Research Reports on Computer Science
https://ojs.wiserpub.com/index.php/RRCS/

UNIVERSAL WISER
PUBLISHER

Research Article

Multi-label Minimax Probability Machine with Multi-manifold Regularisation

Reshma Rastogi, Sambhav Jain[*]

Department of Computer Science, South Asian University, Delhi, India
E-mail: sambhavjain0712@gmail.com

Received: 30 September 2021; **Revised:** 1 December 2021; **Accepted:** 12 December 2021

Abstract: Semi-supervised learning i.e., learning from a large number of unlabelled data and exploiting a small percentage of labelled data has attracted centralised attention in recent years. Semi-supervised problem is handled mainly using graph based Laplacian and Hessian regularisation methods. However, neither the Laplacian method which leads to poor generalisation nor the Hessian energy can properly forecast the data points beyond the range of the domain. Thus, in this paper, the Laplacian-Hessian semi-supervised method is proposed, which can both predict the data points and enhance the stability of Hessian regulariser. In this paper, we propose a Laplacian-Hessian Multi-label Minimax Probability Machine, which is Multi-manifold regularisation framework. The proposed classifier requires mean and covariance information; therefore, assumptions related to the class conditional distributions are not required; rather, a upper bound on the misclassification probability of future data is obtained explicitly. Furthermore, the proposed model can effectively utilise the geometric information via a combination of Hessian-Laplacian manifold regularisation. We also show that the proposed method can be kernelised on the basis of a theorem similar to the representer theorem for handling non-linear cases. Extensive experimental comparisons of our proposed method with related multi-label algorithms on well known multi-label datasets demonstrate the validity and comparable performance of our proposed approach.

***Keywords*:** multi-label semi-supervised classification, Hessian regularisation, Minimax Probability Machine, weighted least squares

1. Introduction

Multi-label classification is a supervised learning scenario wherein each sample belongs to more than one labels simultaneously. Multi-label classification is being used in many fields such as bioinformatics [1], protein function prediction, recommender systems, sentiment classification of microblogs [2], web mining [3], information retrieval [4], etc.

Acquiring labelled data in real life learning scenarios is often a challenging task, either because it is time consuming or too expensive to obtain, while unlabelled data is abundant. This asymmetry is amplifying in multi-label learning scenario when compared with single label case due to complex labelling process. Thus, it is important to consider semi-supervised methods for multi-label learning wherein both labelled and unlabelled data are considered for training, thus can achieve better results.

DOI: https://doi.org/10.37256/rrcs.1120221193

The complexity of a classification problem has grown from *Binary classification* with | m | = 2 (m is the number of class/labels) and *Multi-class classification* where | m | > 2 in which each sample can be associated with only one class or label; to *Multi-label Learning* wherein each sample is associated with more than one relevant label and hence is much more complicated than the first two. Blanco et al. [5] in their paper showed that multi-label classification becomes difficult as label density increases.

There are many semi-supervised learning (SSL) techniques like Gaussian mixture models, self-training (where first a supervised classifier is trained and then the labels of unlabelled training data are predicted and then the classifier is trained again considering the predicted labels), co-training, (which assumes that each example is described using two different feature sets that provide different, complementary information about the instance), graph based methods etc. Here, we intend to use a graph based manifold regularisation for semi-supervised setting.

In order to effectively utilise the unlabelled data, we propose a semi-supervised Minimax Probability Machine (MPM) based approach termed as Semi-Supervised Learning Multi-label Minimax Probability Machine (MLMPM-SSL). We first develop a semi-supervised version of MPM classifier based on Hessian-Laplacian regularisation to effectively utilise geometry of data on the manifold, then show its efficacy and application in Amazon rainforest satellite images.

Let a set of N training instances $X = \{x_1, x_2, \ldots, x_N\}$ each in a n-dimensional space, i.e., $x_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$ be associated with m class labels, $Y = \{y_1, y_2, \ldots, y_m\}$. Thus, each sample x_i can take one or more labels from the m different classes $c_1, \ldots, c_m$ with its corresponding label vector as $y_i = Y_i$. determines its membership to each of these classes. Here, $y_i \in \{1, 0, \text{-}1\}$, meaning the label annotations of sample i can be relevant, missing or irrelevant respectively. The task of semi-supervised multi-label classification is to learn a model utilising both labelled and unlabelled data and assign the proper class label to a test instance.

Recent studies show that the points lie on a manifold space which in turn can be reduced to a low dimensional manifold by locally linear embedding [6, 7] Laplacian eigenmaps, Hessian eigenmaps, etc. A basic assumption is that the sub-manifolds within a manifold are linear and these sub-manifolds can be patched together so as to form a linear manifold also preserving the neighbourhood structure (i.e., local structure) in the graph. Manifold regularisation framework is frequently used for semi-supervised learning which exploits the geometry of the given data on the manifold.

Laplacian regularisation is a popular manifold regularisation based SSL algorithm which approximates the manifold by using graph Laplacian. Another approach that is gaining popularity is Hessian regularisation, which prefers functions whose values vary linearly with respect to geodesic distance. Another recent advancement in manifold learning is the p-Laplacian which acts as a nonlinear generalisation of the standard graph Laplacian. Although p-Laplacian is shown to be effective, yet it is severely limited by its high computational cost to estimate a p-Laplacian.

2. Related work

2.1 *Minimax Probability Machine (MPM)*

The MPM is a state-of-the-art binary classification algorithm, proposed by Lanckriet et al. [8], and has attracted a lot of researchers over the recent years. Some examples are twin minimax probability extreme learning machine [9], twin minimax probability regression [10], structural minimax probability [11], etc. MPM is a generative classifier which exploits statistical information inherent in the data. On the basis of first order and second order moment not only it classifies the data points, but also it aims to maximize the lower bound of accuracy (τ) in the worst case scenario. Unlike other generative classification algorithms, it doesn't require any class distributional assumptions. In case of unavailability of exact values, estimates of means and covariance can also be used.

The central idea for MPM comes from the theorem given by Isii [12], as extended in work by Bertsimas and Sethuraman [13]:

$$\sup_{z\sim(\bar{z},\Sigma_z)} \Pr\{z \in S\} = \frac{1}{1+d^2}$$

$$d^2 = \inf_{z\in S} (z-\bar{z})^T \Sigma_z^{-1} (z-\bar{z}) \tag{1}$$

here, d is the Mahalanobis distance [14], S is a convex set, $\overline{z}$ is the mean of class z, Σ_z is the covariance matrix (assumed to be positive definite for simplicity) of class z.

The Mahalanobis distance is the distance of a sample from the centroid of class, divided by the second order moment of the training data points.

The above theorem helps to find the belongingness of a sample to a class in terms of probability. For example, if the sample has a smaller Mahalanobis distance from the centroid, it would result in a higher probability for that particular class.

MPM finds a hyperplane, which can effectively separate the points of two classes with the highest probability with respect to all distributions. MPM also maximises a lower bound of probability membership value to each of the two classes which is represented by τ.

MPM shares analogy with Support Vector Machines (SVM). SVM tend to find a hyperplane that can separate the two classes with maximum margin, similarly, MPM tries to find a hyperplane such that the classes are separated with maximum probability and also provides a misclassification bound for the worst case, for more details please refer to [15]. In multi-label setting, this could help in focusing more attention on the labels which are difficult to identify as in the case of medical diagnosis of rare diseases.

2.2 *Manifold regularisation*

In SSL, we assume that there are N training samples, out of which l labelled samples can be written as $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabelled samples are represented as $\{(x_i)\}_{i=l+1}^{l+u}$. The labelled samples are generated from the probability distribution whereas the unlabelled samples are drawn from the marginal distribution. It is assumed that if two points p (point p is drawn from the labelled set of samples) and q (point q is drawn from unlabelled set of samples) are close to each other in intrinsic geometry of marginal distribution, then point p and point q would have similar label. The marginal distribution is unknown in most applications, therefore an approximation based on labelled and unlabelled samples using graph Laplacian associated with the samples is used. Belkin et al. [16] showed that when the number of samples approaches infinity, the Laplace-Beltrami operator on the manifold can be approximated by discrete graph Laplacian. The optimisation problem given by Belkin et al. [16] is as follows:

$$\underset{f}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \lambda_1 \|f\|_k^2 + \lambda_2 \|f\|_I^2 \tag{2}$$

here, V is a loss function, $\| f \|_k^2 = f^T K f$ (K is a kernel function) is a smoothness term on f and $\| f \|_I^2$ is the key term to estimate the manifold, λ_1 and λ_2 are the parameters that balance the loss function and the regularisation terms. Here, $\| f \|_I^2$ can be approximated as $\| f \|_I^2 = \frac{1}{(1+u)^2} \sum_{i,j=1}^{1+u} w_{ij} (f(x_i) - f(x_j))^2 = \frac{1}{(1+u)^2} F^T L F$ where $L = D - W$ is the graph Laplacian, $F = [f(x_1), \ldots, f(x_{l+u})]^T$, W is the edge weight matrix whose each element is computed as $w_{ij} = \exp[-\sigma \| x_i - x_j \|_2^2]$, $\sigma > 0$, D is the diagonal matrix computed as $D_{ii} = \sum_{j=1}^{l+u} w_{ij}$.

