involve
a journal of mathematics

Improving multilabel classification via heterogeneous ensemble methods

Yujue Wu and Qing Wang

msp

2019 vol. 12, no. 6

INVOLVE 12:6 (2019)
dx.doi.org/10.2140/involve.2019.12.1035

Improving multilabel classification via heterogeneous ensemble methods

Yujue Wu and Qing Wang

(Communicated by Sat N. Gupta)

We consider the task of multilabel classification, where each instance may belong to multiple labels simultaneously. We propose a new method, called multilabel super learner (MLSL), that is built upon the problem transformation approach using the one-vs-all binary relevance method. MLSL is an ensemble model that predicts multilabel responses by integrating the strength of multiple base classifiers, and therefore it is likely to outperform each base learner. The weights in the ensemble classifier are determined by optimization of a loss function via V-fold cross-validation. Several loss functions are considered and evaluated numerically. The performance of various realizations of MLSL is compared to existing problem transformation algorithms using three real data sets, spanning applications in biology, music, and image labeling. The numerical results suggest that MLSL outperforms existing methods most of the time evaluated by the commonly used performance metrics in multilabel classification.

1. Introduction

Classification is a task of predicting labels of future instances by learning from the patterns of observed instances with known labels [Herrera et al. 2016]. The traditional classification problem, known as single-label classification, considers data sets with only one output attribute. When the single output attribute has two categories, it is referred to as binary classification; when the output attribute has more than two categories, it is called multiclass classification. In this paper we focus on the problem of multilabel classification, where each instance may be associated with more than one label.

The first literature on multilabel classification dates back to [McCallum 1999]; it focuses on the task of text categorization. In recent decades, multilabel classification has become an emerging research area and has been applied to many different disciplines, including image labeling [Duygulu et al. 2002; Boutell et al. 2004],

MSC2010: 62-07.
Keywords: binary relevance, heterogeneous ensemble, multilabel classification, stacking, super learner.

sentiment analysis [Turnbull et al. 2008; Sobol-Shikler and Robinson 2010] and bioinformatics [Elisseeff and Weston 2001; Diplaris et al. 2005]. More recent work on multilabel text categorization can also be found in [Klimt and Yang 2004; Lewis et al. 2004; Crammer et al. 2007; Katakis et al. 2008; Sriram et al. 2010; Charte et al. 2015]. A good overview of multilabel classification and its methods is provided in [Tsoumakas and Katakis 2007; Zhang and Zhou 2014; Gibaja and Ventura 2015; Herrera et al. 2016].

We can formally formulate the problem of multilabel classification as follows [Herrera et al. 2016]: Consider a dataset $\mathcal{D}$ with f input attributes $V_1, \ldots, V_f$. Let $\mathcal{V} = \{V_1, \ldots, V_f\}$ be the set of all input attributes in the dataset and $|\mathcal{V}| = f \geq 1$. Let $\mathcal{X} = V_1 \times V_2 \times V_3 \times \cdots \times V_f$. That is, $\mathcal{X}$ is the input space of the dataset, and $\mathcal{D} \subseteq \mathcal{X}$. Let $\mathcal{L} = \{y_1, \ldots, y_k\}$ be a set of distinct labels for $\mathcal{D}$, where each y_j represents a label. Here $|\mathcal{L}| = k \geq 2$. In single-label classification, including both binary and multiclass classification, each instance $\boldsymbol{x} \in \mathcal{X}$ is associated with one and only one label $y_j \in \mathcal{L}$. However, in multilabel classification, each instance $\boldsymbol{x} \in \mathcal{X}$ is associated with a subset of labels $L \subseteq \mathcal{L}$, where $1 \leq |L| \leq k$. The output space in multilabel classification, denoted by $\mathcal{Y}_{\text{multilabel}}$, is defined as the Cartesian product of k sets of binary values 0 and 1; i.e.,

$$\mathcal{Y}_{\text{multilabel}} = \{0, 1\}_1 \times \{0, 1\}_2 \times \cdots \times \{0, 1\}_k.$$

A multilabel classifier, denoted by $\mathcal{C} : \mathcal{X} \to \mathcal{Y}_{\text{multilabel}}$, learns from the input space $\mathcal{X}$ and predicts outcomes in the output space $\mathcal{Y}_{\text{multilabel}}$.

