

MASTERS THESIS

Hypergraph-driven Semi-Supervised learning approaches for variants of SVMs

by

Dev Nirwal

Under the Supervision of

Dr. Reshma Rastogi

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Science in Computer Science

to the



Department of Computer Science

Faculty of Mathematics and Computer 110068, India

December, 2024

Certificate

This is to certify that the thesis entitled "**Hypergraph-driven Semi-Supervised learning approaches for variants of SVMs**" submitted by **Dev Nirwal** (Enrollment No. SAU/CS(M)/2022/03) the Department of Computer Science, Faculty of Mathematics and Computer Science, South Asian University, New Delhi, 110068, India in partial fulfilment of the requirements for the award of the degree of **Master of Science in Computer Science**, is a record of the bonafide work carried out by him under my supervision and guidance.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree diploma.

Dr. Reshma Rastogi
(Supervisor)

Department of Computer Science
Faculty of Mathematics and Computer Science
South Asian University, New Delhi, India
December, 2024

Declaration

I hereby declare that the thesis entitled "**Hypergraph-driven Semi-Supervised learning approaches for variants of SVMs**" being submitted to the Department of Computer Science, Faculty of Mathematics and Computer Science, South Asian University, New Delhi in partial fulfilment of the requirements for the award of the degree of **Master of Science in Computer Science** contains the original work carried out by me under the supervision of Dr. Reshma Rastogi. The research work reported in this thesis is original and has not been submitted either in part or full to any university or institution for the award of any degree or diploma.

Dev Nirwal

SAU/CS(M)/2022/03

Acknowledgement

First and foremost, I would like to thank my supervisor, Dr Reshma Rastogi, for her continuous optimism concerning this work, providing me with excellent guidance and atmosphere for this project. She let me experience the research and deal with the practical issues beyond the textbooks, patiently corrected my mistakes and helped me develop my background in machine learning.

I would also like to thank my colleague-junior-brother like friend Ankush Bisht and my lab-mates Dr Sanjay Kumar, Sambhav Jain and Mamta Bhattarai Lamsal, and Sayanta Chowdhury for always cheering me up and supporting me throughout.

Dev Nirwal

SAU/CS(M)/2022/03

December, 2024

Abstract

This dissertation explores and extends existing work by developing two novel semi-supervised frameworks based on applications of hypergraphs in machine learning. Improved Hypergraph Laplacian Support Vector Machine (IHLSVM) and Hypergraph Regularized Semi-Supervised Least Squares Twin Support Vector Machine for Multi-label Learning (HMLLSTSVM). The IHLSVM framework combines Laplacian and hypergraph representations to better capture pairwise and higher-order interactions in the data, offering a robust approach to pattern classification. Meanwhile, HMLLSTSVM leverages hypergraph Laplacians and least-squares loss for efficient and accurate multilabel learning, particularly in scenarios with missing or sparse label information. Experimental evaluations on benchmark datasets validate the proposed methods' superior classification and multilabel learning capabilities, highlighting their efficacy in real-world applications like medical diagnosis, text classification, and image annotation.

List of Publications

- R. Rastogi and **Dev Nirwal**, “Hypergraph Regularized Semi-supervised Least Squares Twin Support Vector Machine for Multilabel Classification,” *Lecture Notes in Computer Science*, pp. 223–237, Dec. 2024. (ICPR)
- R. Rastogi and **Dev Nirwal**, “Improved Hypergraph Laplacian Based Semi-supervised Support Vector Machine,” *Lecture Notes in Computer Science*, pp. 242–255, Dec. 2024. (ICPR)

Contents

Certificate	iii
Declaration	iv
Acknowledgement	v
Abstract	vi
List of Publications	vii
List of Figures	xi
List of Tables	xii
1 Introduction	1
2 Literature Review	4
2.1 Hypergraph	4
2.2 Support Vector Machines (SVM)	5
2.3 Twin Support Vector Machines (TSVM)	8
2.4 Least Square Twin Support Vector Machine	11
2.5 Manifold Learning	14

2.6	Laplacian Support Vector Machine (LapSVM)	15
3	Hypergraph regularized Semi-supervised Least Squares Twin Support Vector Machine for Multilabel Classification (HMLLSTSVM)	18
3.1	Local geometric information	18
3.2	Structural information of data	19
3.3	Formulation of the proposed model HMLLSTSVM:	20
3.4	Algorithm	23
3.5	Non-Linear version of HMLLSTSVM	24
3.6	Experiments	25
3.6.1	Compared Algorithms	25
3.6.2	Parameter Selection	26
3.6.3	Datasets	26
3.6.4	Evaluation Metrics	28
3.6.5	Results and Discussion	29
4	Improved Hypergraph-Laplacian-based Semi-Supervised Support Vector Machine (IHLSVM)	32
4.1	Hypergraph Support Vector Machine	32
4.2	Improved Hypergraph Laplacian SVM (IHLSVM)	33
4.3	Multi-category and Multi-label Classification	34
4.4	Experiments and Results	35
4.4.1	UCI Datasets	35
4.4.1.1	Results for Binary Classification.	35
4.4.1.2	Results for Multi-class Classification	36
4.4.2	Results for Multi-label Classification	38
5	Conclusion and Future Works	44

Bibliography	46
--------------	----

List of Figures

- 3.1 Consolidated average rank of participating algorithms across all metrics. 30
- 3.2 HMLLSTSVM comparison with competing algorithms using Nemenyi test, CD=2.32. 31

- 4.1 Results for MNIST Fashion 39
- 4.2 Sensitivity Analysis for binary datasets German and Austrlian. 41
- 4.3 Sensitivity Analysis for multi-category datasets Ecoli and Wine 42
- 4.4 IHLSVM comparison with competing algorithms using the Nemenyi test. 43

List of Tables

3.1	Particulars of the Benchmark multi-label datasets	27
3.2	Results for Multi-Label classification	27
3.3	Results for Multi-Label classification with different percentages of labelled samples.	27
3.4	Results for Multi-Label classification	29
4.1	UCI datasets	36
4.2	Classification Accuracy Results for Binary Datasets	37
4.3	Classification Accuracy Results for Multi-class Datasets	38
4.4	Comparison of Results for MNIST Fashion	39
4.5	Description of Multi-label Datasets	39
4.6	Results of Multi-label Dataset Birds and Flags	39
4.7	Results of Multi-label Dataset Image and Emotions	40
4.8	Results of Multi-label Datasets CAL500 and Enron	40

Chapter 1

Introduction

Support vector machine (SVM) proposed by Vapnik[8], a parallel plane classifier, is a supervised learning algorithm which among all possible supporting hyper-planes identifies a maximum margin classifier for the pattern classification problem. Depending upon the structure of data, either hard margin or soft margin SVMs are obtained via solving a Quadratic Programming Problem (QPP) in the dual space. There exist many non-parallel classifiers like generalized eigenvalue proximal support vector machine (GEPSVM)[26] and Twin Support Vector Machines (TWSVM)[19]. GEPSVM tends to solve a pair of eigenvalue-eigenvector problems and Twin support vector machines[19], which is similar to SVM but solves a pair of QPPs, each smaller in size as compared to QPP solved via SVM, which makes it four times faster than SVM. Although SVM-based classifiers offer several benefits, the primary drawback is requiring a substantial volume of labelled training data. However, in practice, it is challenging to obtain adequately labelled data, which is valid for any real-life application.

Semi-supervised learning (SSL) combines both supervised and unsupervised learning to solve many real-life problems. SSL has applications in computer vision, the medical field, and text classification, where it is challenging to have labelled data. The underlying distribution of unlabelled data captures structural information, enhancing the performance of a learning algorithm[41]. Numerous techniques exist for SSL, such as graph-based approaches, co-training, self-training, and Gaussian mixture models. In many SSL methods, the most prevalent assumptions about data distributions are the Cluster and Manifold assumptions. In the cluster assumption, we can divide the data into several discrete clusters; the data points within a cluster are more likely to have the same output labels. Algorithms such as Transductive SVM[17] or S^3VM

[3] use the cluster assumption to find the optimal hyperplane in the SSL setting. Data points on a manifold structure have significantly lower dimensions compared to the input space[14]. Thus, manifold assumption primarily considers the underlying local characteristics of training data. It represents the local regularity of a decision function, indicating that the samples within a small local region should exhibit similarity.

Laplacian Support Vector Machine (LapSVM)[28] algorithm tries to learn the underlying manifold structure from the unlabeled data and combines it with traditional SVM. A regularisation term was considered to keep the decision function and manifold structure smooth, which avoids overfitting. On the lines of LapSVM[28], several other variations such as Laplacian p-norm proximal SVM (LapPPSVM)[44], Laplacian twin support vector machine (LapTSVM) [33], Laplacian least square twin support vector machine (Lap-LSTSVM)[5], and LapLSTSVM[20], exists. Many researchers proposed another algorithm to overcome this challenge, the Hyper-graph Support vector machine (HGSVM)[42].

Hypergraph-based SVM(HGSVM) is a graph-regularized manifold learning algorithm for semi-supervised settings discussed in [42]. Authors in ([46]) proposed the Structural Least Squares Twin Support Vector Machine (SLSTSVM), which is an amalgamation of the Structural Twin Support Vector Machine (STSVM) [34] and the Least Squares Twin Support Vector Machine (LSTSVM)[23]. SLSTSVM ([46]) has the upper hand over other algorithms as it takes advantage of the structural information of the data, which boosts the model’s generalisation capability. In addition, it also owns less computational complexity as the LSTSVM-based model finds resulting hyperplanes by solving two systems of linear equations rather than one sizeable quadratic programming problem (QPP) or two QPPs. Researchers across the globe utilised the variants of Twin Support Vector Machine in the domain of Human Activity Recognition, such as Robust Least Squares Twin Support Vector Machine ([21]) and Robust Parametric Twin Support Vector Machine ([22]).

Multi-label learning is supervised learning where each instance belongs to one or more than one label simultaneously. It has attracted increasing attention from researchers because of its importance and heavy usage in real life applications such as image annotation [4], [12], video annotation [18], [32], text categorization [27], [49] and bio-informatics [7]. Many multi-label classifiers face challenges with class imbalance, mainly when dealing with a large number of labels and a low density of positive instances— i.e. when there are far more negative examples than

positive ones. This imbalance often leads the classifier to be biased towards the negative class. To address this issue, various research efforts have been made. Some other recent approaches of multilabel learning are discussed by authors in [24, 25, 35, 36, 38, 39].

Chapter 2

Literature Review

In this chapter, we are going to look over existing research, key concepts and theoretical framework relevant to the study.

2.1 Hypergraph

Given a graph $G = (V, E)$, $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_n\}$ are the vertex and hyperedges set. We can define a vertex-edge matrix H such that

$$H = h(v, e) = \begin{cases} 1, & v \in e \\ 0, & v \notin e \end{cases} \quad (2.1)$$

The degree of hyperedge $\delta(e)$ is defined as the number of vertices a hyperedge contains.

$$\begin{aligned} \delta(e) &= \sum_{v \in e} h(v, e) \\ d(v) &= \sum_{v \in e, e \in E} w(e) = \sum_{e \in E} w(e) h(v, e) \end{aligned}$$

The following formula can define the hyperedge weight $w(e)$:

$$w(e) = \frac{1}{\delta(e)(\delta(e) - 1)} \sum_{\{v_i, v_j\} \in e} \exp\left(-\frac{\|x_i - x_j\|^2}{\mu}\right)$$

Similar to a simple graph, the Hypergraph's Laplacian matrix can be defined as [50]:

$$HL = D_v - HWD_e^{-1}H^T \quad (2.2)$$

where D_v, D_e, W are the diagonal matrices composed of $d(v), \delta(e)$ and $w(e)$ respectively. According to Zhou's ([50]), the Laplacian regularized Hypergraph matrix is:

$$HL = I - D_v^{-\frac{1}{2}}HWD_e^{-1}H^TD_v^{-\frac{1}{2}} \quad (2.3)$$

2.2 Support Vector Machines (SVM)

Support Vector Machine (SVM)[8] is a supervised learning algorithm with many applications in image classification, regression, etc. SVM attempts to minimise empirical error while maximising the distance between two bounding hyperplanes. Given a set of training data $X = \{(x_1, y_1) \dots (x_l, y_l)\}$ where $x_i \in R^n$, $y_i \in \{-1, +1\}$, $i = 1, 2, \dots, l$. The SVM solves the following quadratic programming problem.

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2}w^Tw + C \sum_{i=1}^l \xi_i \\ \text{s.t.} \quad & y_i(w^Tx_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \dots, l \end{aligned} \quad (2.4)$$

where the slack variable (error term) $\xi = (\xi_1, \xi_2, \dots, \xi_l)^T$ corresponds to mis-classification error in training data. The separating hyperplane is given by

$$w^Tx + b = 0 \quad (2.5)$$

where $w \in R^n$ and $b \in R$.

In the equation 2.4, $C > 0$ denotes a trade-off parameter where a considerable value of C emphasises the classification error, while a smaller one places more importance on the classification margin. However, in practical situations, rather than solving the primal problem described in 2.4, we solve the dual problem of SVM to get an appropriate soft margin classifier. To obtain the dual

formulation, we construct the Lagrangian and set its gradient w.r.t w , b and ξ equal zero to obtain the Karush-Kuhn-Tucker (K.K.T.) conditions. Thus, the dual problem is given by

$$\begin{aligned}
 \max_{\alpha} \quad & \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j \\
 \text{s.t.} \quad & \sum_{i=1}^l \alpha_i y_i = 0, \\
 & 0 \leq \alpha_i \leq C, \text{ for } i = 1, 2, \dots, l
 \end{aligned} \tag{2.6}$$

where α_i 's are the Lagrange multipliers. SVMs are great for many real-world tasks but suffer from large memory requirements and CPU time when there is an increase in sample points. Classification is done in some high-dimensional inner product space using a non-linear kernel for most real-world datasets that are not linearly separable. The linear classifier in a high-dimension space corresponds to a non-linear decision surface in the input space. To extend the above results to the nonlinear classifier, we define a nonlinear mapping $\phi : R^N \rightarrow H$, where H is some high dimensional feature space. The optimal hyperplane for the transformed training set $\{\phi(x_i), y_i : i = 1, 2, \dots, l\}$ is obtained by solving the following optimisation problem in the feature space

$$\begin{aligned}
 \max_{\alpha} \quad & \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\
 \text{s.t.} \quad & \sum_{i=1}^l \alpha_i y_i = 0, \\
 & 0 \leq \alpha_i \leq C, \text{ for } i = 1, 2, \dots, l
 \end{aligned} \tag{2.7}$$

Here, we observe the inner product of the transformed samples. i.e., $\phi(x_i)^T \phi(x_j)$ in the objective function can further be replaced by the kernel function.

