. SS-kNN and SS-SVM achieve very high MAP values for HIV-1-Human PPI data, which are better than the supervised methods by far. The supervised methods perform worse than SS-kNN and SS-SVM methods overall, since the number of labeled instances is very small. Thus, the training set could not model the true structure of the data.

ST-kNN performs similarly to the supervised k-NN method for the colorectal cancer, HIV-1-Human PPI, and optical digit data. ST-SVM with RBF kernel performs better than the supervised methods for the colorectal cancer and HIV-1-Human PPI data. This algorithm obtains similar performance with supervised methods for the optical digit data with 10 hardly labeled instances. ST-SVM with linear kernel performs slightly worse than the supervised methods for colorectal cancer. This poor performance suggests that the colorectal cancer data might not be linearly separable. On the other hand, ST-SVM with linear kernel performs similar to the supervised methods for the HIV-1-Human PPI and optical digit data.

IW-kNN achieves higher performance than supervised k-NN, but performs poorly for the SVM based methods. The reason might be the insufficient amount of data to

arrange the coefficients of the parameter C more accurately. For the HIV-1-Human data, while IW-kNN performs worse than the supervised method, IW-SVM with linear and RBF kernels achieves better MAP value than the supervised methods. The IW-kNN method performs similarly to the supervised based methods for the optical digit data. The IW-SVM methods could not produce any result for 10 hardly labeled optical digit data. On the other hand, IW-kNN and IW-SVM achieve better MAP values than the supervised methods for 30 hardly labeled data on the optical digit data. IW-SVM with linear kernel achieves similar performance with supervised methods.

As future work, we will investigate designing a kernel, which meets the demands of a binary classification problem with very few instances for SVM based algorithms. Moreover, different similarity functions may be applied or a problem specific new similarity function may be designed. Nowadays, graph based solutions are very popular and promising. Therefore, a graph structure may be designed for the problems and the existing or new graph algorithms may be applied on this graph.

APPENDIX A: Colorectal Cancer Data Results

Table A.1. Colorectal cancer: accuracy values on test data with various parameters of all algorithms.

	Accuracy
k-NN	50.43±6.59
IW-kNN (No Soft)	49.57±3.89
SS-kNN	65.22±3.07
ST-kNN	60.87±-10.65
IW-kNN	66.96±3.89
SVM L	21.74±3.07
IW-SVM L (No Soft)	21.74±3.07
SS-SVM L	60.87±4.35
ST-SVM L	25.22±6.45
IW-SVM L	51.3±3.64
SVM R	87.83±27.22
IW-SVM R (No Soft)	50±5.67
SS-SVM R	100+-0
ST-SVM R	87.83±27.22
IW-SVM R	38.26±5.67

Table A.2. Colorectal cancer: accuracy values on validation data with various parameters of all algorithms.

	Accuracy
k-NN	68.33±9.13
IW-kNN (No Soft)	66.67±10.21
SS-kNN	63.33±9.5
ST-kNN	66.67±10.21
IW-kNN	51.67±3.73
SVM L	63.33±7.45
IW-SVM L (No Soft)	63.33±7.45
SS-SVM L	48.33±6.97
ST-SVM L	56.67±6.97
IW-SVM L	56.67±6.97
SVM R	50±0
IW-SVM R (No Soft)	50±0
SS-SVM R	50±0
ST-SVM R	50±0
IW-SVM R	50±0

Table A.3. Colocteral cancer: k-NN results on 23 test instances with various decision thresholds.

		Decision Threshold				
		0.5	**0.4**	**0.3**	**0.2**	**0.1**
k-NN	**Disgarded**	0±0	0±0	8.2±2.05	8.2±2.05	8.2±2.05
	Accuracy	50.43±6.59	50.43±6.59	45.38±6.65	45.38±6.65	45.38±6.65
	PR-AUC	0.46±0.04	0.46±0.04	0.37±0.11	0.37±0.11	0.37±0.11
	ROC-AUC	0.49±0.05	0.49±0.05	0.46±0.07	0.46±0.07	0.46±0.07
	BE	0.58±0.12	0.58±0.12	0.83±0.09	0.83±0.09	0.83±0.09
IW-kNN(No Soft)	**Disgarded**	0±0	1.4±1.14	2.6±1.34	4±1.22	5.8±2.05
	Accuracy	49.57±3.89	49.17±3.35	49.09±3.23	46.41±2.79	45.45±2.17
	PR-AUC	0.51±0.02	0.47±0.05	0.44±0.05	0.47±0.03	0.49±0.03
	ROC-AUC	0.51±0.02	0.48±0.04	0.47±0.06	0.5±0.03	0.52±0.04
	BE	0.53±0.05	0.49±0.07	0.49±0.08	0.57±0.03	0.65±0.09
SS-kNN	**Disgarded**	0±0	0±0	0±0	0±0	0±0
	Accuracy	65.22±3.07	65.22±3.07	65.22±3.07	65.22±3.07	65.22±3.07
	PR-AUC	0.19±0.02	0.19±0.02	0.19±0.02	0.19±0.02	0.19±0.02
	ROC-AUC	0.65±0.03	0.65±0.03	0.65±0.03	0.65±0.03	0.65±0.03
	BE	0.68±0.04	0.68±0.04	0.68±0.04	0.68±0.04	0.68±0.04
ST-kNN	**Disgarded**	0±0	0±0	9±1	9±1	9±1
	Accuracy	60.87±10.65	60.87±10.65	58.4±11.78	58.4±11.78	58.4±11.78
	PR-AUC	0.25±0.1	0.25±0.1	0.05±0.12	0.05±0.12	0.05±0.12
	ROC-AUC	0.59±0.1	0.59±0.1	0.56±0.1	0.56±0.1	0.56±0.1
	BE	0.63±0.1	0.63±0.1	0.97±0.06	0.97±0.06	0.97±0.06
IW-kNN	**Disgarded**	0±0	3.6±2.07	7±2.92	9.6±2.88	12.4±1.52
	Accuracy	66.96±3.89	63.99±5.62	60.86±5.73	57.33±7.1	54.05±9.21
	PR-AUC	0.45±0.06	0.42±0.07	0.37±0.1	0.32±0.09	0.24±0.12
	ROC-AUC	0.58±0.09	0.51±0.09	0.48±0.14	0.47±0.13	0.54±0.13
	BE	0.6±0.07	0.53±0.1	0.47±0.11	0.53±0.14	0.59±0.15

Table A.4. Colocteral cancer: SVM with linear kernel results on 23 test instances with various decision thresholds.

		Decision Threshold				
		0.5	**0.4**	**0.3**	**0.2**	**0.1**
SVM	**Disgarded**	0±0	2.2±0.84	3.6±1.14	6.2±1.48	9.8±2.17
	Accuracy	21.74±3.07	18.26±1.93	19.62±2.45	16.51±3.86	7.76±1.42
	PR-AUC	0.34±0.01	0.35±0.02	0.34±0.03	0.34±0.01	0.26±0.03
	ROC-AUC	0.12±0.03	0.1±0.04	0.09±0.04	0.06±0.04	0.03±0.02
	BE	0.25±0	0.25±0.06	0.23±0.07	0.21±0.06	0.03±0.07
IW-SVM No Soft	**Disgarded**	0±0	2.2±0.84	3.6±1.14	6.2±1.48	9.8±2.17
	Accuracy	21.74±3.07	18.26±1.93	19.62±2.45	16.51±3.86	7.76±1.42
	PR-AUC	0.34±0.01	0.35±0.02	0.34±0.03	0.34±0.01	0.26±0.03
	ROC-AUC	0.12±0.03	0.1±0.04	0.09±0.04	0.06±0.04	0.03±0.02
	BE	0.25±0	0.25±0.06	0.23±0.07	0.21±0.06	0.03±0.07
SS-SVM	**Disgarded**	0±0	1±0.71	1.6±0.55	1.8±0.84	4.2±3.83
	Accuracy	60.87±4.35	60.09±5.01	58.92±5.72	58.54±5.53	66.9±19.06
	PR-AUC	0.73±0.12	0.73±0.13	0.74±0.13	0.74±0.13	0.76±0.14
	ROC-AUC	0.73±0.12	0.73±0.13	0.75±0.14	0.75±0.14	0.79±0.2
	BE	0.65±0.04	0.64±0.05	0.67±0.07	0.67±0.07	0.73±0.17
ST-SVM	**Disgarded**	0±0	0.6±0.55	1.8±0.84	3.4±1.14	5.4±1.14
	Accuracy	25.22±6.45	24.07±6.42	23.52±8.29	23.51±7.53	20.38±4.45
	PR-AUC	0.35±0.01	0.34±0.02	0.35±0.02	0.33±0.03	0.32±0.01
	ROC-AUC	0.14±0.05	0.13±0.05	0.12±0.06	0.1±0.04	0.08±0.03
	BE	0.3±0.07	0.28±0.09	0.28±0.08	0.23±0.09	0.18±0.07
IW-SVM	**Disgarded**	0±0	3±2	5.4±2.19	6.2±1.92	9.2±1.64
	Accuracy	51.3±3.64	45.65±7.64	44.25±4.47	42.89±6.25	36.23±5.59
	PR-AUC	0.48±0.05	0.46±0.04	0.42±0.08	0.43±0.08	0.42±0.13
	ROC-AUC	0.43±0.05	0.43±0.07	0.43±0.06	0.44±0.07	0.44±0.09
	BE	0.5±0.06	0.43±0.09	0.36±0.08	0.38±0.1	0.36±0.1

Table A.5. Colocteral cancer: SVM with RBF kernel results on 23 test instances with various decision thresholds.

