# Revisiting Multi-Label Propagation: the Case of Small Data

Khadija Musayeva, Mickaël Binois

# REVISITING MULTI-LABEL PROPAGATION: THE CASE OF SMALL DATA

**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 number of labelled data. We demonstrate that the straightforward 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 mainly by a new transition matrix that better captures the underlying manifold structure. Furthermore, we show that when there exists label dependence, we can use the outputs of a competitive learning method as part of the input to the harmonic function to provide improved results over those of the original model. Finally, since we are using multiple metrics 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. This method can be applied to any learning model irrespective of the number of evaluation measures used.

***K*eywords** Multi-label learning · Small data · Label propagation · Label dependence · Multi-objective optimization

## 1 Introduction

Multi-label classification deals with the problems where an instance can be assigned to multiple labels at the same time. The famous examples of such a problem are text and music categorization, semantic annotation of image and video, gene function prediction (see references in [1]). Sometimes, however, only small amount of labelled data are available because labelling can be a costly task, and large amounts of unlabelled data can be obtained rather easily. In this context, semi-supervised learning [2] and transductive learning [3, 4] are particularly suitable learning settings since in both of them one can make use of the unlabelled data to build a predictive model. However, if the former performs learning with the purpose to generalize to out-of-sample examples, the latter aims at finding a suitable classifier for the same unlabelled data it used to learn one. In this paper, we are interested in the latter case. One of such approaches is the family of label propagation algorithms. These are manifold learners [5, 6] where the goal is, using a suitable approximation of the underlying manifold, to propagate the labels from labelled instances to unlabelled ones. Consequently, the successful learning is based on how well the underlying structure is modelled. This is usually done via a weighted graph where the weights represent affinity/similarity of points. The corresponding matrix representation of the graph is called propagation or transition matrix. Label propagation algorithms are decades old and started with the works [7, 8, 9, 10, 11] in the binary case and [12, 13] in the multi-label case. Clearly any binary label propagation can be extended to the $C$-label case in the straightforward fashion by decomposing the data into $C$ binary classification tasks, i.e., performing binary relevance transformation [14], and applying the binary method of interest to each task independently. The purpose of the current work is to bring forward the harmonic function and the iterative label propagation [7, 8] by showing that their binary relevance extensions are competitive multi-label learners. In the setting of interest, due to the multi-dimensionality of the output space it is of central interest to consider the dependency of labels [15, 16, 17], and binary relevance methods are usually criticized for the incapacity of doing so. Although we implement a binary relevance method, we show that this dependence can be leveraged by augmenting the input space by incorporating to it the labels.

In more detail, the harmonic function corresponds to the random walk on the absorbing Markov chain [18], where the absorbing states are the labelled points, and the iterative label propagation can be viewed as doing this walk for a certain number of steps. Unlike the regularized approaches, the labelled instances are not re-labelled. For both methods, we

construct a new transition matrix where the transition probability between any two points, except for the absorbing states, are influenced by the neighbourhood structure of both points. We consider that this construction serves to better capture the clustering structure, particularly by weakening the links between points in different clusters. On the other hand not having any label propagation between the absorbing states serves as an implicit regularization. These two facts are at the basis of the competitive performance of the considered method. This transition matrix can be further improved using the augmented input space of original features and the labels, where for the unlabelled points the predictions of a competitive inductive learner can be used. This is particularly suitable in the case of label dependence, because, although the labels are propagated independently, it is realized on a new structure incorporating inputs from all labels. The experiments show that the interpolated label propagation with the proposed transition matrices is superior to [12, 13, 19], and in particular, to the multi-label extensions of explicitly regularized label propagations [9, 11] and that it is a very effective stacking method improving the performance of an inductive classifier such as ensembles of classifier chains [20] and random k-labelsets [21].

In this paper we evaluate the performances of all models with respect to multiple evaluation metrics, usually of the conflicting nature. To compute a subset of them we need to apply a thresholding function to the real-valued outputs. To this end, we propose using the class-mass normalization method of [8] in the multi-label setting along with the widely used approach of [22]. If the latter, being based on the frequency of relevant labels, improves the family of $F1$-measures, the former is suitable to improve the Hamming loss because it computes the thresholds for each label independently based on their corresponding frequencies. Finally, in most cases it is of interest to find a single solution for multiple metrics. To find the single compromise solution, we propose to use a multi-objective optimization approach based on the game-theoretic method of Kalai and Smorodinsky [23, 24]. Unlike the existing approach in the multi-label setting [25], it can be applied to any model irrespective of the number of metrics considered.

## 2 Theoretical Background

Let $\mathcal{X}$ be an input space and let $\mathcal{L} = \{l_1, \dots, l_C\}$ 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 $\mathbf{x} = (x_1, \dots, x_d) \in \mathcal{X}$ is associated with a subset 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 $\mathbf{y} = (y_1, \dots, y_C) \in \mathcal{Y}$ where $y_i = 1$ if and only if $l_i \in L(\mathbf{x})$. Let $X$ and $\mathbf{Y} = (Y_1, \dots, Y_C)$ 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 $((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n))$ where $\mathbf{y}_i = (y_{i1}, \dots, y_{iC})$, the realization of $n$ independent copies of $(X, \mathbf{Y})$. We will use $Y$ when referring to the binary setting.