Kernel Trick A Kernel is a function $k : R^N \times R^N \rightarrow R$ such that

$$k(x_1, x_2) = \phi(x_1)^T \phi(x_2), \forall x_1, x_2 \in R^N$$

where $\phi : R^N \rightarrow F$ and F is some inner product space, also known as a feature space. The kernel

matrix, K is defined as:

$$[K_{i,j}] = k(x_i, x_j) = \phi(x_i)^T \phi(x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

Some commonly used kernels include the linear kernel given by

$$k(x_i, x_j) = \langle x_i, x_j \rangle \quad (2.8)$$

the polynomial kernel (of degree 'd'), given by

$$k(x_i, x_j) = (a + \langle x_i, x_j \rangle)^d, \text{ where } a > 0 \quad (2.9)$$

and the Gaussian kernel, given by

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \text{ where } \sigma > 0 \quad (2.10)$$

Using a non-linear kernel in the dual problem is equivalent to mapping the original data points into high-dimensional feature space and finding a linear separating hyperplane. Using Mercer's condition satisfiable by any general kernel mapping, finding a separating hyperplane corresponds to finding a linear hyperplane in a higher (possibly infinite) dimensional feature space. With the above set of notations, problem 2.7 mayb be rewritten in terms of a *kernel matrix* K as

$$\max_{\alpha} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (2.11)$$

$$\text{s.t. } \sum_{i=1}^l \alpha_i y_i = 0,$$

$$0 \leq \alpha_i \leq C, \text{ for } i = 1, 2, \dots, l$$

where $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. Once the optimised value of Lagrange multipliers is obtained, which is represented as a column matrix α^* by solving the problem 2.11, the class label of a new data

point $x \in R^N$ is decided as follows.

$$f^*(x) = \begin{cases} 1, & \text{if } \text{sign} \left(\sum_{i=1}^l k(x_i, x) \alpha_i^* + b \right) > 0 \\ -1, & \text{if } \text{sign} \left(\sum_{i=1}^l k(x_i, x) \alpha_i^* + b \right) \leq 0 \end{cases} \quad (2.12)$$

2.3 Twin Support Vector Machines (TSVM)

Twin Support Vector Machine (TWSVM)[15] is a binary classifier that does the binary classification using two non-parallel hyperplanes instead of a single hyperplane as in case of a conventional Support Vector Machine. The two non-parallel hyperplanes are obtained by solving two Quadratic Programming Problems (QPP) of smaller size compared to a single large QPP solved by the conventional SVMs.

Consider a binary classification problem of classifying m_1 data-points belonging to class +1 and m_2 data points belonging to class -1 in the n -dimensional real space R^n . Let matrix A in $R^{m_1 \times n}$ represents m_1 datapoints of class +1 and matrix B in $R^{m_2 \times n}$ represent the m_2 data points of class -1. Given the above stated binary classification problem, the linear TWSVM seeks two non-parallel hyperplanes in R^n .

The two hyperplanes are defined as:

$$x^T w_{(1)} + b_{(1)} = 0 \quad \text{and} \quad x^T w_{(2)} + b_{(2)} = 0 \quad (2.13)$$

such that each hyperplane is closest to the data points of one class and farthest from the data points of the other class. A new data-point is assigned to class +1 or -1 depending upon its proximity to the two non-parallel hyperplanes.

The idea in linear TWSVM [15] is to solve two QPPs (2.14) and (2.15) with the objective function corresponding to one class and constraints corresponding to the other class.

(TWSVM 1)

$$\begin{aligned} \min_{w_{(1)}, b_{(1)}} \quad & \frac{1}{2} (Aw_{(1)} + e_1 b_{(1)})^T (Aw_{(1)} + e_1 b_{(1)}) + C_1 e_2^T y_2 z \\ \text{subject to} \quad & - (Bw_{(1)} + e_2 b_{(1)}) + y_2 \geq e_2, \quad y_2 \geq 0 \end{aligned} \quad (2.14)$$

(TWSVM 2)

$$\begin{aligned} \min_{w_{(2)}, b_{(2)}} \quad & \frac{1}{2} (Bw_{(2)} + e_2 b_{(2)})^T (Bw_{(2)} + e_2 b_{(2)}) + C_2 e_1^T y_1 \\ \text{subject to} \quad & (Aw_{(2)} + e_1 b_{(2)}) + y_1 \geq e_1, \quad y_1 \geq 0 \end{aligned} \quad (2.15)$$

The Wolfe dual of primal problem (2.14) and (2.15) can be given as:

$$\begin{aligned} \max_{\alpha} \quad & e_2^T \alpha - \frac{1}{2} \alpha^T (G(H^T H)^{-1} G^T) \alpha \\ \text{subject to} \quad & 0 \leq \alpha \leq C_1 \end{aligned} \quad (2.16)$$

and

$$\begin{aligned} \max_{\gamma} \quad & e_1^T \gamma - \frac{1}{2} \gamma^T P(Q^T Q)^{-1} P^T \gamma \\ \text{subject to} \quad & 0 \leq \gamma \leq C_2. \end{aligned} \quad (2.17)$$

Solving the Wolfe's dual obtained from equation (2.14) and (2.15), gives:

$$v_1 = -(H^T H)^{-1} G^T \alpha, \quad \text{where } v_1 = \begin{bmatrix} w_{(1)} \\ b_{(1)} \end{bmatrix} \quad (2.18)$$

$$v_2 = -(Q^T Q)^{-1} P^T \beta, \quad \text{where } v_2 = \begin{bmatrix} w_{(2)} \\ b_{(2)} \end{bmatrix} \quad (2.19)$$

where $G = [B \ e]$, $H = [A \ e]$, $P = [A \ e]$, $Q = [B \ e]$, α and β are dual multipliers.

A new point $x \in R^m$ is assigned to class i ($i = +1$ or -1), depending on which of the two hyperplanes the point is closer to, i.e.,

$$Class(i) = \arg \min_{i=1,2} \frac{|x^T w_{(i)} + b_{(i)}|}{\|w_{(i)}\|_2} \quad (2.20)$$

where $|x|$ is the absolute distance of x in R^2 .

For Non-linear TWSVM

For Non-linear TWSVM we try to find the following kernel generated surface:

$$K(x^T, C^T)u_1 + b_1 = 0 \quad \text{and} \quad K(x^T, C^T)u_2 + b_2 = 0 \quad (2.21)$$

where K is an appropriately chosen Kernel and $C^T = [A, B]^T$ by solving the following pair of optimization problems.

2.3.0.0.1 Optimization 1:

$$\min_{u_{(1)}, b_{(1)}} \frac{1}{2} (K(A, C^T)u_1 + eb_1)^T (K(A, C^T)u_1 + eb_1) + C_1 e_2^T y_1 \quad (2.22)$$

$$\text{subject to} \quad - (K(B, C^T)u_1 + eb_1) + y_1 \geq e_2, \quad y_1 \geq 0$$

2.3.0.0.2 Optimization 2:

$$\min_{u_{(2)}, b_{(2)}} \frac{1}{2} (K(B, C^T)u_2 + eb_2)^T (K(B, C^T)u_2 + eb_2) + C_2 e_1^T y_2 \quad (2.23)$$

$$\text{subject to} \quad (K(A, C^T)u_2 + eb_2) + y_2 \geq e_1, \quad y_2 \geq 0.$$

where $C_1 > 0$ and $C_2 > 0$.

Assuming $S = [K(A, C^T) e_1]$, $R = [K(B, C^T), e_2]$ and $z = [u_1, b_1]$, Then Wolfe dual of QPP (2.22) is given by:

$$\max_{\alpha} \quad e_2^T \alpha - \frac{1}{2} \alpha^T R (S^T S)^{-1} R^T \alpha \quad (2.24)$$

$$\text{subject to} \quad 0 \leq \alpha \leq c_1$$

After solving the above dual, we can obtain:

$$z_1 = -(S^T S)^{-1} R^T \alpha \quad (2.25)$$

Similarly, as in (2.22) and (2.3.0.0.2) is solved in the dual space and $z = [u_2, b_2]$ is given as:

$$z = (N^T N)^{-1} L^T \gamma \quad (2.26)$$

where $L = [K(A, C^T) e_1]$, $N = [K(B, C^T), e_2]$, and γ is the optimized dual variable.

Once (2.25) and (2.26) are solved to obtain the surfaces, a new pattern $x_i \in R^n$ is assigned to class +1 or class -1 in a manner similar to the linear case.

2.4 Least Square Twin Support Vector Machine

Twin Support Vector Machine formulation extended in least square sense, which is named as Least Squared Twin Support Vector (LSTSVM). This formulation leads to extremely simple and fast algorithm for generating binary classifiers based on two non-parallel hyperplanes. Here we attempt to solve two modified primal problems reduces to solving tow systems of linear equations as opposed to solving two quadratic programming problems along with two systems of linear equations in TSVM.

TSVM algorithm enhanced in least squares TSVM (LSTSVM) using the idea proposed in Fung and Mangasarian (2001) and Suykens and Vandewalle (1999). The primal QPPs of TSVM in least squares sense and solve them with equality constraints instead of inequalities of TSVM. As a result the solution of LSTSVM follows directly from solving two system of linear equations as opposed to solving two QPPs and two systems of linear equations in TSVM. We extended LSTSVM to handle nonlinear kernels whose solution also leads to systems of linear equations. The algorithm can accurately solve large datasets without any external optimizers. LSTSVM perform better than TSVM. There are two facts behind this:

1. LSTSVM surpasses TSVM in speed and gives very comparable classification accuracy.
2. TSVM has better generalization than conventional SVM and TSVM.

$$x^T w_{(1)} + b_{(1)} = 0 \text{ and } x^T w_{(2)} + b_{(2)} = 0 \quad (2.27)$$

such that each hyperplane is closest to data-points of one class and farthest from the data points of the other class. A new data-point is assigned to class +1 or -1 depending upon its proximity

to the two non-parallel hyperplanes.

The idea in linear LSTSVM is to solve two QPPs 2.28 and 2.29 with the objective function corresponding to one class and constraints corresponding to the other class.

Linear LSTSVM

$$\begin{aligned} \min_{w_{(1)}, b_{(1)}} \quad & \frac{1}{2} (Aw_{(1)} + e_1 b_{(1)})^T (Aw_{(1)} + e_1 b_{(1)}) + C_1 e_2^T y_2 \\ \text{subject to} \quad & (Bw_{(1)} + e_2 b_{(1)}) + y_2 = e_2 \end{aligned} \quad (2.28)$$

$$\begin{aligned} \min_{w_{(2)}, b_{(2)}} \quad & \frac{1}{2} (Bw_{(2)} + e_2 b_{(2)})^T (Bw_{(2)} + e_2 b_{(2)}) + C_2 e_1^T y_1 \\ \text{subject to} \quad & (Aw_{(2)} + e_1 b_{(2)}) + y_1 = e_1 \end{aligned} \quad (2.29)$$

taking gradient of above primal problem 2.28 and 2.29 can be given as

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = \left\{ \begin{bmatrix} B^T \\ e^T \end{bmatrix} [B \ e] + \frac{1}{C_1} \begin{bmatrix} A^T \\ e^T \end{bmatrix} [A \ e] \right\}^{-1} \begin{bmatrix} -B^T e \\ -m_2 \end{bmatrix} \quad (2.30)$$

We define $E = [A \ e]$ and $F = [B \ e]$, and then the solution of w_1 and b_1 computes as:

$$\begin{bmatrix} w^{(1)} \\ b^{(1)} \end{bmatrix} = - \left(F^T F + \frac{1}{C_1} E^T E \right)^{-1} F^T e \quad (2.31)$$

We get the solution of w_2 and b_2 as follows:

$$\begin{bmatrix} w^{(2)} \\ b^{(2)} \end{bmatrix} = \left(E^T E + \frac{1}{C_2} F^T F \right)^{-1} E^T e \quad (2.32)$$

A new point $x \in R^m$ is assigned to class i ($i = +1$ or -1), depending on which of the two

hyperplanes the point is closer to, i.e.,

$$Class(i) = \arg \min_{i=1,2} \frac{|x^T w^{(i)} + b^{(i)}|}{\|w^{(i)}\|_2} \quad (2.33)$$

where $|x|$ is the absolute distance of x in R^2 .

Non-Linear LSTSVM

LSTSVM also extends to the non-linear case, considering the following kernel-generated surfaces:

$$K(x^T, C^T)u_1 + b_1 = 0 \quad \text{and} \quad K(x^T, C^T)u_2 + b_2 = 0 \quad (2.34)$$

where K is an appropriately chosen Kernel and $C^T = [A, B]^T$, with the following pair of optimization problems.

$$K(x^T, C^T)u_1 + b_1 = 0 \quad \text{and} \quad K(x^T, C^T)u_2 + b_2 = 0 \quad (2.35)$$

where K is an appropriately chosen Kernel and $C^T = [A, B]^T$ with the following pair of optimization problems:

$$\begin{aligned} \min_{w^{(1)}, b^{(1)}} \quad & \frac{1}{2} (K(A, C^T)u_1 + e_1 b_1)^T (K(A, C^T)u_1 + e_1 b_1) + C_1 e_1^T y_1 \\ \text{subject to,} \quad & - (K(B, C^T)u_1 + e_2 b_1) + y_2 = e_2, \end{aligned} \quad (2.36)$$

$$\begin{aligned} \min_{w^{(2)}, b^{(2)}} \quad & \frac{1}{2} (K(B, C^T)u_2 + e_2 b_2)^T (K(B, C^T)u_2 + e_2 b_2) + C_2 e_1^T y_2 \\ \text{subject to,} \quad & (K(A, C^T)u_2 + e_1 b_2) + y_2 \geq e_1, \quad y_2 = 0. \end{aligned} \quad (2.37)$$

where $c_1 > 0$ and $c_2 > 0$. Assuming $S = [K(A, C^T) \ e_1]$, $R = [K(B, C^T), e_2]$, and $z = [u_1, b_1]$,

when we solve the gradient of (2.36) and (2.37), we get a solution for the nonlinear case similar to the linear case.