		Decision Threshold				
		0.5	**0.4**	**0.3**	**0.2**	**0.1**
SVM	**Disgarded**	0±0	21±4.47	21±4.47	21±4.47	21±4.47
	Accuracy	87.83±27.22	0±0	0±0	0±0	0±0
	PR-AUC	0.7±0.12	0.1±0.21	0.1±0.21	0.1±0.21	0.1±0.21
	ROC-AUC	0.92±0.18	0±0	0±0	0±0	0±0
	BE	0.9±0.22	0±0	0±0	0±0	0±0
IW-SVM No Soft	**Disgarded**	0±0	13.4±1.14	13.4±1.14	13.4±1.14	13.4±1.14
	Accuracy	38.26±5.67	91.46±4.94	91.46±4.94	91.46±4.94	91.46±4.94
	PR-AUC	0.53±0.06	0.53±0.1	0.53±0.1	0.53±0.1	0.53±0.1
	ROC-AUC	0.62±0.08	0.87±0.07	0.87±0.07	0.87±0.07	0.87±0.07
	BE	0.55±0.07	0.73±0.18	0.73±0.18	0.73±0.18	0.73±0.18
SS-SVM	**Disgarded**	0±0	18.4±10.29	23±0	23±0	23±0
	Accuracy	100±0	0±0	0±0	0±0	0±0
	PR-AUC	0.68±0.2	0.07±0.15	0±0	0±0	0±0
	ROC-AUC	1±0	0±0	0±0	0±0	0±0
	BE	1±0	0±0	0±0	0±0	0±0
ST-SVM	**Disgarded**	0±0	21±4.47	21±4.47	21±4.47	21±4.47
	Accuracy	87.83±27.22	0±0	0±0	0±0	0±0
	PR-AUC	0.7±0.12	0.1±0.21	0.1±0.21	0.1±0.21	0.1±0.21
	ROC-AUC	0.92±0.18	0±0	0±0	0±0	0±0
	BE	0.9±0.22	0±0	0±0	0±0	0±0
IW-SVM	**Disgarded**	0±0	13.4±1.14	13.4±1.14	13.4±1.14	13.4±1.14
	Accuracy	38.26±5.67	91.46±4.94	91.46±4.94	91.46±4.94	91.46±4.94
	PR-AUC	0.51±0.04	0.45±0.08	0.45±0.08	0.45±0.08	0.45±0.08
	ROC-AUC	0.62±0.08	0.87±0.07	0.87±0.07	0.87±0.07	0.87±0.07
	BE	0.55±0.07	0.73±0.18	0.73±0.18	0.73±0.18	0.73±0.18

Table A.6. Colocteral cancer: k-NN results on validation set.

k-NN	**Accuracy**	68.3333±9.1287
	PR-AUC	0.57569±0.12
	ROC-AUC	0.70444±0.13937
	BE	0.7±0.18257
IW-kNN No Soft	**Accuracy**	66.6667±10.2062
	PR-AUC	0.23846±0.12902
	ROC-AUC	0.66667±0.10206
	BE	0.7±0.074536
SS-kNN	**Accuracy**	63.3333±9.5015
	PR-AUC	0.4508±0.11411
	ROC-AUC	0.69278±0.12172
	BE	0.7±0.13944
ST-kNN	**Accuracy**	66.6667±10.2062
	PR-AUC	0.68784±0.099993
	ROC-AUC	0.72611±0.13066
	BE	0.7±0.11785
IW-kNN	**Accuracy**	51.67±3.73
	PR-AUC	0.02±0.04
	ROC-AUC	0.52±0.09
	BE	0.6±0.07

Table A.7. Colocteral cancer: SVM with linear kernel results on validation set.

SVM L	**Accuracy**	63.33±7.45
	PR-AUC	0.63±0.05
	ROC-AUC	0.65±0.04
	BE	0.6±0.09
IW-SVM L No Soft	**Accuracy**	63.33±7.45
	PR-AUC	0.63±0.05
	ROC-AUC	0.65±0.04
	BE	0.6±0.09
SS-SVM L	**Accuracy**	48.33±6.97
	PR-AUC	0.55±0.08
	ROC-AUC	0.53±0.16
	BE	0.5±0.07
ST-SVM L	**Accuracy**	56.67±6.97
	PR-AUC	0.59±0.05
	ROC-AUC	0.61±0.03
	BE	0.53±0.07
IW-SVM L	**Accuracy**	56.67±6.97
	PR-AUC	0.53±0.1
	ROC-AUC	0.52±0.15
	BE	0.47±0.17

Table A.8. Colocteral cancer: SVM with RBF kernel results on validation set.

SVM R	**Accuracy**	50±0
	PR-AUC	0.4±0
	ROC-AUC	0.5±0
	BE	0.6±0
IW-SVM R No Soft	**Accuracy**	50±0
	PR-AUC	0.4±0
	ROC-AUC	0.5±0
	BE	0.6±0
SS-SVM R	**Accuracy**	50±0
	PR-AUC	0.4±0
	ROC-AUC	0.5±0
	BE	0.6±0
ST-SVM R	**Accuracy**	50±0
	PR-AUC	0.4±0
	ROC-AUC	0.5±0
	BE	0.6±0
IW-SVM R	**Accuracy**	50±0
	PR-AUC	0.4±0
	ROC-AUC	0.5±0
	BE	0.6±0

Table A.9. The parameters of colorectal cancer k-NN results.

k-NN	k=3
IW-kNN No Soft	k=5
SS-kNN	k=1,Stage1(+),Stage2(+),Stage3(+)
ST-kNN	k=3,Stage2(+), Threshold=0.9, iterate until finish
IW-kNN	k=3,Stage1(+),Stage2(+)

Table A.10. The parameters of colorectal cancer SVM with linear kernel results.

SVM L	C=0.25
IW-SVM L No Soft	C=0.25
SS-SVM L	C=0.125,θ_{stage1}=1(+),θ_{stage3}=1(+)
ST-SVM L	C=0.125,Stage2(+), Threshold=0.7, iterate
IW-SVM L	C=1,Stage2(+),Stage3(+)

Table A.11. The parameters of colorectal cancer SVM with RBF kernel results.

SVM R	C=0.125,γ=0.03125
IW-SVM R No Soft	C=0.125,γ=0.03125
SS-SVM R	C=0.125,γ=0.03125,θ_{stage1}=1(-),θ_{stage3} = 0.6(+)
ST-SVM R	C=0.125,γ=0.03125,Stage2(+), Threshold=0.9, iterate
IW-SVM R	C=0.125,γ=0.03125,Stage2(+),Stage3(+)

APPENDIX B: HIV-1 and Human PPI Data Results

Table B.1. HIV-1-human PPI: PR-AUC values on test data with various parameters of all algorithms.

	PR-AUC
k-NN	0.42±0.1
WI-kNN (No Soft)	0.28±0.1
SS-kNN	0.34±0.1
ST-kNN	0.41±0.09
WI-kNN	0.23±0.05
SVM Linear	0.05±0.03
WI-SVM Linear (No Soft)	0.43±0.1
SS-SVM Linear	0.94±0
ST-SVM Linear	0.45±0.11
WI-SVM Linear	0.44±0.1
SVM RBF	0.08±0.04
WI-SVM RBF (No Soft)	0.39±0.12
SS-SVM RBF	0.82±0.07
ST-SVM RBF	0.44±0.13
WI-SVM RBF	0.39±0.11

Table B.2. HIV-1-human PPI: k-NN results on test set with various decision thresholds.

		Decision Threshold				
		0.5	**0.4**	**0.3**	**0.2**	**0.1**
k-NN	**Disgarded**	0±0	27.3±4.9	27.3±4.9	39.7±6.38	52±7.82
	Accuracy	79.81±3.14	89.01±2.47	89.01±2.47	93.35±1.45	96.98±1.61
	PR-AUC	0.42±0.1	0.45±0.1	0.45±0.1	0.47±0.12	0.44±0.15
	ROC-AUC	0.85±0.06	0.85±0.06	0.85±0.06	0.85±0.07	0.79±0.09
	BE	0.55±0.08	0.59±0.11	0.59±0.11	0.63±0.12	0.71±0.2
IW-kNN No Soft	**Disgarded**	0±0	1.7±1.06	5.1±2.6	10.7±3.3	24.6±3.1
	Accuracy	93.87±0.68	94±0.67	94.21±0.8	94.72±1.08	95.61±0.87
	PR-AUC	0.28±0.1	0.25±0.09	0.23±0.09	0.19±0.05	0.23±0.08
	ROC-AUC	0.82±0.06	0.81±0.07	0.81±0.07	0.8±0.07	0.79±0.09
	BE	0.29±0.12	0.26±0.1	0.24±0.08	0.2±0.08	0.28±0.12
SS-kNN	**Disgarded**	0±0	10±3.74	22.3±6.06	31.3±6.75	33.2±6.97
	Accuracy	92.97±1.21	94.04±1.33	95.53±1.23	96.1±1.19	96.44±1.11
	PR-AUC	0.34±0.07	0.33±0.08	0.27±0.1	0.29±0.09	0.33±0.07
	ROC-AUC	0.77±0.07	0.75±0.06	0.69±0.07	0.65±0.06	0.64±0.05
	BE	0.44±0.08	0.46±0.11	0.41±0.13	0.61±0.19	0.78±0.12
ST-kNN	**Disgarded**	0±0	7.6±2.27	15.9±3.07	27.1±4.95	38.8±4.89
	Accuracy	90.5±2.35	92.33±2.45	93.24±2.69	94.1±2.56	95.7±2.04
	PR-AUC	0.37±0.12	0.38±0.13	0.39±0.13	0.41±0.14	0.44±0.16
	ROC-AUC	0.82±0.07	0.82±0.07	0.81±0.07	0.81±0.08	0.79±0.09
	BE	0.48±0.09	0.5±0.08	0.53±0.11	0.54±0.1	0.68±0.17
IW-kNN	**Disgarded**	0±0	0.1±0.32	1.4±1.43	3.6±2.8	9.7±3.65
	Accuracy	93.5±0.16	93.5±0.17	93.67±0.26	93.82±0.3	94.13±0.38
	PR-AUC	0.23±0.05	0.23±0.05	0.21±0.06	0.22±0.06	0.25±0.06
	ROC-AUC	0.75±0.07	0.75±0.07	0.75±0.07	0.75±0.07	0.74±0.07
	BE	0.24±0.07	0.25±0.06	0.24±0.08	0.24±0.09	0.32±0.1

Table B.3. HIV-1-human PPI: SVM with linear kernel results on test set with various decision thresholds.

