Improved Multi-Label Propagation for Small Data with Multi-Objective Optimization

Khadija Musayeva, Mickaël Binois

IMPROVED MULTI-LABEL PROPAGATION FOR SMALL DATA WITH MULTI-OBJECTIVE OPTIMIZATION

Khadija Musayeva
Université Côte d'Azur, Inria, CNRS, LJAD, France
khadija.musayeva@inria.fr

Mickael Binois
Université Côte d'Azur, Inria, CNRS, LJAD, France
mickael.binois@inria.fr

ABSTRACT

This paper focuses on multi-label learning from small amounts of labelled data. We demonstrate that the binary-relevance extension of the interpolated label propagation algorithm, the harmonic function, is a competitive learning method with respect to many widely-used evaluation measures. This is achieved by a new transition matrix that better captures the underlying structure useful for classification coupled with the use of data dependent thresholding strategies. Furthermore, we show that in the case of label dependence, one can use the outputs of a competitive learning model as part of the input to the harmonic function to improve the performance of this model. Finally, since we are using multiple measures to thoroughly evaluate the performance of the algorithm, we propose to use the game-theory based method of Kalai and Smorodinsky to output a single compromise solution for all measures. This method can be applied to any learning model irrespective of the number of evaluation metrics used.

***K*eywords** Multi-label classification · Small data · Label propagation · Label dependence · Thresholding strategy · Multi-objective optimization

1 Introduction

Multi-label classification deals with the problems where an instance can be assigned to multiple labels simultaneously. Examples of such a problem are text and music categorization, semantic annotation of image and video, gene function prediction (see references in [39]), soil microbiome prediction [1]. Sometimes, however, only small amounts of labelled data are available because the labelling can be a costly task, and large amounts of unlabelled data can be obtained rather easily. In this context, semi-supervised learning [13] and transductive learning [37, 23] are particularly suitable learning settings as they both allow one to make use of the unlabelled part of the data to construct a predictive model. If in the former setting, the goal of the learning process is to construct a model that makes accurate predictions on out-of-sample examples, in the latter, it is to find an accurate model only for the unlabelled part of the data. The current paper focuses on the latter setting.

One of the transductive learning approaches is the family of label propagation algorithms. These are manifold learners [2,3] where the goal is, using a suitable approximation of the underlying manifold, to propagate the labels from the labelled instances to the unlabelled ones. Label propagation algorithms are decades old and started with the works [42,43,40,41,34] in the binary setting, and continued by [16,33] in the multi-label one. Clearly, any binary label propagation can be straightforwardly extended to the C-label ($C > 2$) case by binary relevance transformation [31], i.e., decomposing the data into C binary classification tasks and applying the binary method of interest to each task

independently. The main purpose of the current work is to show that the binary relevance extension of the harmonic function and its iterative version [42,43] are competitive multi-label learners for small data when they operate on a propagation matrix better aligning with the classification goal.

Roughly speaking, the harmonic function computes the labels based on an absorbing Markov chain [12], where the absorbing states represent the labelled instances with no path between them. One starts from an unlabelled instance and makes transitions unless one reaches an absorbing state. In the iterative approach, one can control the number of transitions made between the unlabelled instances. Each absorbing state contributes to the computation of the label of a given unlabelled instance a proportionally to the probability of reaching that absorbing state from a. Clearly, the performances of these algorithms depend crucially on how well the transition matrix is aligned with the classification task. In this paper, we construct a new transition matrix where the transition probability between any two points is influenced by the neighbourhood structure of both points. Such a construction is useful to weaken the links between the points belonging to different clusters. Although we deal with interpolated approaches, this construction and the absence of paths between the labelled instances can be regarded as an implicit regularization. Furthermore, in the case of label dependence, modeling which is a central issue in the multi-label learning [9,10,11], we can leverage it by computing a transition matrix based on the data combining the original input variables and the labels, where for the unlabelled points the predictions of a competitive learning model can be used. In other words, the harmonic function and the iterative label propagation can be used as stacking methods to improve the performances of competitive external models by operating on a new structure incorporating inputs from all labels. The experiments show that these methods with the proposed transition matrices are superior to [16,33,38], and in particular, to the multi-label extensions of regularized label propagations [40,34]. As stacking methods, they improve performances of inductive classifiers such as ensembles of classifier chains [24] and random k-labelsets [32], and in this context, they are superior to the instance-based logistic regression which can also be used as a stacking method [6].

In the multi-label setting, the performances of learning methods are simultaneously evaluated with respect to multiple evaluation metrics, the majority of which require thresholding the real-valued outputs to $\{0, 1\}$. Our remaining contributions concern this subject. In this paper, along with the widely used thresholding approach of [23], we propose to use the class-mass normalization method of [43] in the multi-label setting. We show that, if the former, being based on the label cardinality, improves the family of $F1$-measures, a heuristic competitive to the exact-$F1$-plug-in classifier [8], the latter computes a threshold separately for each label, and thus is suitable to improve the Hamming loss. Finally, the multiple evaluation metrics used in this paper are usually of conflicting nature, meaning that they might favor different label outputs for a given instance. To find a single compromise solution, we propose to use the multi-objective optimization approach based on the game-theoretic method of Kalai and Smorodinsky [15,4]. Unlike the existing approach in the multi-label setting [28], it can be applied to any model irrespective of the number of evaluation metrics considered.

2 Theoretical Background

We describe the problem of interest of this paper formally as follows.

Let $\mathcal{X}$ be an input space and let $\mathcal{L} = \{l_1, \ldots, l_C\}$, $C > 2$, be a finite set of labels. In this paper, we assume that $\mathcal{X} = \mathbb{R}^d$. In the multi-label classification setting, an instance $\mathbf{x} \in \mathcal{X}$ is associated with a set of labels in $P(\mathcal{L}) = 2^{\mathcal{L}}$. Let $\mathcal{Y} = \{0, 1\}^C$ be the *relevance* set. We map the set of labels $L(\mathbf{x}) \in P(\mathcal{L})$ associated to $\mathbf{x}$ to the corresponding element $\mathbf{y} = (y_1, \ldots, y_C) \in \mathcal{Y}$ where $y_i = 1$ if and only if $l_i \in L(\mathbf{x})$. Let X and $\mathbf{Y} = (Y_1, \ldots, Y_C)$ be random variables taking values in $\mathcal{X}$ and $\mathcal{Y}$, respectively. We assume that a random pair $(X, \mathbf{Y})$ is distributed according to a fixed but unknown probability distribution P on $X \times \mathcal{Y}$. We denote the marginal and conditional distributions as P_X and $P_{\mathbf{Y}|X}$. The only available information we have about P is via an n-sample $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$, where $\mathbf{y}_i = (y_{i1}, \ldots, y_{iC})$, a realization of n independent copies of $(X, \mathbf{Y})$.

In this paper we focus on the problem of learning from the partially labelled data $D = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_l, \mathbf{y}_l), \mathbf{x}_{l+1}, \ldots, \mathbf{x}_n)$ where the goal is to predict the labels of $(\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n)$ based on the information on the joint distribution $P(X, \mathbf{Y})$ as well as that provided by $(\mathbf{x}_i)_{i=1}^n$ on P_X. One of such learning algorithms is the label propagation.

Label propagation exploits the manifold structure of the input data where the manifold is represented as the weighted graph $G = (V, E)$, with $V = \{1, \ldots, n\}$ and $E \subseteq \{(i, j) \in V^2\}$. The default assumption is that if $\mathbf{x}$ and $\mathbf{x}'$ are close in the intrinsic geometry of P_X, then $P_{\mathbf{Y}|X}(\mathbf{y}|\mathbf{x})$ and $P_{\mathbf{Y}|X}(\mathbf{y}|\mathbf{x}')$ should be similar [3], or in terms of the cluster assumption, instances should share similar labels if there is a path connecting them passing through the high density region of P_X [5]. Under such an assumption, we would like the corresponding graph to be a good approximation of this structure and to find a smooth function $f : G \to \mathbb{R}^C$ whose outputs are similar for the instances connected with the edges of large weights. We can express most label propagation approaches as the following graph regularization problem:

$$\min_{\mathbf{f}} tr(\mathbf{f}^T \tilde{\mathbf{L}} \mathbf{f} + (\mathbf{f} - \tilde{\mathbf{y}}^{(n)})^T \mathbf{\Lambda} (\mathbf{f} - \tilde{\mathbf{y}}^{(n)})), \tag{1}$$

where tr is the trace operator, $\mathbf{f} = [f(1)^T \ldots f(n)^T]^T \in \mathbb{R}^{n \times C}$ with $f(i) = (f_1(i), \ldots, f_C(i))$, $\tilde{\mathbf{y}}^{(n)} = [\mathbf{y}_1^T \ldots \mathbf{y}_l^T; \mathbf{0}]^T \in \mathbb{R}^{n \times C}$ with $\mathbf{0}$ being zero matrix of the dimension $C \times (n - l)$, $\mathbf{\Lambda}$ is a diagonal matrix of regularization parameters, $\tilde{\mathbf{L}}$ is usually the normalized Laplacian $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{P}$ with $\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ or $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ [7], $\mathbf{W}$ is the $n \times n$ weight matrix and $\mathbf{D}$ the diagonal matrix with $D_{ii} = \sum_{j=1}^n W_{ij}$. One particular case, on which the current paper focuses, is the interpolated label propagation, where for all labelled points $\Lambda_l = \infty$, and for the unlabelled ones $\Lambda_u = 0$. Consequently, the values of f on the labelled points are constrained to be equal to their labels. This solution is called the harmonic function. Below we show that it corresponds to the random walk on an absorbing Markov chain [12], i.e., a Markov chain with at least one state which once entered cannot be left. The main advantage of such a representation is the unified view of the harmonic function and the iterative label propagation, and the clear view of the structure of the transition matrix for which we provide a new computation.