In this paper we focus on the problem of learning from partially labelled data $D = ((x_1, \mathbf{y}_1), \dots, (x_l, \mathbf{y}_l), x_{l+1}, \dots, x_n)$ where the goal is to predict the labels of $(x_{l+1}, \dots, x_n)$ based on the information on the joint distribution $P(X, \mathbf{Y})$ as well as that provided by $(x_i)_{i=1}^n$ 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 $V = \{1, \dots, n\}$ 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 [6], 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 $P_X$ [26]. Under such an assumption, we would like the corresponding graph to be a good approximation of this structure and 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, $\mathbf{f} = [f(1)^T \dots f(n)^T]^T \in \mathbb{R}^{n \times C}$ with $f(1) = (f_1(1), \dots, f_C(1))$, $\tilde{\mathbf{y}}^{(n)} = [\mathbf{y}_1^T \dots \mathbf{y}_l^T; \mathbf{0}]^T \in \mathbb{R}^{n \times C}$ with $\mathbf{0}$ being zero matrix of 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 $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ [27], $\mathbf{W}$ 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 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 on which the current paper focuses. Below we show that it corresponds to the random walk on an absorbing Markov chain [18], i.e. the chain with at least one state which once entered cannot be left. The main advantage of such a representation is the clear view of the structure of the transition matrix for which we will 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, which have self transitions of probability one and no transition to other states. On such a chain, the labels of the unlabelled points can be computed by a harmonic function which has the following form

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

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

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

which is the solution of the problem (1) with the constraint that $\Lambda_l = \infty$ and $\Lambda_u = 0$, i.e., $\min_{\mathbf{f}} \operatorname{tr}(\mathbf{f}^T\tilde{\mathbf{L}}\mathbf{f}) = \min_{\mathbf{f}} \operatorname{tr}(\mathbf{f}^T\mathbf{f} - \mathbf{f}^T\mathbf{P}\mathbf{f})$.

Now for the $t$-step transition, based on Equation (3), we have $\mathbf{f} = \mathbf{P}\mathbf{f} = \mathbf{P}^{(t)}\mathbf{f}$ and thus

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

whose solution is (4). Consider now the $t$-step transition as an iterative procedure $\mathbf{f}^t = \mathbf{P}\mathbf{f}^{t-1}$, where $\mathbf{f}_u^t$ denotes the values for the iteration $t$. Based on (2), we get similar to (5) expression:

$$\mathbf{f}_u^t = \mathbf{P}_{uu}^{(t)}\mathbf{f}_u^0 + \left(\mathbf{I} + \sum_{i=1}^{t-1}\mathbf{P}_{uu}^{(i)}\right)\mathbf{P}_{ul}\mathbf{f}_l, \tag{6}$$

where we now assume initial values $\mathbf{f}_u^0 = \mathbf{0}$ for unlabelled points (not in terms of labelling them as to make them to belong to the class zero, but to make the first term vanish). By manipulating the number of transitions it is possible to improve a subset of measures compared to that obtained from the harmonic function, having in mind that as $t \to \infty$, (6) converges to (4). It should also be noted that letting $t = 1$ reduces the equality (6) to Nadaraya–Watson kernel regression [28], $\mathbf{f}_u = \mathbf{P}_{ul}\mathbf{f}_l$. Furthermore, it can be shown that the label propagation of [12] is in fact the harmonic function (Appendix A). Note that under the assumption that the $k$-nearest neighbour graph is connected, according to the discrete maximum principle [18], i.e., that the harmonic function attains its maximum and minimum at the boundary which, in our setting, are the set of labelled points, it follows that $f_k(i) \in [0, 1]$, $i \in \{l + 1, \ldots, n\}$.

Now, let us compare (4) with the regularized approach which is based on the propagation matrix of the form

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

and let us suppose that $\Lambda$ in (1) is the same value $\lambda$ both for the labelled and unlabelled points. Then taking the derivative with respect to $\mathbf{f}$ we obtain $\mathbf{f} - \mathbf{P}\mathbf{f} + \lambda(\mathbf{f} - \mathbf{y}^{(n)}) = 0$ whose solution 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 $\mathbf{f} = \sum_{i=0}^{\infty}\mathbf{A}^{(i)}\mathbf{y}^{(n)}$, it follows that

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

Thus, in the regularized approach labels do travel between labelled instances and they are penalized by the regularization parameter (the longer the propagation the weaker the label information that is propagated gets). Although there is no explicit regularization in the harmonic function, the absence of label propagation between the labelled points can be thought of as an implicit regularization. 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

The performance of the label propagation 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 $K_\sigma(\mathbf{x}, \mathbf{x}') = D\left(d(\mathbf{x}, \mathbf{x}')/\lambda\right)$, where $d$ is a metric on $\mathcal{X}$, usually the Euclidean metric, and $\sigma$ is the width of the kernel, for which the most widely used choice is the Gaussian one,

$$D(u) = \exp\left(-u^2/2\right). \tag{7}$$

Applying 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 $\sigma$ dictates how slowly the similarities fade. 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 [29], but the neighbourhood structure they produce are more in the spirit of that obtained by $\epsilon$-neighbourhoods (i.e., instances are neighbours if their similarity is within a fixed threshold). 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 values $W_{i.}$. For all $\mathbf{x}_j \notin \mathcal{N}_k(\mathbf{x}_i)$, we set $W_{ij} = 0$. Furthermore, since in this paper our focus is on the 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 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. Here we propose to column normalize $\mathbf{W}$ and then row normalize the obtained matrix, i.e.,

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}' \tag{8}$$

where $\mathbf{D}$ is the diagonal matrix consisting of row summation of $\mathbf{W}'$ with $D_{ii} = \sum_j^n W'_{ij}$, $\mathbf{W}' = \mathbf{W}\mathbf{D}'^{-1}$, and $\mathbf{D}'$ is another diagonal matrix consisting of column summation of $\mathbf{W}$ with $D'_{jj} = \sum_i^{n-l} W_{ij}$. 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. Consequently, it weakens the label propagation between these clusters. 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 [9, 11] with the propagation matrices inspired from spectral clustering [30], and locally linear embedding [31], respectively.

In the following section we consider the case of label dependence and propose an effective way to incorporate the label information into the transition matrix (8).