		Decision Threshold				
		0.5	**0.4**	**0.3**	**0.2**	**0.1**
SVM	**Disgarded**	0±0	2.9±0.99	5.8±1.93	9.9±3.18	23.9±4.89
	Accuracy	2.1±0.86	1.79±0.81	1.52±0.8	1.28±0.78	0.96±0.59
	PR-AUC	0.05±0.03	0.05±0.03	0.05±0.04	0.05±0.04	0.06±0.06
	ROC-AUC	0.08±0.06	0.09±0.07	0.1±0.08	0.1±0.08	0.15±0.13
	BE	0.06±0.06	0.06±0.07	0.06±0.07	0.07±0.08	0.1±0.12
IW-SVM No Soft	**Disgarded**	0±0	0.7±0.82	1.7±1.25	5.2±2.7	16.8±7.07
	Accuracy	94.04±0.62	93.96±0.61	93.96±0.61	94±0.55	93.91±0.42
	PR-AUC	0.39±0.12	0.35±0.11	0.31±0.13	0.26±0.14	0.13±0.09
	ROC-AUC	0.87±0.06	0.87±0.06	0.86±0.07	0.86±0.06	0.82±0.09
	BE	0.48±0.12	0.46±0.12	0.47±0.13	0.47±0.09	0.38±0.14
SS-SVM	**Disgarded**	0±0	24.6±20.45	156±12.4	213.6±7.11	233.5±4.67
	Accuracy	99.59±0.78	99.8±0.36	100±0	99.68±1.02	70.14±27.58
	PR-AUC	0.94±0	0.93±0	0.92±0.01	0.88±0.03	0.53±0.29
	ROC-AUC	1±0	1±0	1±0	1±0	0.85±0.24
	BE	1±0	0.99±0.03	1±0	0.99±0.03	0.71±0.42
ST-SVM	**Disgarded**	0±0	1.8±0.79	3.9±0.88	7.2±1.62	15.5±4.06
	Accuracy	93.96±0.55	94.41±0.61	94.82±0.56	95.42±0.49	96.44±0.68
	PR-AUC	0.45±0.11	0.42±0.11	0.44±0.12	0.39±0.11	0.34±0.11
	ROC-AUC	0.88±0.07	0.88±0.07	0.88±0.07	0.87±0.08	0.85±0.09
	BE	0.48±0.14	0.45±0.15	0.48±0.14	0.44±0.14	0.41±0.07
IW-SVM	**Disgarded**	0±0	1.2±1.14	2.4±1.71	5.2±4.02	16±8.1
	Accuracy	94.12±0.61	94.16±0.72	94±0.67	93.96±0.58	93.91±0.57
	PR-AUC	0.44±0.1	0.44±0.11	0.42±0.11	0.39±0.12	0.32±0.09
	ROC-AUC	0.88±0.06	0.87±0.07	0.87±0.07	0.86±0.08	0.83±0.09
	BE	0.47±0.12	0.47±0.11	0.45±0.11	0.42±0.11	0.41±0.12

Table B.4. HIV-1-human PPI: SVM with RBF kernel results on test set with various decision thresholds.

		Decision Threshold				
		0.5	**0.4**	**0.3**	**0.2**	**0.1**
SVM	**Disgarded**	0±0	1.7±0.95	3.6±1.17	7.1±2.38	16.2±3.08
	Accuracy	2.14±0.84	1.95±0.87	1.71±0.8	1.27±0.72	0.97±0.68
	PR-AUC	0.08±0.04	0.08±0.04	0.08±0.05	0.08±0.06	0.09±0.06
	ROC-AUC	0.1±0.07	0.1±0.07	0.1±0.08	0.12±0.09	0.15±0.1
	BE	0.09±0.07	0.09±0.07	0.1±0.08	0.11±0.09	0.14±0.1
IW-SVM No Soft	**Disgarded**	0±0	3.7±1.49	7.2±2.66	13.4±3.95	25.7±5.29
	Accuracy	94.04±1.13	94.16±1.04	94.08±0.89	94.12±0.7	93.67±0.28
	PR-AUC	0.39±0.11	0.38±0.12	0.38±0.1	0.3±0.12	0.12±0.09
	ROC-AUC	0.85±0.06	0.85±0.06	0.84±0.06	0.82±0.07	0.74±0.14
	BE	0.43±0.11	0.43±0.12	0.45±0.1	0.36±0.12	0.14±0.16
SS-SVM	**Disgarded**	0±0	41.2±22.74	85.3±19.92	162.1±25.85	173.3±19.42
	Accuracy	80.55±10.07	82.44±9.88	74.06±10.32	49.49±21.59	8.87±6.02
	PR-AUC	0.82±0.07	0.77±0.07	0.65±0.12	0.35±0.13	0±0
	ROC-AUC	0.98±0.02	0.97±0.03	0.94±0.05	0.85±0.1	0±0
	BE	0.87±0.07	0.82±0.1	0.69±0.1	0.54±0.12	0±0
ST-SVM	**Disgarded**	0±0	4.3±2.06	10.6±3.13	15.9±3.98	24.7±4.22
	Accuracy	93.71±1.43	94.52±1.49	95.62±1.26	96.18±1	97.12±0.91
	PR-AUC	0.44±0.13	0.42±0.11	0.4±0.15	0.3±0.12	0.23±0.15
	ROC-AUC	0.89±0.04	0.89±0.04	0.87±0.06	0.85±0.06	0.82±0.09
	BE	0.47±0.13	0.45±0.1	0.45±0.13	0.36±0.13	0.34±0.16
IW-SVM	**Disgarded**	0±0	3.7±1.49	7.2±2.66	13.4±3.95	25.7±5.29
	Accuracy	94.04±1.13	94.16±1.04	94.08±0.89	94.12±0.7	93.67±0.28
	PR-AUC	0.39±0.11	0.38±0.12	0.38±0.1	0.3±0.12	0.12±0.09
	ROC-AUC	0.85±0.06	0.85±0.06	0.84±0.06	0.82±0.07	0.74±0.14
	BE	0.43±0.11	0.43±0.12	0.45±0.1	0.36±0.12	0.14±0.16

Table B.5. The parameters of HIV-1-human PPI results.

k-NN	k=9,θ_+=1,θ_-=0.1
IW-kNN No Soft	k=9,θ_+=Random,θ_-=Random
SS-kNN	k=9,θ_+=1,θ_-=0.5,θ_{soft}=0.001
ST-kNN	k=9,θ_+=1,θ_-=0.1,θ_{soft}=0.01
IW-kNN	k=9,θ_+=Random,θ_-=Random,θ_{soft}=Random
SVM L	C=2, θ_+=1,θ_-=0.5
IW-SVM L No Soft	C=8,θ_+=Random,θ_-=Random
SS-SVM L	C=2, θ_+=1,θ_-=0.5, θ_{soft}=1
ST-SVM L	C=8, θ_+=1,θ_-=0.5, θ_{soft}=0.01
IW-SVM L	C=8, θ_+=Random,θ_-=Random,θ_{soft}=Random
SVM R	C=2,γ=0.015625, θ_+=1,θ_-=1
IW-SVM R No Soft	C=8,γ=1, θ_+=Random,θ_-=Random
SS-SVM R	C=2,γ=0.015625, θ_+=1,θ_-=0.1, θ_{soft}=1
ST-SVM R	C=8,γ=1, θ_+=1,θ_-=0.1, θ_{soft}=1
IW-SVM R	C=8,γ=1, θ_+=Random,θ_-=Random,θ_{soft}=Random

APPENDIX C: Optical Digit Data Results

Table C.1. The parameters of optical digit results on test set: 10 hard no soft.

k-NN	k=1
IW-kNN	k=1
SVM L	C=1
SVM R	C=1 γ=1
IW-SVM L	NA
IW-SVM R	NA

Table C.2. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
No Soft	**k-NN**	79.39±10.54	-	-	-	-	-
	IW-kNN	91.45±13.99	-	-	-	-	-
	SVM L	60.39±8.64	-	-	-	-	-
	IW-SVM L	0±0	-	-	-	-	-
	SVM R	27.16±2.07	-	-	-	-	-
	IW-SVM R	0±0	-	-	-	-	-

Table C.3. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
24 Soft	**SS-kNN**	99.45±0	99.17±0	98.9±0	96.97±0	92.84±0	87.88±0
	ST-kNN	71.63±0	71.35±0	89.26±0	92.29±0	91.18±0	87.88±0
	IW-kNN	70.14±14.47	52.4±13.37	57.14±13.94	70.25±0	74.93±0	54.82±12.42
	SS-SVM L	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM L	57.8±21.02	60.33±16.24	60.17±17.69	57.14±21.84	58.62±20.79	58.62±20.74
	IW-SVM L	NA	NA	NA	NA	NA	NA
	SS-SVM R	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM R	27.38±2.15	27.77±1.62	27.27±1.08	27.6±1.06	28.32±1.96	27.27±1.47
	IW-SVM R	NA	NA	NA	NA	NA	NA

Table C.4. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
50 Soft	**SS-kNN**	99.45±0	99.45±0	99.45±0	99.72±0	97.52±0	95.43±0.15
	ST- kNN	91.18±0	74.66±0	64.74±0	77.14±0	52.89±0	92.84±0
	IW-kNN	81.27±0	60±6.83	53.06±4.11	74.77±14.18	68.32±0	58.95±8.93
	SS-SVM L	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM L	58.46±7.46	59.78±11.54	56.36±19.94	58.46±21.14	60.99±22.24	60.72±8.65
	IW-SVM L	NA	NA	NA	NA	NA	NA
	SS-SVM R	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM R	27.88±1.48	27.71±1.89	27.49±1.84	27.88±2	27.77±1.9	27.77±2.17
	IW-SVM R	NA	NA	NA	NA	NA	NA