Although Laplacian based regularisation is very popular because of its simplicity, yet it has its limitations. The null space of the graph based Laplacian along the manifold is a constant function that results in a poor performance especially under a low percentage of label data and it leads to overfitting the manifold. Hein et al. [17] in their paper proposed a Hessian based regularisation that generalises a smooth manifold which gives better results under a low percentage of labelled data. But the problem with Hessian regularisation is that when labelled data is not scarce (or is sufficiently available), Hessian gives sub-optimal results as the information from the labelled data is sufficient enough to estimate the manifold. In such a case, Laplacian approaches perform better than Hessian regularisation. Therefore, we strongly believe that a combination of these two approaches can better overcome the challenges faced by either.

The main contribution of this paper is summed up as follows:

1. We propose a Laplacian-Hessian based Multi-manifold regulariser semi-supervised Multi-label Minimax Probability Machine (MLMPM-SSL) model that not only uses first order and second order moment of label data but also exploits the intrinsic geometry of the manifold using both Laplacian and Hessian regularisation to

supplement learning.

2. Since the resulting optimisation problems are Second Order Cone Programming (SOCP), we employ weighted least squares method to solve the SOCP problem.
3. Extensive experimental evaluations on well known multi-label datasets based on various evaluation metrics display the effectiveness and validity of our proposed model.

3. Proposed model

Consider a multi-label classification problem with m possible labels in the n-dimensional space. Suppose $X \subseteq \mathbb{R}$ be an input domain of instances and $Y \subseteq \mathbb{R}$ denotes an output domain of m class labels. Multi-label learning aims to obtain the decision function $f(.) : X \rightarrow 2^Y$, which does a mapping from a training set $\mathcal{D} = \{(x_i, y_i)$, where $i = 1, 2, \ldots, N\}$ to the power set of Y, $x_i \in X$ denotes an input instance with the associated label set $y_i \subseteq Y$. Thus, y_{iv} represents the relationship of x_i pattern with v^{th} class and is defined as follows:

$$y_{iv} = \begin{cases} 1, \text{ if } x_i \text{ belongs to } v\text{th label} \\ -1, \text{ if } x_i \text{ does not belong to } v\text{th label} \\ 0, \text{ if } x_i \text{ is unknown or unlabelled for } v\text{th label} \end{cases}$$

where $1 \leq v \leq m$. I_v contains the samples belonging to the class with label v, I_v' contains the rest of the samples which do not belong to the class label v. For convenience, we denote x_r for samples belonging to I_v and x_j for samples belonging to I_v'.

We wish to find m hyperplanes, for each of the m labels, $f_v(z) : (a_v^T z = b_v)$, where a_v and b_v are plane hyperparameters, which can effectively separate the points, represented by random variable $x_r \sim (\overline{x}_r, \Sigma_{x_r})$ and $x_j \sim (\overline{x}_j, \Sigma_{x_j})$, with maximal probability with respect to all distributions.

Here, $\overline{x}_r$ represents means and Σ_{x_r} represents the covariance matrix of samples $x_r \in I_v$. It also maximises τ_v which represents a lower bound of probability membership. For v^{th} hyperplane supervised MPM tends to solve

$$\begin{aligned} &\max_{\tau_v, a_v \neq 0, b_v} \tau_v \\ &\text{st} \quad \inf_{x_r \sim (\overline{x}_r, \Sigma_{x_r})} \Pr\{a_v^T x_r \geq b_v\} \geq \tau_v \\ &\qquad \inf_{x_j \sim (\overline{x}_j, \Sigma_{x_j})} \Pr\{a_v^T x_j \leq b_v\} \geq \tau_v. \end{aligned} \tag{3}$$

With the help of mathematics discussed in Lanckriet et al. [8], aforementioned problem can further be simplified to

$$\begin{aligned} &\max_{\tau_v, a_v \neq 0, b_v} \tau_v \\ &\text{st} \quad a_v^T \overline{x}_j + \kappa_v(\tau_v)\sqrt{a_v^T \Sigma_{x_j} a_v} \leq b_v \leq a_v^T \overline{x}_r + \kappa_v(\tau_v)\sqrt{a_v^T \Sigma_{x_r} a_v} \end{aligned} \tag{4}$$

here, $\kappa_v(\tau_v) = \sqrt{\dfrac{\tau_v}{1-\tau_v}}$. It can be seen that κ_v is directly related to τ_v, therefore we can maximize κ_v without considering τ_v. The upper bound and lower bound of b_v in the above equation are monotonically and unboundedly decreasing and increasing function of κ_v respectively, therefore, we can eliminate b_v at the optimum.

$$\begin{aligned} &\max_{\kappa_v, a_v \neq 0} \kappa_v \\ &\text{st} \quad a_v^T(\overline{x}_r - \overline{x}_j) \geq \kappa_v \left(\sqrt{a_v^T \Sigma_{x_r} a_v} + \sqrt{a_v^T \Sigma_{x_j} a_v}\right). \end{aligned} \tag{5}$$

For $\overline{x}_r = \overline{x}_j$, κ_v would be 0, τ_v would also be 0 and a meaningful solution would not exist. Therefore, we assume $\overline{x}_r \neq \overline{x}_j$, and set $a_v^T(\overline{x}_r - \overline{x}_j,) = 1$ without the loss of generality (for more detail, please refer to [18]). We get the following transformed SOCP, which is convex and bounded below

$$\kappa_{*v}^{-1} = \min \left\| \sum\nolimits_{x_r}^{1/2} a_v \right\|_2 + \left\| \sum\nolimits_{x_j}^{1/2} a_v \right\|_2$$
$$\text{st} \quad a_v^T(\overline{x}_r - \overline{x}_j) = 1. \tag{6}$$

The optimal b_{*v} can be computed as $b_{*v} = \left(a_{*v}^T \overline{x}_r - \kappa_{*v} \sqrt{a_{*v}^T \Sigma_{x_r} a_{*v}} \right) = \left(a_{*v}^T \overline{x}_j - \kappa_{*v} \sqrt{a_{*v}^T \Sigma_{x_j} a_{*v}} \right)$, where κ_{*v}, a_{*v} are the optimal values of κ_v and a_v respectively.

To utilise unlabelled information and taking motivation from Yoshiyama et al. [19], we derive Laplacian-Hessian-MLMPM for SSL. We propose the formulation for linear and kernelised version on the basis of representer theorem and further, solve the optimisation using block coordinate descent.

Let $\{x_r\}_{r=1}^{N_r}$, $\{x_j\}_{j=1}^{N_j}$ be the labelled samples corresponding to index set I_v and I_v' respectively and $\{z_i\}_{i=1}^{N_z}$ be the unlabelled samples which neither belongs to I_v nor I_v'. Here, $f_v(s) = a_v^T s - b_v$, $v = 1, \ldots, m$. Therefore, the optimisation problem in equation (6) can be rewritten as

$$\kappa_{v*}^{-1} = \min_{a_v} \left(\left\| \sum\nolimits_{x_r}^{1/2} a_v \right\|_2 + \left\| \sum\nolimits_{x_j}^{1/2} a_v \right\|_2 + \lambda w_{ik} (f_v(s_i) - f_v(s_k))^2 \right)$$
$$\text{st} \quad a_v^T(\overline{x}_r - \overline{x}_j) = 1. \tag{7}$$

Here, $s \in \{x_r\}_{r=1}^{N_r} \cup \{x_j\}_{j=1}^{N_j} \cup \{z_i\}_{i=1}^{N_z}$, and $(f_v(s_i) - f_v(s_k))^2 = (a_v^T s_i - a_v^T s_k)^2$, λ is the regularisation parameter that balances the effectiveness of the third term. The first two terms have the same interpretation as equation (6), the third term is added to take care of the semi-supervised setting. It propagates label information from the labelled samples to the unlabelled samples, such that if two samples s_i, s_k have similar output (i.e., $(f_v(s_i) - f_v(s_k))$ is small), then they should have similar label.

The above optimisation problem can be rewritten as

$$\kappa_{v*}^{-1} = \min_{a_v} \left(\left\| \sum\nolimits_{x_r}^{1/2} a_v \right\|_2 + \left\| \sum\nolimits_{x_j}^{1/2} a_v \right\|_2 + \lambda a_v^T Z (LH)_v Z^T a_v \right). \tag{8}$$

Here, $Z \in R^{n \times N}$ is a matrix composed of all labelled and unlabelled samples, elements of Z are ordered as $N = N_r + N_j + N_z$ and $(LH)_v$ is the combined Laplacian-Hessian regulariser and is defined as $LH = F^T(\delta_1 \times L)F + F^T(\delta_2 \times H)F$, which is linear combination of L and H. Let $M_v = Z(LH)_v Z^T$, then the final optimisation problem can be written as

$$\kappa_{v*}^{-1} = \min_{a_v} \left(\left\| \sum\nolimits_{x_r}^{1/2} a_v \right\|_2 + \left\| \sum\nolimits_{x_j}^{1/2} a_v \right\|_2 + \left\| \lambda^{1/2} M_v^{1/2} a_v \right\|_2 \right)$$
$$\text{st} \quad a_v^T(\overline{x}_r - \overline{x}_j) = 1. \tag{9}$$

Also the expression of b_{v*} can be rewritten as

$$\max_{\tau_v, a_v \neq 0, b_v} \tau_v$$
$$\text{st} \quad a_v^T \overline{x}_j + \kappa(\tau_v) \left(\sqrt{a_v^T \Sigma_{x_j} a_v} + \mu \sqrt{\lambda a_v^T M_v a_v} \right) \leq b_v \leq a_v^T \overline{x}_r - \kappa(\tau_v) \left(\sqrt{a_v^T \Sigma_{x_r} a_v} + (1-\mu) \sqrt{\lambda a_v^T M_v a_v} \right). \tag{10}$$

Here, $0 \leq \mu \leq 1$ and it balances the intrinsic geometry regularisation term, as mentioned in Yoshiyama et al. [19], it should be proportional to the number of class specific samples to the total number of samples, but for brevity, we take it to be 0.5.