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

When one deals with small labelled data, being a transductive method, the advantage of a label propagation approach over an inductive learner is in its access to the unlabelled part of the data. This opens up a possibility of incorporating the labels into the learning process where for the unlabelled data one can use the predictions of a competitive learning method. In the binary setting for the harmonic function, the authors [8] propose to attach the labels and the predicted labels as new vertices to their corresponding instances in the graph representation. The transition probability from an instance node to its label node is fixed to some value $\eta$ and the remaining transitions are adjusted by multiplying them by $1 - \eta$. Then the label propagation is carried out based on this modified graph. 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 $\mathbf{X}$. The original problem now has a new representation of dimensionality $d + C$. Since this new representation incorporates information from each label, it is suitable to perform label propagation on the resulting graph, which we construct as discussed in Section 3.1, when there exists dependence between all or the majority of the labels.

The important point here is that this work considers real-valued features, and this requires that we transform $\tilde{\mathbf{Y}}$ to a real-valued matrix before treating it as a feature matrix. 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 [32] we propose to modify $\tilde{\mathbf{Y}}$ as $\tilde{\mathbf{Y}}' = \sqrt{\sum_{ij} \tilde{\mathbf{Y}}_{ij}}\mathbf{A}^{-1}\tilde{\mathbf{Y}}\mathbf{B}^{-1/2}$, where $\mathbf{A}$ is the diagonal matrix consisting of row summation of $\tilde{\mathbf{Y}}$,

$A_{ii} = \sum_{j=1}^{n} \tilde{\mathbf{y}}_{ij}$, and $\mathbf{B}$ is the diagonal matrix consisting of column summation, $B_{jj} = \sum_{i=1}^{n} \tilde{\mathbf{y}}_{ij}$. This transformation gives weight to relatively rare labels since their column sums are smaller and consequently, when computing the Euclidean distance in (7) to build the weight matrix they will contribute more to the distance than it is the case with a simple row normalization.

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). In view of this, 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 (7) for any points $(\mathbf{x}, \tilde{\mathbf{y}})$ and $(\mathbf{x}', \tilde{\mathbf{y}}')$. The transition matrix is computed using (8) to which we then apply the harmonic function.

## 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 B.

The label propagation approach that we are dealing with (and in general, most multi-label classification methods) has real-valued outputs. Then to compute the Hamming loss, the subset accuracy and the family of $F1$-measures, a decision rule (12) (in Appendix B) is needed to map them to $\{0, 1\}^C$. In fact, the choice of the threshold in the decision rule can drastically impact the values of these measures. The default approach would be to treat the outputs of the harmonic function $f$ as class posterior probabilities (because they lie in $[0, 1]$) and use the Bayes decision rule, i.e., use the threshold $0.5$ in (12). However this would lead to results of poor quality. For one thing, notice that in our approach we allow self-transitions which have the highest probability values (because of self-similarity), and if the second highest probability is less than $0.5$ and if it is assigned to the absorbing state with the label 1, then thresholding at $0.5$ will yield the label 0 for the unlabelled instance and this is not desirable. This problem can be adressed by calibrating these outputs using probability calibration techniques [33, 34, 35] but they require an additional training step.

In [36] the authors study the way the threshold selection strategy of [37] affects the macro- and micro-$F1$ measures: this strategy finds the optimal threshold value based on the cross-validation procedure. But it is not clear how it would affect the Hamming loss and the subset accuracy. Instead, in this paper we consider two efficient approaches that do not require a learning/cross-validation step and is superior to blindly thresholding at $0.5$. One of them is the class-mass normalization (CMN) proposed in [8]. It uses the class frequency information of the labelled data, and for each $j \in \{1, \ldots, C\}$ (12) takes the form

$$dr_{f_j}(\mathbf{x}_u) = \begin{cases} 1 & \text{if } f_j(u) > \alpha \cdot (1 - f_j(u)) \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where

$$\alpha = \frac{1 - p_j}{p_j} \cdot \frac{\sum_{i=u}^{n} f_j(i)}{\sum_{i=u}^{n} (1 - f_j(i))}$$

and $p_j = \sum_{i=1}^{l} y_{ij}/l$ is the fraction of class 1 for the label $j$. Since we threshold for each label independently, this approach is particularly 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, [20] uses a single threshold for all labels. It is based on the notion of label cardinality computed as $\mathrm{lc}(D_l) = \frac{1}{l} \sum_{i=1}^{l} \sum_{j=1}^{C} y_{ij}$. Then the threshold that minimizes the following difference between the label cardinality of the labelled data and that of the predictions,

$$t^* = \operatorname*{argmin}_{t \in (0,1)} \left| \mathrm{lc}(D_l) - \frac{1}{n - l} \sum_{i=l+1}^{n} \sum_{j=1}^{C} \mathbb{1}_{\{f_{uj}(\mathbf{x}_i) \geq t\}} \right|, \tag{10}$$

is used in (12). 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 (at the cost of slightly decreasing the precision).[1]

[1] To see this let $(1, 0, 1, 1)$ be the true label set. Then, although both $(1, 0, 0, 0)$ and $(1, 1, 1, 0)$ predict two labels incorrectly, the latter provides a higher $F1$ value because it has a higher recall.

It should be noted that these approaches are not suitable in the settings where the relevance set is represented by $\mathcal{Y} = \{-1, 1\}^C$, which is the case, for instance, in linear neighbourhood label propagation [11].

## 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 [15]. 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 [25]. This work is based on the evolutionary algorithm whose computation depends on the structure of the considered approach and might not be applicable to any model; the authors choose a neural network because of the efficiency of the computation. Another drawback of this approach, and in general, of any evolutionary multi-objective optimization method is that, as shown in [38], it does not scale well with the number of objectives.

In this paper, we propose to use the game theory based method of Kalai and Smorodinsky (KS) [23] 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 [24] 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 a learning method over the hyperparameter grid, thus it is independent of the method used. If [25] can optimize up to four objectives, we can apply the KS method to any number of objectives without incurring any computational cost. In this paper, we find the compromise solution for eight objectives.