2.5 Manifold Learning

Given $X = \{(x_i, y_i) | x_i \in R^n\}$ be the set of l -labelled data where n is the number of dimensions for the training data and $y_i \in \{+1, -1\}$ and the set $U = \{x_i, i = l + 1, l + 2, \dots, m\}$ is the set of unlabelled data. Hence, the dataset $S = X \cup U$, where the first l data points are labelled, and the rest of the u data points are unlabelled. $K \in R^{m \times m}$ is the kernel matrix for m points corresponding to $k(x_i, x_j)$ $k : X \times X \rightarrow R$. L is the graph Laplacian associated with the given data S and is given by

$$L = D - W \quad (2.38)$$

where W is the adjacency matrix and D is the diagonal matrix with the degree of each node ($d_{ii} = \sum_{j=1}^m w_{ij}$, w_{ij} is the entry from the matrix W). Graph Laplacian can be mathematically represented in its normalised form as

$$L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \quad (2.39)$$

The target function that the learning algorithm must estimate is denoted as f , such that $f : X \rightarrow R$ where f is the vector of the n values on the training data.

Manifold learning[2] is a non-linear dimensionality reduction technique in which we assume the high dimensional data generally lie on a lower dimension manifold. The semi-supervised manifold regularisation framework can be defined as

$$f^* = \underset{f \in H_k}{\operatorname{argmin}} \sum_{i=1}^l V(x_i, y_i, f) + \delta_A \|f\|_A^2 + \delta_I \|f\|_I^2 \quad (2.40)$$

where $V(x_i, y_i, f)$ is the loss function, $\|f\|_A^2$ is the ambient norm. i.e., the norm of the function f in the Reproducing Kernel Hilbert Space(RKHS) enforces a smoothness condition on the possible outcome. $\|f\|_I^2$ is the intrinsic norm of the function f in the low dimensional manifold, which enforces a smoothness along the samples and is learned from the combination of labelled and unlabeled data. The tradeoff parameter δ_A and δ_I are hyper-parameters that control the weight

of ambient and intrinsic norm. According to the Representer theorem [28] the function f^* will expand in m terms as:

$$f^* = \sum_{i=1}^m \alpha_i^* k(x_i, x) \quad (2.41)$$

2.6 Laplacian Support Vector Machine (LapSVM)

LapSVM [28] is based on the principle of manifold learning and uses the representer theorem to solve the following optimisation similar to (2.40).

$$\begin{aligned} \min_{\alpha \in R^m, \xi \in R^l} \quad & \sum_{i=1}^l \xi_i + \gamma_A \alpha^T K \alpha + \gamma_I \alpha^T K L K \alpha \\ \text{s.t.} \quad & y_i \left(\sum_{j=1}^n \alpha_j k(x_i, x_j) + b \right) \geq 1 - \xi_i, \quad i = 1, 2, \dots, l \\ & \xi_i \geq 0, i = 1, 2, \dots, l \end{aligned}$$

where L is the regularised Graph Laplacian of the dataset derived in (2.39).

The Lagrangian L_l of the above problem is

$$\begin{aligned} L_l(\alpha, \xi, b, \beta, \delta) = & \sum_{i=1}^l \xi_i + \frac{1}{2} \alpha^T (2\gamma_A K + 2\gamma_I K L K) \alpha \\ & - \sum_{i=1}^l \beta_i \left(y_i \left(\sum_{j=1}^n \alpha_j k(x_i, x_j) + b \right) - 1 + \xi_i \right) \\ & - \sum_{i=1}^l \delta_i \xi_i \end{aligned} \quad (2.42)$$

where β & δ are set of n Lagrangian's multipliers. Now,

$$\begin{aligned} \frac{\partial L_l}{\partial b} = 0 & \implies \sum_{i=1}^l \beta_i y_i = 0, \\ \frac{\partial L_l}{\partial \xi_i} = 0 & \implies 1 - \beta_i - \delta_i = 0 \implies 0 \leq \beta_i \leq 1 \end{aligned}$$

Since $\delta_i, \beta_i \geq 0$ are Lagrangian multipliers, there's a bound on β_i . By substituting the above result

to (2.42), we get

$$\begin{aligned}
 L_l(\alpha, \beta) &= \frac{1}{2}\alpha^T(2\gamma_A K + 2\gamma_I K L K)\alpha \\
 &\quad - \beta_i(y_i(\sum_{j=1}^n \alpha_i k(x_i, x_j) + b) - 1)
 \end{aligned} \tag{2.43}$$

Let $J_L = [I \ 0]$ where $I \in R^{l \times l}$ and $0 \in R^{l \times u}$ which implies $J_L \in R^{l \times n}$. Moreover, $Y \in R^{l \times l}$ is a diagonal matrix made up by the labels $y_i, i = 1, \dots, l$. we can rewrite the above equation:

$$\begin{aligned}
 L_l(\alpha, \beta) &= \frac{1}{2}\alpha^T(2\gamma_A K + 2\gamma_I K L K)\alpha \\
 &\quad - \alpha^T K J_L^T Y \beta + \sum_{i=1}^l \beta_i
 \end{aligned} \tag{2.44}$$

Setting the derivative of the above equation wrt α to 0,

$$\begin{aligned}
 \frac{\partial L_l}{\partial \alpha} = 0 &\implies (2\gamma_A K + 2\gamma_I K L K)\alpha - K J_L^T Y \beta = 0 \\
 &\implies \alpha = (2\gamma_A I + 2\gamma_I K L)^{-1} J_L^T Y \beta
 \end{aligned} \tag{2.45}$$

Using the(2.43) and back substituting it in the Lagrangian which leads to β^* as the optimal solution, that is

$$\begin{aligned}
 &\max_{\beta \in R^l} \sum_{i=1}^l \beta_i - \frac{1}{2}\beta^T Q \beta \\
 &\quad s.t. \sum_{i=1}^l \beta_i y_i = 0 \\
 &\quad 0 \leq \beta_i \leq 1, \ i = 1, 2, \dots, l
 \end{aligned}$$

where $Q = Y J_L K (2\gamma_A I + 2\gamma_I K L)^{-1} J_L^T Y$. However, we are going with the primal solution of LapSVM[28], which is

$$\begin{aligned}
 \min_{\alpha \in R^n, b \in R} &\sum_{i=1}^l V(x_i, y_i, k_i^T \alpha + b) + \delta_A \alpha^T K \alpha \\
 &\quad + \delta_I (\alpha^T K + 1^T b) L (K \alpha + 1b)
 \end{aligned} \tag{2.46}$$

2.6 LAPLACIAN SUPPORT VECTOR MACHINE (LAP SVM)

We use the square hinge loss function to make our LapSVM problem continuous and differentiable in f .

$$\begin{aligned} \min_{\alpha \in R^n, b \in R} \frac{1}{2} & \left(\sum_{i=1}^l \max(1 - y_i(k_i^T \alpha + b), 0)^2 + \delta_A \alpha^T K \alpha \right. \\ & \left. + \delta_I (\alpha^T K + 1^T b) L(K \alpha + 1b) \right) \end{aligned} \quad (2.47)$$

We can solve such a convex problem using Newton's method or preconditioned conjugate gradient.

Chapter 3

Hypergraph regularized Semi-supervised Least Squares Twin Support Vector Machine for Multilabel Classification (HMLLSTSVM)

In this chapter we are going to deep-dive into the details of the proposed method. The problem of semi-supervised binary classification considers a set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l), x_{l+1}, \dots, x_n\}$, where $X_l = \{x_i : i = 1, 2, \dots, l\}$ symbolizes the l labeled data points in n dimension corresponding to class labels $Y_l = \{y_i \in [-1, 1] : i = 1, 2, \dots, l\}$ and $X_u = \{x_i : i = l+1, l+2, \dots, m\}$ symbolizes the unlabeled data. Hence, $X = X_l \cup X_u$. The central thought behind the proposed HMLLSTSVM model emerges from MLTSVM ([6]), KNNLSTSVM ([30]) and Laplacian Least Squares TSVM ([5]). Based on the structural information of data and label dimensionality, the proposed model finds l hyperplanes.

3.1 Local geometric information

To extract local geometric information, k -nearest neighbor graph is a mostly used technique ([11]). Here, intra-class and inter-class graphs with weights $W_{wc,ij}$ (W_{wc} represents weights within the class), and $W_{bc,il}$ (W_{bc} represents weights between the classes) respectively are considered for each

class r , $r = 1, 2, \dots, l$, where x_i and x_j is associated to class r , while x_l does not belong to class r .

$$W_{wc,ij} = \begin{cases} 1, & \text{if } x_i \text{ is } k\text{-nearest neighbors of } x_j \\ \text{or } x_j \text{ is } k\text{-nearest neighbors of } x_i \\ 0, & \text{otherwise} \end{cases}$$

$$W_{bc,il} = \begin{cases} 1, & \text{if } x_l \text{ is } k\text{-nearest neighbors of } x_i \\ \text{or } x_i \text{ is } k\text{-nearest neighbors of } x_l \\ 0, & \text{otherwise} \end{cases}$$

The inter-class weighting matrix W_{bc} is redefined with f_{rl} as:

$$f_{rl} = \begin{cases} 1, & \text{if } \exists i, W_{bc,il} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

3.2 Structural information of data

Using Ward's linkage clustering method [11], structural information of the samples is extracted. The mean and covariance matrix of each sub-cluster n_r of class r is given by :

$$u_{C_i} = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \quad (3.2)$$

$$\Sigma_{C_i} = \frac{1}{|C_i|} \sum_{x_j \in C_i} (x_j - u_{C_i})(x_j - u_{C_i})^T \quad (3.3)$$

here C_i , $i = 1, 2, \dots, n_r$ represents the cluster of samples with label r , n_r indicates the number of clusters, and u_{C_i} and Σ_{C_i} are the mean vector and covariance matrix of a particular cluster C_i . Further, we calculate $\Sigma_r = \Sigma_{C_1} + \Sigma_{C_2} + \dots + \Sigma_{C_{n_r}}$, which is the total covariance matrix corresponding to the cluster of samples with label r . This covariance matrix Σ_r comprises of the structural information of class r .

3.3 Formulation of the proposed model HMLLSTSVM:

Working on the lines of LapLSTSVM, we propose a model for multi-label learning which considers geometrical and structural information along with some percentage of known label set of samples. Further, the proposed model is in the spirit of least squares TSVM, which makes it time efficient as well.

$$\begin{aligned} \min_{(w_r, b_r, \xi_j)} \quad & \frac{1}{2} \sum_{i \in I_r} d_{ir} (w_r^T x_i + b_r)^2 + \frac{c_{1r}}{2} \sum_{j \in I'_r} \|\xi_j\|^2 + \frac{c_{2r}}{2} w_r^T \Sigma_r w_r \\ & + \frac{\lambda_r}{2} (\|w_r\|_2 + b_r^2) + \frac{1-\lambda_r}{2} (Xw_r + eb_r)^T HL_a (Xw_r + eb_r) \end{aligned}$$

subject to,

$$-f_{rj} (w_r^T x_j + b_r) + \xi_j = f_{rj}, \quad j \in I'_r \quad (3.4)$$

where $\xi = (\xi_1, \xi_2, \dots, \xi_s)$, s is number of samples in I'_r , $d_{ir} = \sum_{j \in I_r} W_{wc,ij}$, is an element of the intra-class weighting matrix for the samples in the r^{th} label set. The parameters w_r and b_r defines the hyperplane of the r^{th} label, c_{1r} and c_{2r} are the penalty parameters and λ_r is the regularization parameter and Σ_r denotes the covariance matrix of the samples belonging to the r^{th} label. Here, f_{rj} reflects rj^{th} entry of inter-class weighting matrix, λ_1 and λ_2 are the regularization parameters and also, $\lambda_1, \lambda_2 \in (0, 1]$. HL_a is the corresponding Hypergraph Laplacian matrix.

After substituting value of ξ_j from the equality constraint (3.4) in the objective function of the aforementioned problem we obtain the following unconstrained optimization problem:

$$\begin{aligned} L = \frac{1}{2} \sum_{i \in I_r} d_{ir} \|w_r^T x_i + b_r\|^2 + \frac{c_{1r}}{2} \sum_{j \in I'_r} \|f_{rj} (1 + w_r^T x_j + b_r)\|^2 + \frac{c_{2r}}{2} w_r^T \Sigma_r w_r \\ + \frac{\lambda_r}{2} (\|w_r\|^2 + b_r^2) + \frac{1-\lambda_r}{2} (Xw_r + eb_r)^T HL_a (Xw_r + eb_r) \end{aligned} \quad (3.5)$$

After computing the derivatives of the aforementioned Lagrangian w.r.t. w_r and b_r , and equating them equal to zero, we obtain the following equations:

3.3 FORMULATION OF THE PROPOSED MODEL HMLLSTSVM:

$$\begin{aligned} \frac{\partial L}{\partial w_r} = 0 \Rightarrow \sum_{i \in I_r} d_{ir} x_i (w_r^T x_i + b_r) + c_{1r} \sum_{j \in I'r} f_{rj} x_j (1 + w_r^T x_j + b_r) \\ + c_{2r} \Sigma_r w_r + \lambda_r w_r + (1 - \lambda_r) X^T H L_a (X w_r + e b_r) = 0. \end{aligned} \quad (3.6)$$

$$\begin{aligned} \frac{\partial L}{\partial b_r} = 0 \Rightarrow \sum_{i \in I_r} d_{ir} (w_r^T x_i + b_r) + c_{1r} \sum_{j \in I'r} f_{rj} (1 + w_r^T x_j + b_r) + \lambda_r b_r \\ + (1 - \lambda_r) X^T H L_a (X w_r + e b_r) = 0. \end{aligned} \quad (3.7)$$

Let

$$H = \begin{bmatrix} X_{I_r} \\ e \end{bmatrix},$$

where X_{I_r} represents the data with label r . After combining (3.6) and (3.8), we obtain:

$$\begin{aligned} D_r \sum_{i \in I_r} \begin{bmatrix} x_i \\ 1 \end{bmatrix} \begin{bmatrix} x_i^T & 1 \end{bmatrix} \begin{bmatrix} w_r \\ b_r \end{bmatrix} + c_{1r} F_r \sum_{j \in I'r} \begin{bmatrix} x_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^T & 1 \end{bmatrix} \begin{bmatrix} w_r \\ b_r \end{bmatrix} \\ + c_{1r} F_r \sum_{j \in I'r} \begin{bmatrix} x_j \\ 1 \end{bmatrix} + \lambda_r \begin{bmatrix} w_r \\ b_r \end{bmatrix} + c_{2r} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_r \\ b_r \end{bmatrix} \\ + (1 - \lambda_r) H_r H L_a H_r^T \begin{bmatrix} w_r \\ b_r \end{bmatrix} = 0, \end{aligned}$$

Now, denoting

$$z_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}, \quad J_r = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \quad \theta_r = \begin{bmatrix} w_r \\ b_r \end{bmatrix}.$$

$$\begin{aligned} D_r \sum_{i \in I_r} z_i z_i^T \theta_r + c_{1r} F_r \sum_{j \in I'r} z_j z_j^T \theta_r + c_{1r} F_r \sum_{j \in I'r} z_j + \\ \lambda_r \theta_r + c_{2r} J_r \theta_r + (1 - \lambda_r) H_r H L_a H_r^T \theta_r = 0 \end{aligned} \quad (3.8)$$

3.3 FORMULATION OF THE PROPOSED MODEL HMLLSTSVM:

On rearranging the terms in equation (3.8), we get

$$\begin{aligned} \left(D_r \sum_{i \in I_r} z_i z_i^T + c_{1r} F_r \sum_{j \in I'_r} z_j z_j^T + \lambda_r I + c_{2r} J_r + (1 - \lambda_r) H_r H L_a H_r^T \right) \theta_r \\ = c_{1r} F_r \sum_{j \in I'_r} z_j \end{aligned} \quad (3.9)$$

Hence, after deriving the dual by applying Kuhn-Tucker conditions, we determine the equation of r^{th} hyperplane as :

$$\begin{aligned} \theta_r = c_{1r} \left(H_r^T D_r H_r + c_{1r} G_r^T F_r G_r + \lambda_r I + c_{2r} J_r \right. \\ \left. + (1 - \lambda_r) H_r^T H L_a H_r \right)^{-1} G_r^T F_r e, \end{aligned} \quad (3.10)$$

where

- $\theta_r = \begin{bmatrix} w_r & b_r \end{bmatrix}^T$, which represents the r th hyperplane,
- $H_r = [X_{I_r}, e_{1r}]$ and $G_r = [X_{I'_r}, e_{2r}]$,
- e_{1r} and e_{2r} are vectors of ones of suitable dimensions,
- $D_r = \text{diag}(d_{1r}, d_{2r}, \dots, d_{Nr})$, $N = |I_r|$,
- $J_r = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}$,
- c_{1r} is the pre-specified penalty parameter,
- $F_r = \text{diag}(f_{r1}, f_{r2}, \dots, f_{rM})$, $M = |I'_r|$,
- e is a vector of ones of appropriate dimension.