3 Label Propagation on Absorbing Markov Chain

Representing the labelled points by absorbing states, the label propagation on an absorbing Markov chain is a process that starts in non-absorbing states, i.e., the states corresponding to the unlabelled points and makes transitions based on the following probabilities:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{P}_{ul} & \mathbf{P}_{uu} \end{bmatrix}, \tag{2}$$

where $\mathbf{I}$ is $l \times l$ identity matrix, and $\mathbf{P}_{uu}$ and $\mathbf{P}_{ul}$ stand for the sub-matrices of $\mathbf{P}$ denoting the transition probabilities from the unlabelled to unlabelled, and from the unlabelled to labelled instances, respectively. The first row of this matrix represents the absorbing states, i.e., the labelled instances, with self-transitions of probability one and no transitions to other states. On such a chain, the labels of the unlabelled points can be computed by the harmonic function:

$$\mathbf{f} = \mathbf{P}\mathbf{f}. \tag{3}$$

Partitioning $\mathbf{f}$ into the labelled and unlabelled parts as $\mathbf{f} = [\mathbf{f}_l \ \mathbf{f}_u]^T$ and solving the system (3) for $\mathbf{f}_u$ gives

$$\mathbf{f}_u = (\mathbf{I} - \mathbf{P}_{uu})^{-1} \mathbf{P}_{ul} \mathbf{f}_l. \tag{4}$$

This is exactly the solution of the problem (1) with the constraints $\Lambda_l = \infty$ and $\Lambda_u = 0$ yielding $\min_{\mathbf{f}} \mathrm{tr}(\mathbf{f}^T \tilde{\mathbf{L}} \mathbf{f}) = \min_{\mathbf{f}} \mathrm{tr}(\mathbf{f}^T \mathbf{f} - \mathbf{f}^T \mathbf{P} \mathbf{f})$.

Now for the t-step transition, $t \geqslant 2$, based on (3), we have $\mathbf{f} = \mathbf{P}^{(t)}\mathbf{f}$, where $\mathbf{A}^{(t)}$ denotes the t-th power of $\mathbf{A}$, and thus using (2) we get $\mathbf{f}_u = \mathbf{P}_{uu}^{(t)} \mathbf{f}_u + \left(\mathbf{I} + \sum_{i=1}^{t-1} \mathbf{P}_{uu}^{(i)}\right) \mathbf{P}_{ul} \mathbf{f}_l$ whose solution is again (4) (to see this, consider the equality

$(\mathbf{I} - \mathbf{P}_{uu})^{-1} = \sum_{i=0}^{\infty} \mathbf{P}_{uu}^{(i)}$). If we consider the t-step transition as an iterative procedure $\mathbf{f}^t = \mathbf{P}\mathbf{f}^{t-1}$ where $\mathbf{f}^t$ denotes the values for the iteration t, based on (2), we get the expression similar to the one just above:

$$\mathbf{f}_u^t = \mathbf{P}_{uu}^{(t)}\mathbf{f}_u^0 + \left(\mathbf{I} + \sum_{i=1}^{t-1} \mathbf{P}_{uu}^{(i)}\right) \mathbf{P}_{ul}\mathbf{f}_l, \tag{5}$$

where we initiate $\mathbf{f}_u^0 = \mathbf{0}$. Note that the harmonic function subsumes the infinite number of transitions (as $t \to \infty$, the expression (5) converges to (4)). However, as we show in the experiments section, in some cases, only a finite number of transitions is needed to improve the classification performance.

Although the harmonic function is an interpolated approach, the manipulation of the transition matrix (2) on which it operates can be seen as an implicit regularization which makes the harmonic function a competitive learning method. In contrast, the regularized approaches are based on propagation matrices where there are transitions between the labelled points, and the regularization can be seen as penalizing the labels increasingly as they get propagated (see Appendix C).

Our goal now is to compute the second row of the transition matrix (2). We provide a new way of doing so in the following section.

3.1 Computing Transition Probabilities

As we mentioned before, the performance of the label propagation algorithms depends crucially on how well the weighted graph models the manifold structure. The standard way of building such a graph is based on the use of a weighting function or kernel, in a general form given as $K_\sigma(\mathbf{x}, \mathbf{x}') = D\left(d(\mathbf{x}, \mathbf{x}')/\sigma\right)$, where d is a metric on $\mathcal{X}$, usually the Euclidean metric, and σ is the width of the kernel, for which the most widely used choice is the Gaussian one,

$$D(u) = \exp\left(-u^2/2\right). \tag{6}$$

Applying it to the data $(\mathbf{x}_i)_{i=1}^n$ gives the weight matrix $\mathbf{W}$ with $W_{ij} = K_\sigma(\mathbf{x}_i, \mathbf{x}_j)$. The Gaussian kernel has infinite support where the parameter σ dictates how slowly the similarities fade. Then applying the k-nearest neighbour approach, we can cancel the long-range relationships which can be helpful in capturing the underlying clustering structure. This might look similar to using compact kernels, such as Epanechnikov or tricube kernel [14], but the neighbourhood structure they produce are more in the spirit of that obtained by ϵ-neighbourhoods. Now let $\mathcal{N}_k(\mathbf{x}_i)$ denote the set of points which are the k-nearest neighbours of an unlabelled point $\mathbf{x}_i$ chosen based on the row $W_{i.}$ of $\mathbf{W}$. For all $\mathbf{x}_j \notin \mathcal{N}_k(\mathbf{x}_i)$, we set $W_{ij} = 0$. Furthermore, since this paper focuses on an absorbing Markov chain, we consider the neighbourhood structure only of the unlabelled instances. Thus we set $W_{ij} = 0$ for all labelled points $\mathbf{x}_i, \mathbf{x}_j$, but allow self-loops for all points. Notice that the obtained weight matrix is not necessarily symmetric, i.e., $W_{ij} \neq W_{ji}$, yielding a directed graph.

The next step is to construct the transition probabilities which will turn our graph into an absorbing Markov chain. The standard way of doing so is to row normalize $\mathbf{W}$. Then the transition probability from $\mathbf{x}_i$ to $\mathbf{x}_j$ is $P_{ij} = W_{ij} / \sum_{j=1}^n W_{ij}$, which we call the standard transition matrix. Instead, we propose to column normalize $\mathbf{W}$ first and then row normalize the obtained matrix, i.e.,

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}' \tag{7}$$

where $\mathbf{D}$ is the diagonal matrix consisting of row summation of $\mathbf{W}'$ with $D_{ii} = \sum_{j=1}^n W'_{ij}$, $\mathbf{W}' = \mathbf{W}\mathbf{D}'^{-1}$, and $\mathbf{D}'$ is another diagonal matrix consisting of column summation of $\mathbf{W}$ with $D'_{jj} = \sum_{i=1}^n W_{ij}$. The column normalization can be understood as adjusting the transition from $\mathbf{x}_i$ to $\mathbf{x}_j$ by taking into account all incoming links to $\mathbf{x}_j$. Thus if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in different clusters but there is an edge between them, then the weight of this edge will be reduced. This weakens the label propagation between these clusters. Our experiments demonstrate that constructing the transition matrix in this way produces decisively better results (with respect to the majority of evaluation metrics) compared to those obtained by the standard transition matrix. In our experiments, we also compare this approach to the regularized ones of [40,34] whose propagation matrices are inspired from spectral clustering [19], and locally linear embedding [26], respectively.

Furthermore, in the case of label dependence, we can improve the performance of the harmonic function by incorporating the label information into our transition matrix (7). This is demonstrated in the next section.

4 Leveraging Label Dependence: Interpolated Label Propagation as a Stacking Method

If the labels are correlated, leveraging this information can improve the predictive power of the learning method [11]. One of the ways of doing so, that we follow in this paper, is to treat the labels as features [6,23].