3.1 *Nonlinear MLMPM-SSL*

Working on the line of Lanckriet et al. [8] and Yoshiyama et al. [19] the non-linear MLMPM-SSL is briefly described in this section. Let ϕ be the mapping from the input space R^n to R^F such that the nature of the data becomes linear in the higher dimensional space. The kernel matrix $K(.\,,.)$ is defined as

$$K(x_1, x_2) = \phi(x_1)^T \phi(x_2) \tag{11}$$

where the data is mapped as

$$\begin{aligned} x_r &\rightarrow \phi(x_r) \sim \left(\overline{\phi(x_r)}, \Sigma_{\phi(x_r)}\right) \\ x_j &\rightarrow \phi(x_j) \sim \left(\overline{\phi(x_j)}, \Sigma_{\phi(x_j)}\right). \end{aligned} \tag{12}$$

The optimisation problem (equation (9)) can be rewritten in kernelised form as

$$\begin{aligned} \kappa_{v*}^{-1} &= \min_{\gamma_v} \left(\sqrt{\gamma_v^T G_{x_r}^T G_{x_r} \gamma_v} + \sqrt{\gamma_v^T G_{x_j}^T G_{x_j} \gamma_v} + \sqrt{\lambda \gamma_v^T K(LH)_v K^T \gamma_v} \right) \\ &\text{st} \quad \gamma_v^T (E_{x_r}^T - E_{x_j}) = 1 \end{aligned} \tag{13}$$

where

$$\begin{aligned} (E_{x_r}^T)_i &= \frac{1}{N_r} \sum^{N_r} K(x_r, t_i) \\ (E_{x_j}^T)_i &= \frac{1}{N_j} \sum^{N_j} K(x_j, t_i) \end{aligned} \tag{14}$$

here, $i \in \{1, 2...N_r, N_r + 1, ..., N_r + N_j, ..., N_r + N_j + N_z\}$, $t_i \in \{x_1, x_2...x_{N_r}, x_1, ..., x_{N_j}\}$, $N_r(N_j)$ represents the number of samples in class x_r (class x_j), respectively and N_z represents the number of unlabelled samples. The kernel matrix K is defined as $K = \begin{pmatrix} K_{x_r} \\ K_{x_j} \\ K_z \end{pmatrix}$, here the first $N_r(N_j)$ rows are represented as $K_r(K_j)$ respectively. The Gram matrix P is given by

$$P = \begin{pmatrix} K_{x_r} - 1_{N_r} E_{x_r}^T \\ K_{x_j} - 1_{N_j} E_{x_j}^T \end{pmatrix} = \begin{pmatrix} \sqrt{N_r} G_{x_r} \\ \sqrt{N_j} G_{x_j} \end{pmatrix}. \tag{15}$$

Here, G_{x_r} and G_{x_j} represents the Gram matrix w.r.t. to class x_r and x_j, 1_{N_r}, 1_{N_j} represents vector of ones of dimension N_r and N_j respectively. The upper bound misclassification probability is given by

$$
\begin{aligned}
1-\tau_{v*} &= \frac{1}{1+\kappa_{v*}^{2}} \\
&= \frac{\left(\sqrt{\gamma_{v*}^{T}G_{x_r}^{T}G_{x_r}\gamma_{v*}}+\sqrt{\gamma_{v*}^{T}G_{x_j}^{T}G_{x_j}\gamma_{v*}}+\sqrt{\lambda\gamma_{v*}^{T}K(LH)_{v}K^{T}\gamma_{v*}}\right)^{2}}{1+\left(\sqrt{\gamma_{v*}^{T}G_{x_r}^{T}G_{x_r}\gamma_{v*}}+\sqrt{\gamma_{v*}^{T}G_{x_j}^{T}G_{x_j}\gamma_{v*}}+\sqrt{\lambda\gamma_{v*}^{T}K(LH)_{v}K^{T}\gamma_{v*}}\right)^{2}}.
\end{aligned}
\tag{16}
$$

The class of testing data z_{new} can be evaluated with the sign $\left(\left(\sum_{i=1}^{N_r+N_j}[\gamma_{v*}]_i K(t_i, z_{\text{new}})\right)-b_{v*}\right)$ where $b_{v*}=\gamma_{v*}^{T}E_{x_r}$ $-\kappa_{v*}(\sqrt{\gamma_{v*}^{T}G_{x_r}^{T}G_{x_r}\gamma_{v*}}+\mu\sqrt{\lambda\gamma_{v*}^{T}K(LH)_{v}K^{T}\gamma_{v*}})$ and γ_{v*}, κ_{v*} are the optimal values of γ_v and κ_v, respectively.

3.2 *Computation of Hessian matrix*

The Hessian matrix H can be computed by using the following steps.

1. For each training point x_i, find its k nearest neighbours to define a neighbourhood matrix N_i.
2. Estimate the orthonormal coordinate system of the tangent space $T_{x_i}(Q)$ by performing a singular value decomposition of $X_i = UDV^T$ on N_i.
3. Perform Gram-Schimidt orthogonalisation process on the matrix $Q_i = [U_1 ... U_d\ U_{11}\ U_{12} ... U_{dd}]$ and resulting H_i. The Frobenius norm of H_i is $H_i H_i^T$.
4. Finally, construct Hessian by summing up $H_i^T H_i$ over all samples.

3.3 *Computation of Laplacian matrix*

The Laplacian matrix L is computed using the following steps.

1. Get nearest neighbours of each sample.
2. Get must link and cannot link (side information) from labelled samples.
3. Compute edge weight matrix W, by using kernel similarity of nearest neighbours and must link edges, set cannot link weight to zero.
4. Finally, construct normalised Laplacian as $L = I - D^{-1/2}WD^{-1/2}$, D is the diagonal matrix containing sum of each row at diagonal entries.

3.4 *Multi-manifold regularisation using Hessian and Laplacian regularisation*

Multi-manifold regularisation can be effectively calculated as in [20-22].

$$LH = F^{T}(\delta_1 \times L)F + F^{T}(\delta_2 \times L)F$$

where L and H represent the Laplacian and Hessian regularisation respectively and optimal values of δ_1 and δ_2 are searched in [0.01, 0.1, 1].

4. Algorithm

The optimisation of MLMPM-SSL i.e, the problem defined in equation (9) can be solved by block coordinate descent [23] in a similar manner as in [18]. For that, we define $a_v = a_{v0} + Fu$, where $u \in R^{n-1}$, $a_{v0} = \dfrac{\overline{x}_r - \overline{x}_j}{\left\|\overline{x}_r - \overline{x}_j\right\|_2^2}$, $F \in R^{n\times(n-1)}$ is an orthogonal matrix whose columns span the subspace of vectors orthogonal to $(\overline{x}_r - \overline{x}_j)$. After eliminating the constraint from the equation, the aforementioned optimisation equation can be rewritten as an unconstrained SOCP.

$$\min_{u,\beta_v>0,n_v>0,\gamma_v>0} \beta_v + \frac{1}{\beta_v}\left\|\sum\nolimits_{x_i}^{1/2}(a_{v0}+Fu)\right\|_2 + \eta_v + \frac{1}{\eta_v}\left\|\sum\nolimits_{xj}^{1/2}(a_{v0}+Fu)\right\|_2 + \gamma_v + \frac{\lambda}{\gamma_v}\left\|(LH)^{1/2}(a_{v0}+Fu)\right\|_2^2 \tag{17}$$

In the iterative procedure, we update β_v, η_v, γ_v and u alternatively. The pseudocode for algorithm is presented in Algorithm 1 [8, 18, 19].