It can be seen from our formulation of the proposed model that when the regularization parameter $\lambda_r = 1$, KNN-Based Multi-Label Least Squares Twin Support Vector Machine (KNNMLLSTSVM) is the limiting case of HMLLSTSVM. Also, unlike Multi-Label Structural Least Squares Twin Support Vector Machine ([1]), our proposed model HMLLSTSVM considers local geometric information and structural information among the samples. Hence, HMLLSTSVM is less sensitive to outliers.

3.4 Algorithm

Inputs: A training dataset \mathcal{D} and a testing dataset $\mathcal{D}^* = \{(X_i, Y_i)\}$, where $i = 1, 2, \dots, n$, and $Y_i \in \{1, -1\}$ for multi-label classification.

Output: A relevant set of labels for each test sample.

1. **For** each label Y_r in the label matrix Y , repeat steps 2 to 5:
2. Create a set of instances for each column Y_r .
3. Extract local geometric and structural information from the input samples for each column Y_r using Equations (3.1), (3.1), (3.1), and (3.2).
4. Compute the Hypergraph Laplacian matrix HL using:

$$HL = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}},$$

where D_v is the degree matrix, and W is the weighted adjacency matrix.

5. Utilizing the structural and geometric information ingrained in the data, determine the hyperplane for every label.
6. **For** each test sample t_i , repeat steps 7 and 8:
7. Calculate the distances from the hyperplane of each label using the equation:

$$d_r = \frac{|w_r^T t_i + b_r|}{\|w_r\|_2}, \quad r = 1, 2, \dots, L.$$

8. Find a relevant set of labels for a test point t_i based on an appropriate threshold value δ , defined as:

$$\delta = \min_{r=1, \dots, L} \frac{1}{\|w_r\|}.$$

For each label Y_r , if the distance between the test point t_i and the hyperplane is less than δ , assign the test point t_i to the r^{th} label.

3.5 Non-Linear version of HMLLSTSVM

In real-life classification problems, linear classifiers do not work very well. Therefore, we extend the linear HMLLSTSVM to its non-linear version. We can determine non-linear MLLSTSVM by using kernel-generated hyperplanes. The kernel takes its input vectors in the original space, and gives the dot product of vectors in the feature space. Formally, let we have data $x, x_2 \in X$, and a map $\phi : X \rightarrow \mathbb{R}^N$ then $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$ is a kernel function, \langle, \rangle is a inner product and u_r and b_r define the r^{th} hyper-surface parameter. Thus, the optimization problem of non-linear HMLLSTSVM is as follows:

$$\begin{aligned}
& \min_{(u_r, b_r, \xi_j)} \quad \frac{1}{2} \sum_{i \in I_r} d_{ir} (u_r^T \langle \phi(X), \phi(x_i) \rangle + b_r)^2 + \\
& \quad c_{1r} \sum_{j \in I'_r} \|\xi_j\|^2 + \frac{1}{2} \lambda_r (\|u_r\|_2^2 + b_r^2) + \frac{c_{2r}}{2} \sum_{j \in I'_r} u_r^T \Sigma_r^\phi u_r \\
& \quad + \frac{1-\lambda_r}{2} (Xw_r + eb_r)^T HL_a (Xu_r + eb_r) \\
& \text{subject to,} \\
& \quad -f_{rj} (u_r^T x_j + b_r) + \xi_j = f_{rj}, \quad j \in I'_r.
\end{aligned} \tag{3.11}$$

where $\xi = (\xi_1, \xi_2, \dots, \xi_s)$, s is number of samples in I'_r , Σ^ϕ is the covariance matrix obtained by the kernel Ward's linkage method in the kernel space of samples. After solving the optimizing problem, the parameters u_r and b_r are obtained as shown in the following formula:

$$\begin{aligned}
\begin{bmatrix} u_r \\ b_r \end{bmatrix} &= c_{1r} \left(H_r^{\phi T} D_r H_r^\phi + c_{1r} G_r^{\phi T} F_r G_r^\phi + \lambda_r I + c_{2r} J_r^\phi \right. \\
& \quad \left. + (1 - \lambda_r) H_r^{\phi T} HL_a H_r^\phi \right)^{-1} G_r^{\phi T} F_r e,
\end{aligned} \tag{3.12}$$

where $H_r^\phi = [\langle \phi(X_{I_r}), \phi(X) \rangle, e_{1r}]$, $G_r^\phi = [\langle \phi(X_{I'_r}), \phi(X) \rangle, e_{2r}]$, $J_r^\phi = \begin{bmatrix} \Sigma_r^\phi & 0 \\ 0 & 0 \end{bmatrix}$.

Here, c_{1r} is the penalty factor, and F_r , e_{1r} , and e_{2r} are similar to the linear case.

3.6 Experiments

The experiments are performed using five-fold cross-validation in MATLAB version 2020a under a Microsoft Windows environment on a machine with a 3.40 GHz CPU and 16 GB RAM on six well-known multi-label datasets, which are taken from the site <http://mulan.sourceforge.net/datasets-mlc.html>.

Since our proposed model is based on a multi-label twin support vector machine. So, we compared the algorithms, which are based on a multi-label twin support vector machine.

3.6.1 Compared Algorithms

1. **Multi-label Twin Support Vector Machine (MLTSVM) ([6]):** MLTSVM is an extension of TSVM in multilabel domain. It discovers multiple non-parallel hyperplanes to procure multi-label information embedded in the data.
2. **KNN-based Multi-Label Twin Support Vector Machine (KNNMLTSVM) ([29]):** KNNMLTSVM is an enhanced version of MLTSVM to tackle multi-label classification. It determines multiple non-parallel hyperplanes, which contain the structural and local geometric information ingrained in the data.
3. **Multi-label learning with label-specific features(LIFT) [48]** To discover label-specific features, it employs a clustering approach. As the basic learner in this case, they employed linear kernel SVM. $r = 0.1$.
4. **Learning Label Specific Features(LLSF) [13]** To train the multi-label classification model, it makes use of the idea of learning label-specific characteristics. The tuning for the parameter λ is $2^{-10}, \dots, 2^{10}$.
5. **KNN-based Multi-Label Least Squares Laplacian Twin Support Vector Machine [11]:** It is an improved version of KNNMLTSVM. It is computationally fast, as it solves a system of linear equations rather than QPPs. Thus, it considers the square loss function instead of the hinge loss function. we have used LAP as an acronym while reporting results.
6. **Discriminatory Label-specific Weights for Multi-label Learning with Missing Labels (CIMML) [37]:** Multi-label learning algorithms that handle the datasets class

imbalance issue by assigning favourable weights for positive labels to compensate for their scarce occurrence.

7. Hypergraph based Least Squares Twin Support Vector Machine (HMLLSTSVM)

(Proposed): This is our proposed semi-supervised algorithm as it learns from both labelled as well as unlabeled information using Hypergraph; we have used HYP as acronym while reporting results.

3.6.2 Parameter Selection

We used the RBF kernel, and the value of kernel parameter and the penalty factor is tuned for the set $\{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4\}$, and $\{2^{-5}, \dots, 2^5\}$ respectively. The value of k (number of neighbors) is tuned for the set $\{2, 3, 4, 5, 6, 8, 10\}$. Expressly, we set $k=4, 6, 8, 2, 4, 8$ for emotions, flags, yeast, and image respectively. The regulariserWe for variance-covariance is tuned for the set $\{2^{-5}, \dots, 2^5\}$. The regularizer for Hypergraph Laplacian, which is λ is kept low, and is set to e^{-4} . Once, the optimal set of parameters is obtained, it is used to learn the final decision function.

3.6.3 Datasets

In our experiments, we consider four UCI multi-label datasets shown in the Table 3.1 below. The datasets such as **Image** and **flags** are from the image domain. **Emotions** dataset belongs to the music domain. **Yeast** dataset is associated with a biological field, where each instance represents a yeast gene, and labels denote the functional group of the corresponding yeast gene. The number of instances, the number of features, the number of class labels, and the cardinality of each dataset is observed in Table 3.1. The computation of cardinality is performed using the equation below:

$$Cardinality = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^L y_{ij} \quad (3.13)$$

where N is the number of total instances of a dataset, L is the number of labels, and y_{ij} is a binary variable indicating the relevance of j^{th} label for the i^{th} instance.

Dataset	Instances	Features	Labels	Cardinality
Emotions	593	72	6	1.87
Flags	194	19	7	3.392
Image	2000	294	5	1.24
Yeast	2417	103	14	4.24

Table 3.1: Particulars of the Benchmark multi-label datasets

Dataset	Metric	MLTSVM	CIMML	MLTSVMKNN	LLSF	LIFT	LAP	HYP
Flags	HL(↓)	0.3888	0.4235	0.3991	0.3532	0.4017	0.3165	0.3098
	F1(↑)	0.7048	0.6546	0.7007	0.6705	0.5969	0.7369	0.7407
	AP(↑)	0.5873	0.6174	0.5894	0.6028	0.4792	0.5934	0.6007
	RL(↓)	0.8385	0.4941	0.8458	0.9382	0.9742	0.8257	0.8090
	AUC(↑)	0.5362	0.4979	0.5329	0.5581	0.6018	0.5559	0.5590
Emotions	HL(↓)	0.3454	0.2569	0.3772	0.2175	0.3394	0.4194	0.3384
	F1(↑)	0.6267	0.4757	0.6096	0.6218	0.4588	0.5882	0.6277
	AP(↑)	0.5283	0.7051	0.5072	0.6095	0.3227	0.4847	0.5343
	RL(↓)	0.5184	0.27	0.5563	0.7622	0.9157	0.6179	0.5174
	AUC(↑)	0.7097	0.7023	0.6907	0.7179	0.6051	0.6601	0.7105
Image	HL(↓)	0.4947	0.3525	0.5321	0.2821	0.2952	0.5295	0.4102
	F1(↑)	0.4814	0.3349	0.4703	0.2649	0.2659	0.4622	0.5129
	AP(↑)	0.3851	0.5885	0.3641	0.3767	0.2664	0.3647	0.4247
	RL(↓)	0.7075	0.41	0.7339	0.8690	0.8840	0.7627	0.6249
	AUC(↑)	0.5862	0.5547	0.5703	0.5352	0.5417	0.5697	0.6536
yeast	HL(↓)	0.2485	0.2903	0.2594	0.2410	0.2484	0.2581	0.2005
	F1(↑)	0.666	0.5279	0.6607	0	0	0.664	0.6796
	AP(↑)	0.3882	0.6446	0.3842	1	1	0.3911	0.4194
	RL(↓)	0.7497	0.2479	0.7525	0.9582	0.9330	0.7310	0.7439
	AUC(↑)	0.5986	0.5547	0.5962	0.5389	0.6973	0.6043	0.6068

Table 3.2: Results for Multi-Label classification

Metric	LAP	HYP	LAP	HYP	LAP	HYP	LAP	HYP	LAP	HYP
Labelled	90%	90%	80%	80%	70%	70%	60%	60%	50%	50%
Flags:										
HL (↓)	0.3055	0.2996	0.3047	0.3018	0.3040	0.3017	0.3092	0.3129	0.3196	0.3248
F1 (↑)	0.7433	0.7461	0.7428	0.7444	0.7420	0.7420	0.7382	0.7357	0.7334	0.7315
AP (↑)	0.6015	0.6042	0.6018	0.6039	0.5992	0.6062	0.5979	0.5958	0.5962	0.5981
RL (↓)	0.7940	0.7891	0.7946	0.7823	0.7914	0.7791	0.8083	0.8088	0.8118	0.7962
AUC (↑)	0.5692	0.5736	0.5691	0.5725	0.5716	0.5733	0.5622	0.5606	0.5540	0.5610
Emotions:										
HL (↓)	0.3485	0.2923	0.3499	0.3066	0.3738	0.3342	0.3696	0.3488	0.3848	0.3679
F1 (↑)	0.6245	0.6542	0.6243	0.6446	0.6096	0.6297	0.6122	0.6184	0.5995	0.6084
AP (↑)	0.5269	0.5585	0.5258	0.5501	0.5072	0.5253	0.5055	0.5202	0.5016	0.5107
RL (↓)	0.5246	0.4590	0.5247	0.4749	0.5575	0.5056	0.5516	0.5223	0.5818	0.5500
AUC (↑)	0.7033	0.7398	0.7042	0.7326	0.6904	0.7174	0.6968	0.7076	0.6816	0.6977
Image:										
HL (↓)	0.4181	0.2865	0.3975	0.3296	0.4650	0.4201	0.4702	0.4826	0.5030	0.5584
F1 (↑)	0.5189	0.5921	0.5156	0.5486	0.4370	0.4780	0.4045	0.4473	0.3632	0.4289
AP (↑)	0.4117	0.4876	0.4172	0.4378	0.3391	0.3746	0.3386	0.3369	0.2960	0.3108
RL (↓)	0.6025	0.4584	0.5972	0.5252	0.7546	0.6555	0.8250	0.7221	0.9223	0.7881
AUC (↑)	0.6538	0.7485	0.6858	0.7249	0.5794	0.6679	0.5685	0.6592	0.5338	0.5967
Yeast:										
HL (↓)	0.2571	0.2029	0.2656	0.2061	0.2741	0.2384	0.2706	0.2209	0.2741	0.2384
F1 (↑)	0.6636	0.6781	0.6580	0.6791	0.6509	0.6662	0.6533	0.6746	0.6509	0.6662
AP (↑)	0.3896	0.4158	0.3841	0.4126	0.3777	0.3932	0.3790	0.4024	0.3777	0.3932
RL (↓)	0.7362	0.7478	0.7453	0.7463	0.7698	0.7427	0.7606	0.7423	0.7698	0.7427
AUC (↑)	0.6019	0.6043	0.5976	0.6054	0.5891	0.5995	0.5920	0.6030	0.5891	0.5995

Table 3.3: Results for Multi-Label classification with different percentages of labelled samples.