Table C.5. Optical digit: accuracy values on test data with various parameters of all algorithms for 10 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
100 Soft	**SS-kNN**	99.72±0	99.72±0	99.56±0.15	98.79±0.42	97.52±0	95.87±0
	ST-kNN	66.39±0	66.67±0	60.33±0	54.82±0	58.95±0	53.17±0
	IW-kNN	68.32±0	50.14±0	56.91±7.63	61.49±9.88	50.19±0.74	63.8±5.22
	SS-SVM L	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM L	60.5±8.61	59.45±17.72	57.14±21.84	61.54±19.91	58.68±26.14	59.89±24.36
	IW-SVM L	NA	NA	NA	NA	NA	NA
	SS-SVM R	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM R	27.77±1.13	27.38±0.79	28.1±1.38	27.99±0.77	27.33±2.12	27.11±1.91
	IW-SVM R	NA	NA	NA	NA	NA	NA

Table C.6. Optical digit results on test set: 10 hard no soft.

k-NN	**Accuracy**	79.39±10.54
	PR-AUC	0.08±0.07
	ROC-AUC	0.79±0.11
	Break Even	0.99±0.01
	Break Even(odd)	0.61±0.2
IW-kNN	**Accuracy**	91.45±13.99
	PR-AUC	0±0
	ROC-AUC	0.3±0.27
	Break Even	0.6±0.55
	Break Even(odd)	0.88±0.21
SVM L	**Accuracy**	60.39±8.64
	PR-AUC	0.61±0.11
	ROC-AUC	0.67±0.12
	Break Even	0.62±0.1
	Break Even(odd)	1±0
SVM R	**Accuracy**	27.16±2.07
	PR-AUC	0±0
	ROC-AUC	0.5±0
	Break Even	0.01±0
	Break Even(odd)	0.01±0
IW-SVM L	**Accuracy**	0±0
	PR-AUC	0±0
	ROC-AUC	0±0
	Break Even	0±0
	Break Even(odd)	1±0
IW-SVM R	**Accuracy**	0±0
	PR-AUC	0±0
	ROC-AUC	0±0
	Break Even	0±0
	Break Even(odd)	0±0

Table C.7. Optical digit: k-NN results on test set: 10 hard 24 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-kNN	**Accuracy**	99.45±0	99.17±0	98.9±0	96.97±0	92.84±0	87.88±0
	PR-AUC	0.05±0	0.28±0	0.46±0	0.91±0	0.89±0	0.86±0
	ROC-AUC	1±0	1±0	1±0	0.99±0	0.95±0	0.94±0
	Break Even(even)	0.99±0	0.99±0	0.99±0	0.98±0	0.94±0	0.94±0
	Break Even(odd)	1±0	0.99±0	1±0	0.97±0	0.98±0	0.87±0
ST-kNN	**Accuracy**	71.63±0	71.35±0	89.26±0	92.29±0	91.18±0	87.88±0
	PR-AUC	0.02±0.01	0.29±0.11	0.16±0.15	0.01±0.01	0.12±0.05	0.01±0.01
	ROC-AUC	0.79±0	0.77±0	0.99±0	0.97±0	0.97±0	0.94±0
	Break Even(even)	0.86±0	0.8±0	0.98±0	0.98±0	0.96±0	0.94±0
	Break Even(odd)	0.72±0	0.84±0	0.96±0	0.88±0	0.91±0	0.87±0
IW-kNN	**Accuracy**	70.14±14.47	52.4±13.37	57.14±13.94	70.25±0	74.93±0	54.82±12.42
	PR-AUC	0.61±0.19	0.49±0.25	0.52±0.09	0.05±0	0.02±0	0.49±0.33
	ROC-AUC	0.7±0.24	0.41±0.33	0.66±0.09	0.7±0	0.75±0	0.82±0.06
	Break Even(even)	0.66±0.19	0.42±0.26	0.63±0.1	0.91±0	0.97±0	0.79±0.05
	Break Even(odd)	0.97±0.06	1±0	1±0	0.58±0	0.55±0	1±0

Table C.8. Optical digit: SVM with linear kernel results on test set: 10 hard 24 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM L	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0.99±0	0.99±0	0.99±0	0.99±0	0.99±0	0.99±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM L	**Accuracy**	57.8±21.02	60.33±16.24	60.17±17.69	57.14±21.84	58.62±20.79	58.62±20.74
	PR-AUC	0.64±0.27	0.64±0.28	0.63±0.23	0.66±0.27	0.67±0.24	0.67±0.24
	ROC-AUC	0.62±0.36	0.6±0.38	0.63±0.32	0.64±0.37	0.66±0.36	0.66±0.36
	Break Even(even)	0.59±0.3	0.58±0.3	0.6±0.24	0.61±0.3	0.63±0.29	0.63±0.29
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
IW-SVM L	**Accuracy**	NA	NA	NA	NA	NA	NA
	PR-AUC	NA	NA	NA	NA	NA	NA
	ROC-AUC	NA	NA	NA	NA	NA	NA
	Break Even(even)	NA	NA	NA	NA	NA	NA
	Break Even(odd)	NA	NA	NA	NA	NA	NA

Table C.9. Optical digit: SVM with RBF kernel results on test set: 10 hard 24 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM R	**Accuracy**	27.38±2.15	27.77±1.62	27.27±1.08	27.6±1.06	28.32±1.96	27.27±1.47
	PR-AUC	0±0	0±0	0±0	0±0	0.6±0.33	0±0
	ROC-AUC	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
IW-SVM R	**Accuracy**	NA	NA	NA	NA	NA	NA
	PR-AUC	NA	NA	NA	NA	NA	NA
	ROC-AUC	NA	NA	NA	NA	NA	NA
	Break Even(even)	NA	NA	NA	NA	NA	NA
	Break Even(odd)	NA	NA	NA	NA	NA	NA

Table C.10. The parameters of optical digit results: 10 hard 24 soft.

	Noise Proportion					
	None	**10**	**20**	**30**	**40**	**50**
SS-kNN	k=5, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=1	k=7, θ_6=0.01,θ_5=1	k=9, θ_6=1,θ_5=1	k=5, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=1
ST-kNN	k=9, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=1
IW-kNN	k=3, θ_6=1	k=3, θ_6=1	k=1, θ_6=1	k=1, θ_6=1	k=3, θ_6=1	k=1, θ_6=1
SS-SVM L	C=8,θ_5=0.5	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1
ST-SVM L	C=8,θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1
IW-SVM L	NA	NA	NA	NA	NA	NA
SS-SVM R	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1θ_5=1	C=8,γ=1, θ_6=1,θ_5=1
ST-SVM R	C=2,γ=0.031, θ_5=1	C=0.125,γ=0.031 θ_6=1	C=0.25,γ=0.25, θ_6=1,θ_5=1	C=0.5,γ=0.031, θ_6=1,θ_5=1	C=2,γ=0.25, θ_5=1	C=2,γ=0.063, θ_5=1
IW-SVM R	NA	NA	NA	NA	NA	NA

Table C.11. Optical digit: k-NN results on test set: 10 hard 50 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-kNN	**Accuracy**	99.45±0	99.45±0	99.45±0	99.72±0	97.52±0	95.43±0.15
	PR-AUC	0.04±0	0.06±0	0.22±0	0.64±0	0.91±0	0.9±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	0.99±0
	Break Even(even)	0.99±0	0.99±0	0.99±0	0.99±0	0.98±0	0.97±0
	Break Even(odd)	1±0	1±0	1±0	1±0	0.98±0	0.98±0
ST-kNN	**Accuracy**	91.18±0	74.66±0	64.74±0	77.14±0	52.89±0	92.84±0
	PR-AUC	0.13±0	0.37±0	0.84±0	0.32±0	0.56±0	0.74±0
	ROC-AUC	0.91±0	0.75±0	0.78±0	0.77±0	0.49±0	0.99±0
	Break Even(even)	0.99±0	0.99±0	0.95±0	0.98±0	0.65±0	0.97±0
	Break Even(odd)	0.84±0	0.51±0	0.65±0	0.58±0	0.63±0	0.98±0
IW-kNN	**Accuracy**	81.27±0	60±6.83	53.06±4.11	74.77±14.18	68.32±0	58.95±8.93s
	PR-AUC	0.11±0	0.41±0.09	0.54±0.23	0.44±0.07	0.11±0	0.48±0.25
	ROC-AUC	0.81±0	0.57±0.09	0.76±0.16	0.77±0.16	0.68±0	0.78±0.08
	Break Even(even)	0.83±0	0.57±0.08	0.78±0.06	0.74±0.15	0.81±0	0.74±0.05
	Break Even(odd)	0.96±0	1±0	0.92±0.19	1±0	0.75±0	0.94±0.14

Table C.12. Optical digit: SVM with linear kernel results on test Set: 10 hard 50 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM L	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0.99±0	0.99±0	0.99±0	0.99±0	0.99±0	0.99±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM L	**Accuracy**	58.46±7.46	59.78±11.54	56.36±19.94	58.46±21.14	60.99±22.24	60.72±8.65
	PR-AUC	0.64±0.18	0.6±0.16	0.68±0.25	0.64±0.26	0.66±0.29	0.62±0.21
	ROC-AUC	0.7±0.2	0.64±0.19	0.67±0.36	0.64±0.36	0.62±0.39	0.65±0.25
	Break Even(even)	0.66±0.15	0.61±0.15	0.63±0.29	0.61±0.3	0.61±0.32	0.61±0.2
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
IW-SVM L	**Accuracy**	NA	NA	NA	NA	NA	NA
	PR-AUC	NA	NA	NA	NA	NA	NA
	ROC-AUC	NA	NA	NA	NA	NA	NA
	Break Even(even)	NA	NA	NA	NA	NA	NA
	Break Even(odd)	NA	NA	NA	NA	NA	NA

Table C.13. Optical digit: SVM with RBF kernel results on test set: 10 hard 50 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM R	**Accuracy**	27.88±1.48	27.71±1.89	27.49±1.84	27.88±2	27.77±1.9	27.77±2.17
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
IW-SVM R	**Accuracy**	NA	NA	NA	NA	NA	NA
	PR-AUC	NA	NA	NA	NA	NA	NA
	ROC-AUC	NA	NA	NA	NA	NA	NA
	Break Even(even)	NA	NA	NA	NA	NA	NA
	Break Even(odd)	NA	NA	NA	NA	NA	NA

Table C.14. The parameters of optical digit results: 10 hard 50 soft.