Algorithm 1. Pseudocode for the iterative procedure for MLMPM-SSL

```
Output: label of $X_{\text{test}}$
For each label $v$
    Get estimates
        $\bar{x}_r, \bar{x}_j, \Sigma_{x_r}, \Sigma_{x_j}, (LH)_v$
    Compute
        $a_0 \leftarrow (\bar{x}_r - \bar{x}_j) / \| \bar{x}_r - \bar{x}_j \|_2^2$
        Let the columns of $F$ be orthogonal to $(\bar{x}_r - \bar{x}_j)$
        $S \leftarrow F^T \Sigma_{x_r} F, \quad H \leftarrow F^T \Sigma_{x_j} F, \quad M \leftarrow \lambda F^T (LH)_v F$
        $s \leftarrow F^T \Sigma_{x_r} a_{v0}, \quad h \leftarrow F^T \Sigma_{x_j} a_{v0}, \quad m \leftarrow \lambda F^T (LH)_v a_{v0}$
    Initialise $\beta_{v1} = 1$, $\eta_{v1} = 1$, $\gamma_{v1} = 1$
    Repeat
        $M_{LS} \leftarrow S / \beta_{vk} + H / \eta_{vk} + M / \gamma_{vk} + \delta I$
        $b_{LS} \leftarrow - \left( s / \beta_{vk} + h / \eta_{vk} + m / \gamma_{vk} \right)$
        solve $M_{LS} u_k = b_{LS}$ w.r.t. $u_k$
        $a_{vk} \leftarrow a_{v0} + F u_k$
        $\beta_{vk+1} \leftarrow \sqrt{a_{vk}^T \Sigma_{x_r} a_{vk}}, \eta_{vk+1} \leftarrow \sqrt{a_{vk}^T \Sigma_{x_j} a_{vk}}, \gamma_{vk+1} \leftarrow \sqrt{\lambda a_{vk}^T M a_{vk}}$
        $vk \leftarrow vk + 1$
    Stop when ( $\beta_{vk} + \eta_{vk} + \gamma_{vk}$ ) is small or maximum iteration is reached.
    Assign
        $a_v \leftarrow a_{vk}$
        $\kappa_v \leftarrow 1 / \ (\beta_{vk} + \eta_{vk} + \gamma_{vk})$
        $b_v \leftarrow a_v^T \bar{x}_r - \kappa_v \left( \beta_{vk} + \frac{1}{2} \gamma_{vk} \right)$
    Report optimal parameter $a_{*v}$, $b_{*v}$ which are used to predict the label of testing patterns $X_{\text{test}}$.
    $\tau = \sum_{v=1}^{m} \frac{\kappa_v^2}{\kappa_v^2 + 1}$
end For
Outlabel = sign( $X_{\text{test}} * a_{*1} - b_{*1}$), sign($X_{\text{test}} * a_{*2} - b_{*2}$), …, sign($X_{\text{test}} * a_{*m} - b_{*m}$)
```

5. Experiments

The experiments are performed on well-known multi-label datasets using 10-fold cross-validation in MATLAB version 9.4 under Microsoft Windows environment on a machine with 3.40 GHz CPU and 16 GB RAM. For each fold training data, we randomly hide some percent of the labels of the training samples so as to adapt to semi-supervised learning scenario. We used the following hyperparameter settings for comparison of our proposed model with related algorithms: For all algorithms, where required we searched best value of *k*-nearest neighbour in [2, 4, ..., 10], kernel parameter σ was searched in $\{2^4, ..., 2^{-4}\}$. The hyperparameters were tuned corresponding to the best accuracy on the labelled training set.

Estimating missing labels: We fix 20% samples, i.e. 10% positive and 10% negative for each label, but by doing so a small fraction of missing label samples are created. We complete such missing labels by estimation of their likelihood by the following:

First, we calculate label correlation matrix L ($L \in R^{m \times m}$) from the available label information.

$$L(c_1, c_2) = \frac{\left|Y_{c_1} \cap Y_{c_2}\right| + s}{\left|Y_{c_1} + 2s\right|} \quad \text{where} \quad 1 \le c_1, c_2 \le m \tag{18}$$

and Y_{c_1} set of labelled instances annotated with label c_1 and s is a small constant. Now, estimating the likelihood of missing label c for ith instance

$$\tilde{y}_{ic} = \begin{cases} y_i^T L(\cdot, c), & \text{if} \quad y_{ic} = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{19}$$

Then, final label of ith instance for label c is given as $y_{ic} = \dfrac{\tilde{y}_{ic}}{\left\|\tilde{y}_{ic}\right\|}$ for $y_{ic} = 0$.

5.1 *Compared algorithms*

We have compared MLMPM with popular plane based classifiers along with algorithms which considers label co-occurency matrix while training along with some popular multi-label algorithms.

1. MLMPM-SSL (**Proposed**): Kernelised MLMPM-SSL model, Gaussian kernel is used. The hyperparameters were tuned corresponding to the best accuracy on the training set. Kernel parameter σ was searched in $\{2^4, ..., 2^{-4}\}$. We search the optimal number of nearest neighbour in [2, 4, 6, 8].
2. MLTSVM [24]: Multi-label Twin Support Vector Machines (MLTSVM) using rbf kernel. MLTSVM is an extension of twin support vector machines to multi-label learning. It finds several non-parallel hyperplanes to infer multi-label information embedded in the data. The parameters c_1 and c_2 are taken to be unity. Kernel parameter σ for MLTSVM was searched in $\{2^4, ..., 2^{-4}\}$.
3. MLSVM(K): Kernelised Multi-label Support Vector Machines (fitcsvm [25]). MLSVM is another plane based classifier, extension of SVM to multi-label learning, which aims to construct a plane so as to separate the classes with maximum margin.
4. MLKNN [26]: Multi-label lazy learning approach using *k*-nearest neighbours to infer multi-label information from the data. It is not a plane based classifier which works on the underlying principle of maximum apriori. We search the optimal number of nearest neighbour in [2, 4, 6, 8] and set smoothness value as 1.
5. TRAM [27]: i.e., a transductive multi-label classification algorithm via label set propagation. We implement TRAM in a supervised scenario and take the default setting for *k* as 10.
6. CPNL [28]: Cost-sensitive multi-label learning with positive and negative label pairwise correlations. An approach to deal with imbalance multi-label classification. We find the parameters λ_1, λ_2, λ_3 in $\{10^{-4}, ..., 10^2\}$ on

training data. Kernel parameter σ for CPNL was searched in $\{2^4, ..., 2^{-4}\}$.

7. C2AE [29]: Canonical Correlated AutoEncoder (C2AE). This model integrates canonical correlation and autoencoders using deep learning architecture to exploit label dependencies in multi-label learning. We follow the parameter settings as described in the paper and tune α in $\{0.1, 0.2, ..., 2, ..., 10\}$. The number of epoch is set to 50.

5.2 *Computational complexity*

The main optimisation of MPM is a second order cone programming problem which has a complexity of $O(N^3)$ in the worst case. Considering the computation for mean, variance, and geometric information, the total complexity of MLMPM-SSL is $O(mN^3 + mN^2 + mN^2)$ which is approximately $O(mN^3)$. Here, m is the number of labels and N is the total number of samples.

5.3 *Datasets*

The characteristics of multi-label datasets used are described in Table 1. Cardinality measures the average number of labels associated with each instance, and density is defined as cardinality divided by the number of labels.

We selected medium size datasets (large datasets have not been used because of large computation time and lack of resources in the COVID-19 lockdown), although the model can be extended to large dataset easily, out of these Emotions, Image, Yeast and Scene datasets have numerical features, PlantGO, GnegativeGO have binary features. Also, we normalise each dataset except PlantGO, GnegativeGO as they have binary features. We have carefully selected these datasets so as to check the performance of the proposed algorithm in diverse conditions.

Table 1. Datasets used in the experiment

Dataset	Instance	Features	Label	Cardinality	Density	Domain
Emotions [30]	593	72	6	1.869	0.311	Music
Image [30]	2000	294	5	1.240	0.247	Image
Scene [30]	2407	294	6	1.074	0.179	Image
Yeast [30]	2417	103	14	4.237	0.303	Biology
PlantGO [31]	978	3091	12	1.079	0.090	Biology
GnegativeGO [31]	1392	1717	8	1.046	0.131	Biology
TMC [31]	2000	500	22	2.158	0.098	Text

5.4 *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 ith 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 criterion indicates the fraction of labels that are incorrectly predicted to the total number of labels.

$$\text{HL} = \frac{1}{N_t \times m} \sum_{i=1}^{N_t} \sum_{j=1}^{m} [h_j(x_i) \neq y_{ij}] \tag{20}$$

2. **Exact match** (EM): This metric evaluates the fraction of examples for which the predicted label set is same as the ground truth label set.

$$\text{EM} = \frac{1}{N_t}\sum_{i=1}^{N_t}[h(x_i) = y_i] \tag{21}$$

3. **F1-example** (F1): This metric evaluates the harmonic mean of precision and recall averaged for all instances [32].

$$\text{F1} = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{2|h(x_i) \cap y_i|}{|h(x_i)| + |y_i|} \tag{22}$$

$$\text{MacroF1} = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{2\sum_{j=1}^{m}\hat{y}_i^j y_i^j}{\sum_{j=1}^{l}\hat{y}_i^j + \sum_{j=1}^{l}y_i^j}$$

$$\text{MicroF1} = \frac{2\sum_{j=1}^{m}\sum_{i=1}^{N_t}\hat{y}_i^j y_i^j}{\sum_{j=1}^{m}\sum_{i=1}^{N_t}\hat{y}_i^j + \sum_{j=1}^{m}\sum_{i=1}^{N_t}y_i^j} \tag{23}$$

Here, $\hat{y}_i^j$ is the predicted value and y_i^j is the original value as in ground truth label. MicroF1 computes across each label and then averages them, macroF1 computes across each sample and then averages them.