Generally speaking, there are two fundamental approaches to realize multilabel classification: *problem transformation* and *algorithm adaption* [Herrera et al. 2016]. The problem transformation methodology, at its core, converts a multilabel data set into several single-label data sets, thereby allowing the transformed data sets to be modeled using existing binary or multiclass classification methods. For example, one of the ways to realize problem transformation is through the *one-vs-all binary relevance* method, where a multilabel data set with k labels is converted into k binary data sets, one for each label. On the other hand, the algorithm adaption methodology transforms a single-label classification algorithm so that it can be applied to the original multilabel data set.

In this paper we propose a new method, called multilabel super learner (MLSL), which is an improved multilabel classification algorithm following the problem transformation approach, and is built upon the one-vs-all binary relevance method. MLSL is an ensemble model that makes predictions based on an integration of multiple base classifiers. The weights in the ensemble classifier are determined by optimizing a loss function. Several widely used loss functions are considered and evaluated numerically in this paper. The performance of the proposal is compared to existing problem transformation algorithms using real data sets in Section 4.

The numerical results suggest that MLSL outperforms existing binary relevance algorithms evaluated by almost all of the commonly used performance metrics in multilabel classification. To the best of our knowledge, none of the previous research considers implementing ensemble methods of this kind in multilabel classification.

The rest of the paper is structured as follows. In Section 2 we introduce the two general approaches to realize multilabel classification, and focus on the binary relevance method that the proposed MLSL is built upon. In Section 3 we detail the proposed MLSL algorithm, followed by numerical studies in Section 4. Finally, we conclude the paper with discussions of some future work in Section 5.

2. Existing methods for multilabel classification

We now introduce commonly used methods in multilabel classification. The two main approaches for multilabel classification are problem transformation and algorithm adaption. Problem transformation can be realized in two possible ways: (1) by converting the multilabel dataset into multiple binary data sets, (2) by converting the multilabel data set into one multiclass data set. These two approaches are often referred to as *binary relevance* and *label powerset* respectively. After the conversion, the altered data sets are suitable for single-label classification. Individually predicted labels are obtained from each of these single-label data sets, and then combined to produce the desired multilabeled outputs as the final predictions.

In algorithm adaption, existing single-label classification methods are altered so that they can be applied to multilabel data sets. Common methods under this framework include instance-based and logistic regression (IBLR-ML) [Cheng and Hüllermeier 2009], which is adapted from k-nearest neighbor (kNN) [Cover and Hart 1967] and logistic regression [Cox 1958], MODEL-x [Boutell et al. 2004], which is derived from support vector machines (SVM) [Cortes and Vapnik 1995], and the multilabel k-nearest neighbor lazy learning algorithm (ML-kNN) [Zhang and Zhou 2007]. Figure 1 displays an overview of the methods mentioned above. A detailed introduction of multilabel classification methods, including problem transformation and algorithm adaption, can be found in [Herrera et al. 2016; Tsoumakas and Katakis 2007].

Since our proposal is built upon the binary relevance method, we provide more details of this method in the following subsections. In binary relevance, a multilabel data set is converted into multiple single-label data sets. Such a data conversion process can be realized in two different ways: one-vs-all or one-vs-one.

2.1. ***One-vs-all binary relevance.*** The one-vs-all binary relevance approach [Herrera et al. 2016], showcased in Figure 2, transforms a multilabel data set $\mathcal{D}$, associated with k labels, into k unique binary-response data sets — one for each label. One then applies k single-label classifiers to the k binary data sets. The