In the binary setting of the harmonic function, the authors [43] propose to attach the predicted labels as new vertices to the corresponding unlabelled instance nodes in the graph representation, where the transition probabilities from the unlabelled instance nodes to their label nodes are fixed to some value (which is the same for all these nodes). In this paper, instead of attaching the labels directly to the graph, we propose to use the original labels, $\mathbf{y}^{(l)} = [\mathbf{y}_1^T \dots \mathbf{y}_l^T] \in \mathcal{Y}^{C\times l}$, and the predictions $\mathbf{y}^{(u)} = [\bar{\mathbf{y}}_{l+1}^T \dots \bar{\mathbf{y}}_n^T] \in \mathcal{Y}^{C\times u}$ of an external classifier, i.e., the combined label matrix $\tilde{\mathbf{Y}} = [\mathbf{y}^{(l)}; \mathbf{y}^{(u)}]^T \in \mathcal{Y}^{n\times C}$ as new features and join them to the original input data $\mathbf{X} = [\mathbf{x}_1^T \dots \mathbf{x}_n^T]^T \in \mathbb{R}^{n\times d}$. The original problem now has a new representation of the dimensionality $d + C$. Since this new representation incorporates the information from all labels, it is suitable to perform the label propagation on the resulting graph, which we construct as discussed in Section 3.1, when there exists inter-dependencies between the majority of the labels.

The important point here is that we are dealing with the real-valued data $\mathbf{X}$, and thus we need to transform the relevance matrix $\tilde{\mathbf{Y}}$ to a real-valued one before joining it to $\mathbf{X}$. The straightforward approach would be to row-normalize it, but this transformation is indifferent to the frequency of labels. Instead, inspired from the ecological study [18], we propose to do a chi-square metric based transformation of $\tilde{\mathbf{Y}}$ as $\tilde{\mathbf{Y}}' = (\sum_{i=1}^{n}\sum_{j=1}^{C}\tilde{\mathbf{Y}}_{ij})^{1/2}\mathbf{A}^{-1}\tilde{\mathbf{Y}}\mathbf{B}^{-1/2}$, where $\mathbf{A}$ is the diagonal matrix consisting of row summation of $\tilde{\mathbf{Y}}$, i.e., $A_{ii} = \sum_{j=1}^{C}\tilde{\mathbf{Y}}_{ij}$, and $\mathbf{B}$ is the diagonal matrix consisting of column summation of $\tilde{\mathbf{Y}}$, i.e., $B_{jj} = \sum_{i=1}^{n}\tilde{\mathbf{Y}}_{ij}$. The advantage of this transformation is that it is sensitive to the frequencies of the labels where the relatively rare ones get more weight since their column sums are smaller. Consequently, when building the weight matrix (6), they contribute more to the Euclidean distance than it is the case with a simple row normalization.

It should be noted that the matrix $\tilde{\mathbf{Y}}$ is noisy because it combines the predictions of an external classifier (the original labels might also be assumed to be noisy). Taking this into account, we control the contribution from $\tilde{\mathbf{Y}}$ by multiplying it by a regularization parameter $\alpha \in (0, 1)$. This simply reduces to $d((\mathbf{x}, \mathbf{x}'))^2 + \alpha \cdot d((\tilde{\mathbf{y}}, \tilde{\mathbf{y}}'))^2$ in the computation of the squared Euclidean distance in (6) for any points $(\mathbf{x}, \tilde{\mathbf{y}})$ and $(\mathbf{x}', \tilde{\mathbf{y}}')$. The transition matrix is computed using (7) to which we then apply the harmonic function or the iterative label propagation.

5 Evaluation Measures and Thresholding Strategies

Since multi-label classification constitutes a more complex setting than the single-label one, the performance of a multi-label classifier is usually evaluated simultaneously according to multiple evaluation metrics, each of which captures different aspect of this performance. This paper considers the following most widely used metrics: subset accuracy, Hamming loss, average precision, $F1$-measure, and the *macro* and *micro* variants of the latter as well as that of the area under the ROC curve (AUC). For the completeness of the paper, their definitions are given in Appendix A.

The label propagation approach that we are dealing with produces outputs in $[0, 1]$. Then, to compute the Hamming loss, the subset accuracy and the family of $F1$-measures, a decision rule/thresholding strategy is needed to map them to $\{0, 1\}$. In fact, the choice of the decision rule can drastically impact the values of these measures [35,13]. The default approach would be to treat the outputs of the harmonic function as class posterior probabilities and use the standard thresholding at 0.5, however this would lead to poor results if the outputs are not well calibrated probability estimates. We can address it by probability calibration techniques [21,36,17], but they require an additional training step. Instead,

in this paper we consider two efficient data-dependent approaches that do not require a learning/cross-validation step and is superior to the standard thresholding. Their definitions are delegated to Appendix B.

One of these approaches is the class-mass normalization (CMN) proposed in [43] which can be extended straightforwardly to the multi-label setting (9). It computes a threshold for each label (and for each instance) independently based on its frequency and thus it is suitable to optimize the Hamming loss. To the best of our knowledge, this strategy has not yet been considered in the multi-label classification setting. On the other hand, [24] uses a single threshold for all labels and is based on the notion of label cardinality (10). We refer to this rule as the label cardinality optimizer (LCO). If the cardinality of the unlabelled data is similar to that of the labelled data, this heuristic is suitable to improve the family of $F1$-measures, because by optimizing the label cardinality it improves the recall[1]. In our experiments, we also compare the $F1$-values obtained by this strategy to that obtained by the exact-$F1$-plug-in classifier (EFC) [8] designed to optimize just mentioned measure.

6 Finding Compromise Solution for Multiple Evaluation Measures

The problem that arises from using multiple evaluation metrics is that their optimal values might favor different label outputs. For instance, the conflicting nature of the Hamming loss and the subset accuracy has been demonstrated rigorously in [9]. However, it might be desirable to find a single output for multiple metrics without much compromising on any of them. This problem falls into the field of multi-objective optimization and can be handled in the context of *Pareto dominance*. Without loss of generality, consider a minimization problem. The solution $\mathbf{a} = (a_1, \ldots, a_p) \in \mathbb{R}^p$ is said to dominate the solution $\mathbf{b} = (b_1, \ldots, b_p) \in \mathbb{R}^p$ if for all $i, a_i \leq b_i$ and there exists i such that $a_i < b_i$, and the corresponding set of non-dominated solutions is called the Pareto front. The goal then is to select a single solution from this set. To the best of our knowledge, the only work that addresses this task in the context of multi-label classification is [28]. It is based on the evolutionary algorithm whose computation depends on the considered approach and might not be applicable to any model: the authors choose a neural network due to the efficiency of the computation.

In this paper, we propose to use the game theory based method of Kalai and Smorodinsky (KS) [15] which can be briefly described as follows. This method searches for the solution which, for a chosen set of evaluation metrics or objectives of equal importance, is the solution $\mathbf{s} = (s_1, \ldots, s_p) \in \mathbb{R}^p$ centrally located on the Pareto front, where s_i is the value of the i-th evaluation metric and p is the number of the metrics considered. Given the minimum of each objective $\mathbf{u} = (u_1, \ldots, u_p) \in \mathbb{R}^p$ and a *disagreement* point $\mathbf{d} = (d_1, \ldots, d_p) \in \mathbb{R}^p$, defaulted to the worse value of each objective on the Pareto front, the idea is to move from $\mathbf{d}$ towards $\mathbf{u}$ while equally improving all objectives. More precisely, the idea is to improve the corresponding benefit ratios defined as $r(\mathbf{s}, i) = \dfrac{d_i - s_i}{d_i - u_i},\ 1 \leqslant i \leqslant p$, for any solution $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^p$ and objective i. If it exists, this process leads to the KS solution, i.e., the Pareto optimal solution for which all benefit ratios are equal, and if it does not, the authors [4] propose the efficient maxmin solution: it is the Pareto optimal solution maximizing the smallest benefit ratio over objectives, i.e., $\mathbf{s}^*_{KS} \in \arg\max_{\mathbf{s}\in\mathcal{S}} \min_{1\leqslant i\leqslant p} r(\mathbf{s}, i)$.

This method is applied to the outputs of evaluation measures over a hyperparameter grid, thus it is independent of the learning method used. If [28] can optimize up to four objectives, the KS method can be applied to any number of objectives scaling only linearly. In this paper, we find the compromise solution for eight objectives/evaluation metrics mentioned in the previous section.