3.6.4 Evaluation Metrics

Given a test dataset $T_s = \{x_i, y_i\}_{i=1}^{N_t}$ where $y_i \in \{-1, 1\}^m$. Let N_t , m , y_i , \hat{y}_i denote, respectively, the number of test data, the number of labels, the set of labels relevant to the i^{th} instance and the set of labels that are irrelevant to it. In addition, the function $f_y(x)$ is a real-valued function ($f : \mathbf{X} \times \mathbf{Y} \rightarrow \mathbb{R}$) that returns the confidence of being proper label of x and $\text{rank}_f(x, y)$ returns the rank of y in \mathbf{Y} based on the descending order induced from $f_y(x)$ and $h(\cdot)$ be the learned multi-label classifier.

We have used the following evaluation criteria to compare the performance of different algorithms

1. **Hamming Loss (HL)**: This metric indicates the fraction of incorrectly classified labels to the total number of labels.

$$HL = \frac{1}{N_t \times m} \sum_{i=1}^{N_t} \sum_{j=1}^m [h_j(x_i) \neq y_{ij}] \quad (3.14)$$

2. **F1 Score (F1)**

This measure is the harmonic mean between recall and precision as defined in Eq. (3.16):

$$\text{prec}(h) = \frac{1}{m'} \sum_{i=1}^{m'} \frac{|h(x_i) \cap Y_i|}{|h(x_i)|} \quad (3.15)$$

$$F1 = \frac{1}{m'} \sum_{i=1}^{m'} \frac{2 \times |h(x_i) \cap Y_i|}{|h(x_i)| + |Y_i|} \quad (3.16)$$

3. **Average Precision (AP)**: Average precision evaluates the average fraction of relevant labels ranked higher than a particular label $y \in \mathbf{y}_i$.

$$AP = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{1}{y_i} \sum_{y \in \mathbf{y}_i} \frac{\{(y' \in \mathbf{y}_i | \text{rank}_f(x_i, y') \leq \text{rank}_f(x_i, y)\}}{\text{rank}_f(x_i, y)} \quad (3.17)$$

4. **Coverage (AUC)**: Coverage is described as the distance to cover all possible labels assigned to a sample x . It is loosely linked to precision at the level of perfect recall. The smaller the

Dataset Labelled	Metric	LAP 50%	HYP 50%	LAP 70%	HYP 70%	LAP 90%	HYP 90%	MLTS -SVM	KNNML -TSVM
Flags	HL(↓)	0.3456	0.3456	0.3381	0.3337	0.3365	0.3350	0.2894	0.3027
	F1(↑)	0.7215	0.7242	0.7292	0.7327	0.7336	0.7352	0.7396	0.7510
	AP(↑)	0.5912	0.5976	0.5913	0.5941	0.5935	0.5993	0.6184	0.6275
	RL(↓)	0.8523	0.8296	0.8452	0.8327	0.8204	0.8084	0.6801	0.7558
	AUC(↑)	0.5395	0.5480	0.5439	0.5508	0.5524	0.5551	0.6026	0.5781
Emotions	HL(↓)	0.3791	0.3670	0.3622	0.3238	0.3444	0.2886	0.2825	0.2926
	F1(↑)	0.5997	0.6063	0.6187	0.6364	0.6289	0.6562	0.6520	0.6511
	AP(↑)	0.4931	0.5020	0.5071	0.5343	0.5270	0.5590	0.5747	0.5568
	RL(↓)	0.5792	0.5500	0.5385	0.4886	0.5165	0.4536	0.4596	0.4607
	AUC(↑)	0.6820	0.6921	0.7006	0.7228	0.7105	0.7421	0.7304	0.7352
Image	HL(↓)	0.5027	0.5567	0.4642	0.4269	0.4232	0.2883	0.3365	0.3706
	F1(↑)	0.3614	0.4291	0.4373	0.4737	0.5163	0.5915	0.5655	0.5476
	AP(↑)	0.2967	0.3101	0.3412	0.3696	0.4092	0.4885	0.4748	0.4511
	RL(↓)	0.9230	0.7871	0.7533	0.6656	0.6079	0.4587	0.5083	0.5424
	AUC(↑)	0.5319	0.6234	0.5807	0.6639	0.6505	0.7480	0.6996	0.6769
Yeast	HL(↓)	0.2796	0.2410	0.2667	0.2125	0.2533	0.1996	0.2457	0.2582
	F1(↑)	0.6457	0.6635	0.6560	0.6788	0.6675	0.6827	0.6700	0.6627
	AP(↑)	0.3736	0.3930	0.3825	0.4070	0.3946	0.4219	0.3971	0.3892
	RL(↓)	0.7786	0.7466	0.7557	0.7476	0.7319	0.7438	0.7427	0.7491
	AUC(↑)	0.5840	0.5967	0.5942	0.6041	0.6055	0.6074	0.6021	0.5984
CHD	HL(↓)	0.5399	0.5360	0.5399	0.5345	0.5405	0.5342	0.5207	0.5060
	F1(↑)	0.6300	0.6309	0.6298	0.6310	0.6295	0.6318	0.6346	0.6382
	AP(↑)	0.4743	0.4750	0.4810	0.4819	0.4771	0.4794	0.4751	0.4769
	RL(↓)	0.9995	0.9915	0.9991	0.9880	1.0000	0.9904	0.9705	0.9706
	AUC(↑)	0.5127	0.5140	0.5137	0.5144	0.5111	0.5125	0.5166	0.5200

Table 3.4: Results for Multi-Label classification

value of coverage is, the better the performance is.

$$Coverage = \frac{1}{N_t} \sum_{i=1}^{N_t} maxrank_f(x_i, y) - 1 \quad (3.18)$$

5. **Ranking Loss (RL)**: This criterion is utilized for ranking-based algorithms and measures the average fraction of label pairs that are reversely ordered. For example, an irrelevant label is ranked higher than a relevant label.

$$RL = \frac{1}{N_t} \sum_{i=1}^{N_t} \left(\frac{1}{|y_i|, |\bar{y}_i|} |\{(y' y'') | f_{y' \in y_i}(x_i) \leq f_{y'' \in \bar{y}_i}(x_i)\}| \right) \quad (3.19)$$

3.6.5 Results and Discussion

We investigate the classification performances of the proposed algorithm in a supervised framework and compare it with the aforementioned algorithms in Table 2, further, we have also compared performance with the Laplacian version of our algorithm under 50%, 70%, 90% labelled data in the training dataset. Figure 3.1 reflects the overall rank of the compared and proposed algorithms. From the figure, we can conclude that the overall rank of the proposed algorithm is higher when compared with other approaches.

In the semi-supervised setting, as shown in Table 3.3 and Table 3.4, our proposed model performs up to the mark with less percentage of labeled samples.

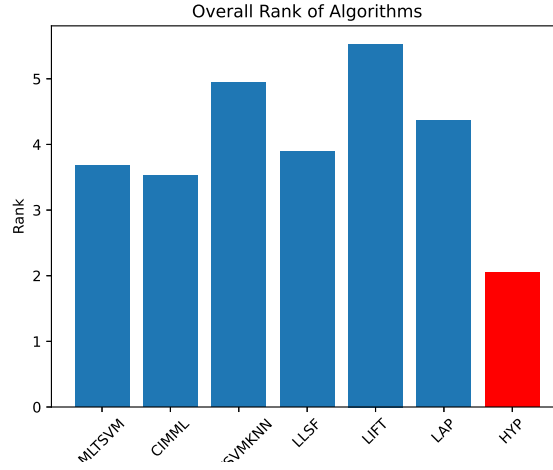
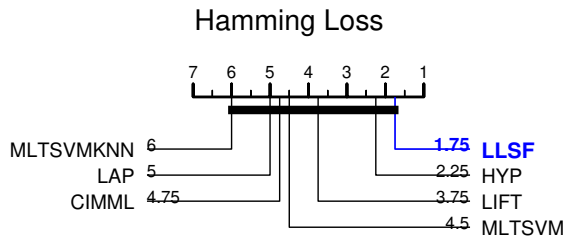
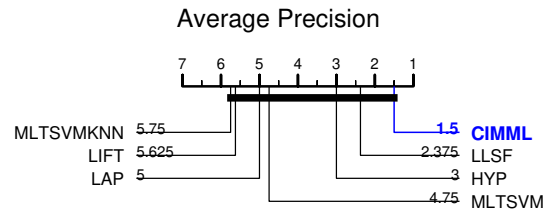


Figure 3.1: Consolidated average rank of participating algorithms across all metrics.

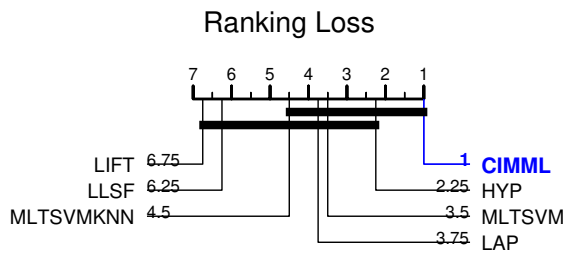
We further analyze pairwise comparisons using the Nemenyi test [9]. For six classifiers, the critical value, $\alpha_{0.1}$ is 2.589, and the CD is 2.32. Significant differences exist between the best and worst-performing algorithms across all metrics Figure 3.2. Overall, HMLLSTSVM demonstrates competitive and often superior performance.



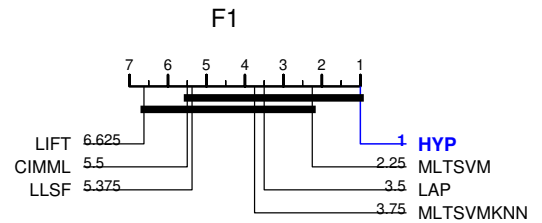
(a) Hamming Loss



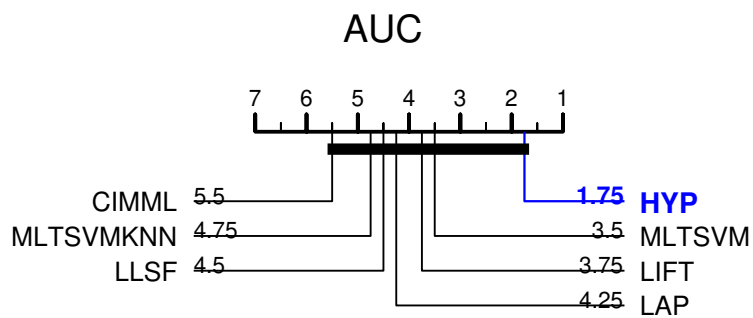
(b) Average Precision



(c) Ranking Loss



(d) F1



(e) AUC

Figure 3.2: HMLLSTSVM comparison with competing algorithms using Nemenyi test, $CD=2.32$.

Chapter 4

Improved Hypergraph-Laplacian-based Semi-Supervised Support Vector Machine (IHLSVM)

4.1 Hypergraph Support Vector Machine

Laplacian Support Vector Machine assumes a pairwise association between the samples, which generally doesn't seem to hold in practice. Graph Laplacian is unable to capture multivariate and higher-order relationships between the samples. Authors in [43] express an order relationship with Hypergraph, which captures the association between two or more two vertices. Thus, the formulation of HGSVM is a quadratic programming problem given by

$$\begin{aligned} & \min_{\alpha \in R^n, \xi \in R^l} \sum_{i=1}^l \xi_i + \gamma_A \alpha^T K \alpha + \gamma_I \alpha^T K H K \alpha \\ & \text{s.t. } y_i \left(\sum_{j=1}^n \alpha_j k(x_i, x_j) + b \right) \geq 1 - \xi_i, \quad i = 1, 2, \dots, l, \\ & \xi_i \geq 0, \quad i = 1, 2, \dots, l. \end{aligned} \tag{4.1}$$

where H is the Laplacian regularised Hyper-Graph of the dataset as defined in (2.3). The solution of the optimization, as mentioned earlier, is obtained by solving one quadratic programming problem followed by solving a system of linear equations which is not scalable.

$$\begin{aligned}
& \max_{\beta \in R^l} \sum_{i=0}^l \beta_i - \frac{1}{2} \beta^T Q \beta \\
& \text{s.t. } \sum_{i=0}^l \beta_i y_i = 0, \\
& 0 \leq \beta_i \leq 1, \quad i = 1, 2, \dots, l,
\end{aligned} \tag{4.2}$$

Where

$$Q = Y J K (2\gamma_A I + 2\gamma_I K L)^{-1} J_L^T Y,$$

I is the identity matrix, $J = [I \ 0]_{l \times (l+u)}$, and $Y = \text{diag}(y_1, y_2, \dots, y_l)$.

4.2 Improved Hypergraph Laplacian SVM (IHLSVM)

In this section, we propose solving HG SVM, i.e., the aforementioned quadratic programming problem in primal space, by replacing hinge loss with a square-hinged loss function, which leads to solving the following unconstrained optimisation problem in the primal space.

$$\begin{aligned}
& \min_{\alpha \in R^m, b \in R} \frac{1}{2} \left(\sum_{i=1}^l \max(1 - y_i(k_i^T \alpha + b), 0)^2 + \delta_A \alpha^T K \alpha + \right. \\
& \quad \left. \delta_I (\alpha^T K + 1^T b) H L (K \alpha + 1b) \right)
\end{aligned} \tag{4.3}$$

Since the above optimization is a convex problem, Newton's method is used to find the optimal solution whose complexity is $O(n^3)$.