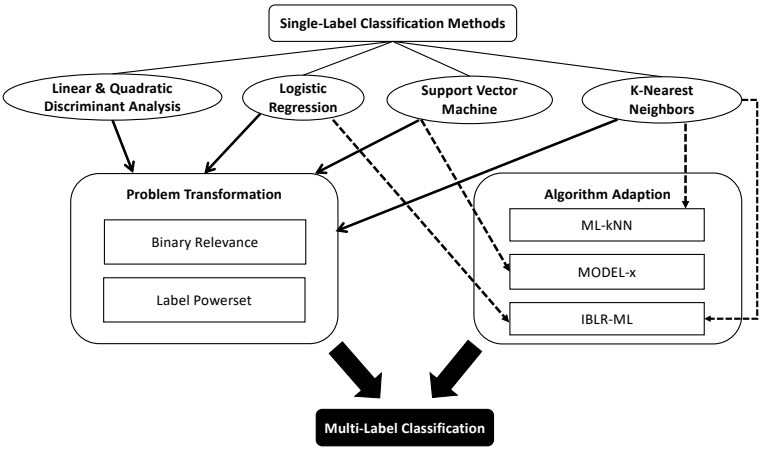

Figure 1. Illustration of some multilabel classification methods and their relationships with single-label classification methods.

k single-label classifiers are often set to be the same classification method, such as support vector machines, although this is not a strict requirement. In the prediction process, each test sample with unknown labels is considered as input for each of the binary classifiers, and based on the inputs, the i-th ($1 \leq i \leq k$) binary classifier produces a binary output, 0 or 1, indicating whether the test sample is associated with label $y_i \in \mathcal{L}$. All outputs generated by the trained binary classifiers will then be combined to form a final multilabel prediction.

The one-vs-all binary relevance approach is easy to implement. In addition, it offers a flexible family of methods in the sense that any binary classifier can be considered and used in the process. However, it suffers from two main disadvantages [Herrera et al. 2016]: First, since the single-label classifiers are independently trained, any potential correlations between labels are not taken into account in producing multilabel predictions. Intuitively, label correlations are valuable information that could help improve the accuracy of multilabel prediction. Second, it is possible that the transformed binary training data sets are more imbalanced than the original multilabel data set. As a result, some challenges may arise in the training stage due to the data conversion.

2.2. ***One-vs-one binary relevance.*** In the one-vs-one approach [Herrera et al. 2016], a multilabel data set is transformed to binary data sets, each of which is associated with a pair of labels in the label space $\mathcal{L}$. That is, given a data set with k unique labels, one considers $\binom{k}{2}$ binary data sets where each data set is associated with labels y_i and y_j ($y_i, y_j \in \mathcal{L}$ and $i \neq j$). Additionally, any instance that is not categorized by either of the two labels under consideration, or is categorized by both labels, is discarded from the corresponding binary data set.

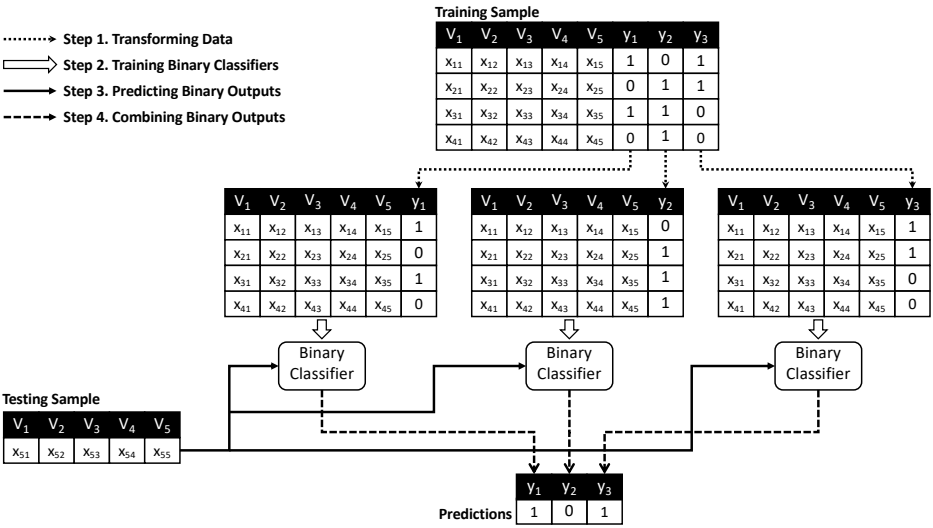

Figure 2. Illustration of one-vs-all binary relevance transformation, assuming five input variables and three possible labels.