7 Experiments

The main goal of the experiments is to compare the overall performance on multiple evaluation measures, i.e., the KS compromise solution of the harmonic function (HF) and the iterative label propagation (ILP) using the proposed

[1] To see this let $(1, 0, 1, 1)$ be the true label set. Then, although both $(1, 0, 0, 0)$ and $(1, 1, 0, 1)$ predict two labels incorrectly, the latter provides a higher $F1$ value because it provides higher recall without much degrading the precision.

transition matrices to those of other label propagation (LP)/local methods and to evaluate its stacking performance using several competitive inductive learning methods[2].

7.1 Experimental Setup

To the best of our knowledge, in this paper we considered all existing LP methods: the straightforward multi-label extension of regularized label propagation approaches, consistency method (CM) [40] and linear neighbourhood label propagation (LNP) [34], and multi-label propagation approaches, dynamic label propagation (DLP) [33] and TRAM [16]. In TRAM, we do not row-normalize the label matrix which leads to a better performance, and also, since we work with data of small to moderate dimensionality we do not apply any dimensionality reduction approach as it is done in the original work. As the considered LP approaches model the manifold based on the k-nearest neighbour graph, we also compare their performances to ML-KNN [38].

For the stacking performance of HF/ILP, we use the predictions of the following competitive multi-label classifiers: binary relevance method (BR) [30], ensembles of classifier chains (ECC) [24] and random k-labelsets (RAKEL) [32]. For these methods the support-vector machine is chosen as the base algorithm. Since BR is tailored to improve the label-wise performance, it improves the Hamming loss. On the other hand, ECC and RAKEL are capable of modeling the label dependencies, and thus they improve the subset accuracy [9,11]. Then, the general view of their performances gives an idea about possible label dependencies. More precisely, when there is no label dependence, we can expect the Hamming loss and the subset accuracy to be optimized simultaneously by BR. In this case, using HF/ILP as a stacking method is useless. If there is a label dependence, then ECC or RAKEL is expected to improve the subset accuracy, and we use the predictions of the best performing (on most measures) method in HF/ILP as discussed in Section 4.

We compare the stacking performance of HF/ILP to that of the instance-base logistic regression (IBLR) method: first, we test the performance of IBLR as a standalone method, then apply it to the predictions of the inductive methods mentioned above. The neighbourhood structure in IBLR is based on the similarity matrix derived from the Gaussian kernel (6).

As a decision rule, for LNP, since the labels are in $\{-1, 1\}$, we use the sign function. TRAM method implements a label propagation to find the number of relevant labels for each unlabelled point: as we show in Appendix F, TRAM is equivalent to using HF with the standard transition matrix, where the same procedure is also used to find the number of relevant labels. Since IBLR is based on logistic regression, we threshold at 0.5. The remaining methods use both CMN 9 and LCO 10 thresholding approaches.

For the quadratic optimization problem of LNP we use `OSQP` package [29], and for the KS solution the `GPGame` package [20]. We use the `utiml` package [25] for BR, ECC, RAKEL and ML-KNN. For the logistic regression problem of IBLR and EFC we use `glm` package. All methods are implemented in `R` programming language [22].

We selected five publicly available data sets[3] with the moderate number n of observations and the low average class imbalance defined as $\mathrm{avgImb}(D) = \frac{1}{C}\sum_{j=1}^{C}\frac{\max(F(j),n-F(j))}{\min(F(j),n-F(j))}$ where $F(j) = \sum_{i=1}^{n} y_{ij}$ [37]. Table 1 shows the values of these properties along with the number C of labels, the number ls of unique labelsets, and the label density $\mathrm{ld} = \frac{1}{nC}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij}$.

For all LP methods and ML-KNN, we normalize the features of all data sets to the range $[0, 1]$, because they are based on the use of the Euclidean distance which is sensitive to the magnitude of the values. Since the current paper focuses on small data, we sampled only half of the observations from Scene and Yeast datasets by stratified sampling based on the label powerset approach [27]. This guarantees similar label distribution, thus similar ls, ld and avgImb to those of the entire dataset. We used this same approach in 5×2-fold cross validation.

[2] The code and the data are available at github.com/kmusayeva/M-LP.

[3] All datasets except for Fungi are taken from https://www.uco.es/kdis/mllresources/. The Fungi dataset has been kindly provided to us by C. Averill [1].

Table 1: The properties of the datasets used in the experiments.

Data set	n	d	C	ls	ld	avgImb
Emotions	593	72	6	27	0.311	2.146
Fungi	240	9	12	147	0.344	5.451
Scene	2407	294	6	15	0.179	4.662
Yeast	2417	103	14	198	0.325	8.867

Finally, in HF, ILP and TRAM, the number of neighbours takes values in the set $\{15, 20, \ldots, 50\}$. In general, LNP favors small number of neighbours: this is due to the fact that each point and its neighbourhood is represented by a locally linear patch of the underlying manifold. Thus for LNP, and also for ML-KNN, $k \in \{3, 5, \ldots, 50\}$. The α parameter in LNP and CM is in $\{0.2, 0.4, \ldots, 0.99\}$ and that in HF as a stacking method is in $\{0.01, 0.1, 0.3, 0.5\}$. In DLP method, α takes its values in $\{0.001, 0.0001\}$ and λ in $\{0.01, 0.1\}$, and the number of iterations are set from $\{1, 2, \ldots, 10\}$. In LP methods the width σ of the kernel is chosen from $\{0.1, 0.2, \ldots, 3\}$ and in SVM (used in the inductive approaches) from $\{10^{-3}, 10^{-2}, \ldots, 10\}$.

7.2 Results

We first report the superior performance of HF using our new transition matrix (7) to that using the standard one in Table 2. Tables 3-6 report the results of the experiments comparing the performances of learning methods discussed in the previous section (the standard deviations are delegated to Appendix E).

In all experiments, we observed CMN to be a better minimizer of the Hamming loss compared to LCO. The latter, on the other hand, improves the family of $F1$-measures at the cost of degrading the Hamming loss. For simplicity of the presentation, we report the performances corresponding to one of these decision rules (homogeneous for all methods, except for LNP, TRAM and IBLR which use their own decision functions): we choose CMN over LCO if it substantially improves the Hamming loss and the subset accuracy without much degrading the family of $F1$-measures. We report both CMN and LCO results only for HF in Table 8 in Appendix D; the observed tendency holds for all methods.

Table 2: Comparison of the new transition matrix with the standard one for HF. The reported values are the KS compromise solutions. The up/down arrow next to the name of an evaluation metric means the higher/lower the value the better the performance. The best results are underlined.

	Emotions		Fungi	
Measures	new	standard	new	standard
Hamming loss ↓	0.1891 ± 0.0085	0.1997 ± 0.0107	0.2047 ± 0.0095	0.2060 ± 0.0099
Subset-accuracy ↑	0.3305 ± 0.0151	0.3156 ± 0.0235	0.1534 ± 0.0199	0.1517 ± 0.0190
F1 ↑	0.6730 ± 0.0133	0.6576 ± 0.0145	0.6654 ± 0.0174	0.6631 ± 0.0178
Macro-F1 ↑	0.6792 ± 0.0146	0.6622 ± 0.0171	0.5455 ± 0.0233	0.5511 ± 0.0189
Micro-F1 ↑	0.6975 ± 0.0141	0.6801 ± 0.0167	0.6696 ± 0.0154	0.6711 ± 0.0139
Macro-AUC ↑	0.8503 ± 0.0087	0.8408 ± 0.0105	0.8046 ± 0.0128	0.8024 ± 0.0093
Micro-AUC ↑	0.8662 ± 0.0081	0.8578 ± 0.0105	0.8612 ± 0.0064	0.8599 ± 0.0061
Average precision↑	0.8081 ± 0.0056	0.8027 ± 0.0093	0.8434 ± 0.0082	0.8423 ± 0.0068

	Scene		Yeast	
Measures	new	standard	new	standard
Hamming loss ↓	0.0935 ± 0.0030	0.1031 ± 0.0049	0.2018 ± 0.0046	0.2066 ± 0.0057
Subset-accuracy ↑	0.6879 ± 0.0104	0.6623 ± 0.0156	0.2051 ± 0.0097	0.2066 ± 0.0107
F1 ↑	0.7338 ± 0.0094	0.7064 ± 0.0144	0.6515 ± 0.0063	0.6429 ± 0.0097
Macro-F1 ↑	0.7428 ± 0.0077	0.7156 ± 0.0129	0.4248 ± 0.0084	0.4389 ± 0.0116
Micro-F1 ↑	0.7297 ± 0.0087	0.7021 ± 0.0140	0.6676 ± 0.0065	0.6604 ± 0.0092
Macro-AUC ↑	0.9333 ± 0.0026	0.9292 ± 0.0033	0.7122 ± 0.0110	0.7058 ± 0.0130
Micro-AUC ↑	0.9342 ± 0.0036	0.9283 ± 0.0050	0.8439 ± 0.0037	0.8389 ± 0.0045
Average precision↑	0.8556 ± 0.0056	0.8372 ± 0.0085	0.7579 ± 0.0061	0.7529 ± 0.0068

As Tables 3-6 demonstrate, the interpolated LP approach, HF, is superior to the regularized approaches, CM and LNP, with respect to most evaluation metrics on all datasets, except for Scene, which is indicative of the fact that, on these datasets, the transition matrix (7) is capable of capturing the structure useful for classification. On Scene dataset, which has the lowest label density, i.e., a sparse label matrix, in view of Equation 12, the regularized label propagation approach, CM, achieves a superior performance if we highly penalize the fitting to the labels. This, however, comes at the cost of degraded AUC measures. Also, on this dataset, ILP performs better than HF, which means that by keeping the number of transitions between the unlabelled points small, one can respect the sparsity of the label matrix.