The graph Laplacian L carries the pairwise relationship between the training instances, while the multivariate and complex relations are carried by the Hyper-graph Laplacian matrix HL . However, it is challenging to identify which representation would be optimal for the given dataset. Therefore, in this section, we define an improved Hypergraph Laplacian matrix, which is a weighted linear combination of Laplacian as well as Hypergraph Laplacian and solve it as an unconstrained optimisation problem which can be further solved in primal space using Preconditioned Conjugate Gradient method [28] whose order of complexity is $O(kn^2)$, where k is number of iterations and n is the number of training samples.

$$IHL = \lambda L + (1 - \lambda)HL \quad (4.4)$$

In the IHL matrix, L is from (2.38) and H matrix is from (4.4). λ is the hyper-parameter which work as a balancing factor between both of them.

Thus, considering squared-hinged loss function.

$$V(x_i, y_i, f) = \max(1 - y_i(k_i^T \alpha + b), 0)^2,$$

regulariser term as $\|f\|_A^2 = \alpha^T K \alpha$, $\|f\|_I^2 = \alpha^T K(IHL)K \alpha$. IHLSVM problem would solve the following unconstrained optimisation problem

$$\min_{\alpha \in R^m, b \in R} \frac{1}{2} \left(\sum_{i=1}^l \max(1 - y_i(k_i^T \alpha + b), 0)^2 + \delta_A \alpha^T K \alpha + \delta_I (\alpha^T K + 1^T b)(IHL)(K \alpha + 1b) \right) \quad (4.5)$$

We can solve (4.5) using the Preconditioned Conjugate Gradient (PCG) method[28]. The detailed algorithm is mentioned below.

Algorithm 1: Algorithm for IHLSVM

- 1 Dataset; $S = X \cup U = \{x_t, y_t\}_{t=1}^l \cup \{x_t\}_{t=l+1}^{l+u}$, X denotes a set of l labeled data, U denotes set of u unlabeled examples and parameters set $\sigma, \gamma_1, \gamma_2, \lambda$; classsifier $f(x) = \text{sign}(\sum_{i=1}^{l+u} \alpha_i^* K(x, x_i))$
 - 2 Initialisation: Calculate Graph Laplacian L ;
 - 3 Calculate the Laplacian Regularised Hypergraph $HL = I - D_v^{-\frac{1}{2}} F W D_e^{-1} F^T D_v^{-\frac{1}{2}}$;
 - 4 Calculate Improved Hypergraph Laplacian matrix $IHL = \lambda L + (1 - \lambda)H$;
 - 5 Find kernel function $K(x_i, x_j)$;
 - 6 Calculate α^* by solving primal problem (4.5) by using PCG Method;
-

4.3 Multi-category and Multi-label Classification

In semi-supervised learning, multi-class or multi-label classification is a challenging task due to the unknown geometrical structure of the data. There are many problems like speech recognition[16], face recognition[45], and many staged diseases which are multi-class problems. There are many

methods to solve multi-class like OVO(One-vs-One)[10], OVR(One-vs-Rest)[40] or DAG for multi-class classification[31], and Binary Relevance(BR)[47] for multi-label classification. We have adopted the OVR classification strategy for multiclass and binary relevance for multi-label classification for experiment purposes.

4.4 Experiments and Results

We ran a large set of experiments to analyse the Improved Hypergraph Laplacian Support Vector Machine (IHLSVM) proposed solution strategies and compared them with other algorithms like LapSVM and HGSVM. In this section, we discuss data sets, our experimental protocol and the details of parameter selection strategy. We perform numerical experiments on UCI datasets. All the experiments are performed in MATLAB R2023a on a PC with AMD Ryzen 7 5800U with Radeon Graphics 1.90 GHz and 8GB RAM.

4.4.1 UCI Datasets

We ran tests on 12 UCI ¹ benchmark datasets to look into the classification performance of our proposed approach with the square-hinge loss version of LapSVM and HGSVM. Table 4.1 presents specifics of the 12 UCI benchmark datasets and multilabel datasets ². Every dataset is standardised with mean zero and staa mean zero and a standard deviation of ndard deviation 1.

4.4.1.1 Results for Binary Classification.

We have compared IHLSVM with square-hinge loss version of LapSVM and HGSVM, on seven UCI binary datasets. We have used 10-fold cross-validation to report the accuracy of models, where two regularisation parameters have range $\{10^{-6}, 10^{-5}, \dots, 1, 10, 100\}$ and kernel parameters for RBF have the range $[0, 1]$. Table 4.2 report the mean accuracy with standard deviation. We can observe that the classification performance of our proposed IHLSVM is better than that of the other two methods on all binary datasets. In addition, we investigate the classification performances of IHLSVM under 0%, 10%, 20%, 30%, 40, 50%, 60%, 70% unlabelled data used while training.

¹<https://archive.ics.uci.edu/>

²<https://www.uco.es/kdis/mlresources>

Table 4.1: UCI datasets

Dataset	Size	Features	Classes
Australian	690	14	2
Breast Cancer	277	9	2
CMC	1473	9	2
German	1000	24	2
Hearts	270	13	2
Ionosphere	351	34	2
WDBC	569	30	2
Ecoli	3186	3	3
Dna	3186	180	3
Glass	214	9	7
Iris	150	4	3
Wine	178	13	3

4.4.1.2 Results for Multi-class Classification

We investigate the classification performance of our proposed method on five UCI multi-class datasets and have evaluate its performance on Fashion MNIST data. We have used the OVR[40] strategy and compared it with the square-hinge loss version of LapSVM and HGSVM. We are using 10-fold cross-validation to report the classification accuracy. Each experiment is also repeated ten times. Table 4.3 reports the mean accuracy and its standard deviation. We can observe that the classification performance of our proposed IHLSVM is better than that of the other two methods on all multi-class datasets. In addition, we investigate the classification performances of IHLSVM under 0%, 10%, 20%, 30% unlabelled data used while training. Fashion-MNIST ³ is a dataset of Zalando’s article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image associated with a label from 10 classes. Zalando intends Fashion-MNIST to be a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits. We investigate the classification performances of IHLSVM under 0%, 10%, 20%, 30%40%, 50% unlabelled data in the training dataset. Mean and standard deviation are reported in Table 4.4 and fig. 4.1 illustrates the performance for various percentages of unlabelled data. 4.2 and 4.3 reports an ablation study to discuss the performance of the proposed algorithm (IHLSVM) for different values of hyperparameters, i.e. kernel parameter

³<https://tech.zalando.com>t

Table 4.2: Classification Accuracy Results for Binary Datasets

Dataset	Unlabelled	LapSVM	HGSVM	IHLSVM
Australian	0%	80.29 \pm 6.70%	80.00 \pm 6.62%	80.87 \pm6.10%
	10%	81.01 \pm 5.48%	80.87 \pm 5.33%	81.16 \pm5.63%
	20%	82.46 \pm 5.65%	82.46 \pm 5.65%	82.75 \pm5.53%
	30%	80.87 \pm 5.19%	81.01 \pm 5.18%	81.45 \pm4.20%
	50%	81.30 \pm 4.50%	81.30 \pm 4.50%	81.45 \pm4.62%
	60%	79.71 \pm 4.83%	81.30 \pm 4.50%	81.88 \pm4.39%
	70%	86.36 \pm 1.17%	86.33 \pm 1.20%	86.49 \pm1.12%
German	0%	71.90 \pm 4.70%	72.10 \pm 4.43%	72.50 \pm4.43%
	10%	71.30 \pm 4.42%	71.30 \pm 4.57%	71.80 \pm4.44%
	20%	71.70 \pm 4.37%	71.30 \pm 4.79%	71.80 \pm3.82%
	30%	72.00 \pm 3.50%	72.00 \pm 3.83%	72.20 \pm2.97%
	50%	70.80 \pm 5.35%	71.00 \pm 4.99%	71.10 \pm4.89%
	60%	70.10 \pm 4.61%	70.10 \pm 4.61%	70.40 \pm4.45%
	70%	71.60 \pm 4.30%	71.80 \pm 4.59%	71.90 \pm4.77%
Ionosphere	0%	94.32 \pm 3.24%	94.31 \pm 3.28%	94.60 \pm3.36%
	10%	94.02 \pm 3.68%	94.02 \pm3.68%	93.74 \pm 3.75%
	20%	93.47 \pm 4.98%	94.04 \pm 5.23%	93.75 \pm5.11%
	30%	92.04 \pm 3.69%	92.04 \pm 3.69%	92.33 \pm3.75%
	50%	92.03 \pm 4.19%	92.03 \pm 4.60%	92.32 \pm4.45%
	60%	90.33 \pm 4.26%	90.33 \pm 4.04%	90.90 \pm3.70%
	70%	64.95 \pm 7.64%	64.95 \pm 7.64%	65.24 \pm7.61%
Breast Cancer	0%	70.74 \pm 5.80%	70.74 \pm5.80%	71.10 \pm 5.71%
	10%	71.14 \pm 6.20%	71.14 \pm 6.20%	71.85 \pm6.19%
	20%	65.36 \pm 7.33%	65.56 \pm 7.33%	66.43 \pm8.09%
	30%	69.70 \pm 9.04%	63.92 \pm 7.91%	70.04 \pm7.73%
	50%	71.85 \pm 3.64%	67.87 \pm 7.72%	72.21 \pm4.13%
	60%	67.53 \pm 5.54%	64.47 \pm 7.58%	67.87 \pm6.69%
	70%	72.91 \pm 6.04%	67.50 \pm 5.91%	72.94 \pm5.64%
CMC	0%	66.05 \pm 4.04%	65.57 \pm 4.38%	66.12 \pm3.84%
	10%	66.46 \pm 3.42%	66.19 \pm 3.48%	66.52 \pm3.90%
	20%	64.56 \pm 2.56 %	65.78 \pm 2.53%	66.19 \pm3.17%
	30%	65.92 \pm 3.72%	65.37 \pm 2.75%	66.05 \pm3.21%
	50%	65.17 \pm 4.15%	65.24 \pm 3.84%	65.44 \pm4.81%
	60 %	62.65 \pm 4.49%	62.58 \pm 5.48%	63.40 \pm4.43%
	70 %	61.78 \pm 3.81%	60.82 \pm 4.41%	62.05 \pm3.73%
Hearts-statlog	0%	76.30 \pm 8.04%	75.19 \pm 6.99%	77.41 \pm7.29%
	10%	76.67 \pm 8.56%	75.93 \pm 8.42%	78.89 \pm8.38%
	20%	77.78 \pm 9.56%	76.67 \pm 12.10%	78.89 \pm8.74%
	30%	79.26 \pm 5.58%	78.52 \pm 9.53%	80.37 \pm6.54%
	50%	80.00 \pm 6.10%	80.37 \pm 5.80%	81.48 \pm7.20%
	60%	80.00 \pm 8.04%	80.37 \pm 8.91%	81.11 \pm8.27%
	70%	81.11 \pm 7.90%	80.00 \pm 8.94%	81.48 \pm7.20%
WDBC	0%	96.14 \pm 3.77%	95.09 \pm 3.39%	96.32 \pm4.09%
	10%	96.66 \pm 3.14%	95.08 \pm 3.18%	96.84 \pm2.96%
	20%	96.49 \pm 2.61%	96.31 \pm 2.67%	96.66 \pm2.40%
	30%	96.66 \pm 2.40%	96.49 \pm 1.85%	96.84 \pm1.61%
	50%	96.32 \pm 2.67%	96.49 \pm 2.48%	97.02 \pm2.20%
	60%	96.49 \pm 3.20%	96.32 \pm 3.14%	96.67 \pm3.14%
	70%	96.14 \pm 3.39%	96.14 \pm 2.96%	96.32 \pm3.45%

$(\sigma), \delta_A, \delta_I, \lambda$.

Table 4.3: Classification Accuracy Results for Multi-class Datasets

Dataset	Unlabelled Percentage	LapSVM	HGSVM	IHLSVM
Ecoli	0%	$85.71 \pm 3.76\%$	$85.71 \pm 3.77\%$	$86.01 \pm 3.92\%$
	10%	$85.11 \pm 3.84\%$	$84.52 \pm 3.77\%$	$85.71 \pm 3.77\%$
	20%	$85.71 \pm 3.90\%$	$85.40 \pm 4.68\%$	$86.30 \pm 4.30\%$
	30%	$81.23 \pm 5.88\%$	$81.23 \pm 6.51\%$	$83.32 \pm 3.76\%$
Iris	0%	$92.00 \pm 5.06\%$	$93.33 \pm 5.27\%$	$94.67 \pm 5.06\%$
	10%	$92.67 \pm 5.48\%$	$91.33 \pm 9.60\%$	$95.33 \pm 3.80\%$
	20%	$92.67 \pm 10.90\%$	$90.00 \pm 8.16\%$	$94.67 \pm 5.06\%$
	30%	$92.67 \pm 7.96\%$	$94.00 \pm 4.35\%$	$94.67 \pm 4.47\%$
Wine	0%	$98.30 \pm 1.55\%$	$98.32 \pm 1.54\%$	$98.87 \pm 1.54\%$
	10%	$97.73 \pm 2.39\%$	$97.75 \pm 1.26\%$	$98.30 \pm 1.55\%$
	20%	$98.86 \pm 1.56\%$	$98.87 \pm 1.54\%$	$99.43 \pm 1.28\%$
	30%	$93.81 \pm 2.37\%$	$93.25 \pm 1.56\%$	$93.83 \pm 1.21\%$
Glass	0%	$66.36 \pm 7.80\%$	$66.38 \pm 6.97\%$	$67.30 \pm 8.66\%$
	10%	$64.99 \pm 5.92\%$	$64.03 \pm 3.74\%$	$65.46 \pm 5.80\%$
	20%	$58.89 \pm 8.55\%$	$57.96 \pm 8.75\%$	$60.29 \pm 8.81\%$
	30%	$53.28 \pm 6.90\%$	$52.34 \pm 6.26\%$	$53.73 \pm 6.28\%$
DNA	0%	$50.25 \pm 1.95\%$	$50.22 \pm 1.96\%$	$50.38 \pm 1.93\%$
	10%	$50.22 \pm 1.90\%$	$50.16 \pm 1.72\%$	$50.25 \pm 1.74\%$
	20%	$50.66 \pm 1.97\%$	$50.56 \pm 1.78\%$	$50.75 \pm 1.87\%$
	30%	$51.07 \pm 1.84\%$	$51.13 \pm 1.75\%$	$51.22 \pm 1.66\%$

4.4.2 Results for Multi-label Classification

We investigate the classification performance of our proposed method along with the square hinge loss version of LapSVM and HGSVM on four multi-label datasets, i.e. Birds, Cal500, Flags, and Enron discussed in Table 4.5. We have used the five evaluation criteria, Hamming Loss, Average Precision, Ranking Loss, AUC and F1 Score, to compare the performance of multi-label classification algorithms and results are discussed in Table 4.6 - Table 4.8 for performances of IHLSVM under 0%, 10%, 30, 50% unlabelled data used while training. In Figure 4.2 and Figure 4.3, we have reported the Nemenyi Test[9] in Figure 4.4 for comparing the performance of IHLSVM with the other two algorithms on five metrics.