In the prediction process, as illustrated in Figure 3, the test sample is considered as input. The output of each binary classifier is then used as "votes", and subsequently a ranking of labels produced by the votes will be generated to decide which labels are to be included in the final multilabel prediction. Examples of ranking algorithms include ranking by pairwise comparison [Hüllermeier et al. 2008] and calibrated label ranking [Fürnkranz et al. 2008].

The one-vs-one binary relevance has the same drawbacks as the one-vs-all binary relevance approach: lack of considerations of label correlations and imbalance in training datasets. In addition, the one-vs-one binary relevance method is likely to be less efficient than the one-vs-all binary relevance method due to the following two reasons: First, any given multilabel dataset with k labels, $k > 2$ and $\binom{k}{2} \geq k$. Thus, the one-vs-one method fits a larger number of binary classifiers than the one-vs-all method. Second, in the prediction process, since the one-vs-one approach incorporates ranking algorithms, it requires additional computation and is therefore likely to introduce errors to the final predictions. As a result, when considering the binary relevance approach, one often prefers the one-vs-all method.

3. Our proposal: multilabel super learner

We propose a stacking-based heterogeneous ensemble method, multilabel super learner (MLSL). MLSL is a multilabel classification algorithm that combines the prediction power of several one-vs-all binary relevance multilabel classification algorithms through an ensemble algorithm, *super learner* [van der Laan et al. 2007].

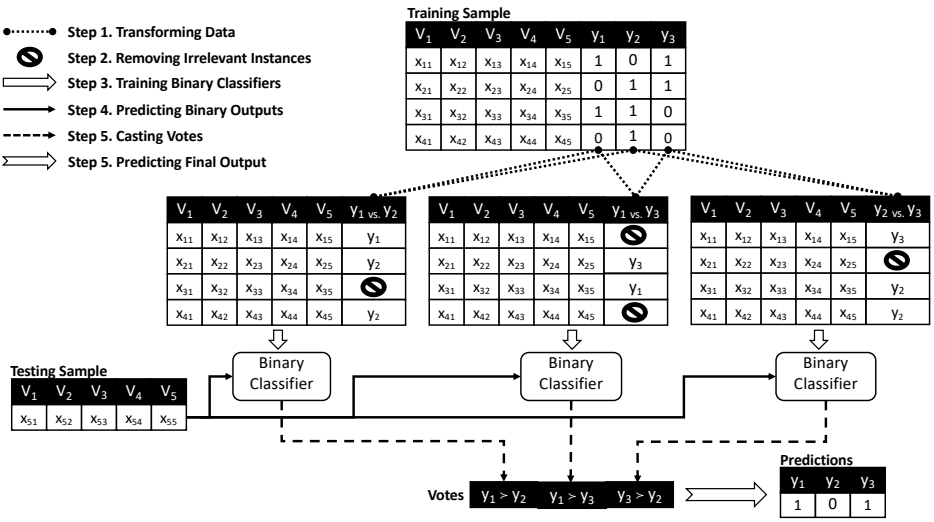

Figure 3. Illustration of one-vs-one binary relevance transformation, assuming five input variables and three possible labels.

In the following we first provide some background for the development of MLSL, followed by a step-by-step description of the MLSL algorithm. In the end we discuss its properties based on the theorems of super learner [van der Laan et al. 2007].

3.1. ***Background.*** MLSL is rooted in stacking, which dates back to the discussion of stacked generalization in [Wolpert 1992]. Stacked generalization combines information from multiple generalizers and minimizes the generalization error rate or biases of the generalizers. This model was later studied by Breiman [1996] in the context of regression and is referred to as stacked regression. Later, Freund et al. [1997] and Hansen [1998] adopted the same idea and proposed combining base learners from different methods to form a single learner. Following in the footsteps of previous work, van der Laan and Dudoit [2003] provided a unified framework to select the optimal combination of the set of base learners through cross-validation; they refer to the optimal solution as a "super learner". More recently, van der Laan et al. [2007] improved the previously proposed super learner by (1) extending it to include more flexible base learning algorithms, and (2) controlling over-fitting of the algorithm using cross-validation. Both [van der Laan and Dudoit 2003] and [van der Laan et al. 2006] show that under some regularity conditions the super learner in regression and single-label classification perform asymptotically as well as or even better than any of the base learning algorithms.