In particular, HF is superior over all LP methods as well as ML-KNN on Emotions and Yeast datasets. This is particularly remarkable for the former dataset: it contains a small number of unique labelsets, the lowest class imbalance among the datasets considered, and exhibits label dependence which can also be concluded from the performance of the RAKEL method. Such a dependence is also observed for Scene and Yeast datasets. On these datasets, HF effectively leverages this dependence as a stacking method using the predictions of RAKEL.

Table 3: The results for Emotions dataset. The up/down arrow next to the name of an evaluation metric means the higher/lower the value the better the performance. The best results are underlined. The decision function used is LCO.

A) HF and LP/ML-KNN

Measures	HF	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss ↓	0.1891	0.2270	0.2330	0.2251	0.2387	0.2029
Subset-accuracy ↑	0.3305	0.2090	0.2630	0.2472	0.2091	0.2906
F1 ↑	0.6730	0.6210	0.5780	0.6519	0.6072	0.6482
Macro-F1 ↑	0.6792	0.6420	0.6000	0.6660	0.6123	0.6471
Micro-F1 ↑	0.6975	0.6530	0.6080	0.6757	0.6365	0.6746
Macro-AUC ↑	0.8503	0.7870	0.7840	0.8431	0.7631	0.8229
Micro-AUC ↑	0.8662	0.8000	0.7950	0.8598	0.7825	0.8484
Average precision↑	0.8081	0.8080	0.7700	0.8034	0.7930	0.7968

B) Stacking: HF vs IBLR

Measures	BR	ECC	RAKEL	RAKEL+HF	IBLR	RAKEL+IBLR
Hamming loss↓	0.1873	0.1857	0.1841	0.1799	0.1937	0.1877
Subset accuracy↑	0.3292	0.3410	0.3568	0.3703	0.3197	0.3382
F1↑	0.6747	0.6710	0.6809	0.6858	0.6419	0.6615
Macro-F1↑	0.6822	0.6869	0.6905	0.6986	0.6515	0.6799
Micro-F1↑	0.7003	0.6995	0.7060	0.7114	0.6748	0.6989
Macro-AUC↑	0.8508	0.8385	0.8044	0.8515	0.8417	0.8463
Micro-AUC↑	0.8704	0.8510	0.8162	0.8646	0.8602	0.8651
Average precision↑	0.8244	0.8117	0.8001	0.8241	0.8107	0.8117

Table 4: The results for Fungi dataset. The up/down arrow next to the name of an evaluation metric means the higher/lower the value the better the performance. The best results are underlined. The decision function used is CMN.

A) HF and LP/ML-KNN

Measures	HF	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss ↓	0.2020	0.2060	0.2140	0.2034	0.2051	0.2255
Subset-accuracy ↑	0.1517	0.1509	0.1360	0.1537	0.1410	0.1200
F1 ↑	0.6690	0.6435	0.6440	0.6661	0.6672	0.6158
Macro-F1 ↑	0.5192	0.4859	0.5710	0.5248	0.5689	0.4878
Micro-F1 ↑	0.6708	0.6495	0.6600	0.6720	0.6809	0.6286
Macro-AUC ↑	0.8051	0.8042	0.7600	0.8034	0.8047	0.7485
Micro-AUC ↑	0.8605	0.8581	0.8190	0.8611	0.8623	0.8316
Average precision↑	0.8440	0.8412	0.8280	0.8417	0.8480	0.7929

B) Stacking: HF vs IBLR

Measures	BR	RAKEL	ECC	ECC+HF	HF+HF	IBLR	ECC+IBLR
Hamming loss ↓	0.2070	0.2095	0.2078	0.2065	0.2018	0.2265	0.2056
Subset-accuracy ↑	0.1483	0.1548	0.1552	0.1567	0.1600	0.1165	0.1533
F1 ↑	0.6453	0.638	0.6559	0.6593	0.6745	0.6238	0.6579
Macro-F1↑	0.4828	0.5112	0.5626	0.5840	0.5708	0.5357	0.5561
Micro-F1 ↑	0.6460	0.6463	0.6639	0.6749	0.6796	0.6417	0.6672
Macro-AUC ↑	0.7364	0.7150	0.7638	0.7631	0.7956	0.7717	0.8084
Micro-AUC ↑	0.8285	0.7974	0.8409	0.8373	0.8611	0.8327	0.8570
Average precision↑	0.8092	0.7916	0.8170	0.8226	0.8372	0.8011	0.8361

Table 5: The results for Scene dataset. The up/down arrow next to the name of an evaluation metric means the higher/lower the value the better the performance. The best results are underlined. The decision function used is CMN. On this dataset ILP performs better than HF.

A) ILP and LP/ML-KNN

Measures	ILP	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss ↓	0.0930	0.0920	0.0958	0.1009	0.0937	0.1030
Subset-accuracy ↑	0.6879	0.6928	0.6851	0.6691	0.6884	0.6573
F1 ↑	0.7338	0.7414	0.7301	0.7121	0.7329	0.7087
Macro-F1 ↑	0.7297	0.7351	0.7336	0.7079	0.7287	0.7047
Micro-F1 ↑	0.7428	0.7465	0.7245	0.7205	0.7413	0.7150
Macro-AUC ↑	0.9333	0.8521	0.9225	0.9301	0.9097	0.8995
Micro-AUC ↑	0.9342	0.8553	0.9264	0.9303	0.9126	0.9140
Average precision↑	0.8556	0.8620	0.8538	0.8411	0.8564	0.8320

B) Stacking: ILP vs IBLR

Measures	BR	ECC	RAKEL	RAKEL+ILP	IBLR	RAKEL+IBLR
Hamming loss ↓	0.0848	0.0862	0.0805	0.0796	0.0989	0.0923
Subset-accuracy ↑	0.6998	0.6936	0.7237	0.7304	0.6599	0.6838
F1 ↑	0.7683	0.7653	0.7750	0.7752	0.7255	0.7439
Macro-F1 ↑	0.7620	0.7588	0.7699	0.7702	0.7203	0.7378
Micro-F1 ↑	0.7720	0.7693	0.7797	0.7794	0.7302	0.7470
Macro-AUC ↑	0.9346	0.9247	0.8954	0.9192	0.9272	0.9290
Micro-AUC ↑	0.9443	0.9287	0.8938	0.9248	0.9337	0.9360
Average precision↑	0.8753	0.8624	0.8499	0.8742	0.8487	0.8583

Table 6: The results for Yeast dataset. The up/down arrow next to the name of an evaluation metric means the higher/lower the value the better the performance. The best results are underlined. The decision function used is LCO.

A) HF and LP/ML-KNN

Measures	HF	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss ↓	0.2018	0.2164	0.2030	0.2061	0.2090	0.2143
Subset-accuracy ↑	0.2051	0.1584	0.1840	0.1799	0.1709	0.1599
F1 ↑	0.6515	0.6150	0.6010	0.6278	0.6405	0.6333
Macro-F1 ↑	0.4248	0.4317	0.3930	0.3968	0.3899	0.3838
Micro-F1 ↑	0.6676	0.6426	0.6300	0.6438	0.6569	0.6480
Macro-AUC ↑	0.7122	0.6906	0.6690	0.7063	0.7088	0.6521
Micro-AUC ↑	0.8439	0.8298	0.8220	0.8400	0.8335	0.8261
Average precision↑	0.7579	0.7611	0.7500	0.7544	0.7486	0.7460

B) Stacking: HF vs IBLR

Measures	BR	ECC	RAKEL	RAKEL+HF	IBLR	RAKEL+IBLR
Hamming loss ↓	0.2019	0.2062	0.1992	0.1980	0.2038	0.2005
Subset-accuracy ↑	0.1838	0.1883	0.2243	0.2477	0.1604	0.2008
F1 ↑	0.6502	0.6470	0.6565	0.6595	0.6009	0.6312
Macro-F1 ↑	0.4180	0.4341	0.4200	0.4183	0.3517	0.3779
Micro-F1 ↑	0.6672	0.6633	0.6712	0.6736	0.6271	0.6518
Macro-AUC ↑	0.7029	0.6552	0.6194	0.6909	0.6687	0.6856
Micro-AUC ↑	0.8406	0.8152	0.7910	0.8385	0.8296	0.8366
Average precision↑	0.7663	0.7577	0.7602	0.7585	0.7527	0.7596

Fungi dataset contains the smallest number of observations, and HF, TRAM and DLP, being transductive methods, outperform all inductive methods with respect to all evaluation metrics except subset accuracy because they do not take the label dependence into account. It also has the highest ratio ls/n of unique labelsets to the number of observations, thus the inferior performance demonstrated by RAKEL. Moreover, this dataset contains independent subsets of correlated labels, and as can be judged from the value of the subset accuracy, ECC and RAKEL are capable of leveraging it. The predictions of ECC are further improved by HF. Since HF outperforms the inductive methods, for comparison we also use its own predictions in the stacking approach. Compared to the standalone HF, incorporating the label information simultaneously improves the Hamming loss, the subset accuracy and the family of $F1$ measures.