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

$$\text{AP} = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{1}{y_i}\sum_{y\in \mathrm{y}_i}\frac{(y' \in \mathrm{y}_i \mid \text{rank}(x_i, y') \leq \text{rank}_f(x_i, y)}{\text{rank}_f(x_i, y)} \tag{24}$$

5.5 *Results and statistical analysis*

Table 2 reports the results of the proposed model at different labelled percentages of training data. Best results are highlighted in bold. Table 3 reports the average results (mean ± standard deviation) of the comparing algorithms versus the proposed model at 100 percent labelled training data. As mentioned in various papers, Hessian which is a second order differential operator gives the best performance under low percentage of labelled data whereas Laplacian which is a first order differential overfits the data and therefore performs poorly under low label data. Experimentation on various datasets suggest that although Laplacian can give better training accuracy but generally it underperforms in testing phase. Due to brevity, the experimental details are avoided here. A general observation is that when label information is abundant Laplacian and Hessian based manifold regularisation give nearly the same results and using a multi-manifold regularisation generally yields the best results. The results shown confirm the same.

The proposed model has superior performance on Emotions, Image, GnegativeGO, Scene and PlantGO datasets. This clearly indicates MLMPM-SSL model is able to effectively utilise the geometric and statistical information inherent in the multi-label datasets. The proposed model has competitive performance on Yeast dataset where MLSVM due to its discriminating nature along with kernelisation is able to achieve superior results. CPNL has average performance on the datasets and it shows to be less effective for binary feature datasets. CPNL's performance is restricted because of its sensitivity to its parameters and also due to inaccurate modelling of label correlations. MLTSVM adopts twin support vector machines to multi-label setting, authors set the threshold for each proximal hyperplane to $\min\left(\frac{1}{\|w_v\|}\right)$. This causes the MLTSVM to have improper prediction on rare labels compared with the proposed model and thus gets a lesser exact match and macroF1. MLKNN achieves lesser results on GnegativeGo and PlantGo due to high dimensionality of these datasets. TRAM gets comparable results with the proposed model except for macroF1 due to improper prediction on rare labels. A general trend is observed that the proposed model's performance increases as the percentage of label data increases. Also, when the label data is 20 percent, the model gradually approaches the worst case performance but never falls below worst case accuracy obtained explicitly on the training data. The Hessian-Laplacian multi-manifold regularisation enables the proposed model to effectively capture the information from unlabelled samples on the underlying manifold. Overall, the MLMPM-SSL model has good performance on real world datasets and exhibits superior performance in most cases. Also, the worst case accuracy (τ) as predicted by the MLMPM-SSL model is highly accurate as the Hamming loss obtained on testing data is always lesser than $1 - \tau$ (worst case Hamming loss) obtained in the training phase.

Table 2(A). Results of MLMPM-SSL on Emotions, Image, GnegativeGO and Scene dataset

Label data	20%	30%	40%	50%	60%	70%	80%	90%	100%
					Emotions				
Exact match (↑)	0.209±0.013	0.229±0.017	0.234±0.018	0.251±0.0144	0.254±0.021	0.268±0.015	0.298±0.022	0.303±0.019	**0.322±0.022**
Hamming loss (↓)	0.270±0.007	0.235±0.009	0.227±0.009	0.213±0.007	0.207±0.008	0.206±0.008	0.201±0.008	0.198±0.008	**0.185±0.006**
MacroF1 (↑)	0.624±0.010	0.617±0.016	0.605±0.016	0.626±0.012	0.627±0.016	0.634±0.015	0.647±0.016	0.649±0.015	**0.678±0.014**
MicroF1 (↑)	0.638±0.010	0.634±0.017	0.628±0.017	0.648±0.013	0.650±0.014	0.656±0.014	0.665±0.014	0.668±0.014	**0.693±0.012**
Avg precision (↑)	0.706±0.010	0.713±0.013	0.721±0.012	0.742±0.010	0.742±0.010	0.741±0.012	0.748±0.011	0.757±0.008	**0.774±0.009**
Worst case accuracy	0.660	0.662	0.630	0.621	0.596	0.588	0.574	0.573	0.562
					Image				
Exact match (↑)	0.114±0.007	0.17±0.008	0.226±0.009	0.283±0.01	0.313±0.008	0.335±0.006	0.413±0.013	0.443±0.008	**0.468±0.008**
Hamming loss (↓)	0.413±0.005	0.348±0.004	0.296±0.003	0.242±0.004	0.223±0.002	0.208±0.002	0.18±0.003	0.17±0.003	**0.16±0.003**
MacroF1 (↑)	0.485±0.006	0.51±0.006	0.525±0.005	0.538±0.006	0.555±0.004	0.576±0.006	0.616±0.008	0.638±0.006	**0.656±0.008**
MicroF1 (↑)	0.487±0.006	0.51±0.006	0.524±0.006	0.54±0.006	0.555±0.004	0.575±0.005	0.612±0.008	0.635±0.007	**0.655±0.008**
Avg precision (↑)	0.643±0.007	0.688±0.009	0.712±0.008	0.724±0.007	0.732±0.007	0.745±0.006	0.756±0.009	0.755±0.006	**0.77±0.007**
Worst case accuracy	0.632	0.626	0.635	0.631	0.628	0.617	0.62	0.624	0.655
					GnegativeGO				
Exact match (↑)	0.504±0.009	0.539±0.011	0.752±0.017	0.872±0.011	0.909±0.006	0.908±0.007	0.902±0.006	0.909±0.005	**0.915±0.006**
Hamming loss (↓)	0.118±0.003	0.088±0.003	0.040±0.002	0.019±0.001	0.015±0.001	0.015±0.001	0.016±0.001	**0.014±0.001**	**0.014±0.001**
MacroF1 (↑)	0.532±0.012	0.637±0.024	0.733±0.027	0.770±0.027	0.780±0.025	0.780±0.025	0.778±0.024	0.782±0.023	**0.789±0.022**
MicroF1 (↑)	0.682±0.006	0.742±0.008	0.860±0.008	0.927±0.005	0.940±0.004	0.940±0.004	0.937±0.003	0.941±0.003	**0.945±0.004**
Avg precision (↑)	0.826±0.006	0.895±0.009	0.921±0.006	0.953±0.003	0.956±0.005	0.956±0.005	0.954±0.005	0.957±0.004	**0.958±0.004**
Worst case accuracy	0.786	0.762	0.752	0.733	0.691	0.686	0.694	0.677	0.666
					Scene				
Exact match (↑)	0.212±0.007	0.417±0.008	0.507±0.011	0.621±0.009	0.626±0.005	0.628±0.012	0.643±0.009	0.660±0.007	**0.664±0.006**
Hamming loss (↓)	0.255±0.003	0.160±0.003	0.112±0.002	0.088±0.001	0.083±0.001	0.083±0.002	0.079±0.002	0.076±0.001	**0.074±0.001**
MacroF1 (↑)	0.572±0.004	0.680±0.006	0.734±0.005	0.753±0.005	0.758±0.005	0.763±0.006	0.769±0.007	0.780±0.006	**0.787±0.004**
MicroF1 (↑)	0.557±0.004	0.654±0.006	0.714±0.006	0.744±0.005	0.751±0.004	0.755±0.006	0.763±0.007	0.774±0.005	**0.777±0.004**
Avg Precision (↑)	0.71±0.004	0.780±0.004	0.807±0.006	0.824±0.006	0.828±0.003	0.831±0.004	0.841±0.006	**0.847±0.005**	0.841±0.005
Worst case accuracy	0.629	0.612	0.613	0.605	0.606	0.605	0.584	0.583	0.588

Table 2(B). Results of MLMPM-SSL on Yeast and PlantGO dataset

Label data	20%	30%	40%	50%	60%	70%	80%	90%	100%
					Yeast				
Exact match (↑)	0.070±0.002	0.102±0.004	0.12±0.004	0.140±0.007	0.151±0.004	0.158±0.006	**0.168±0.006**	0.168±0.005	0.167±0.006
Hamming loss (↓)	0.342±0.003	0.273±0.001	0.247±0.001	0.232±0.001	0.227±0.001	0.219±0.001	0.215±0.002	0.213±0.001	**0.208±0.001**
MacroF1 (↑)	0.442±0.004	**0.453±0.005**	0.452±0.007	0.452±0.005	0.446±0.005	0.449±0.005	0.45±0.007	0.45±0.007	0.45±0.006
MicroF1 (↑)	0.524±0.004	0.574±0.003	0.6±0.004	0.613±0.002	0.617±0.003	0.629±0.002	0.633±0.003	0.633±0.003	**0.640±0.002**
Avg precision (↑)	0.575±0.005	0.633±0.004	0.656±0.004	0.667±0.003	0.672±0.004	0.679±0.004	0.683±0.004	0.684±0.005	**0.687±0.003**
Worst case accuracy	0.604	0.593	0.599	0.583	0.567	0.558	0.556	0.57	0.564
					PlantGO				
Exact match (↑)	0.392±0.022	0.515±0.014	0.597±0.013	0.628±0.013	0.626±0.015	**0.644±0.017**	0.620±0.017	0.625±0.015	0.639±0.017
Hamming loss (↓)	0.076±0.004	0.058±0.002	0.047±0.001	0.043±0.001	0.042±0.001	**0.04±0.001**	0.043±0.001	0.042±0.001	0.042±0.001
MacroF1 (↑)	0.619±0.012	0.662±0.014	**0.675±0.015**	0.657±0.016	0.649±0.014	0.656±0.015	0.622±0.014	0.610±0.015	0.606±0.015
MicroF1 (↑)	0.676±0.011	0.718±0.008	0.737±0.006	0.755±0.008	0.761±0.009	**0.764±0.01**	0.745±0.012	0.748±0.011	0.749±0.012
Avg precision (↑)	**0.825±0.007**	0.816±0.008	0.755±0.006	0.777±0.008	0.782±0.01	0.780±0.011	0.769±0.012	0.772±0.012	0.772±0.01
Worst case accuracy	0.666	0.668	0.656	0.653	0.645	0.638	0.64	0.637	0.640