4.4.2. RESULTS FOR MULTI-LABEL CLASSIFICATION

Table 4.4: Comparison of Results for MNIST Fashion

Unlabelled	LapSVM	HGSVM	IHLSVM
0 %	76.57± 0.79	76.48± 1.41	76.89± 0.84
10 %	76.45± 0.84	75.88± 0.81	76.41± 0.72
20 %	75.07± 0.98	75.38± 1.18	75.43± 1.10
30 %	74.55± 1.56	74.75± 0.52	74.70± 1.41
40 %	73.91± 0.96	73.21± 1.82	74.16± 0.98
50 %	72.62± 1.28	71.82± 1.23	72.88± 1.25

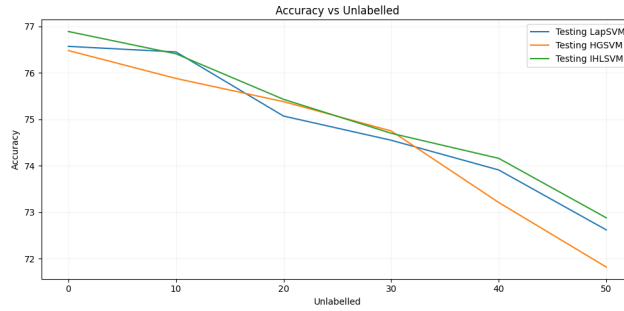


Figure 4.1: Results for MNIST Fashion

Table 4.5: Description of Multi-label Datasets

Dataset	Instances	Features	Labels	Card	Domain
Birds	645	260	19	1.014	Audio
CAL500	502	68	174	26.044	Music
Emotions	593	72	6	1.868	Music
Flags	194	19	7	3.392	Image
Image	2000	294	5	1.236	Image
Enron	1702	1001	53	3.378	text

Table 4.6: Results of Multi-label Dataset Birds and Flags

Results of Multi-label Dataset Birds

Unlabelled	Metric	LapSVM	HGSVM	IHLSVM
0%	HL(↓)	0.0458	0.0468	0.0452
	F1(↑)	0.61581	0.6095	0.6200
	AP(↑)	0.1429	0.1524	0.1429
	RL(↓)	0.7675	0.7669	0.7632
	AUC(↑)	0.6726	0.6731	0.6743
10%	HL(↓)	0.0458	0.0474	0.0455
	F1(↑)	0.6097	0.5927	0.6131
	AP(↑)	0.1167	0.1362	0.1262
	RL(↓)	0.7908	0.8069	0.7901
	AUC(↑)	0.6494	0.6382	0.6481
30%	HL(↓)	0.0477	0.0480	0.0470
	F1(↑)	0.5519	0.5581	0.5635
	AP(↑)	0.0581	0.0771	0.0386
	RL(↓)	0.8153	0.8353	0.8137
	AUC(↑)	0.6240	0.6201	0.6265
50%	HL(↓)	0.0531	0.0537	0.0530
	F1(↑)	0.4980	0.4945	0.4959
	AP(↑)	0.0776	0.0781	0.0581
	RL(↓)	0.8432	0.8698	0.8486
	AUC(↑)	0.5873	0.5801	0.5854

Results of Multi-label Dataset Flags

Unlabelled	Metric	LapSVM	HGSVM	IHLSVM
0%	HL(↓)	0.3084	0.3077	0.3062
	F1(↑)	0.6640	0.6641	0.6671
	AP(↑)	0.0571	0.0571	0.0571
	RL(↓)	0.7014	0.6979	0.6983
	AUC(↑)	0.5773	0.5789	0.5787
10%	HL(↓)	0.3056	0.3027	0.3019
	F1(↑)	0.6768	0.6794	0.6821
	AP(↑)	0.1143	0.1143	0.1143
	RL(↓)	0.6881	0.6849	0.6803
	AUC(↑)	0.5928	0.5952	0.5987
30%	HL(↓)	0.3175	0.3168	0.3146
	F1(↑)	0.6672	0.6689	0.6705
	AP(↑)	0.0571	0.0571	0.0571
	RL(↓)	0.7097	0.7084	0.7055
	AUC(↑)	0.5774	0.5780	0.5792
50%	HL(↓)	0.3180	0.3225	0.3158
	F1(↑)	0.6713	0.6671	0.6748
	AP(↑)	0.0286	0.0286	0.0286
	RL(↓)	0.7189	0.7270	0.7175
	AUC(↑)	0.5745	0.5701	0.5759

4.4.2. RESULTS FOR MULTI-LABEL CLASSIFICATION

Table 4.7: Results of Multi-label Dataset Image and Emotions

(a) Multi-label Dataset Image

Unlabelled	Metric	LapSVM	HGSVM	IHLSVM
0%	HL(↓)	0.1664	0.1649	0.1670
	F1(↑)	0.5804	0.5851	0.5784
	AP(↑)	0	0	0
	RL(↓)	0.5516	0.5471	0.5545
	AUC(↑)	0.7053	0.7078	0.7040
10%	HL(↓)	0.1632	0.1647	0.1631
	F1(↑)	0.5880	0.5872	0.5871
	AP(↑)	0	0	0
	RL(↓)	0.5467	0.5461	0.5487
	AUC(↑)	0.7047	0.7053	0.7032
30%	HL(↓)	0.1677	0.1678	0.1677
	F1(↑)	0.5605	0.5635	0.5607
	AP(↑)	0	0	0
	RL(↓)	0.5800	0.5768	0.5800
	AUC(↑)	0.6926	0.6954	0.6936
50%	HL(↓)	0.1839	0.1824	0.1828
	F1(↑)	0.5135	0.5222	0.5181
	AP(↑)	0	0	0
	RL(↓)	0.6237	0.6145	0.6198
	AUC(↑)	0.6686	0.6721	0.6697

(b) Multi-label Dataset Emotions

Unlabelled	Metric	LapSVM	HGSVM	IHLSVM
0%	HL(↓)	0.1855	0.1866	0.1858
	F1(↑)	0.6815	0.6821	0.6824
	AP(↑)	0	0	0
	RL(↓)	0.4485	0.4440	0.4450
	AUC(↑)	0.7437	0.7453	0.7450
10%	HL(↓)	0.1936	0.1942	0.1936
	F1(↑)	0.6651	0.6662	0.6654
	AP(↑)	0	0	0
	RL(↓)	0.4775	0.4734	0.4767
	AUC(↑)	0.7355	0.7375	0.7361
30%	HL(↓)	0.1889	0.1917	0.1880
	F1(↑)	0.6651	0.6626	0.6670
	AP(↑)	0	0	0
	RL(↓)	0.4702	0.4714	0.4683
	AUC(↑)	0.7381	0.7383	0.7395
50%	HL(↓)	0.2018	0.2035	0.2004
	F1(↑)	0.6452	0.6443	0.6482
	AP(↑)	0	0	0
	RL(↓)	0.4946	0.4945	0.4917
	AUC(↑)	0.7301	0.7294	0.7312

Table 4.8: Results of Multi-label Datasets CAL500 and Enron

Results of Multi-label Dataset CAL500

Unlabelled	Metric	LapSVM	HGSVM	IHLSVM
0%	HL(↓)	0.1386	0.1388	0.1387
	F1(↑)	0.3086	0.3101	0.3096
	AP(↑)	0.0959	0.1055	0.1055
	RL(↓)	0.9901	0.9886	0.9892
	AUC(↑)	0.4712	0.4715	0.4714
10%	HL(↓)	0.1381	0.1380	0.1380
	F1(↑)	0.3054	0.3072	0.3071
	AP(↑)	0.1243	0.1339	0.1255
	RL(↓)	0.9910	0.9898	0.9902
	AUC(↑)	0.4718	0.4726	0.4725
30%	HL(↓)	0.1390	0.1395	0.1395
	F1(↑)	0.3101	0.3097	0.3110
	AP(↑)	0.0754	0.0827	0.0790
	RL(↓)	0.9916	0.9912	0.9909
	AUC(↑)	0.4737	0.4731	0.4733
50%	HL(↓)	0.1400	0.1394	0.1401
	F1(↑)	0.3064	0.3080	0.3091
	AP(↑)	0.0685	0.0553	0.0649
	RL(↓)	0.9919	0.9923	0.9915
	AUC(↑)	0.4752	0.4764	0.4755

Results of Multi-label Dataset Enron

Unlabelled	Metric	LapSVM	HGSVM	IHLSVM
0%	HL(↓)	0.0583	0.0585	0.0583
	F1(↑)	0.4266	0.4282	0.4272
	AP(↑)	0.2412	0.2538	0.2412
	RL(↓)	0.9304	0.9295	0.9295
	AUC(↑)	0.4860	0.4854	0.4858
10%	HL(↓)	0.0586	0.0580	0.0574
	F1(↑)	0.3592	0.4405	0.4410
	AP(↑)	0.1764	0.1724	0.1724
	RL(↓)	0.9311	0.9291	0.9289
	AUC(↑)	0.4868	0.4871	0.4889
30%	HL(↓)	0.0583	0.0587	0.0583
	F1(↑)	0.3252	0.4251	0.3258
	AP(↑)	0.2160	0.2161	0.2119
	RL(↓)	0.9420	0.9424	0.9418
	AUC(↑)	0.4863	0.4848	0.4862
50%	HL(↓)	0.0611	0.0584	0.0586
	F1(↑)	0.1326	0.3815	0.2721
	AP(↑)	0.1341	0.1425	0.1425
	RL(↓)	0.9478	0.9459	0.9438
	AUC(↑)	0.4694	0.4745	0.4760

4.4.2. RESULTS FOR MULTI-LABEL CLASSIFICATION

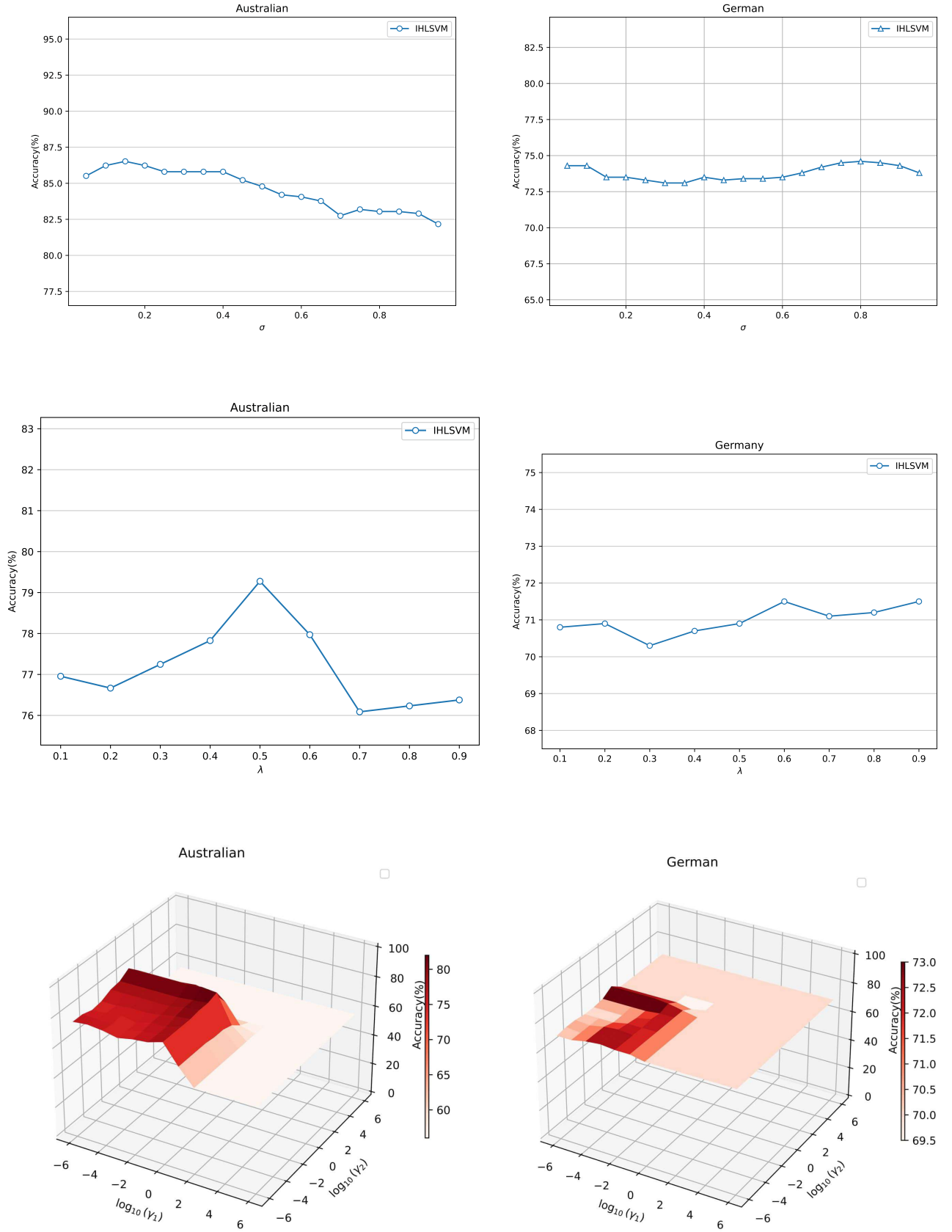


Figure 4.2: Sensitivity Analysis for binary datasets German and Austrlian.