## 7 Experiments

### 7.1 Experimental Setup

The goal of the experiments is to compare the overall performance on multiple evaluation measures, the KS compromise solution, of the harmonic function (HF) and the iterative version (ILP) (6), using the new transition matrix (8), to other label propagation (LP)/local methods as well as to evaluate its stacking performance using several competitive inductive learning methods.

We consider (to the best of our knowledge) all existing LP methods: the straightforward multi-label extension of regularized label propagation approaches, consistency method (CM) [9] and linear neighbourhood label propagation (LNP) [11], and multi-label propagation approaches, dynamic label propagation (DLP) [11] and TRAM [12]. In the case of TRAM method, we do not row-normalize the label matrix which leads to a better performance, and also, since we work with the data of small dimensionality we do not apply any dimensionality reduction approach as it is done in the original work. As the considered LP approach models the manifold by the $k$-nearest neighbour graph, we also compare its performance to ML-KNN [19].

For stacking performance of HF/ILP, we consider the following competitive multi-label classifiers: binary relevance method (BR) [39], ensembles of classifier chains (ECC) [20] and random k-labelsets (RAKEL) [21]. For these methods the support-vector machine is chosen as a base algorithm. Since BR is tailored for label-wise performance, it improves the Hamming loss, on the other hand, ECC and RAKEL are capable of modeling label dependencies, and thus improve the subset accuracy [15, 17]. Then the general view of their performances gives an idea about possible label dependencies. More precisely, when there is no label dependency, based on [15, 17], we can expect Hamming loss and subset accuracy to be optimized simultaneously by BR method in which case using HF/ILP as a stacking

method is useless. If this is not the case, 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.

As a decision rule, for LNP, since the labels are in $\{-1, 1\}$, we use the sign function. TRAM method uses label propagation to find the number of labels with the highest score for each unlabelled point: as shown in Appendix A, TRAM is equivalent to using HF with the standard row normalized matrix where the same procedure is also used to find the number of relevant labels. The remaining methods use both CMN (9) and LCO (10).

For the quadratic optimization problem of LNP we use `OSQP` package [40], and for the KS solution the `GPGame` package [41]. We use the `utiml` package [42] for BR, ECC, RAKEL and ML-KNN. We implemented all label propagation methods in `R` programming language [43].

We selected five publicly available data sets[2] with the moderate number $n$ of observations and the low average class-imbalance level defined as $\mathrm{avgImb}(D) = \dfrac{1}{C}\displaystyle\sum_{j=1}^{C} \dfrac{\max(F(j), n - F(j))}{\min(F(j), n - F(j))}$ where $F(j) = \sum_{i=1}^{n} y_{ij}$ [45]. Table 1 shows the values of these properties along with the number $C$ of labels, the number $\mathrm{ls}$ of unique labelsets, and the label density $\mathrm{ld} = \mathrm{lc}(D)/C$. 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 susceptible to the magnitude of values.

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 | 7 | 12 | 147 | 0.344 | 5.451 |
| SCENE | 2407 | 294 | 6 | 15 | 0.179 | 4.662 |
| YEAST | 2417 | 103 | 14 | 198 | 0.325 | 8.867 |

Since the current paper focuses on small data, we have sampled only half of the observations from Scene and Yeast datasets by stratified sampling based on the label powerset approach [46] to guarantee the similar label distribution, and thus similar $\mathrm{ls}, \mathrm{ld}$ and $\mathrm{avgImb}$ to those of the entire dataset. This same approach is also used in $5 \times 2$-fold cross validation to tune the hyperparameters.

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

We report the average performance and the standard deviation.

### 7.2 Results

The results of the experiments are summarized in Tables 2-5. They reflect the KS compromise solution for multiple metrics. In all experiments we observed CMN to be a better Hamming loss minimizer compared to LCO. The latter, on the other hand, improved the family of $F1$-measures at the cost of degrading the Hamming loss. For simplicity of presentation we report the performance corresponding to one of these decision rules (homogeneous for all methods, except for LNP and TRAM): 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 6 in Appendix C; the observed tendency holds for all methods. We reiterate the fact that these decision rules do not affect the values of the macro- and micro-AUC measures and the average precision since they are based on the ranking of the real-valued outputs.

Firstly, notice that interpolated LP approaches, HF/ILP, are superior to the regularized approaches, CM and LNP, on almost all metrics and on all datasets which is indicative of the fact that while capturing the underlying structure, the transition matrix (8) controls the noise better. We also did not observe any remarkable performance by ML-KNN. In particular, HF has a clear superiority over all LP methods as well as ML-KNN on Emotions and Yeast dataset. This is particularly remarkable for the former dataset: it contains a small number of unique labelsets, the lowest class imbalance

[2] All datasets except Fungi are taken from https://www.uco.es/kdis/mllresources/. The Fungi dataset has been kindly provided to us by C. Averill [44].

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. The Fungi dataset, on the other hand, contains the smallest number of observations, and being transductive methods, HF, TRAM and DLP outperform all inductive methods with respect to all evaluation metrics except the subset accuracy. This dataset also has the highest ratio $\mathrm{ls}\,/n$ of unique labelsets to the number of observations, thus the inferior performance demonstrated by RAKEL. Furthermore, it contains independent subsets of correlated fungi species and ECC leverages it as can be judged from the value of the subset accuracy. The predictions of ECC are further improved by HF, however the performance of HF as a stacking method is not as remarkable as it is for the remaining datasets due to independent labels. Since HF outperforms the inductive methods, for comparison we also use its own predictions in the stacking approach. Compared to standalone HF, incorporating the label information substantially improves the subset accuracy and the family of $F1$ measures at the cost of slightly deteriorating the Hamming loss.

Table 2: 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 decision function used is LCO.