HF/ILP also outperforms IBLR on all datasets. Using the predictions of the inductive methods, IBLR does improve its performance with respect to all measures, however this does not provide an improvement over that of the inductive method.

Finally, the competitive performance of (stacking) HF using LCO to that of EFC method with respect to $F1$-measure is given in Table 7. EFC, being sensitive to class imbalance, performed poorly on Fungi dataset, and thus this dataset is excluded from the summary. The performance gain of EFC with respect to $F1$-measure comes at the cost of the deteriorated Hamming loss compared to HF. If standalone HF outperforms EFC with respect to both measures simultaneously only on Yeast dataset, for stacking HF, this holds true on all datasets.

Table 7: Comparison of EFC with the KS solution of HF and Stacking HF (SHF). SHF uses the predictions of RAKEL method. The best results are underlined.

Emotions

Measures	EFC	HF	SHF
Hamming loss ↓	0.2345 ± 0.0038	0.1886 ± 0.0082	<u>0.1846 ± 0.0065</u>
F1 ↑	0.6754 ± 0.0070	0.6740 ± 0.0120	<u>0.6784 ± 0.0095</u>

Scene

Measures	EFC	HF	SHF
Hamming loss ↓	0.1163 ± 0.0031	0.0986 ± 0.0045	<u>0.0808 ± 0.0029</u>
F1 ↑	0.7497 ± 0.0052	0.7409 ± 0.0122	<u>0.7817 ± 0.0089</u>

Yeast

Measures	EFC	HF	SHF
Hamming loss ↓	0.2334 ± 0.0037	0.2016 ± 0.0041	<u>0.1982 ± 0.0034</u>
F1 ↑	0.6457 ± 0.0065	0.6519 ± 0.0073	<u>0.6587 ± 0.0058</u>

8 Conclusions

This paper extends the harmonic function and its iterative version to the multi-label setting via the binary relevance transformation. In particular, we construct a new transition matrix which better aligns with the classification task than the standard transition matrix. Furthermore, although it is a binary relevance approach, we can leverage the label dependence by incorporating the labels into this transition matrix, where for the unlabelled points the predictions of a competitive learning method can be used. We evaluated the performances of all models considered in the paper via multiple evaluation metrics. A subset of these measures requires a thresholding strategy to be applied to the real-valued outputs. Since it computes a threshold for each label independently based on its frequency, we propose using the class-mass normalization method in the multi-label setting to improve the Hamming loss. Finally, for multiple evaluation metrics, we report a single compromise solution using the game-theory based multi-objective optimization approach. This approach can be applied to any number of metrics, scaling only linearly. The obtained results show that, despite its simplicity, the label propagation on an absorbing Markov chain with the proposed transition matrices is a competitive approach capable of improving the outputs of an external model when there exists label dependence.

References

1. Averill, C., Werbin, Z., Atherton, K., Bhatnagar, J., Dietze, M.: Soil microbiome predictability increases with spatial and taxonomic scale. Nature Ecology & Evolution **5**(6), 747–756 (2021)
2. Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifolds. Machine learning **56**(1), 209–239 (2004)
3. 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)
4. Binois, M., Picheny, V., Taillandier, P., Habbal, A.: The Kalai-Smorodinsky solution for many-objective Bayesian optimization. J. Mach. Learn. Res. **21**(150), 1–42 (2020)
5. Chapelle, O., Weston, J., Schölkopf, B.: Cluster kernels for semi-supervised learning. In: Advances in neural information processing systems. Citeseer (2002)

6. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning **76**, 211–225 (2009)
7. Chung, F.: Spectral graph theory, vol. 92. American Mathematical Soc. (1997)
8. Dembczyński, K., Jachnik, A., Kotlowski, W., Waegeman, W., Hüllermeier, E.: Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In: International conference on machine learning. pp. 1130–1138. PMLR (2013)
9. Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, W.: Regret analysis for performance metrics in multi-label classification: the case of Hamming and subset zero-one loss. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 280–295. Springer (2010)
10. Dembczyński, K.K., Cheng, W., Hüllermeier, E.: Bayes optimal multilabel classification via probabilistic classifier chains. In: ICML (2010)
11. Dembczyński, K.K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence in multilabel classification. In: LastCFP: ICML Workshop on learning from multi-label data. Ghent University, KERMIT, Department of Applied Mathematics, Biometrics (2010)
12. Doyle, P., Snell, J.: Random walks and electric networks, vol. 22. American Mathematical Soc. (1984)
13. Fan, R., Lin, C.: A study on threshold selection for multi-label classification. Department of Computer Science, National Taiwan University pp. 1–23 (2007)
14. Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer (2009)
15. Kalai, E., Smorodinsky, M.: Other solutions to Nash's bargaining problem. Econometrica: Journal of the Econometric Society pp. 513–518 (1975)
16. Kong, X., Ng, M., Zhou, Z.: Transductive multilabel learning via label set propagation. IEEE Transactions on Knowledge and Data Engineering **25**(3), 704–719 (2011)
17. Leathart, T., Frank, E., Holmes, G., Pfahringer, B.: Probability calibration trees. In: Asian Conference on Machine Learning. pp. 145–160 (2017)
18. Legendre, P., Gallagher, E.: Ecologically meaningful transformations for ordination of species data. Oecologia **129**(2), 271–280 (2001)
19. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems **14** (2001)
20. Picheny, V., Binois, M.: GPGame: Solving Complex Game Problems using Gaussian Processes (2022), `https://github.com/vpicheny/GPGame`, R package version 1.2.0
21. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3), 61–74 (1999)
22. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2021), `https://www.R-project.org/`
23. Read, J., Pfahringer, B., Holmes, G.: Generating synthetic multi-label data streams. In: ECML/PKKD 2009 Workshop on Learning from Multi-label Data (MLD'09). pp. 69–84. Citeseer (2009)
24. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine learning **85**(3), 333–359 (2011)
25. Rivolli, A., de Carvalho, A.: The utiml package: Multi-label classification in R. R J. **10**(2), 24 (2018)
26. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
27. Sechidis, K., Tsoumakas, G., Vlahavas, I.: On the stratification of multi-label data. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 145–158. Springer (2011)
28. Shi, C., Kong, X., Yu, P., Wang, B.: Multi-objective multi-label classification. In: Proceedings of the 2012 SIAM International Conference on Data Mining. pp. 355–366. SIAM (2012)

29. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation **12**(4), 637–672 (2020)

30. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM) **3**(3), 1–13 (2007)

31. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data mining and knowledge discovery handbook, pp. 667–685. Springer (2009)

32. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE transactions on knowledge and data engineering **23**(7), 1079–1089 (2010)

33. Wang, B., Tu, Z., Tsotsos, J.: Dynamic label propagation for semi-supervised multi-class multi-label classification. In: Proceedings of the IEEE international conference on computer vision. pp. 425–432 (2013)

34. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering **20**(1), 55–67 (2007)

35. Yang, Y.: A study of thresholding strategies for text categorization. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 137–145 (2001)

36. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 694–699 (2002)

37. Zhang, M., Li, Y., Yang, H., Liu, X.: Towards class-imbalance aware multi-label learning. IEEE Transactions on Cybernetics (2020)

38. Zhang, M., Zhou: ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition **40**(7), 2038–2048 (2007)

39. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering **26**(8), 1819–1837 (2013)

40. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in neural information processing systems. pp. 321–328 (2004)

41. Zhou, D., Schölkopf, B.: Learning from labeled and unlabeled data using random walks. In: Joint Pattern Recognition Symposium. pp. 237–244. Springer (2004)

42. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002)

43. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th International conference on Machine learning (ICML-03). pp. 912–919 (2003)

A Evaluation Measures

In what follows, we denote $h = (h_1, \ldots, h_C) : \mathcal{X} \to \mathcal{Y}$, and $h^s = (h_1^s, \ldots, h_C^s) : \mathcal{X} \to \mathbb{R}^C$.