	Noise Proportion					
	None	**10**	**20**	**30**	**40**	**50**
SS-kNN	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=0.5,θ_5=1	k=1, θ_6=0.5,θ_5=	k=1, θ_6=0.5,θ_5=1
ST-kNN	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1	k=9, θ_6=1,θ_5=1
IW-kNN	k=1, θ_6=1	k=3, θ_6=1	k=5, θ_6=1	k=3, θ_6=1	k=1, θ_6=1	k=5, θ_6=1
SS-SVM L	C=8,θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1
ST-SVM L	C=0.25, θ_6=1	C=0.5, θ_5=1	C=0.125, θ_5=1	C=0.25, θ_6=1,θ_5=1	C=0.125, θ_5=1	C=0.25, θ_6=1,θ_5=1
IW-SVM L	NA	NA	NA	NA	NA	NA
SS-SVM R	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5
ST-SVM R	C=2,γ=0.031, θ_6=1,θ_5=1	C=2,γ=0.063, θ_5=1	C=4,γ=1, θ_6=1	C=0.25,γ=1, θ_6=,θ_5=1	C=8,γ=1, θ_5=1	C=2,γ=1, θ_6=1
IW-SVM R	NA	NA	NA	NA	NA	NA

Table C.15. Optical digit: k-NN results on test set: 10 hard 100 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-kNN	**Accuracy**	99.72±0	99.72±0	99.56±0.15	98.79±0.42	97.52±0	95.87±0
	PR-AUC	0.01±0	0.02±0	0.13±0	0.28±0	0.75±0	0.99±0
	ROC-AUC	1±0	1±0	1±0	1±0	0.99±0	0.99±0
	Break Even(even)	0.99±0	0.99±0	0.99±0	0.99±0	0.98±0	0.96±0
	Break Even(odd)	0.99±0	0.99±0	0.99±0	0.98±0	0.98±0	0.98±0
ST-kNN	**Accuracy**	66.39±0	66.67±0	60.33±0	54.82±0	58.95±0	53.17±0
	PR-AUC	66.39±0	66.67±0	60.33±0	54.82±0	58.95±0	53.17±0
	ROC-AUC	0.73±0	0.67±0	0.6±0	0.62±0	0.59±0	0.53±0
	Break Even(even)	1±0	1±0	1±0	0.61±0	1±0	0.99±0
	Break Even(odd)	0.46±0	0.34±0	0.21±0	0.98±0	0.18±0	0.08±0
IW-kNN	**Accuracy**	68.32±0	50.14±0	56.91±7.63	61.49±9.88	50.19±0.74	63.8±5.22
	PR-AUC	0.31±0	0±0	0.58±0.13	0.64±0.24	0.46±0.12	0.54±0.08
	ROC-AUC	0.68±0	0.5±0	0.64±0.18	0.74±0.23	0.47±0.16	0.61±0.1
	Break Even(even)	0.85±0	1±0	0.61±0.13	0.68±0.21	0.48±0.11	0.59±0.08
	Break Even(odd)	0.68±0	0.01±0	1±0	1±0	1±0	1±0

Table C.16. Optical digit: SVM with linear kernel tesults on test set: 10 hard 100 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM L	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0.99±0	0.99±0	0.99±0	0.99±0	0.99±0	0.99±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM L	**Accuracy**	60.5±8.61	59.45±17.72	57.14±21.84	61.54±19.91	58.68±26.14	59.89±24.36
	PR-AUC	0.61±0.11	0.65±0.28	0.66±0.27	0.63±0.22	0.65±0.3	0.62±0.27
	ROC-AUC	0.67±0.12	0.61±0.38	0.64±0.37	0.66±0.23	0.62±0.38	0.63±0.33
	Break Even(even)	0.62±0.1	0.59±0.29	0.61±0.3	0.64±0.21	0.62±0.33	0.61±0.27
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
IW-SVM L	**Accuracy**	NA	NA	NA	NA	NA	NA
	PR-AUC	NA	NA	NA	NA	NA	NA
	ROC-AUC	NA	NA	NA	NA	NA	NA
	Break Even(even)	NA	NA	NA	NA	NA	NA
	Break Even(odd)	NA	NA	NA	NA	NA	NA

Table C.17. Optical digit: SVM with RBF kernel results on test set: 10 hard 100 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM R	**Accuracy**	27.77±1.13	27.38±0.79	28.1±1.38	27.99±0.77	27.33±2.12	27.11±1.91
	PR-AUC	0±0	0.3±0.41	0±0	0±0	0±0	0±0
	ROC-AUC	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
IW-SVM R	**Accuracy**	NA	NA	NA	NA	NA	NA
	PR-AUC	NA	NA	NA	NA	NA	NA
	ROC-AUC	NA	NA	NA	NA	NA	NA
	Break Even(even)	NA	NA	NA	NA	NA	NA
	Break Even(odd)	NA	NA	NA	NA	NA	NA

Table C.18. The parameters of optical digit results: 10 hard 100 soft.

	Noise Proportion					
	None	**10**	**20**	**30**	**40**	**50**
SS-kNN	k=3, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=9, θ_6=1,θ_5=0.5	k=5, θ_6=1,θ_5=0.5	k=5, θ_6=1,θ_5=0.01	k=7, θ_6=1,θ_5=0.5
ST-kNN	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
IW-kNN	k=1, θ_6=1	k=9, θ_6=1,θ_5=1	k=3, θ_6=1	k=5, θ_6=1	k=7, θ_6=1	k=3, θ_6=1
SS-SVM L	C=8, θ_5=0.001	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_6=0.5, θ^5:0.5
ST-SVM L	C=8, θ_6=1,θ_5=1	C=2, θ_5=1	C=4, θ_6=1,θ_5=1	C=0.5, θ_6=1,θ_5=1	C=4, θ_5=1	C=0.25, θ_6=1,θ_5=1
IW-SVM L	NA	NA	NA	NA	NA	NA
SS-SVM R	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5	C=2,γ=1, θ_6=1θ_5=0.5	C=2,γ=1, θ_6=1,θ_5=0.5
ST-SVM R	C=0.125γ=0.125, θ_6=1	C=0.125,γ=0.25, θ_6=1	C=0.5,γ=0.031, θ_5=1	C=0.125,γ=0.031 θ_6=1,θ_5=1	C=0.125,γ=0.063 θ_5=1	C=0.5,γ=0.031, θ_6=1,θ_5=1
IW-SVM R	NA	NA	NA	NA	NA	NA

Table C.19. Optical digit: accuracy values on test data with various parameters of all algorithms for 30 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
No Soft	**k-NN**	96.69±1.47	-	-	-	-	-
	IW-kNN	96.69±1.47	-	-	-	-	-
	SVM L	50.25±0.54	-	-	-	-	-
	IW-SVM L	51.52±1.9	-	-	-	-	-
	SVM R	27.38±0.31	-	-	-	-	-
	IW-SVM R	100±0	-	-	-	-	-

Table C.20. Optical digit: accuracy values on test data with various parameters of all algorithms for 30 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
24 Soft	**SS-kNN**	99.17±0.19	99.45±0.19	99.28±0.42	96.64±3.58	97.3±1.49	97.25±0.93
	ST-kNN	98.84±0.53	98.46±0.72	98.35±0.62	96.64±3.58	97.3±1.49	97.25±0.93
	IW-kNN	98.18±0.79	97.19±1.88	97.63±1.29	96.64±3.58	96.42±1.72	97.25±0.93
	SS-SVM L	53±2.02	59.39±7.44	56.86±5.09	55.32±1.25	54.1±2.38	57.8±6.77
	ST-SVM L	61.93±21.4	65.95±34.24	71.41±26.11	70.85±26.62	60.11±22.31	60.28±22.33
	IW-SVM L	67±25.93	55.76±26.23	64.02±20.42	54.55±4.19	59.06±18.33	54.55±8
	SS-SVM R	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM R	27.49±2.48	28.54±2.54	27.27±1.92	27.11±1.48	28.04±1.6	27.55±1.05
	IW-SVM R	100±0	100±0	100±0	100±0	100±0	100±0

Table C.21. Optical digit: accuracy values on test data with various parameters of all hlgorithms for 30 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
50 Soft	**SS-kNN**	99.72±0	99.78±0.23	98.84±0.79	98.02±1.24	96.03±2.61	95.98±4.5
	ST- kNN	97.58±1.72	97.25±1.64	98.46±0.5	96.86±3.58	96.03±2.61	95.98±4.5
	IW-kNN	96.8±1.88	95.1±2.64	96.42±2.64	96.86±3.58	96.03±2.61	95.98±4.5
	SS-SVM L	61.32±6.47	61.93±2.07	59.06±6.68	57.69±5.97	57.69±4.57	60.33±8.28
	ST-SVM L	60.06±8.36	69.64±27.76	61.98±21.49	55.21±6.09	58.62±23.39	60.77±22
	IW-SVM L	64.3±20.86	60.72±23.27	55.26±17.65	51.24±12.24	74.71±26.64	69.86±25.74
	SS-SVM R	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM R	27.22±2.16	28.1±1.42	27.66±2.2	27.22±1.19	27.55±2	27.88±2.6
	IW-SVM R	100±0	100±0	100±0	100±0	100±0	100±0

Table C.22. Optical digit: accuracy values on test data with various parameters of all algorithms for 30 hardly labeled.