A) HF and LP/ML-KNN

| Measures | HF | CM | LNP | TRAM | DLP | ML-KNN |
|---|---|---|---|---|---|---|
| Hamming loss ↓ | 0.1891 ± 0.0085 | 0.2039 ± 0.0104 | 0.2024 ± 0.0076 | 0.2251 ± 0.0109 | 0.2387 ± 0.0060 | 0.2029 ±0 .0088 |
| Subset accuracy ↑ | 0.3305 ± 0.0151 | 0.2758 ± 0.0335 | 0.2941 ± 0.0164 | 0.2472 ± 0.0176 | 0.2091 ± 0.0174 | 0.2906 ± 0.0258 |
| F1 ↑ | 0.6730 ± 0.0133 | 0.6484 ± 0.0206 | 0.6193 ± 0.0141 | 0.6519 ± 0.0151 | 0.6072 ± 0.0082 | 0.6482 ± 0.0124 |
| Macro-F1 ↑ | 0.6792 ± 0.0146 | 0.6596 ± 0.0182 | 0.6390 ± 0.0148 | 0.6660 ± 0.0167 | 0.6123 ± 0.0109 | 0.6471 ± 0.0151 |
| Micro-F1 ↑ | 0.6975 ± 0.0141 | 0.6760 ± 0.0165 | 0.6534 ± 0.0142 | 0.6757 ± 0.0163 | 0.6365 ± 0.0089 | 0.6746 ± 0.0131 |
| Macro-AUC ↑ | 0.8503 ± 0.0087 | 0.8337 ± 0.0100 | 0.7369 ± 0.0088 | 0.8431 ± 0.0103 | 0.7631 ± 0.0119 | 0.8229 ± 0.0110 |
| Micro-AUC ↑ | 0.8662 ± 0.0081 | 0.8491 ± 0.0088 | 0.7474 ± 0.0084 | 0.8598 ± 0.0106 | 0.7825 ± 0.0113 | 0.8484 ± 0.0108 |
| Average precision↑ | 0.8081 ± 0.0056 | 0.8074 ± 0.0068 | 0.7454 ± 0.0097 | 0.8034 ± 0.0097 | 0.7930 ± 0.0042 | 0.7968 ± 0.0077 |

B) HF as a stacking method

| Measures | BR | ECC | RAKEL | RAKEL+HF |
|---|---|---|---|---|
| Hamming loss ↓ | 0.1873 ± 0.0077 | 0.1857 ± 0.0068 | 0.1841 ± 0.0069 | 0.1799 ± 0.0094 |
| Subset accuracy ↑ | 0.3292 ± 0.0219 | 0.3410 ± 0.0196 | 0.3568 ± 0.0138 | 0.3703 ± 0.0159 |
| F1 ↑ | 0.6747 ± 0.0144 | 0.6710 ± 0.0135 | 0.6809 ± 0.0130 | 0.6858 ± 0.0144 |
| Macro-F1 ↑ | 0.6822 ± 0.0147 | 0.6869 ± 0.0141 | 0.6905 ± 0.0119 | 0.6986 ± 0.0158 |
| Micro-F1 ↑ | 0.7003 ± 0.0123 | 0.6995 ± 0.0126 | 0.7060 ± 0.0118 | 0.7114 ± 0.0155 |
| Macro-AUC ↑ | 0.8508 ± 0.0085 | 0.8385 ± 0.0111 | 0.8044 ± 0.0078 | 0.8515 ± 0.0083 |
| Micro-AUC ↑ | 0.8704 ± 0.0082 | 0.8510 ± 0.0097 | 0.8162 ± 0.0071 | 0.8646 ± 0.0091 |
| Average precision↑ | 0.8244 ± 0.0133 | 0.8117 ± 0.0124 | 0.8001 ± 0.0075 | 0.8241 ± 0.0087 |

Table 3: 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 ↓ | 0.0938 ± 0.0084 | 0.0990 ± 0.0075 | 0.1290 ± 0.0095 | 0.1070 ± 0.0085 | 0.0941 ± 0.0051 | 0.1084 ± 0.0076 |
| Subset accuracy ↑ | 0.6894 ± 0.0225 | 0.6765 ± 0.0202 | 0.5878 ± 0.0257 | 0.6519 ± 0.0221 | 0.6874 ± 0.0138 | 0.6493 ± 0.0201 |
| F1 ↑ | 0.7346 ± 0.0244 | 0.7189 ± 0.0217 | 0.6326 ± 0.0279 | 0.6948 ± 0.0242 | 0.7351 ± 0.0144 | 0.6910 ± 0.0218 |
| Macro-F1 ↑ | 0.7412 ± 0.0226 | 0.7258 ± 0.0207 | 0.6425 ± 0.0290 | 0.7026 ± 0.0225 | 0.7383 ± 0.0135 | 0.6968 ± 0.0208 |
| Micro-F1 ↑ | 0.7301 ± 0.0242 | 0.7137 ± 0.0216 | 0.6303 ± 0.0280 | 0.6907 ± 0.0246 | 0.7299 ± 0.0145 | 0.6865 ± 0.0220 |
| Macro-AUC ↑ | 0.9168 ± 0.0050 | 0.7517 ± 0.0068 | 0.7458 ± 0.0154 | 0.9268 ± 0.0062 | 0.8238 ± 0.0071 | 0.8901 ± 0.0110 |
| Micro-AUC ↑ | 0.9138 ± 0.0046 | 0.7579 ± 0.0062 | 0.7440 ± 0.0154 | 0.9216 ± 0.0068 | 0.8344 ± 0.0055 | 0.9062 ± 0.0086 |
| Average precision↑ | 0.8531 ± 0.0145 | 0.8452 ± 0.0128 | 0.7352 ± 0.0214 | 0.8295 ± 0.0149 | 0.8544 ± 0.0084 | 0.8232 ± 0.0134 |