The subset accuracy is the generalization of the traditional indicator loss function to the multi-label classification setting, and as such is the strictest evaluation measure since it penalizes the output of h if it does not exactly match the true labels:

$$l_{SA}(\mathbf{Y}, h(X)) = \mathbb{1}_{\{h(X) \neq \mathbf{Y}\}}.$$

Unlike the subset accuracy, the Hamming loss penalizes the misclassifications for each label independently:

$$l_{HL}(\mathbf{Y}, h(X)) = \frac{1}{C} \sum_{j=1}^{C} \mathbb{1}_{\{h_j(X) \neq Y_j\}}.$$

The AUC measure evaluates the capacity of multi-label classifier for each instance to score higher the relevant labels than the irrelevant ones:

$$l_{AUC}(\mathbf{Y}, h^s(X)) = \frac{S}{l^r l^{irr}},$$

where S is the number of pairs (r, irr) of all relevant and irrelevant labels for which $h^s_r(X) \geqslant h^s_{irr}(X)$, and l^r and l^{irr} are the number of relevant and irrelevant labels for X.

Primarily used in information retrieval problems, the $F1$-measure is the harmonic mean of precision, $\frac{\sum_{j=1}^C h_j(X)Y_j}{\sum_{j=1}^C h_j(X)}$, and recall, $\frac{\sum_{j=1}^C h_j(X)Y_j}{\sum_{j=1}^C Y_j(X)}$:

$$l_{F1}(\mathbf{Y}, h(X)) = \frac{2\sum_{j=1}^C h_j(X)Y_j}{\sum_{j=1}^C Y_j + \sum_{j=1}^C h_j(X)}.$$

Due to the multi-dimensionality of outputs, on an n-sample, the $F1$-measure and AUC can be averaged differently with respect to labels and instances. These are called *macro* and *micro* averaging.

Macro-$F1$ is primarily used in class-imbalance setting to evaluate the performance of classifier on rare labels:

$$macro\text{-}F1((\mathbf{Y}_i)_{i=1}^n, h(X_i)_{i=1}^n) = \frac{1}{C}\sum_{j=1}^C \frac{2\sum_{i=1}^n h_j(X_i)Y_{ij}}{\sum_{i=1}^n Y_{ij} + \sum_{i=1}^n h_j(X_i)},$$

where Y_{ij} denotes the j-th element of the vector $\mathbf{Y}_i$. Micro-$F1$ measure, on the other hand, is not sensitive to rare labels and is defined as

$$micro\text{-}F1((\mathbf{Y}_i)_{i=1}^n, h(X_i)_{i=1}^n) = \frac{2\sum_{i=1}^n \sum_{j=1}^C h_j(X_i)Y_{ij}}{\sum_{i=1}^n \sum_{j=1}^C Y_{ij} + \sum_{i=1}^n \sum_{j=1}^C h_j(X_i)}.$$

Macro- and micro-AUC measures are defined as follows:

$$macro\text{-}AUC((\mathbf{Y}_i)_{i=1}^n, h^s(X_i)_{i=1}^n) = \frac{1}{C}\sum_{j=1}^C \frac{S^j}{n_j^r n_j^{irr}},$$

where n_j^r and n_j^{irr} denote the number of relevant and irrelevant instances for the given label j, and S^j is the number of all pairs (X_r, X_{irr}) of relevant and irrelevant instances for which $h^s_j(X_r) \geqslant h^s_j(X_{irr})$, and

$$micro\text{-}AUC((\mathbf{Y}_i)_{i=1}^n, h^s(X_i)_{i=1}^n) = \frac{S}{n^r n^{irr}},$$

where n^r and n^{irr} denote the total number of relevant and irrelevant labels for all instances, and S is the number of all quadruples $(X_r(i), X_{irr}(j), i, j)$ for which $h^s_i(X_r(i)) \geqslant h^s_j(X_{irr}(j))$.

The average precision is the average fraction of labels ranked above a given label k in the set R of relevant labels:

$$l_{ap}((\mathbf{Y}_i)_{i=1}^n, h^s(X_i)_{i=1}^n) = \frac{1}{n}\sum_{i=1}^n \frac{1}{|R|}\sum_{k\in R} \frac{|\{k' \in R : h^s_{k'}(X_i) \leqslant h^s_k(X_i)\}|}{h^s_k(X_i)},$$

where $|S|$ stands the cardinality of the set S.

B Decision Function/Thresholding Strategy

Assume h^s is defined as in the previous section. For any unlabelled point $\mathbf{x}_i$ and any $j \in \{1, 2, \ldots, C\}$, $z^i_j = h^s_j(\mathbf{x}_i)$ can be mapped to an element in $\mathcal{Y}$ using a decision function

$$dr_{t_j}(z^i_j) = \begin{cases} 1 & \text{if } z^i_j \geqslant t_j \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $t_j \geqslant 0$ is a threshold. In class-mass normalization method, t_j is computed as

$$t_j = \frac{1-p_j}{p_j} \cdot \frac{\sum_{p=l+1}^{n} z_j^p}{\sum_{p=l+1}^{n}(1-z_j^p)} \cdot (1-z_j^i), \tag{9}$$

where $p_j = \sum_{i=1}^{l} y_{ij}/l$ is the fraction of class 1 for the label j. In [25], a single threshold is used for all labels which is found by minimizing the following difference:

$$t^* = \underset{t\in(0,1)}{\operatorname{argmin}} \left| \operatorname{lc}(D_l) - \frac{1}{n-l} \sum_{i=l+1}^{n} \sum_{j=1}^{C} \mathbb{1}_{\{z_j^i \geqslant t\}} \right|, \tag{10}$$

where the quantity

$$\operatorname{lc}(D_l) = \frac{1}{l} \sum_{i=1}^{l} \sum_{j=1}^{C} y_{ij} \tag{11}$$

is the label cardinality. Then, in (8), $t_j = t^*$ for all j.

C Regularized Label Propagation

The regularized approach is based on the propagation matrix of the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{ll} & \mathbf{P}_{lu} \\ \mathbf{P}_{ul} & \mathbf{P}_{uu} \end{bmatrix}.$$

Let us now suppose that Λ in 1 is the same value λ for both the labelled and unlabelled points. Then, taking the derivative of 1 with respect to $\mathbf{f}$ we obtain $\mathbf{f} - \mathbf{P}\mathbf{f} + \lambda(\mathbf{f} - \mathbf{y}^{(n)}) = 0$, the solution of which is equivalent to $\mathbf{f} = \left(\mathbf{I} - \frac{1}{1+\lambda}\mathbf{P}\right)^{-1} \mathbf{y}^{(n)}$. Now, let $\mathbf{A} = \lambda'\mathbf{P}$ with $\lambda' = \frac{1}{1+\lambda}$. Since $\mathbf{f}$ can be expressed as $\mathbf{f} = \sum_{i=0}^{\infty} \mathbf{A}^{(i)}\mathbf{y}^{(n)}$, taking into account that the labels of the unlabelled points are initialized to zero, it follows that

$$\mathbf{f}_u = \mathbf{P}_{ul}(\lambda' \cdot \mathbf{y}_l^{(n)}) + (\mathbf{P}_{ul}\mathbf{P}_{ll} + \mathbf{P}_{uu}\mathbf{P}_{ul})((\lambda')^2 \cdot \mathbf{y}_l^{(n)}) + \dots. \tag{12}$$

Since $\lambda' \in (0, 1)$, the more the labels travel the less their contributions become.

D Comparison of CMN and LCO Thresholding Approaches for HF

Table 8: The KS solution for HF. The up/down arrow next to the name of an evaluation metric means the higher/lower the value the better the performance. Only the measures that requires a thresholding strategy are reported.