4.4.2. RESULTS FOR MULTI-LABEL CLASSIFICATION

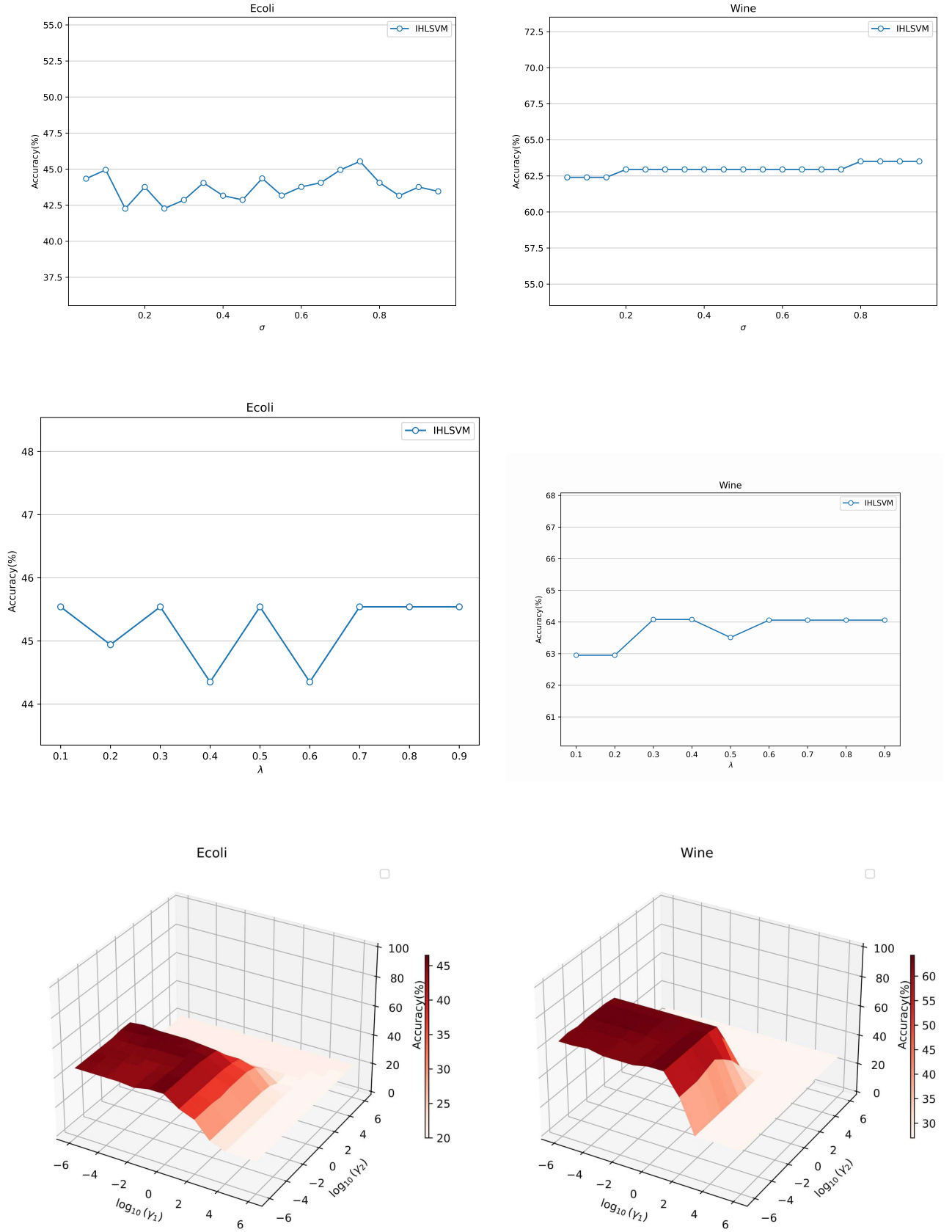
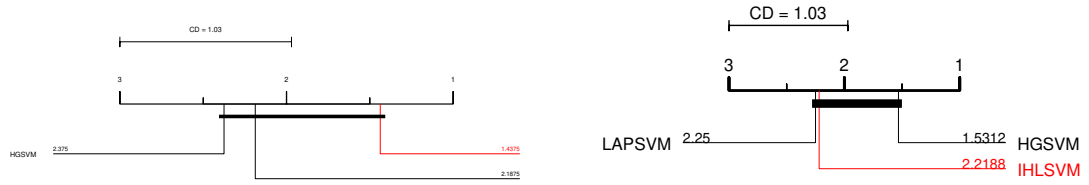


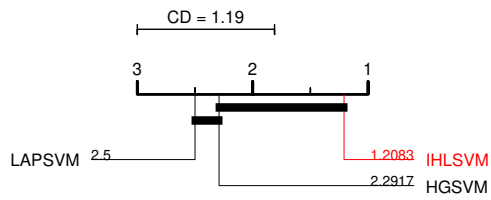
Figure 4.3: Sensitivity Analysis for multi-category datasets Ecoli and Wine

4.4.2. RESULTS FOR MULTI-LABEL CLASSIFICATION

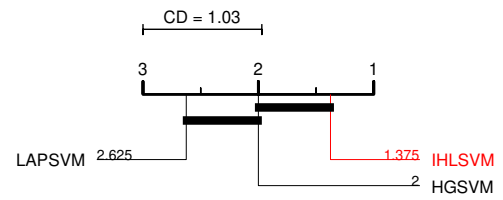


(a) Hamming Loss

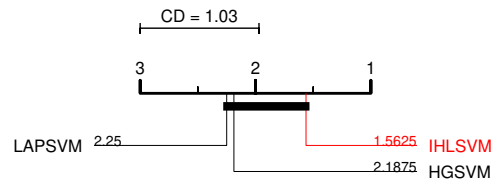
(b) Average Precision



(c) Ranking Loss



(d) F1



(e) AUC

Figure 4.4: IHLSVM comparison with competing algorithms using the Nemenyi test.

Chapter 5

Conclusion and Future Works

This dissertation presents two novel approaches for leveraging hypergraph structures in machine learning: the Hypergraph Least Squares Twin Support Vector Machine (HMLLSTSVM) for multi-label learning and the Improved Hypergraph Regularized Semi-Supervised Support Vector Machine (IHLSVM). The HMLLSTSVM introduces a weight assignment mechanism based on distances from neighbors, effectively exploiting information from both labeled and unlabeled samples through hypergraph-based higher-order relationships. By replacing the hinge loss function with a square loss function, this method enhances noise tolerance. It reduces sensitivity to outliers, solving a system of linear equations instead of relying on quadratic programming problems (QPPs). Furthermore, integrating cluster-based structural and local geometric information improves performance across various multi-label datasets.

The IHLSVM, the other hand, incorporates a weighted combination of graph Laplacian and hypergraph representations to explore the multivariate manifold structure embedded in data. This approach addresses scenarios where individual representations may fall short. Using the Square Hinge loss function and Preconditioned Conjugate Gradient method expedites the training process, demonstrating robust performance on binary, multi-category, and multi-label datasets, including an application on the MNIST Fashion dataset. An ablation study further highlights the robustness of the proposed algorithm.

Experimental results validate the efficacy of both models across multiple datasets and performance measures, showcasing their capability to handle diverse learning problems.

Future research could focus on incorporating label correlations better to leverage the peculiar

properties of multi-label learning problems. Additionally, establishing relationships within the label space that exhibit higher-order relationships, alongside feature space analysis, could provide a significant advancement in modeling multi-label datasets. These directions can potentially refine and expand the applicability of hypergraph-based learning frameworks in complex data scenarios.

Bibliography

- [1] Azad-Manjiri, M., Amiri, A., Sedghpour, A.S.: Ml-sltsvm: A new structural least square twin support vector machine for multi-label learning. *Pattern Analysis and Applications* **23**(1), 295–308 (2020) 22
- [2] Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research* **7**(11) (2006) 14
- [3] Bennett, K., Demiriz, A.: Semi-supervised support vector machines. *Advances in Neural Information processing systems* **11** (1998) 2
- [4] Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recognition* **37**(9), 1757–1771 (2004) 2
- [5] Chen, W.J., Shao, Y.H., Deng, N.Y., Feng, Z.L.: Laplacian least squares twin support vector machine for semi-supervised classification. *Neurocomputing* **145**, 465–476 (2014) 2, 18
- [6] Chen, W.J., Shao, Y.H., Li, C.N., Deng, N.Y.: Mltsvm: a novel twin support vector machine to multi-label learning. *Pattern Recognition* **52**, 61–74 (2016) 18, 25
- [7] Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: *eEuropean Conference on Principles of Data Mining and Knowledge Discovery*. pp. 42–53. Springer (2001) 2
- [8] Cortes, C., Vapnik, V.: Support-vector networks. *Machine learning* **20**(3), 273–297 (1995) 1, 5

- [9] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* **7**(Jan), 1–30 (2006) 30, 38
- [10] Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: Drcw-ovo: distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems. *Pattern recognition* **48**(1), 28–42 (2015) 35
- [11] Hanifelou, Z., Adibi, P., Monadjemi, S.A., Karshenas, H.: Knn-based multi-label twin support vector machine with priority of labels. *Neurocomputing* **322**, 177–186 (2018) 18, 19, 25
- [12] Hong, R., Wang, M., Gao, Y., Tao, D., Li, X., Wu, X.: Image annotation by multiple-instance learning with discriminative feature mapping and selection. *IEEE Transactions on Cybernetics* **44**(5), 669–680 (2013) 2
- [13] Huang, J., Li, G., Huang, Q., Wu, X.: Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* **28**(12), 3309–3323 (2016) 25
- [14] Izenman, A.J.: Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics* **4**(5), 439–446 (2012) 2
- [15] Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(5), 905–910 (2007) 8
- [16] Jin, Y., Li, P.: Performance and robustness of bio-inspired digital liquid state machines: A case study of speech recognition. *Neurocomputing* **226**, 145–160 (2017) 34
- [17] Joachims, T.: Transductive learning via spectral graph partitioning. In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. pp. 290–297 (2003) 1
- [18] Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. vol. 2, pp. 1719–1726. IEEE (2006) 2
- [19] Khemchandani, R., Chandra, S., et al.: Twin support vector machines for pattern classification. *IEEE Transactions on pattern analysis and machine intelligence* **29**(5), 905–910 (2007) 1

- [20] Khemchandani, R., Pal, A.: Multi-category laplacian least squares twin support vector machine. *Applied Intelligence* **45**, 458–474 (2016) 2
- [21] Khemchandani, R., Sharma, S.: Robust least squares twin support vector machine for human activity recognition. *Applied Soft Computing* **47**, 33–46 (2016) 2
- [22] Khemchandani, R., Sharma, S.: Robust parametric twin support vector machine and its application in human activity recognition. In: *Proceedings of International Conference on Computer Vision and Image Processing*. pp. 193–203. Springer (2017) 2
- [23] Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. *Expert Systems with Applications* **36**(4), 7535–7543 (2009) 2
- [24] Kumar, S., Ahmadi, N., Rastogi, R.: Multi-label learning with missing labels using sparse global structure for label-specific features. *Applied Intelligence* **53**(15), 18155–18170 (2023) 3
- [25] Kumar, S., Rastogi, R.: Low rank label subspace transformation for multi-label learning with missing labels. *Information Sciences* **596**, 53–72 (2022) 3
- [26] Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE transactions on pattern analysis and machine intelligence* **28**(1), 69–74 (2005) 1
- [27] McCallum, A.K.: Multi-label text classification with a mixture model trained by em. In: *AAAI 99 Workshop on Text Learning*. pp. 1–7. Citeseer (1999) 2
- [28] Melacci, S., Belkin, M.: Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research* **12**(3) (2011) 2, 15, 16, 33, 34
- [29] Mir, A., Nasiri, J.A.: Knn-based least squares twin support vector machine for pattern classification. *Applied Intelligence* **48**(12), 4551–4564 (2018) 25
- [30] Pan, X., Luo, Y., Xu, Y.: K-nearest neighbor based structural twin support vector machine. *Knowledge-Based Systems* **88**, 34–44 (2015) 18
- [31] Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin dags for multiclass classification. *Advances in neural information processing systems* **12** (1999) 35

- [32] Qi, G.J., Hua, X.S., Rui, Y., Tang, J., Mei, T., Zhang, H.J.: Correlative multi-label video annotation. In: Proceedings of the 15th ACM International Conference on Multimedia. pp. 17–26 (2007) 2
- [33] Qi, Z., Tian, Y., Shi, Y.: Laplacian twin support vector machine for semi-supervised classification. *Neural networks* **35**, 46–53 (2012) 2
- [34] Qi, Z., Tian, Y., Shi, Y.: Structural twin support vector machine for classification. *Knowledge-Based Systems* **43**, 74–81 (2013) 2
- [35] Rastogi, R., Jain, S.: Multi-label learning via minimax probability machine. *International Journal of Approximate Reasoning* **145**, 1–17 (2022) 3
- [36] Rastogi, R., Kumar, S.: Discriminatory label-specific weights for multi-label learning with missing labels. *Neural Processing Letters* **55**(2), 1397–1431 (2023) 3
- [37] Rastogi, R., Kumar, S.: Discriminatory label-specific weights for multi-label learning with missing labels. *Neural Processing Letters* (2), 1397–1431 (Apr 2023). <https://doi.org/10.1007/s11063-022-10945-z>, <https://doi.org/10.1007/s11063-022-10945-z> 25
- [38] Rastogi, R., Mortaza, S.: Multi-label classification with missing labels using label correlation and robust structural learning. *Knowledge-Based Systems* **229**, 107336 (2021) 3
- [39] Rastogi, R., Mortaza, S.: Imbalance multi-label data learning with label specific features. *Neurocomputing* **513**, 395–408 (2022) 3
- [40] Rifkin, R., Klautau, A.: In defense of one-vs-all classification. *The Journal of Machine Learning Research* **5**, 101–141 (2004) 35, 36
- [41] Song, Z., Yang, X., Xu, Z., King, I.: Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems* **34**(11), 8174–8194 (2022) 1
- [42] Sun, Y., Ding, S., Zhang, Z., Zhang, C.: Hypergraph based semi-supervised support vector machine for binary and multi-category classifications. *International Journal of Machine Learning and Cybernetics* pp. 1–18 (2022) 2

- [43] Sun, Y., Ding, S., Zhang, Z., Zhang, C.: Hypergraph based semi-supervised support vector machine for binary and multi-category classifications. *International Journal of Machine Learning and Cybernetics* pp. 1–18 (2022) 32
- [44] Tan, J., Zhen, L., Deng, N., Zhang, Z.: Laplacian p-norm proximal support vector machine for semi-supervised classification. *Neurocomputing* **144**, 151–158 (2014) 2
- [45] Wei, J., Jian-Qi, Z., Xiang, Z.: Face recognition method based on support vector machine and particle swarm optimization. *Expert Systems with Applications* **38**(4), 4390–4393 (2011) 34
- [46] Xu, Y., Pan, X., Zhou, Z., Yang, Z., Zhang, Y.: Structural least square twin support vector machine for classification. *Applied Intelligence* **42**(3), 527–536 (2015) 2
- [47] Zhang, M.L., Li, Y.K., Liu, X.Y., Geng, X.: Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science* **12**, 191–202 (2018) 35
- [48] Zhang, M.L., Wu, L.: Lift: Multi-label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**(1), 107–120 (2014) 25
- [49] Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering* **18**(10), 1338–1351 (2006) 2
- [50] Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems* **19**(1), 1601–1608 (2006) 5