Table 3(A). Results on Emotion, Image and Scene

Evaluation metric	MLMPM-SSL	MLTSVM	MLSVM(K)	MLKNN	TRAM	CPNL	C2AE
				Emotions, $\tau = 0.562$			
Time	0.370 ± 0.025	0.194 ± 0.002	0.086 ± 0.001	0.0559 ± 0.0002	0.0425 ± 0.0001	0.3801 ± 0.0138	7.660±0.019
Exact match(↑)	**0.322±0.022**	0.285±0.023	0.298±0.023	0.270±0.026	0.261±0.0177	0.290±0.018	0.020±0.010
Hamming loss(↓)	**0.185±0.006**	0.195±0.009	0.186±0.007	0.201±0.009	0.217±0.008	0.194±0.007	0.515±0.011
MacroF1(↑)	**0.678±0.014**	0.674±0.019	0.633±0.016	0.601±0.020	0.645±0.014	0.652±0.017	0.327±0.005
MicroF1(↑)	0.693±0.012	**0.694±0.017**	0.663±0.015	0.642±0.019	0.656±0.014	0.680±0.015	0.447±0.010
Avg precision(↑)	**0.774±0.009**	0.761±0.016	0.752±0.014	0.739±0.013	0.748±0.012	0.754±0.015	0.537±0.016
				Image, $\tau = 0.655$			
Time	6.625 ± 0.319	8.671 ± 0.130	0.544 ± 0.0136	1.217 ± 0.002	0.9923 ± 0.0042	0.3582 ± 0.0689	26.918± 0.738
Exact match(↑)	**0.468±0.008**	0.311±0.028	0.346±0.020	0.356±0.026	0.461±0.019	0.308±0.026	0.187±0.014
Hamming loss(↓)	**0.16±0.003**	0.226±0.012	0.180±0.007	0.186±0.008	0.191±0.009	0.203±0.007	0.298±0.009
MacroF1(↑)	**0.656±0.008**	0.610±0.019	0.483±0.019	0.502±0.023	0.581±0.017	0.479±0.031	0.401±0.017
MicroF1(↑)	**0.655±0.008**	0.613±0.02	0.501±0.020	0.513±0.023	0.583±0.018	0.495±0.031	0.455±0.016
Avg precision(↑)	**0.77±0.007**	0.741±0.017	0.666±0.010	0.685±0.011	0.734±0.012	0.654±0.021	0.623±0.014
				Scene, $\tau = 0.588$			
Time	13.908 ± 0.143	10.581 ± 0.057	0.559 ± 0.009	1.787 ± 0.008	1.417 ± 0.0017	13.8681 ± 0.0969	34.588±0.092
Exact match(↑)	0.664±0.006	0.546±0.011	0.636±0.013	0.633±0.009	**0.697±0.006**	0.388±0.012	0.371± 0.010
Hamming loss(↓)	**0.074±0.001**	0.105±0.003	0.075±0.003	0.085±0.002	0.091±0.002	0.146±0.003	0.137±0.003
MacroF1(↑)	**0.787±0.004**	0.769±0.006	0.759±0.010	0.742±0.007	0.741±0.006	0.649±0.007	0.561±0.007
MicroF1(↑)	**0.777±0.004**	0.746±0.007	0.757±0.010	0.740±0.007	0.736±0.005	0.644±0.006	0.572±0.007
Avg precision(↑)	**0.841±0.005**	0.820±0.006	0.823±0.006	0.819±0.005	0.831±0.003	0.753±0.008	0.683±0.009

Table 3(B). Results on GnegativeGO,Yeast and PlantGO

Evaluation metric	MLMPM-SSL	MLTSVM	MLSVM(K)	MLKNN	TRAM	CPNL	C2AE
			GnegativeGO, $\tau = 0.666$				
Time	4.928 ± 0.171	5.746 ± 0.122	2.486 ± 0.022	7.365 ± 0.024	3.762 ± 0.008	3.777 ± 0.033	32.829±1.375
Exact match(↑)	**0.915±0.006**	0.846±0.012	0.900±0.006	0.887±0.007	0.902±0.006	0.382±0.008	0.648±0.017
Hamming loss(↓)	**0.014±0.001**	0.024±0.002	0.016±0.001	0.0196±0.001	0.019±0.001	0.155±0.002	0.053±0.002
MacroF1(↑)	**0.789±0.022**	0.580±0.004	0.755±0.019	0.727±0.020	0.708±0.017	0.071±0.001	0.540±0.015
MicroF1(↑)	**0.945±0.004**	0.910±0.006	0.934±0.005	0.922±0.005	0.925±0.004	0.391±0.01	0.790±0.011
Avg Precision(↑)	**0.958±0.004**	0.951±0.004	0.943±0.004	0.937±0.004	0.945±0.004	0.529±0.007	0.828±0.011
			Yeast, $\tau = 0.564$				
Time	09.344 ± 0.514	28.638 ± 0.082	2.176 ± 0.021	0.989 ± 0.009	0.815 ± 0.001	13.954 ± 0.059	31.754±0.093
Exact match(↑)	0.167±0.006	0.138±0.005	**0.195±0.007**	0.180±0.006	0.153±0.005	0.143±0.007	0.124±0.006
Hamming loss(↓)	0.208±0.001	0.210±0.003	**0.187±0.003**	0.194±0.002	0.213±0.003	0.203±0.003	0.222±0.002
MacroF1(↑)	**0.45±0.006**	0.296±0.002	0.369±0.005	0.371±0.005	0.409±0.005	0.318±0.002	0.408±0.008
MicroF1(↑)	0.640±0.002	0.617±0.006	**0.654±0.006**	0.639±0.006	0.646±0.007	0.623±0.006	0.633±0.004
Avg Precision(↑)	0.687±0.003	0.663±0.005	0.682±0.005	0.670±0.005	**0.693±0.007**	0.662±0.005	0.680±0.004
			PlantGO, $\tau = 0.640$				
Time	1.848 ± 0.099	5.152 ± 0.040	4.23 ± 0.029	5.115 ± 0.005	3.474 ± 0.011	1.650 ± 0.039	44.779 ± 0.061
Exact match(↑)	**0.639±0.017**	0.549±0.019	0.566±0.018	0.576±0.015	0.614±0.016	0.142±0.046	0.334±0.009
Hamming loss(↓)	**0.042±0.001**	0.066±0.002	0.046±0.002	0.052±0.001	0.060±0.002	0.153±0.015	0.074±0.002
MacroF1(↑)	**0.606±0.015**	0.248±0.004	0.449±0.020	0.468±0.019	0.503±0.017	0.056±0.006	0.163±0.010
MicroF1(↑)	**0.749±0.012**	0.648±0.012	0.696±0.015	0.685±0.010	0.65±0.013	0.289±0.015	0.450±0.013
Avg Precision(↑)	**0.772±0.01**	0.728±0.012	0.696±0.014	0.720±0.013	0.711±0.013	0.393±0.02	0.499±0.009

We select 2,000 random samples from TMC dataset and plot the results for Laplacian, Hessian and Laplacian-Hessian MPM models. In Figures 1 to 5, the y-axis is the metric score and x-axis represents the percentage of labelled training data available. It can be seen that when label data percentage is high, Laplacian is able to achieve better results, meanwhile Hessian performs better under low percentage of labelled data. Using a combination of the two approaches, generally yields stable results.