		Noise Proportion on Softly Labeled Data					
		None	**10**	**20**	**30**	**40**	**50**
100 Soft	**SS-kNN**	99.72±0	99.72±0	99.28±0.25	98.35±1.12	96.97±3.09	98.24±1.53
	ST-kNN	99.45±0	95.76±1.93	98.13±1.48	98.35±1.12	96.97±1.8	98.24±1.53
	IW-kNN	96.64±2.51	96.03±1.7	96.53±3.93	98.35±1.12	96.97±1.8	98.24±1.53
	SS-SVM L	56.42±5.79	55.59±5.51	52.89±2.2	56.31±0.74	55.81±3.73	60.17±5.54
	ST-SVM L	57.36±7.12	56.25±5.92	56.42±24.76	56.36±1.75	54.88±9.36	52.23±3.16
	IW-SVM L	70.58±26.7	60.88±16.68	51.13±10.46	66.39±20.88	57.3±21.65	50.47±32.79
	SS-SVM R	100±0	100±0	100±0	100±0	100±0	100±0
	ST-SVM R	27.44±1.08	27.11±2.02	27.38±1.5	27.6±1.3	28.1±1.31	27.66±1.68
	IW-SVM R	100±0	100±0	100±0	100±0	100±0	100±0

Table C.23. Optical digit results on test set: 30 hard no soft.

k-NN	**Accuracy**	96.69±1.47
	PR-AUC	0.01±0.01
	ROC-AUC	0.97±0.01
	Break Even	0.99±0.01
	Break Even(odd)	0.9±0.04
IW-kNN	**Accuracy**	96.69±1.47
	PR-AUC	0.01±0.01
	ROC-AUC	0.97±0.01
	Break Even	0.99±0.01
	Break Even(odd)	0.95±0.03
SVM L	**Accuracy**	50.25±0.54
	PR-AUC	0.63±0.06
	ROC-AUC	0.71±0.06
	Break Even	0.66±0.05
	Break Even(odd)	1±0
SVM R	**Accuracy**	27.38±0.31
	PR-AUC	0.6±0.33
	ROC-AUC	0.5±0
	Break Even	1±0
	Break Even(odd)	0.01±0
IW-SVM L	**Accuracy**	51.52±1.9
	PR-AUC	0.7±0.21
	ROC-AUC	0.75±0.2
	Break Even	0.7±0.18
	Break Even(odd)	1±0
IW-SVM R	**Accuracy**	100±0
	PR-AUC	0±0
	ROC-AUC	1±0
	Break Even	1±0
	Break Even(odd)	1±0

Table C.24. The parameters of optical digit results on test set: 30 hard no soft.

k-NN	k=1
IW-kNN	k=1
SVM L	C=2
SVM R	C=4 γ=0.03125
IW-SVM L	C=1
IW-SVM R	C=8 γ=1

Table C.25. Optical digit: k-NN results on test set: 30 hard 24 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-kNN	**Accuracy**	99.17±0.19	99.45±0.19	99.28±0.42	96.64±3.58	97.3±1.49	97.25±0.93
	PR-AUC	0.08±0.01	0.1±0.02	0.3±0.05	0.01±0.01	0.12±0.05	0.01±0.01
	ROC-AUC	1±0	1±0	1±0	0.97±0.04	0.99±0	0.97±0.01
	Break Even(even)	1±0	1±0	0.99±0	0.99±0.01	0.99±0	0.99±0.01
	Break Even(odd)	0.99±0	1±0	0.99±0.01	0.94±0.07	0.96±0.03	0.96±0.02
ST-kNN	**Accuracy**	98.84±0.53	98.46±0.72	98.35±0.62	96.64±3.58	97.3±1.49	97.25±0.93
	PR-AUC	0.02±0.01	0.29±0.11	0.16±0.15	0.01±0.01	0.12±0.05	0.01±0.01
	ROC-AUC	0.99±0.01	1±0	1±0	0.97±0.04	0.99±0	0.97±0.01
	Break Even(even)	1±0	0.99±0	0.99±0	0.99±0.01	0.99±0	0.99±0.01
	Break Even(odd)	0.98±0.01	0.99±0.01	0.98±0.01	0.94±0.07	0.96±0.03	0.96±0.02
IW-kNN	**Accuracy**	98.18±0.79	97.19±1.88	97.63±1.29	96.64±3.58	96.42±1.72	97.25±0.93
	PR-AUC	0.38±0.14	0.39±0.15	0.01±0	0.01±0.01	0.12±0.05	0.01±0.01
	ROC-AUC	0.99±0	0.99±0	0.98±0.01	0.97±0.04	0.99±0.01	0.97±0.01
	Break Even(even)	0.99±0.01	0.98±0.01	0.99±0	0.99±0.01	0.96±0.02	0.99±0.01
	Break Even(odd)	0.99±0.01	1±0	0.97±0.03	0.94±0.07	1±0	0.96±0.02

Table C.26. Optical digit: SVM with linear kernel results on test set: 30 hard 24 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM L	**Accuracy**	53±2.02	59.39±7.44	56.86±5.09	55.32±1.25	54.1±2.38	57.8±6.77
	PR-AUC	0.88±0.12	0.97±0.03	0.97±0.03	0.98±0.01	0.95±0.04	0.93±0.11
	ROC-AUC	0.91±0.11	0.98±0.02	0.98±0.02	0.99±0.01	0.97±0.03	0.95±0.08
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM L	**Accuracy**	61.93±21.4	65.95±34.24	71.41±26.11	70.85±26.62	60.11±22.31	60.28±22.33
	PR-AUC	0.86±0.15	0.8±0.29	0.92±0.09	0.91±0.09	0.68±0.19	0.68±0.28
	ROC-AUC	0.89±0.14	0.76±0.43	0.94±0.06	0.93±0.07	0.74±0.16	0.7±0.29
	Break Even(even)	0.84±0.16	0.74±0.39	0.9±0.1	0.89±0.11	0.7±0.18	0.68±0.27
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
IW-SVM L	**Accuracy**	67±25.93	55.76±26.23	64.02±20.42	54.55±4.19	59.06±18.33	54.55±8
	PR-AUC	0.69±0.29	0.58±0.28	0.84±0.17	0.89±0.09	0.72±0.24	0.67±0.27
	ROC-AUC	0.7±0.31	0.55±0.33	0.87±0.14	0.93±0.05	0.74±0.25	0.7±0.27
	Break Even(even)	0.7±0.28	0.55±0.3	0.81±0.16	0.87±0.07	0.71±0.23	0.66±0.22
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0

Table C.27. Optical digit: SVM with RBF kernel results on test set: 30 hard 24 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM R	**Accuracy**	27.49±2.48	28.54±2.54	27.27±1.92	27.11±1.48	28.04±1.6	27.55±1.05
	PR-AUC	0±0	0±0	0±0	0±0	0.6±0.33	0±0
	ROC-AUC	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
IW-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0

Table C.28. The parameters of optical digit results: 30 hard 24 soft.

	Noise Proportion					
	None	**10**	**20**	**30**	**40**	**50**
SS-kNN	k=5, θ_6=1,θ_5=1	k=5, θ_6=1,θ_5=1	k=5, θ_6=0.01,θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
ST-kNN	k=1, θ_6=1,θ_5=1	k=3, θ_5=1	k=3, θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
IW-kNN	k=3, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
SS-SVM L	C=2, θ_5=0.5	C=0, θ_5=0.001	C=1, θ_5=1	C=0.125, θ_5=0.001	C=0.125, θ_5=0.001	C=0.125, θ_5=0.5
ST-SVM L	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1	C=8, θ_5=1
IW-SVM L	C=0.5, θ_6=1,θ_5=1	C=8, θ_6=1,θ_5=1	C=8, θ_6=1	C=0.125, θ_6=1	C=0.25, θ_6=1,θ_5=1	C=0.125, θ_5=1
SS-SVM R	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5
ST-SVM R	C=8,γ=0.25, θ_5=1	C=2,γ=0.125, θ_5=1	C=0.5,γ=1, θ_6=1	C=2,γ=0.0625, θ_5=1	C=8,γ=0.03125, θ_5=1	C=0.5,γ=0.25, θ_5=1
IW-SVM R	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1

Table C.29. Optical digit: k-NN results on test Set: 30 hard 50 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-kNN	**Accuracy**	99.72±0	99.78±0.23	98.84±0.79	98.02±1.24	96.03±2.61	95.98±4.5
	PR-AUC	0.03±0	0.02±0	0.07±0.02	0.01±0.01	0.01±0.01	0.01±0.01
	ROC-AUC	1±0	1±0	1±0	0.98±0.01	0.96±0.03	0.96±0.04
	Break Even(even)	1±0	1±0	1±0.01	0.99±0.01	0.99±0.01	0.99±0.01
	Break Even(odd)	0.99±0	1±0	0.99±0	0.98±0.02	0.94±0.05	0.94±0.08
ST-kNN	**Accuracy**	97.58±1.72	97.25±1.64	98.46±0.5	96.86±3.58	96.03±2.61	95.98±4.5
	PR-AUC	0.02±0	0.02±0	0.12±0.08	0.01±0.01	0.01±0.01	0.01±0.01
	ROC-AUC	1±0.01	0.99±0.01	1±0	0.97±0.04	0.96±0.03	0.96±0.04
	Break Even(even)	0.99±0	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01	0.99±0.01
	Break Even(odd)	0.98±0.01	0.99±0	0.99±0.01	0.95±0.07	0.94±0.05	0.94±0.08
IW-kNN	**Accuracy**	96.8±1.88	95.1±2.64	96.42±2.64	96.86±3.58	96.03±2.61	95.98±4.5
	PR-AUC	0.99±0	0.42±0.2	0.12±0.12	0.01±0.01	0.01±0.01	0.01±0.01
	ROC-AUC	0.99±0	0.99±0	1±0	0.97±0.04	0.96±0.03	0.96±0.04
	Break Even(even)	0.98±0.02	0.97±0.02	0.98±0.01	0.99±0.01	0.99±0.01	0.99±0.01
	Break Even(odd)	1±0.01	0.99±0.01	1±0	0.95±0.07	0.94±0.05	0.94±0.08