B) ILP as a stacking method

| Measures | BR | ECC | RAKEL | RAKEL+ILP |
|---|---|---|---|---|
| Hamming loss ↓ | 0.0892 ± 0.0053 | 0.0887 ± 0.0043 | 0.0876 ± 0.0054 | 0.0844 ± 0.0051 |
| Subset accuracy ↑ | 0.7038 ± 0.0152 | 0.7068 ± 0.0123 | 0.7104 ± 0.0160 | 0.7182 ± 0.0142 |
| F1 ↑ | 0.7477 ± 0.0154 | 0.7496 ± 0.0126 | 0.7531 ± 0.0162 | 0.7620 ± 0.0149 |
| Macro-F1 ↑ | 0.7523 ± 0.0145 | 0.7502 ± 0.0113 | 0.7538 ± 0.0148 | 0.7651 ± 0.0137 |
| Micro-F1 ↑ | 0.7421 ± 0.0152 | 0.7434 ± 0.0123 | 0.7468 ± 0.0156 | 0.7559 ± 0.0147 |
| Macro-AUC ↑ | 0.9311 ± 0.0035 | 0.9137 ± 0.0059 | 0.8666 ± 0.0064 | 0.9067 ± 0.0085 |
| Micro-AUC ↑ | 0.9418 ± 0.0029 | 0.9138 ± 0.0067 | 0.8621 ± 0.0062 | 0.9143 ± 0.0076 |
| Average precision↑ | 0.8639 ± 0.0087 | 0.8511 ± 0.0069 | 0.8299 ± 0.0103 | 0.8648 ± 0.0097 |

Table 4: 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 ↓ | 0.2030 ± 0.0047 | 0.2164 ± 0.0037 | 0.2306 ± 0.0069 | 0.2237 ± 0.0053 | 0.2081 ± 0.0025 | 0.2165 ± 0.0064 |
| Subset accuracy ↑ | 0.2151 ± 0.0087 | 0.1584 ± 0.0065 | 0.1340 ± 0.0262 | 0.0667 ± 0.0100 | 0.1696 ± 0.0121 | 0.1611 ± 0.0189 |
| F1 ↑ | 0.6473 ± 0.0073 | 0.6150 ± 0.0108 | 0.5685 ± 0.0141 | 0.6398 ± 0.0067 | 0.6425 ± 0.0057 | 0.6286 ± 0.0102 |
| Macro-F1 ↑ | 0.4471 ± 0.0071 | 0.4317 ± 0.0125 | 0.3166 ± 0.0133 | 0.4499 ± 0.0104 | 0.3735 ± 0.0056 | 0.3840 ± 0.0131 |
| Micro-F1 ↑ | 0.6653 ± 0.0072 | 0.6426 ± 0.0084 | 0.5947 ± 0.0117 | 0.6565 ± 0.0073 | 0.6574 ± 0.0044 | 0.6439 ± 0.0096 |
| Macro-AUC ↑ | 0.7157 ± 0.0112 | 0.6906 ± 0.0091 | 0.5893 ± 0.0106 | 0.7121 ± 0.0125 | 0.7072 ± 0.0054 | 0.6514 ± 0.0061 |
| Micro-AUC ↑ | 0.8446 ± 0.0045 | 0.8298 ± 0.0049 | 0.7879 ± 0.0067 | 0.8421 ± 0.0049 | 0.8315 ± 0.0025 | 0.8239 ± 0.0043 |
| Average precision↑ | 0.7566 ± 0.0068 | 0.7611 ± 0.0055 | 0.7162 ± 0.0068 | 0.7548 ± 0.0073 | 0.7489 ± 0.0033 | 0.7434 ± 0.0055 |

B) HF as a stacking method

| Measures | BR | ECC | RAKEL | RAKEL+HF |
|---|---|---|---|---|
| Hamming loss ↓ | 0.1969 ± 0.0046 | 0.2098 ± 0.0030 | 0.2009 ± 0.0047 | 0.1975 ± 0.0051 |
| Subset accuracy ↑ | 0.1684 ± 0.0125 | 0.1696 ± 0.0094 | 0.2084 ± 0.0171 | 0.2381 ± 0.0141 |
| F1 ↑ | 0.6104 ± 0.0090 | 0.6329 ± 0.0062 | 0.6555 ± 0.0085 | 0.6570 ± 0.0089 |
| Macro-F1 ↑ | 0.3645 ± 0.0131 | 0.3919 ± 0.0117 | 0.4244 ± 0.0086 | 0.4557 ± 0.0116 |
| Micro-F1 ↑ | 0.6364 ± 0.0092 | 0.6494 ± 0.0064 | 0.6698 ± 0.0081 | 0.6746 ± 0.0085 |
| Macro-AUC ↑ | 0.6967 ± 0.0129 | 0.6421 ± 0.0079 | 0.6083 ± 0.0052 | 0.7202 ± 0.0108 |
| Micro-AUC ↑ | 0.8377 ± 0.0031 | 0.8091 ± 0.0042 | 0.7796 ± 0.0069 | 0.8468 ± 0.0048 |
| Average precision↑ | 0.7649 ± 0.0057 | 0.7523 ± 0.0076 | 0.7562 ± 0.0065 | 0.7614 ± 0.0071 |