However, none of the previous literature considers applying ensemble methods of this kind to multilabel classification problems. Hence, the main contribution of our paper is to propose a multilabel super learner that integrates the strength and power

of several base multilabel classifiers through an optimal linear combination of them that minimizes some cross-validated risk. More specifically, we adapt the one-vs-all binary relevance method following the problem transformation approach, and implement super learner to realize binary classification based on each transformed binary data set. The weights in the linear combination of the binary base learners are optimized by cross-validated risk, which guards against over-fitting. In the end, predictions from binary super learners are combined to form multilabel output. The detailed algorithm of our proposed MLSL method is presented in the next subsection.

3.2. ***Methodology.*** Suppose we have an input space $\mathcal{X}$ and its associated label space $\mathcal{L}$. In multilabel prediction, one takes any instance $\boldsymbol{x} \in \mathcal{X}$ as an input and predicts an array of outputs

$$\boldsymbol{Y} = [Y_1 \;\; Y_2 \;\; \cdots \;\; Y_k],$$

where

$$Y_j = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is labeled by } y_j \in \mathcal{L}, \\ 0 & \text{otherwise} \end{cases} \qquad (1 \le j \le k).$$

Under this setting, the MLSL algorithm can be realized by the following five steps:

Step 1: selecting base learners. Define a library of m ($m \ge 2$) *base learners* $\{\phi_1, \ldots, \phi_m\}$. Candidates for the base learners include any binary classifier, ranging from simple models, such as support vector machine (SVM) and k-nearest neighbors (kNN), to multistep algorithms that may involve covariate screening, parameter optimization, or model selection.

Step 2: transforming multilabel dataset. Given a training dataset $\mathcal{D} \subseteq \mathcal{X}$, we transform the multilabel datasets into $|\mathcal{L}| = k$ binary datasets, following the one-vs-all binary relevance method. Denote these k transformed binary datasets by $\mathcal{D}_1, \ldots, \mathcal{D}_k$.

Step 3: training single-label super learners. For each binary data set $\mathcal{D}_j$ ($1 \le j \le k$), we realize the single-label super learner as follows:

(1) We first randomly split the j-th binary dataset $\mathcal{D}_j$ into V equally sized subsets, denoted by $\mathcal{D}_j^1, \mathcal{D}_j^2, \ldots, \mathcal{D}_j^V$. Without loss of generality, assume $|\mathcal{D}_j|$ is divisible by V. Denote the number of observations in each data subset D_j^v by $\tilde{n} = |\mathcal{D}_j| / V$. For $v \in \{1, \ldots, V\}$, let $\mathcal{D}_j^v$ be the validation sample and the remaining data be the training sample. Denote the v-th training set by $\mathcal{D}_j^{-v}$ so that $\mathcal{D}_j^{-v} = \mathcal{D}_j \setminus \mathcal{D}_j^v$.

(2) For each v ($1 \le v \le V$), we fit base learners $\phi_h \in \{\phi_1, \ldots, \phi_m\}$ on $\mathcal{D}_j^{-v}$. Denote the fitted classifiers, trained on $\mathcal{D}_j^{-v}$, as $\hat{\phi}_{h,\mathcal{D}_j^{-v}}$ for $1 \le h \le m$. Write the prediction for label j based on the s-th instance $\boldsymbol{x}_s \in \mathcal{D}_j^v$ as $\hat{\phi}_h^j(\boldsymbol{x}_s) := \hat{\phi}_{h,\mathcal{D}_j^{-v}}(\boldsymbol{x}_s)$ ($1 \le h \le m$).