Table C.30. Optical digit: SVM with linear kernel results on test set: 30 hard 50 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM L	**Accuracy**	61.32±6.47	61.93±2.07	59.06±6.68	57.69±5.97	57.69±4.57	60.33±8.28
	PR-AUC	0.99±0.01	0.99±0	0.98±0.03	0.98±0.03	0.97±0.02	0.96±0.05
	ROC-AUC	1±0.01	1±0	0.98±0.03	0.99±0.02	0.98±0.02	0.97±0.04
	Break Even(even)	0.98±0.02	0.98±0.01	0.95±0.05	0.96±0.04	0.95±0.04	0.94±0.07
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM L	**Accuracy**	60.06±8.36	69.64±27.76	61.98±21.49	55.21±6.09	58.62±23.39	60.77±22
	PR-AUC	0.86±0.19	0.74±0.27	0.75±0.25	0.89±0.1	0.64±0.28	0.74±0.28
	ROC-AUC	0.88±0.17	0.77±0.25	0.79±0.26	0.92±0.07	0.63±0.35	0.76±0.3
	Break Even(even)	0.85±0.18	0.74±0.25	0.75±0.24	0.86±0.1	0.62±0.29	0.73±0.28
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
IW-SVM L	**Accuracy**	64.3±20.86	60.72±23.27	55.26±17.65	51.24±12.24	74.71±26.64	69.86±25.74
	PR-AUC	0.82±0.21	0.78±0.28	0.66±0.29	0.68±0.31	0.81±0.28	0.82±0.27
	ROC-AUC	0.85±0.18	0.77±0.37	0.62±0.41	0.65±0.4	0.82±0.31	0.81±0.35
	Break Even(even)	0.8±0.17	0.75±0.31	0.59±0.39	0.63±0.35	0.82±0.28	0.79±0.31
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0

Table C.31. Optical digit: SVM with RBF kernel results on test set: 30 hard 50 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM R	**Accuracy**	27.22±2.16	28.1±1.42	27.66±2.2	27.22±1.19	27.55±2	27.88±2.6
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
IW-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0

Table C.32. The parameters of optical digit results: 30 hard 50 soft.

	Noise Proportion					
	None	**10**	**20**	**30**	**40**	**50**
SS-kNN	k=3, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=5, θ_6=0.01,θ_5=1	k=1, θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
ST-kNN	k=3, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=3, θ_5=1	k=1, θ_6=1	k=1, θ_5=1	k=1, θ_6=1,θ_5=1
IW-kNN	k=5, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_5=1	k=1, θ_5=1	k=1, θ_6=1,θ_5=1
SS-SVM L	C=1, θ_5=1	C=0.5, θ_5=0.01	C=0.25, θ_5=1	C=0.125, θ_5=0.001	C=0.125, θ_5=0.001	C=0.125, θ_5=0.001
ST-SVM L	C=0.25, θ_5=1	C=0.5, θ_5=1	C=2, θ_5=1	C=4, θ_5=1	C=8, θ_5=1	C=8, θ_5=1
IW-SVM L	C=0.25, θ_5=1	C=4, θ_5=1	C=8, θ_6=1	C=0.25, θ_6=1,θ_5=1	C=1, θ_6=1,θ_5=1	C=0.25, θ_6=1
SS-SVM R	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5
ST-SVM R	C=1,γ=0.25, θ_6=1,θ_5=1	C=4,γ=0.5, θ_5=1	C=1,γ=0.063, θ_5=1	C=0.25,γ=0.032, θ_6=1	C=0.125,γ=0.25, θ_6=1	C=0.125,γ=0.032 θ_6=1
IW-SVM R	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1

Table C.33. Optical digit: k-NN results on test set: 30 hard 100 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-kNN	**Accuracy**	99.72±0	99.72±0	99.28±0.25	98.35±1.12	96.97±3.09	98.24±1.53
	PR-AUC	0.02±0	0.05±0.01	0.08±0.04	0.01±0.01	0.1±0.04	0±0.01
	ROC-AUC	1±0	1±0	1±0	0.98±0.01	1±0.01	0.98±0.02
	Break Even(even)	0.99±0	0.99±0	1±0	0.99±0.01	1±0.01	1±0.01
	Break Even(odd)	1±0	1±0	1±0	0.98±0.02	0.96±0.03	0.97±0.03
ST-kNN	**Accuracy**	99.45±0	95.76±1.93	98.13±1.48	98.35±1.12	96.97±1.8	98.24±1.53
	PR-AUC	0.02±0	0.06±0.03	0.17±0.18	0.01±0.01	0.01±0.01	0±0.01
	ROC-AUC	1±0	0.96±0.02	1±0	0.98±0.01	0.97±0.02	0.98±0.02
	Break Even(even)	0.99±0	1±0	0.99±0	0.99±0.01	0.99±0.01	1±0.01
	Break Even(odd)	1±0	0.92±0.04	0.98±0.01	0.98±0.02	0.95±0.03	0.97±0.03
IW-kNN	**Accuracy**	96.64±2.51	96.03±1.7	96.53±3.93	98.35±1.12	96.97±1.8	98.24±1.53
	PR-AUC	0.63±0.1	0.47±0.22	0.01±0.01	0.01±0.01	0.01±0.01	0±0.01
	ROC-AUC	0.98±0.01	0.98±0.01	0.97±0.04	0.98±0.01	0.97±0.02	0.98±0.02
	Break Even(even)	1±0	0.99±0.02	0.99±0.01	0.99±0.01	0.99±0.01	1±0.01
	Break Even(odd)	0.97±0.02	0.97±0.03	0.95±0.09	0.98±0.02	0.95±0.03	0.97±0.03

Table C.34. Optical digit: SVM with linear kernel results on test set: 30 hard 100 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM L	**Accuracy**	56.42±5.79	55.59±5.51	52.89±2.2	56.31±0.74	55.81±3.73	60.17±5.54
	PR-AUC	0.93±0.09	0.95±0.07	0.92±0.06	0.99±0.01	0.93±0.1	0.95±0.04
	ROC-AUC	0.95±0.06	0.96±0.07	0.94±0.05	0.99±0.01	0.95±0.07	0.96±0.04
	Break Even(even)	0.91±0.09	0.91±0.11	0.89±0.07	0.97±0.03	0.9±0.1	0.91±0.06
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM L	**Accuracy**	57.36±7.12	56.25±5.92	56.42±24.76	56.36±1.75	54.88±9.36	52.23±3.16
	PR-AUC	0.91±0.14	0.95±0.06	0.55±0.26	0.97±0.03	0.66±0.3	0.81±0.24
	ROC-AUC	0.94±0.09	0.95±0.07	0.52±0.32	0.98±0.03	0.58±0.4	0.83±0.24
	Break Even(even)	0.9±0.11	0.9±0.11	0.53±0.3	0.95±0.05	0.61±0.34	0.78±0.22
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
IW-SVM L	**Accuracy**	70.58±26.7	60.88±16.68	51.13±10.46	66.39±20.88	57.3±21.65	50.47±32.79
	PR-AUC	0.75±0.31	0.73±0.25	0.68±0.29	0.82±0.2	0.64±0.26	0.5±0.28
	ROC-AUC	0.73±0.35	0.74±0.25	0.68±0.35	0.86±0.16	0.67±0.28	0.41±0.38
	Break Even(even)	0.73±0.3	0.72±0.25	0.67±0.32	0.82±0.18	0.65±0.26	0.42±0.37
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0

Table C.35. Optical digit: SVM with RBF kernel results on test set: 30 hard 100 soft.

		Noise Proportion					
		None	**10**	**20**	**30**	**40**	**50**
SS-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0
ST-SVM R	**Accuracy**	27.44±1.08	27.11±2.02	27.38±1.5	27.6±1.3	28.1±1.31	27.66±1.68
	PR-AUC	0±0	0.3±0.41	0±0	0±0	0±0	0±0
	ROC-AUC	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0	0.5±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0	0.01±0
IW-SVM R	**Accuracy**	100±0	100±0	100±0	100±0	100±0	100±0
	PR-AUC	0±0	0±0	0±0	0±0	0±0	0±0
	ROC-AUC	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(even)	1±0	1±0	1±0	1±0	1±0	1±0
	Break Even(odd)	1±0	1±0	1±0	1±0	1±0	1±0

Table C.36. The parameters of optical digit results: 30 hard 100 soft.

	Noise Proportion					
	None	**10**	**20**	**30**	**40**	**50**
SS-kNN	k=3, θ_6=1,θ_5=1	k=5, θ_6=1,θ_5=1	k=5, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
ST-kNN	k=3, θ_6=1,θ_5=1	k=1, θ_5=1	k=3, θ_5=1	k=1, θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
IW-kNN	k=3, θ_6=1,θ_5=1	k=3, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1	k=1, θ_6=1,θ_5=1
SS-SVM L	C=1, θ_5=0.001	C=4, θ_5=1	C=8, θ_5=1	C=0.125, θ_5=0.001	C=0.25, θ_5=0.001	C=2, θ_6=0.5, θ^5:0.5
ST-SVM L	C=0.125, θ_5=1	C=0.25 ,θ_5=1	C=8, θ_6=1,θ_5=1	C=4, θ_5=1	C=0.5, θ_6=1,θ_5=1	C=0.5, θ_5=1
IW-SVM L	C=8, θ_6=1,θ_5=1	C=0.125, θ_5=1	C=2, θ_5=1	C=0.5, θ_6=1	C=8, θ_6=1,θ_5=1	C=0.25, θ_6=1,θ_5=1
SS-SVM R	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5	C=1,γ=1, θ_6=1θ_5=0.5	C=1,γ=1, θ_6=1,θ_5=0.5
ST-SVM R	C=2,γ=0.5, θ_5=1	C=0.5,γ=0.031, θ_5=1	C=0.25,γ=0.031, θ_6=1,θ_5=1	C=0.125,γ=0.5, θ_6=1,θ_5=1	C=1,γ=0.125 θ_6=1	C=0.125,γ=0.125 θ_6=1,θ_5=1
IW-SVM R	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1	C=8,γ=1, θ_6=1,θ_5=1

REFERENCES

1. Winslow, T., 2005, *Colorectal Cancer: Tumor Sizes*, http://www.terese winslow.com, accessed at March 2014.

2. Qi, Y., O. Tastan, J. G. Carbonell, J. Klein-Seetharaman and J. Weston, "Semi-Supervised Multi-Task Learning for Predicting Interactions Between HIV-1 and Human Proteins", *Bioinformatics*, Vol. 26, pp. 1645-1652, 2010.