Table 5: 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 ↓ | 0.2018 ± 0.0088 | 0.2066 ± 0.0127 | 0.2141 ± 0.0086 | 0.2020 ± 0.0071 | 0.2046 ± 0.0086 | 0.2204 ± 0.0123 |
| Subset accuracy ↑ | 0.1509 ± 0.0204 | 0.1509 ± 0.0247 | 0.1292 ± 0.0228 | 0.1521 ± 0.0147 | 0.1499 ± 0.0192 | 0.1238 ± 0.0346 |
| F1 ↑ | 0.6696 ± 0.0158 | 0.6374 ± 0.0258 | 0.6489 ± 0.0137 | 0.6692 ± 0.0105 | 0.6600 ± 0.0161 | 0.6281 ± 0.0247 |
| Macro-F1 ↑ | 0.5201 ± 0.0132 | 0.4493 ± 0.0511 | 0.5813 ± 0.0217 | 0.5430 ± 0.0087 | 0.5045 ± 0.0201 | 0.4875 ± 0.0376 |
| Micro-F1 ↑ | 0.6708 ± 0.0127 | 0.6344 ± 0.0276 | 0.6643 ± 0.0131 | 0.6736 ± 0.0082 | 0.6607 ± 0.0141 | 0.6389 ± 0.0269 |
| Macro-AUC ↑ | 0.8089 ± 0.0103 | 0.8088 ± 0.0142 | 0.7005 ± 0.0134 | 0.8041 ± 0.0113 | 0.8102 ± 0.0112 | 0.7482 ± 0.0141 |
| Micro-AUC ↑ | 0.8644 ± 0.0048 | 0.8607 ± 0.0088 | 0.7500 ± 0.0104 | 0.8628 ± 0.0050 | 0.8634 ± 0.0066 | 0.8353 ± 0.0065 |
| Average precision↑ | 0.8433 ± 0.0096 | 0.8437 ± 0.0085 | 0.7498 ± 0.0086 | 0.8412 ± 0.0067 | 0.8444 ± 0.0092 | 0.8008 ± 0.0103 |

B) HF as a stacking method

| Measures | BR | ECC | RAKEL | ECC+HF | HF+HF |
|---|---|---|---|---|---|
| Hamming loss ↓ | 0.2064 ± 0.0100 | 0.2086 ± 0.0100 | 0.2183 ± 0.0121 | 0.2058 ± 0.0112 | 0.2021 ± 0.0082 |
| Subset accuracy ↑ | 0.1460 ± 0.0215 | 0.1576 ± 0.0249 | 0.1502 ± 0.0209 | 0.1583 ± 0.0171 | 0.1577 ± 0.0221 |
| F1 ↑ | 0.6468 ± 0.0182 | 0.6533 ± 0.0188 | 0.6231 ± 0.0107 | 0.6653 ± 0.0201 | 0.6730 ± 0.0131 |
| Macro-F1 ↑ | 0.4886 ± 0.0299 | 0.5525 ± 0.0258 | 0.4841 ± 0.0346 | 0.5696 ± 0.0167 | 0.5594 ± 0.0183 |
| Micro-F1 ↑ | 0.6466 ± 0.0137 | 0.6605 ± 0.0165 | 0.6288 ± 0.0173 | 0.6723 ± 0.0145 | 0.6769 ± 0.0104 |
| Macro-AUC ↑ | 0.7594 ± 0.0200 | 0.7653 ± 0.0123 | 0.6980 ± 0.0235 | 0.7933 ± 0.0128 | 0.8022 ± 0.0156 |
| Micro-AUC ↑ | 0.8358 ± 0.0036 | 0.8403 ± 0.0049 | 0.7828 ± 0.0199 | 0.8614 ± 0.0060 | 0.8656 ± 0.0055 |
| Average precision↑ | 0.8145 ± 0.0117 | 0.8098 ± 0.0103 | 0.7930 ± 0.0175 | 0.8278 ± 0.0107 | 0.8424 ± 0.0085 |

## 8 Conclusions

This paper extended the harmonic function and its iterative version to the multi-label setting via the binary relevance transformation. In particular, we constructed a new transition matrix which better captures the underlying clustering structure for most real-world datasets. 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 the thresholds for each label independently based on their corresponding frequencies, we proposed using the class-mass normalization method in the multi-label setting to improve the Hamming loss. Finally, without favoring one metric over the other, we reported the single compromise output using the game-theory based multi-objective optimization approach. The obtained results show that, despite its simplicity, performing label propagation on an absorbing Markov chain with our transition matrix provides a competitive approach capable of improving the outputs of an external learning model when there exists dependence between the labels.

## References


[1] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[2] E. J. E. Van and H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[3] V. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.

[4] D. Pechyony and R. El-Yaniv, *Theory and practice of transductive learning*. PhD thesis, Computer Science Department, Technion, 2009.

[5] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine learning*, vol. 56, no. 1, pp. 209–239, 2004.

[6] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. 11, 2006.

[7] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," 2002.

[8] X. Zhu, Z. Ghahramani, and J. Lafferty, "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.

[9] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, pp. 321–328, 2004.

[10] D. Zhou and B. Schölkopf, "Learning from labeled and unlabeled data using random walks," in *Joint Pattern Recognition Symposium*, pp. 237–244, Springer, 2004.

[11] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2007.

[12] X. Kong, M. Ng, and Z. Zhou, "Transductive multilabel learning via label set propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 704–719, 2011.

[13] B. Wang, Z. Tu, and J. Tsotsos, "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.

[14] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*, pp. 667–685, Springer, 2009.

[15] K. Dembczyński, W. Waegeman, W. Cheng, and W. Hüllermeier, "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.

[16] K. K. Dembczyński, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *ICML*, 2010.

[17] K. K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "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.

[18] P. Doyle and J. Snell, *Random walks and electric networks*, vol. 22. American Mathematical Soc., 1984.

[19] M. Zhang and Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[20] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[21] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE transactions on knowledge and data engineering*, vol. 23, no. 7, pp. 1079–1089, 2010.

[22] J. Read, B. Pfahringer, and G. Holmes, "Generating synthetic multi-label data streams," in *ECML/PKKD 2009 Workshop on Learning from Multi-label Data (MLD'09)*, pp. 69–84, Citeseer, 2009.

[23] E. Kalai and M. Smorodinsky, "Other solutions to Nash's bargaining problem," *Econometrica: Journal of the Econometric Society*, pp. 513–518, 1975.

[24] M. Binois, V. Picheny, P. Taillandier, and A. Habbal, "The Kalai-Smorodinsky solution for many-objective Bayesian optimization," *J. Mach. Learn. Res.*, vol. 21, no. 150, pp. 1–42, 2020.