(3) Create a $|\mathcal{D}_j| \times m$ prediction matrix by combining the predictions from the m base learners $\phi_1, \ldots, \phi_m$ over the V validation sets. Denote the prediction matrix for label j by Z_j $(1 \le j \le k)$:

$$Z_j = \begin{bmatrix} \hat{\phi}_1^j(\boldsymbol{x}_1) & \hat{\phi}_2^j(\boldsymbol{x}_1) & \cdots & \hat{\phi}_m^j(\boldsymbol{x}_1) \\ \hat{\phi}_1^j(\boldsymbol{x}_2) & \hat{\phi}_2^j(\boldsymbol{x}_2) & \cdots & \hat{\phi}_m^j(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\phi}_1^j(\boldsymbol{x}_{|\mathcal{D}_j|}) & \hat{\phi}_2^j(\boldsymbol{x}_{|\mathcal{D}_j|}) & \cdots & \hat{\phi}_m^j(\boldsymbol{x}_{|\mathcal{D}_j|}) \end{bmatrix}.$$

Note that each prediction $\hat{\phi}_k^j(\boldsymbol{x}_s)$ in Z_j is obtained by training on a data subset $\mathcal{D}_j^{-v}$ for $v \in \{1, \ldots, V\}$, where $\boldsymbol{x}_s \in \mathcal{D}_j^v$.

(4) For each label j, let $\{\alpha_{j1}, \ldots, \alpha_{jm}\}$ be a set of weights $(\alpha_{jh} \in \mathbb{R},\ 1 \le h \le m)$. Additional constraints on the weights, such as nonnegativity, may be applied but are not required. For any instance $\boldsymbol{x}_s \in \mathcal{D}_j$, define the predicted single-label output as

$$\hat{Y}_{js}(\boldsymbol{x}_s) := \sum_{h=1}^{m} \alpha_{jh}\, \hat{\phi}_h^j(\boldsymbol{x}_s), \tag{3-1}$$

or $\hat{Y}_{js}$ for short. The coefficients, $\alpha_{j1}, \ldots, \alpha_{jm}$, are obtained under some optimization criterion via cross-validation, such as V-fold cross-validation. For instance, if we denote by $L(Y_{js}, \hat{Y}_{js})$ a loss function that evaluates the closeness between Y_{js} and $\hat{Y}_{js}$, then

$$(\hat{\alpha}_{j1}, \ldots, \hat{\alpha}_{jm}) = \arg \min_{\alpha_1, \ldots, \alpha_m} \sum_{v=1}^{V} \sum_{\boldsymbol{x}_s \in \mathcal{D}_j^v} L(Y_{js}, \hat{Y}_{js}(\boldsymbol{x}_s)). \tag{3-2}$$

(5) The predicted probability of label j for instance $\boldsymbol{x}_s$ is thus given by

$$\hat{Y}_j^{\sup}(\boldsymbol{x}_s) = \sum_{h=1}^{m} \hat{\alpha}_h\, \hat{\phi}_h^j(\boldsymbol{x}_s).$$

Given a discriminating threshold c $(0 < c < 1)$, such as 0.5, one determines the classification output. Instances with predicted probabilities greater than the threshold would be classified as 1 (i.e., associated with label j) and as 0 (i.e., not associated with label j) otherwise. Denote the final predicted outcome of label j for instance $\boldsymbol{x}_s$ by $\mathcal{C}_j^{\sup}$, given by

$$\mathcal{C}_j^{\sup}(\boldsymbol{x}_s) = \begin{cases} 1 & \text{if } \hat{Y}_j^{\sup}(\boldsymbol{x}_s) \ge c, \\ 0 & \text{otherwise.} \end{cases}$$

Step 4: predicting future instances. In the prediction process, given an unknown instance $\boldsymbol{x}_t$, the multilabel output is given by combining all k binary outputs

predicted by the k binary super learners:

$$[\mathcal{C}_1^{\sup}(\boldsymbol{x}_t)\ \ \mathcal{C}_2^{\sup}(\boldsymbol{x}_t)\ \ \cdots\ \ \mathcal{C}_k^{\sup}(\boldsymbol{x}_t)].$$

3.3. ***Properties.*** As noted in [van der Laan and Dudoit 2003; van der Laan et al. 2006], the binary super learner is shown to perform asymptotically at least as well as any of the base binary classifiers. As a result, the binary super learner, i.e., $\mathcal{C}_j^{\sup}$ $(1 \le j \le k)$ in Step 4, is asymptotically at least as good as any of the base binary classifiers in $\{\phi_1, \ldots, \phi_m\}$. Therefore, the multilabel prediction, i.e., $[\mathcal{C}_1^{\sup}(\boldsymbol{x}_t)\ \ \cdots\ \ \mathcal{C}_k^{\sup}(\boldsymbol{x}_t)]$, should perform at least as well as, or even better than, the one-vs-all binary relevance multilabel classifier based on ϕ_h for all $h \in \{1, \ldots, m\}$.