3. Design, P. S., 2013, *How to Choose a Classifier for a Multi-Modal Problem*, http://perclass.com/doc/kb/29.html, accessed at March 2014.

4. Elworthy, D., "Does Baum-Welch Re-estimation Help Taggers", *Proceedings of the Fourth Conference on Applied Natural Language Processing*, Stuttgart, 13-15 October 1994, pp. 53-58, Association for Computational Linguistics, Stroudsburg, 1994.

5. Zhu, X., *Semi-Supervised Learning Literature Survey*, Technical Report, Computer Sciences Department, University of Wisconsin Madison, 2006.

6. World Health Organization, *Globocan 2012 - Latest World Cancer Statistics*, International Agency for Research on Cancer, Lyon, 2013.

7. Shi, M. and B. Zhang, "Semi-Supervised Learning Improves Gene Expression-Based Prediction of Cancer Recurrence", *Bioinformatics*, Vol. 27, No. 21, pp. 3017–3023, 2011.

8. Dupuy, A. and R. M. Simon, "Critical Review of Published Microarray Studies for Cancer Outcome and Guidelines on Statistical Analysis and Reporting", *Oxford Journals Journal of The National Cancer Institute*, Vol. 99, No. 2, pp. 147–57, 2007.

9. Weston, J., C. Leslie, I. E, D. Zhou, A. Elisseeff and W. Noble, "Semi-Supervised Protein Classification Using Cluster Kernels", *Bioinformatics*, Vol. 21, No. 15, pp. 3241-3247, 2005.

10. Ernst, J., Q. K. Beg, K. A. Kay, G. Balazsi, Z. N. Oltvai and Z. Bar-Joseph, "A Semi-Supervised Method for Predicting Transcription Factor-Gene Interactions in Escherichia Coli", *Public Library of Science Computational Biology*, Vol. 4, No. 3, 2008.

11. Chapelle, O. and A. Zien, "Semi-Supervised Classification by Low Density Separation", *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Barbados, 6-8 January 2005, Vol., pp.57-64, 2005.

12. National Cancer Intelligence Network, 2009, *Colorectal Cancer Survival by Stage*, http://www.ncin.org.uk/databriefings, accessed at March 2014.

13. World Health Organization, 2013, *HIV/AIDS Fact Sheet No 360*, http://www.who.int/mediacentre/factsheets/fs360/en, accessed at March 2014.

14. EM, P. and S. Fields, "Protein-Protein Interactions: Methods for Detection and Analysis", *Microbiological Reviews*, Vol. 59, No. 1, pp. 94–123, 1995.

15. Shoemaker, A. B. and A. R. Panchenko, "Deciphering Protein,Protein Interactions Part I Experimental Techniques and Databases", *Public Library of Science Computational Biology*, Vol. 3, No. 3, p. e42, 2007.

16. Shoemaker, A. B. and A. R. Panchenko, "Deciphering Protein,Protein Interactions Part II Computational Methods to Predict Protein and Domain Interaction Partners", *Public Library of Science Computational Biology*, Vol. 3, No. 4, p. e43, 2007.

17. Qi, Y., J. Klein-Seetharaman, Z. Bar-Joseph, "Random Forest Similarity for Protein-Protein Interaction Prediction From Multiple Sources", *Pacific Symposium*

on Biocomputing, Hawaii, 4-8 January 2005, pp. 531–542, 2005.

18. Jansen, R., H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, A. E. Sambath Chung, M. Snyder, J. F. Greenblatt and M. Gerstein, "A Bayesian Networks Approach for Predicting Protein-Protein Interactions From Genomic Data", *Science*, Vol. 302, No. 5644, pp. 449–453, 2003.

19. Yamanishi, Y., J.-P. Vert and M. Kanehis, "Protein Network Inference From Multiple Genomic Data: A Supervised Approach", *Bionformatics*, Vol. 20, No. 1, pp. i363–i370, 2004.

20. Koike, A. and T. Takagi, "Prediction of Protein-Protein Interaction Sites Using Support Vector Machines", *Protein Engineering, Design and Selection*, Vol. 17, No. 2, pp. 165–173, 2004.

21. Tastan, O., Y. Qi, J. G. Carbonell and J. Klein-Seetharaman, "Predictions of Interactions Between HIV-1 and Human Proteins by Information Integration", *Pacific Symposium on Biocomputing*, Vol. 13, pp. 516–527, 2009.

22. Cusick, M. E., H. Yu, A. Smolyar, K. Venkatesan, A.-R. Carvunis, N. S. andJean François Rual, H. Borick, P. Braun, J. V. Matija Dreze, M. Galli, J. Yazaki, D. E. Hill, J. R. Ecker, F. P. Roth and M. Vidal, "Literature-Curated Protein Interaction Datasets", *Nature Methods*, Vol. 6, No. 1, pp. 39–46, 2009.

23. Gene Expression Omnibus, 2010, *GSE17536 and GSE14333 Datasets*, http://www.ncbi.nlm.nih.gov/geo/, accessed at March 2014.

24. Fu, W., B. E. Sanders, K. S. Katz, D. R. Maglott, Pruitt K. D. and R. G. Ptak, "Human Immunodeficiency Virus Type 1, Human Protein Interaction Database at NCBI", *Nucleic Acids Research*, Vol. 37, pp. D417–D422, 2008.

25. Japkowicz, N. and S. Stephen, "The Class Imbalance Problem: A Systematic Study", *Intelligent Data Analysis*, Vol. 6, No. 5, pp. 429–449, 2002.

26. LeCun, Y., C. Cortes and C. J. Burges, 1998, *The Mnist Database of Handwritten Digits*, http://yann.lecun.com/exdb/mnist/, accessed at March 2014.

27. Alpaydin, E. and C. Kaynak, 1998, *Optical Recognition of Handwritten Digits*, https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits, accessed at March 2014.

28. Jahromi, M. Z., E. Parvinnia and R. John, "A Method of Learning Weighted Similarity Function to Improve the Performance of Nearest Neighbour", *Information Sciences*, Vol. 179, No. 17, pp. 2964–2973, 2009.

29. Boser, B. E., I. M. Guyon and V. N. Vapnik, "A Trainning Algorithm for Optimal Margin Classifiers", *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, Pittsburgh, 27-29 July 1992, pp.144-152, ACM, New York, 1992.

30. Platt, J. C., "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods", *Advances In Larger Margin Classifiers*, pp. 61–74, 1999.

31. Davis, J. and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves", *Proceedings of the Twenty-Third International Conference on Machine Learning* , Pittsburgh, 25-29 June 2006, pp. 233-240, ACM, New York, 2006.

32. Chang, C.-C. and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines", *ACM Transactions Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 27:1–27:27, 2011.

33. Fawcett, T., "An Introduction to ROC Analysis", *Pattern Recognition Letters*, Vol. 27, No. 8, pp. 861–874, 2006.

34. Fawcett, T., *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*, Technical Report, Intelligent Enterprise Technologies Laboratory, HP

Laboratories Palo Alto, 2007.

35. Irsoy, O., O. T. Yildiz and E. Alpaydin, "Design and Analysis of Classifier Learning Experiments in Bioinformatics: Survey and Case Studies", *IEEE ACM Transactions on Computational Biology Bioinformatics*, Vol. 9, No. 6, pp.1663-1675, 2012.

36. Dudani, S., "The Distance Weighted k Nearest Neighbour Rule", IEEE Transactions on Systems, *Man and Cybernetics*, Vol. SMC-6, No. 4, pp. 325–327, 1975.

37. Liu, Q., X. Liao and L. Carin, "Learning Classifiers on a Partially Labeled Data Manifold", *Acoustics, Speech and Signal Processing*, Vol. 2, pp. 621–624, 2007.

38. Xie, J. and T. Xiong, "Stochastic Semi-Supervised Learning on Partially Labeled Imbalanced Data", *Journal of Machine Learning Research Workshop and Conference Proceedings*, Vol. 16, Workshop on Active Learning and Experimental Design, 2011.

39. Ying Liu, X., J. Wu and Z. hua Zhou, "Exploratory Under-Sampling for Class-Imbalance Learning", *Industrial Conference on Data Mining*, pp. 965–969, 2006.

40. Kohavi, R., P. Langley and Y. Yun, "The Utility of Feature Weighting in Nearest-Neighbor Algorithms", *Proceedings of the Ninth European Conference on Machine Learning*, Prague, 23-25 April 1997, pp. 85-92, Springer, London, 1997.

41. Zhang, J. and I. Mani, "kNN Approach to Unbalanced Data Distributions: A Case Study involving Information Extraction", *Worksop on Learning from Imbalanced Datasets II*, 2003.

42. Sun Kim, M., "An Effective Under-Sampling Method for Class Imbalance Data Problem", *Proceedings of the Eighth Symposium on Advanced Intelligent Systems*, Sockcho City, 6-8 September 2007, pp. 825–829, 2007.

43. Cour, T., B. Sapp and B. Taskar, "Learning From Partial Labels", *Journal of*

Machine Learning Research, Vol. 999999, pp. 1501–1536, 2011.

44. Zhu, X. and Z. Ghahramani, *Learning From Labeled and Unlabeled Data with Label Propagation*, Technical Report, Carnegie Mellon University, 2002.