[25] C. Shi, X. Kong, P. Yu, and B. Wang, "Multi-objective multi-label classification," in *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 355–366, SIAM, 2012.

[26] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *Advances in neural information processing systems*, Citeseer, 2002.

[27] F. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.

[28] E. A. Nadaraya, *Nonparametric estimation of probability densities and regression curves*. Springer, 1989.

[29] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[30] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[31] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[32] P. Legendre and E. Gallagher, "Ecologically meaningful transformations for ordination of species data," *Oecologia*, vol. 129, no. 2, pp. 271–280, 2001.

[33] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[34] B. Zadrozny and C. Elkan, "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.

[35] T. Leathart, E. Frank, G. Holmes, and B. Pfahringer, "Probability calibration trees," in *Asian Conference on Machine Learning*, pp. 145–160, 2017.

[36] R. Fan and C. Lin, "A study on threshold selection for multi-label classification," *Department of Computer Science, National Taiwan University*, pp. 1–23, 2007.

[37] Y. Yang, "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.

[38] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, pp. 2419–2426, IEEE, 2008.

[39] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[40] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[41] V. Picheny and M. Binois, *GPGame: Solving Complex Game Problems using Gaussian Processes*, 2022. R package version 1.2.0.

[42] A. Rivolli and A. de Carvalho, "The utiml package: Multi-label classification in R.," *R J.*, vol. 10, no. 2, p. 24, 2018.

[43] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[44] C. Averill, Z. Werbin, K. Atherton, J. Bhatnagar, and M. Dietze, "Soil microbiome predictability increases with spatial and taxonomic scale," *Nature Ecology & Evolution*, vol. 5, no. 6, pp. 747–756, 2021.

[45] M. Zhang, Y. Li, H. Yang, and X. Liu, "Towards class-imbalance aware multi-label learning," *IEEE Transactions on Cybernetics*, 2020.

[46] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 145–158, Springer, 2011.

## A The Approach of [12] is the Harmonic Function

The optimization problem of [12] is as follows:

$$\min_{\mathbf{f}} \sum_{i=l+1}^{n} \sum_{k=1}^{C} \left( f_k(i) - \sum_{j=1}^{n} P_{ij} f_k(j) \right)^2 \tag{11}$$

where $P_{ij}$ corresponds to the $(i,j)$-th entry in our 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 (3). The fact that $f_k(i) \in [0,1]$, $i \in \{l+1, \ldots, n\}$ follows from the discrete maximum principle.

## B Decision Function and Evaluation Measures

The multi-label classifier is usually defined either as the set $H$ of functions $h = (h_1, \ldots, h_C) : \mathcal{X} \to \mathcal{Y}$, or as the set $H^s$ of functions $h^s = (h_1^s, \ldots, h_C^s) : \mathcal{X} \to \mathbb{R}^C$ outputting *soft* labels or confidence values for an instance $\mathbf{x}$. Any $z = h_j^s(\mathbf{x}_i)$ can be mapped to $\mathcal{Y}$ using a decision function

$$dr_t(z) = \begin{cases} 1 & \text{if } z \geq t \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where $t \geq 0$ is a threshold. To keep the notation simple, in the sequel, we will assume that $h(x)$ is a value obtained after applying such a decision function.

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, thus it is less restrictive:

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

The AUC measure evaluates the capacity of multi-label classifier for each instance to score higher the relevant labels than 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 $h_r^s(X) \geqslant h_{irr}^s(X)$, 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 $h_j^s(X_r) \geqslant h_j^s(X_{irr})$, 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 $h_i^s(X_r(i)) \geqslant h_j^s(X_{irr}(j))$.

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

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

## C Comparison of CMN and LCO Thresholding Approaches for Harmonic Function

Table 6: The KS compromise 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.

| | Emotions | | Scene | |
|---|---|---|---|---|
| Measures | CMN | LCO | CMN | LCO |
| Hamming loss ↓ | 0.1856 ± 0.0049 | 0.1891 ± 0.0085 | 0.0956 ± 0.0077 | 0.1016 ± 0.0069 |
| Subset accuracy ↑ | 0.3167 ± 0.0075 | 0.3305 ± 0.0151 | 0.6828 ± 0.0215 | 0.6229 ± 0.0163 |
| F1 ↑ | 0.6475 ± 0.0100 | 0.6730 ± 0.0133 | 0.7292 ± 0.0222 | 0.7339 ± 0.0205 |
| Macro-F1 ↑ | 0.6511 ± 0.0111 | 0.6792 ± 0.0146 | 0.7360 ± 0.0202 | 0.7404 ± 0.0178 |
| Micro-F1 ↑ | 0.6792 ± 0.0099 | 0.6975 ± 0.0141 | 0.7246 ± 0.0220 | 0.7248 ± 0.0194 |

| | Yeast | | Fungi | |
|---|---|---|---|---|
| Measures | CMN | LCO | CMN | LCO |
| Hamming loss ↓ | 0.1961 ± 0.0048 | 0.2030 ± 0.0047 | 0.2018 ± 0.0088 | 0.2077 ± 0.0082 |
| Subset accuracy ↑ | 0.2036 ± 0.0085 | 0.2151 ± 0.0087 | 0.1509 ± 0.0204 | 0.1398 ± 0.0174 |
| F1 ↑ | 0.6113 ± 0.0096 | 0.6473 ± 0.0073 | 0.6696 ± 0.0158 | 0.6760 ± 0.0158 |
| Macro-F1 ↑ | 0.4070 ± 0.0103 | 0.4471 ± 0.0071 | 0.5201 ± 0.0132 | 0.5758 ± 0.0211 |
| Micro-F1 ↑ | 0.6415 ± 0.0089 | 0.6653 ± 0.0072 | 0.6708 ± 0.0127 | 0.6853 ± 0.0120 |