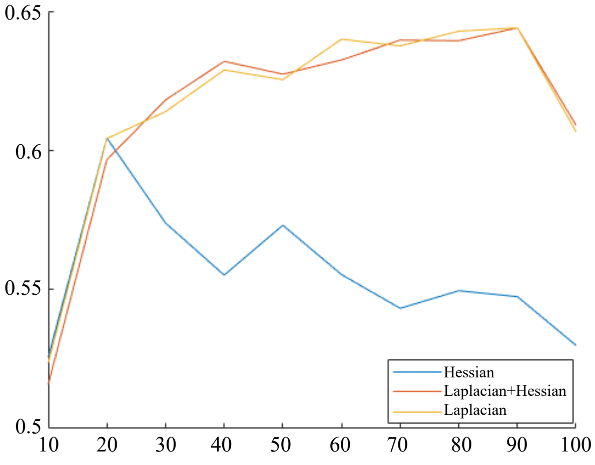

Figure 1. Average precision scores on TMC dataset

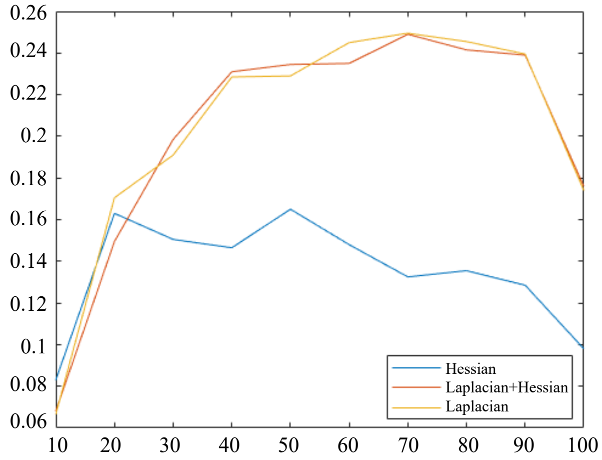

Figure 2. Exact match scores on TMC dataset

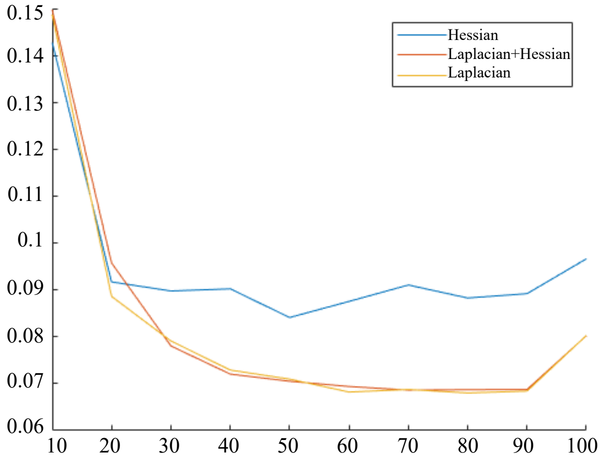

Figure 3. Hamming loss scores on TMC dataset

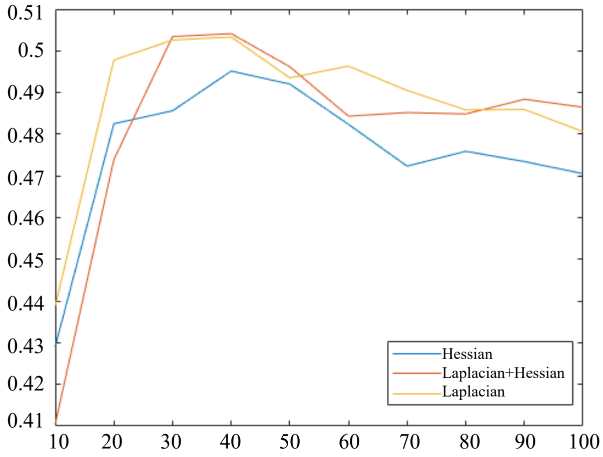

Figure 4. MacroF1 scores on TMC dataset

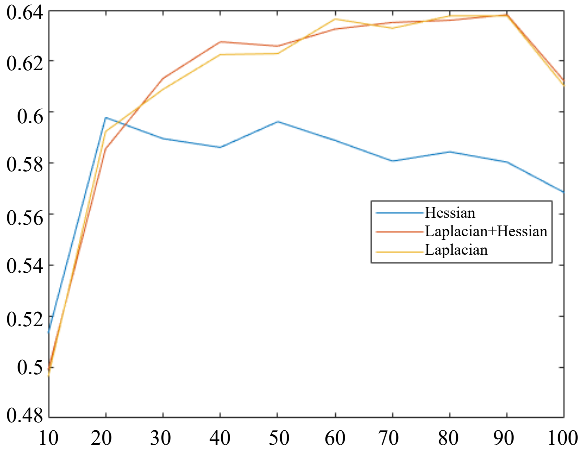

Figure 5. MicroF1 scores on TMC dataset

6. Application to predict Amazon rainforest satellite images

The Amazon rainforest is home to rich and diverse flora and fauna, some of which are limited to the region only. The thick forest cover and richness of wildlife in the region has attracted a lot of wildlife researchers, enthusiasts, environmentalists etc. But in the past decades, the rising deforestation in the area has been a major concern for governments and like minded communities. Planet, a satellite imaging company, recently released a dataset of more than 100,000 images from the Amazon basin and hosted a Kaggle competition involving labelling the atmosphere and ground features in the images [33]. The high-resolution Planet images enable identification of specific causes of deforestation and differentiation of legal and illegal human developments in the region.

The Amazon satellite images dataset is a multi-label dataset, has nearly 40,000 images in the training set and the rest in the testing set. The dataset has 17 labels which can be broadly categorised as follows.

1. Atmospheric labels: clear, partly cloudy, cloudy and hazy
2. Common labels: primary, water, habitation, agriculture, road, cultivation and bare ground
3. Rare labels: artisinal mine, blooming, blow down, conventional mine, selective logging and slash burn.

Manual labelling of these images is a cumbersome task which requires intense human efforts. Also, the manual task of image annotation may introduce human error. In many images, the labels are noisy. Therefore, we assume a semi-supervised scenario and then apply our proposed model to effectively learn from labelled and unlabelled images.

6.1 *Generation of features from images*

Due to the limitations of our system during the COVID-19 lockdown, we select a subset of 3,000 sample images for testing the efficacy of the plane based classifiers. We choose the tiff format of images to construct the features for the images. The images have 4 colour bands corresponding to the CMYK (cyan, magenta, yellow and key/black) representation. The label cardinality of dataset is 2.87.

To generate features from image, we follow a similar approach as in [34]. We divide the image into smaller 64 grids and report the colour moments, i.e. mean and standard deviation of each grid as features of that image. The number of features for each image are 64 × 2 × 4 which is 512. The process is illustrated in Figure 6.

6.2 *Results and discussion on Amazon rainforest's satellite dataset*

We evaluate the Amazon rainforest's satellite dataset on the proposed multi-label classifier MLMPM-SSL under label percentage varying from 20% to 100% with 10% interval, results which are reported in Table 4.

As it can be inferred from the results, the metrics show better performance when more label data is available. The MLMPM-SSL not only achieve good results when label data is spare but also provides a worst case accuracy which is desirable for large datasets and can provide valuable insights into future predictions.

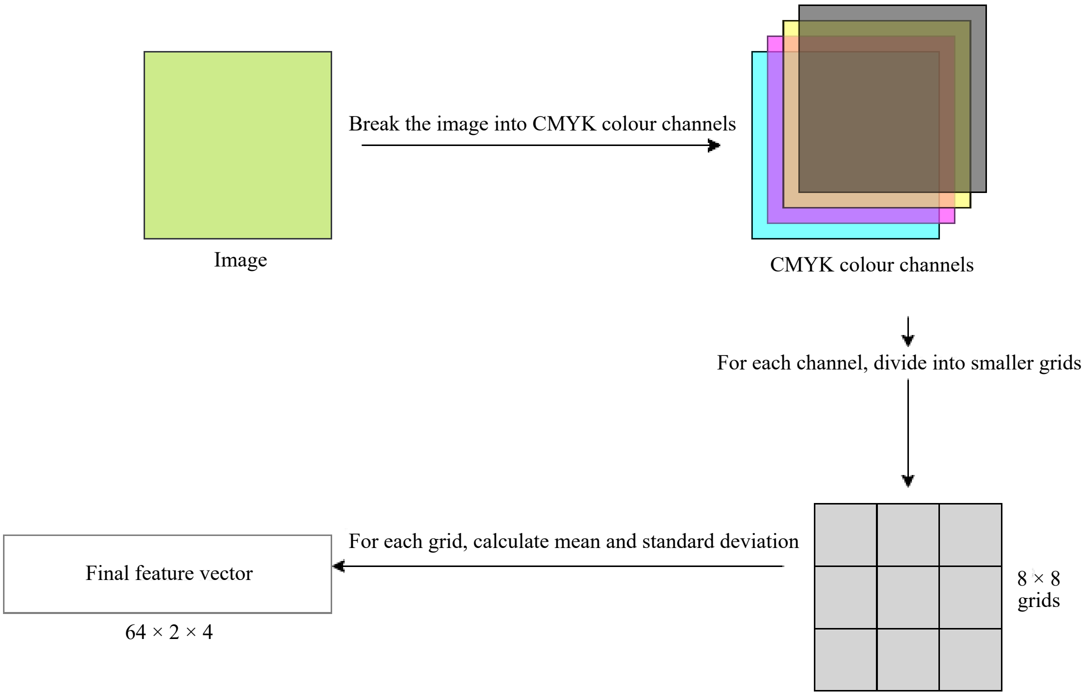
Image
Break the image into CMYK colour channels
CMYK colour channels
For each channel, divide into smaller grids
8 × 8 grids
For each grid, calculate mean and standard deviation
Final feature vector
64 × 2 × 4

Figure 6. Feature generation from an image

Table 4. Results of MLMPM-SSL on Amazon rainforest's satellite images dataset

Label data	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Exact match (↑)	0.165±0.009	0.207±0.003	0.182±0.013	0.228±0.011	0.262±0.013	0.276±0.009	0.288±0.007	0.309±0.004	**0.329±0.003**
Hamming loss (↓)	0.187±0.002	0.171±0.001	0.171±0.004	0.134±0.002	0.125±0.002	0.123±0.002	0.12±0.002	0.114±0.002	**0.107±0.001**
MacroF1 (↑)	0.287±0.003	0.275±0.005	0.263±0.005	0.262±0.006	0.275±0.007	0.286±0.006	0.306±0.004	0.318±0.005	**0.333±0.006**
MicroF1 (↑)	0.56±0.007	0.571±0.005	0.573±0.007	0.63±0.004	0.649±0.005	0.661±0.005	0.674±0.006	0.687±0.004	**0.706±0.003**
Avg precision (↑)	0.625±0.009	0.653±0.005	0.656±0.007	0.697±0.003	0.708±0.005	0.716±0.004	0.725±0.003	0.737±0.003	**0.751±0.002**
Worst case accuracy	0.801	0.825	0.818	0.778	0.742	0.721	0.705	0.698	0.696

7. Discussion and conclusions

In this paper, we have proposed MLMPM-SSL model to handle multi-label semi-supervised learning, wherein we effectively utilise the unlabelled samples along with labelled samples by a multi-manifold regularisation of Hessian-Laplacian regularisation. Although our proposed model is effective at learning from semi-supervised setting in multi-label learning yet exploiting label correlation still remains. Many similar models rely on self-training to predict the labels in training data and then consider label correlation, but this approach is highly inefficient when there is less label data and the self-training approach fails miserably. Also, if they achieve higher training accuracy, it does lead to overfitting and hence poor results on testing data. In our opinion, there is a need for an approach to mine label correlation directly from the feature space and being less dependent on the label data at the same time as adopted in Cheng et al. [35].