	Emotions		Fungi	
Measures	CMN	LCO	CMN	LCO
Hamming loss ↓	0.1856 ± 0.0049	0.1891 ± 0.0085	0.2049 ± 0.0078	0.2062 ± 0.0062
Subset accuracy ↑	0.3167 ± 0.0075	0.3305 ± 0.0151	0.1506 ± 0.0190	0.1423 ± 0.0163
F1 ↑	0.6475 ± 0.0100	0.6730 ± 0.0133	0.6662 ± 0.0149	0.6772 ± 0.0129
Macro-F1 ↑	0.6511 ± 0.0111	0.6792 ± 0.0146	0.5521 ± 0.0193	0.5789 ± 0.0239
Micro-F1 ↑	0.6792 ± 0.0099	0.6975 ± 0.0141	0.6727 ± 0.0117	0.6880 ± 0.0090
	Scene		Yeast	
Measures	CMN	LCO	CMN	LCO
Hamming loss ↓	0.0926 ± 0.0042	0.0986 ± 0.0045	0.1965 ± 0.0052	0.2024 ± 0.0043
Subset accuracy ↑	0.6892 ± 0.0132	0.6335 ± 0.0146	0.2056 ± 0.0094	0.2046 ± 0.0123
F1 ↑	0.7369 ± 0.0122	0.7409 ± 0.0122	0.6118 ± 0.0098	0.6505 ± 0.0067
Macro-F1 ↑	0.7448 ± 0.0114	0.7482 ± 0.0099	0.3967 ± 0.0138	0.4315 ± 0.0102
Micro-F1 ↑	0.7325 ± 0.0119	0.7328 ± 0.0114	0.6404 ± 0.0095	0.6672 ± 0.0066

E Standard Deviations for Tables 3-6.

Table 9: Standard deviations for Emotions dataset.

A) HF and LP/ML-KNN

Measures	HF	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss	0.0085	0.0090	0.0076	0.0109	0.0060	0.0088
Subset-accuracy	0.0151	0.0250	0.0164	0.0176	0.0174	0.0258
F1	0.0133	0.0160	0.0141	0.0151	0.0082	0.0124
Macro-F1	0.0146	0.0120	0.0148	0.0167	0.0109	0.0151
Micro-F1	0.0141	0.0120	0.0142	0.0163	0.0089	0.0131
Macro-AUC	0.0087	0.0090	0.0088	0.0103	0.0119	0.0110
Micro-AUC	0.0081	0.0100	0.0084	0.0106	0.0113	0.0108
Average precision	0.0056	0.0090	0.0097	0.0097	0.0042	0.0077

B) Stacking: HF vs IBLR

Measures	BR	ECC	RAKEL	RAKEL+HF	IBLR	RAKEL+IBLR
Hamming loss	0.0077	0.0068	0.0069	0.0094	0.0082	0.0059
Subset accuracy	0.0219	0.0196	0.0138	0.0159	0.0165	0.0193
F1	0.0144	0.0135	0.0130	0.0144	0.0134	0.0098
Macro-F1	0.0147	0.0141	0.0119	0.0158	0.0155	0.0100
Micro-F1	0.0123	0.0126	0.0118	0.0155	0.0147	0.0098
Macro-AUC	0.0085	0.0111	0.0078	0.0083	0.0107	0.0093
Micro-AUC	0.0082	0.0097	0.0071	0.0091	0.0097	0.0082
Average precision	0.0133	0.0124	0.0075	0.0087	0.0099	0.0081

Table 10: Standard deviations for Fungi dataset.

A) HF and LP/ML-KNN

Measures	HF	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss	0.0090	0.0082	0.0081	0.0087	0.0090	0.0122
Subset-accuracy	0.0203	0.0234	0.0153	0.0132	0.0213	0.0257
F1	0.0159	0.0150	0.0148	0.0105	0.0154	0.0255
Macro-F1	0.0141	0.0285	0.0226	0.0107	0.0244	0.0340
Micro-F1	0.0133	0.0170	0.0143	0.0107	0.0126	0.0181
Macro-AUC	0.0098	0.0162	0.0134	0.0118	0.0162	0.0098
Micro-AUC	0.0056	0.0096	0.0104	0.0063	0.0072	0.0078
Average precision	0.0101	0.0104	0.0090	0.0079	0.0084	0.0197

B) Stacking: HF vs IBLR

Measures	BR	ECC	RAKEL	ECC+HF	HF+HF	IBLR	ECC+IBLR
Hamming loss	0.0095	0.0102	0.0081	0.0101	0.0107	0.0077	0.0115
Subset-accuracy	0.0220	0.0264	0.0251	0.0263	0.0178	0.0166	0.0192
F1	0.0187	0.0162	0.0202	0.0194	0.0149	0.0149	0.0182
Macro-F1	0.0408	0.0240	0.0191	0.0220	0.0141	0.0165	0.0244
Micro-F1	0.0174	0.0177	0.0162	0.0164	0.0133	0.0156	0.0172
Macro-AUC	0.0179	0.0142	0.0130	0.0163	0.0140	0.0106	0.0102
Micro-AUC	0.0089	0.0092	0.0131	0.0132	0.0067	0.0060	0.0077
Average precision	0.0124	0.0167	0.0206	0.0101	0.0116	0.0178	0.0095

Table 11: Standard deviations for Scene dataset.

A) ILP and LP/ML-KNN

Measures	ILP	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss	0.0030	0.0022	0.0042	0.0054	0.0032	0.0030
Subset-accuracy	0.0104	0.0074	0.0136	0.0159	0.0109	0.0146
F1	0.0094	0.0070	0.0117	0.0160	0.0100	0.0093
Macro-F1	0.0087	0.0065	0.0115	0.0155	0.0093	0.0083
Micro-F1	0.0077	0.0061	0.0101	0.0147	0.0084	0.0079
Macro-AUC	0.0026	0.0041	0.0066	0.0036	0.0037	0.0058
Micro-AUC	0.0036	0.0055	0.0066	0.0050	0.0044	0.0051
Average precision	0.0056	0.0050	0.0084	0.0096	0.0046	0.0066

B) Stacking: ILP vs IBLR

Measures	BR	ECC	RAKEL	RAKEL+ILP	IBLR	RAKEL+IBLR
Hamming loss	0.0051	0.0041	0.0029	0.0041	0.0038	0.0037
Subset-accuracy	0.0175	0.0163	0.0084	0.0110	0.0152	0.0112
F1	0.0142	0.0124	0.0082	0,0126	0.0119	0.0103
Macro-F1	0.0139	0.0118	0.0080	0.0121	0.0104	0.0105
Micro-F1	0.0139	0.0119	0.0074	0.0120	0.0096	0.0101
Macro-AUC	0.0044	0.0037	0.0054	0.0067	0.0033	0.0042
Micro-AUC	0.0044	0.0036	0.0056	0.0057	0.0035	0.0039
Average precision	0.0066	0.0064	0.0069	0.0079	0.0069	0.0069

Table 12: Standard deviations for Yeast dataset.

A) HF and LP/ML-KNN

Measures	HF	CM	LNP	TRAM	DLP	ML-KNN
Hamming loss	0.0046	0.0037	0.0060	0.0047	0.0031	0.0046
Subset-accuracy	0.0097	0.0065	0.0054	0.0096	0.0127	0.0181
F1	0.0063	0.0108	0.0094	0.0082	0.0048	0.0080
Macro-F1	0.0084	0.0125	0.0097	0.0114	0.0067	0.0126
Micro-F1	0.0065	0.0084	0.0093	0.0081	0.0042	0.0074
Macro-AUC	0.0110	0.0091	0.0057	0.0120	0.0087	0.0054
Micro-AUC	0.0037	0.0049	0.0060	0.0043	0.0030	0.0029
iAverage precision	0.0061	0.0055	0.0109	0.0070	0.0038	0.0061

B) Stacking: HF vs IBLR

Measures	BR	ECC	RAKEL	RAKEL+HF	IBLR	RAKEL+IBLR
Hamming loss	0.0031	0.0047	0.0038	0.0041	0.0037	0.0031
Subset-accuracy	0.0096	0.0111	0.0146	0.0084	0.0146	0.0085
F1	0.0059	0.0091	0.0052	0.0070	0.0101	0.0083
Macro-F1	0.0084	0.0100	0.0094	0.0068	0.0145	0.0117
Micro-F1	0.0055	0.0080	0.0057	0.0065	0.0084	0.0073
Macro-AUC	0.0080	0.0073	0.0054	0.0095	0.0065	0.0092
Micro-AUC	0.0031	0.0053	0.0050	0.0030	0.0028	0.0033
Average precision	0.0055	0.0063	0.0080	0.0046	0.0078	0.0067

F TRAM Method [16] is Harmonic Function

The optimization problem of TRAM method [17] is as follows:

$$\min_{\mathbf{f}} \sum_{i=l+1}^{n} \sum_{k=1}^{C} \left(f_k(i) - \sum_{j=1}^{n} P_{ij} f_k(j) \right)^2$$

where P_{ij} corresponds to the (i, j)-th entry in the transition matrix $\mathbf{P}$ 2, with the constraint that $f_k(i) = y_{ik}$. The solution follows by taking the derivative with respect to $\mathbf{f}$: $f_k(i) = \sum_{j=1}^{n} P_{ij} f_k(j)$ which in the matrix form gives the harmonic function (3). The fact that $f_k(i) \in [0, 1]$, $i \in \{l + 1, \ldots, n\}$ follows from the discrete maximum principle.