Acknowledgements

Authors of this manuscript would like to thank Prof Shao for providing the code of MLTSVM and authors of TRAM and MLKNN for contributing their codes on public domain which in a way helped us in our implementations. We would also like to express our gratitude to the anonymous reviewers who have helped us to improve the quality of the manuscript.

Conflict of interest

There are no conflicts of interest in this study.

Ethical approval

This research paper does not involve in any studies with human participants or animals performed by any of the authors.

Informed consent

Informed consent was obtained from all participants involved in this study.

References

[1] Elisseeff A, Weston J. A Kernel Method for Multi-Labelled Classification. In: Dietterich TG, Becker S, Ghahramani Z. (eds.) *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, MA, United States: MIT Press; 2001. p.681-687. Available from: https://dl.acm.org/doi/10.5555/2980539.2980628 [Accessed 5th December 2021].

[2] Liu SM, Chen JH. A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*. 2015; 42(3): 1083-1093. Available from: doi: 10.1016/j.eswa.2014.08.036.

[3] Tang L, Rajan S, Narayanan VK. Large Scale Multi-Label Classification via Metalabeler. In: *Proceedings of the 18th international conference on World wide web*. New York, NY, United States: Association for Computing Machinery; 2009. p.211-220. Available from: doi: 10.1145/1526709.1526738.

[4] Gopal S, Yang Y. Multilabel classification with meta-level features. In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval.* New York, NY, United States: Association for Computing Machinery; 2010. p.315-322. Available from: doi: 10.1145/1835449.1835503.

[5] Blanco A, Casillas A, Pérez A, de Ilarraza AD. Multi-label clinical document classification: Impact of label-density. *Expert Systems with Applications*. 2019; 138: 112835. Available from: doi: 10.1016/j.eswa.2019.112835.

[6] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science*. 2000; 290(5500): 2323-2326. Available from: doi: 10.1126/science.290.5500.2323.

[7] Donoho DL, Grimes C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*. 2003; 100(10): 5591-5596. Available from: doi: 10.1073/pnas.1031596100.

[8] Lanckriet G, Ghaoui LE, Bhattacharyya C, Jordan MI. Minimax probability machine. In: Dietterich TG, Becker S, Ghahramani Z. (eds.) *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic.* Cambridge, MA, United States: MIT Press; 2001. p.801-807. Available from: https://dl.acm.org/doi/abs/10.5555/2980539.2980643 [Accessed 5th December 2021].

[9] Ma J, Yang L, Wen Y, Sun Q. Twin minimax probability extreme learning machine for pattern recognition. *Knowledge-Based Systems.* 2020; 187: 104806. Available from: doi: 10.1016/j.knosys.2019.06.014.

[10] Ma J, Shen J. A novel twin minimax probability machine for classification and regression. *Knowledge-Based Systems*. 2020; 196: 105703. Available from: doi: 10.1016/j.knosys.2020.105703.

[11] Gu B, Sun X, Sheng VS. Structural minimax probability machine. *IEEE Transactions on Neural Networks and Learning Systems*. 2016; 28(7): 1646-1656. Available from: doi: 10.1109/TNNLS.2016.2544779.

[12] Isii K. On sharpness of Tchebycheff-type inequalities. *Annals of the Institute of Statistical Mathematics.* 1962; 14: 185-197. Available from: doi: 10.1007/BF02868641.

[13] Bertsimas D, Sethuraman J. Moment problems and semidefinite optimization. In: Wolkowicz H, Saigal R, Vandenberghe L. (eds.) *Handbook of Semidefinite Programming. International Series in Operations Research & Management Science, vol 27*. Boston, MA: Springer; 2000. p.469-509. Available from: doi: 10.1007/978-1-4615-4381-7_16.

[14] Mahalanobis PC. On the generalized distance in statistics. *Proceedings of National Institute of Science of India.* 1936; 2(1): 49-55.

[15] Nath JS, Bhattacharyya C. Maximum margin classifiers with specified false positive and false negative error rates. In: *Proceedings of the 2007 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics; 2007. p.35-46. Available from: doi: 10.1137/1.9781611972771.4.

[16] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Dietterich TG, Becker S, Ghahramani Z. (eds.) *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, MA, United States: MIT Press; 2001. p.585-591. Available from: https://dl.acm.org/doi/abs/10.5555/2980539.2980616 [Accessed 5th December 2021].

[17] Kim KI, Steinke F, Hein M. Semi-supervised regression using Hessian energy with an application to semi-supervised dimensionality reduction. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A. (eds.) *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. Red Hook, NY, United States: Curran Associates Inc; 2009. p.979-987. Available from: https://dl.acm.org/doi/abs/10.5555/2984093.2984204 [Accessed 5th December 2021].

[18] Lanckriet GR, Ghaoui LE, Bhattacharyya C, Jordan MI. A robust minimax approach to classification. *The Journal of Machine Learning Research*. 2003; 3: 555-582. Available from: doi: 10.1162/153244303321897726.

[19] Yoshiyama K, Sakurai A. Laplacian minimax probability machine. *Pattern Recognition Letters*. 2014; 37: 192-200. Available from: doi: 10.1016/j.patrec.2013.01.004.

[20] Liu W, Tao D. Multiview hessian regularization for image annotation. *IEEE Transactions on Image Processing.* 2013; 22(7): 2676-2687. Available from: doi: 10.1109/TIP.2013.2255302.

[21] Lei Y, Cen L, Chen X, Xie Y. A hybrid regularization semi-supervised extreme learning machine method and its application. *IEEE Access*. 2019; 7: 30102-30111. Available from: doi: 10.1109/ACCESS.2019.2900267.

[22] Liu H, Liu W, Tao D, Wang Y. Laplacian-Hessian regularization for semi-supervised classification. In: *Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*. IEEE; 2014. p.203-207. Available from: doi: 10.1109/SPAC.2014.6982685.

[23] Vandenberghe L, Boyd S. *Convex optimization*. Cambridge: Cambridge University Press; 2004. Available from: doi: 10.1017/CBO9780511804441.

[24] Chen WJ, Shao YH, Li CN, Deng NY. MLTSVM: A novel twin support vector machine to multi-label learning. *Pattern Recognition*. 2016; 52: 61-74. Available from: doi: 10.1016/j.patcog.2015.10.008.

[25] MathWorks. *MATLAB*. (Version 9.4.0.813654 (R2018a)) [Software] Natick, Massachusetts: The MathWorks, Inc. 2018.

[26] Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*. 2007; 40(7): 2038-2048. Available from: doi: 10.1016/j.patcog.2006.12.019.

[27] Kong X, Ng MK, Zhou ZH. Transductive multilabel learning via label set propagation. *IEEE Transactions on Knowledge and Data Engineering.* 2011; 25(3): 704-719. Available from: doi: 10.1109/TKDE.2011.141.

[28] Wu G, Tian Y, Liu D. Cost-sensitive multi-label learning with positive and negative label pairwise correlations. *Neural Networks*. 2018; 108: 411-423. Available from: doi: 10.1016/j.neunet.2018.09.003.

[29] Yeh CK, Wu WC, Ko WJ, Wang YCF. Learning deep latent spaces for multi-label classification. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press; 2017. p.2838-2844. Available from: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14166 [Accessed 5th December 2021].

[30] Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I. *Mulan: A java library for multi-label learning*. [Dataset] http://mulan.sourceforge.net/datasets.html. 2011.

[31] Knowledge Discovery and Intelligent Systems (KDIS) research group. *Multi-Label Classification Dataset Repository*. [Dataset] Knowledge Discovery and Intelligent Systems. http://www.uco.es/kdis/mllresources/. 2017.

[32] Wu G, Tian Y, Liu D. Cost-sensitive multi-label learning with positive and negative label pairwise correlations. *Neural Networks*. 2018; 108: 411-423. Available from: doi: 10.1016/j.neunet.2018.09.003.

[33] Planet. *Planet: Understanding the Amazon from Space*. https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/ [Accessed 6th December 2021].

[34] Boutell MR, Luo J, Shen X, Brown CM. Learning multi-label scene classification. *Pattern recognition*. 2004; 37(9): 1757-1771. Available from: doi: 10.1016/j.patcog.2004.03.009.

[35] Cheng Z, Zeng Z. Joint label-specific features and label correlation for multi-label learning with missing label. *Applied Intelligence*. 2020; 50: 4029-4049. Available from: doi: 10.1007/s10489-020-01715-2.