4. Numerical comparison

We now empirically examine the performance of the MLSL algorithm introduced in Section 3. We consider four different criteria in determining the optimal weights in (3-2). In the context of V-fold cross-validation, the optimal weights under each criterion are selected as follows:

(1) Nonnegative least squares criterion (MLSL-NNLS): For $1 \le j \le k$,

$$(\alpha_{j1}, \ldots, \alpha_{jm}) = \arg\min \sum_{v=1}^{V} \sum_{\boldsymbol{x}_s \in \boldsymbol{D}_j^v} (Y_{js} - \widehat{Y}_{js}(\boldsymbol{x}_s))^2$$

subject to $\alpha_{j\ell} \ge 0$ $(1 \le \ell \le m)$ and $\sum_{\ell=1}^{h} \alpha_{j\ell} = 1$. Here $\widehat{Y}_{js}(\boldsymbol{x}_s)$ represents the predicted response for label j given an instance $\boldsymbol{x}_s \in \mathcal{D}_j^v$, where the base learners used to define $\widehat{Y}_{js}$ (3-1) are trained on data D_j^{-v}.

(2) Nonnegative binomial likelihood maximization (MLSL-NNloglik): For $1 \le j \le k$,

$$(\alpha_{j1}, \ldots, \alpha_{jm}) = \arg\max \sum_{v=1}^{V} \sum_{\boldsymbol{x}_s \in \mathcal{D}_j^v} [Y_{js} \log \widehat{Y}_{js}(\boldsymbol{x}_s) + (1 - Y_{js}) \log(1 - \widehat{Y}_{js}(\boldsymbol{x}_s))],$$

subject to $\alpha_{j\ell} \ge 0$ $(1 \le \ell \le h)$.

(3) Negative binomial log-likelihood minimization on the logistic scale using convex combination of weights (MLSL-CC_nloglik): For $1 \le j \le k$,

$$(\alpha_{j1}, \ldots, \alpha_{jm}) = \arg\min \sum_{v=1}^{V} \sum_{\boldsymbol{x}_s \in \mathcal{D}_j^v} [-Y_{js} \log \widehat{Y}_{js}(\boldsymbol{x}_s) + (Y_{js} - 1) \log(1 - \widehat{Y}_{js}(\boldsymbol{x}_s))],$$

subject to $\alpha_{j\ell} \ge 0$ $(1 \le \ell \le h)$ and $\sum_{\ell=1}^{h} \alpha_{j\ell} = 1$.

(4) Area under the ROC (receiver operating characteristic) curve maximization (MLSL-AUC): For $1 \le j \le k$,

$$(\alpha_{j1}, \ldots, \alpha_{jm}) = \arg \max_{\alpha_{j1},\ldots,\alpha_{jm}} \mathrm{AUC}(\boldsymbol{\mathcal{C}}_j^{\mathrm{sup}}),$$

where AUC stands for the area under the ROC curve computed based on predictions $\boldsymbol{\mathcal{C}}_j^{\mathrm{sup}} = \{\mathcal{C}_j^{\mathrm{sup}}(\boldsymbol{x}_s) : \boldsymbol{x}_s \in \mathcal{D}_j\}$. For more on ROC and AUC, see [Metz 1978; Swets 1973; Fawcett 2006].

In [van der Laan et al. 2007] it is shown, both theoretically and numerically, that the super learner [van der Laan et al. 2007] for single-label classification yields a result that is at least as good as that obtained from any of the base learners. As a result, the ensemble binary classifier $\mathcal{C}_j^{\mathrm{sup}}$ $(1 \le j \le k)$ should produce predictions that are at least as good as the binary predictions from the one-vs-all binary relevance method. Thus, following the proposed MLSL algorithm and combining the ensemble binary outputs to form multilabel predictions, we expect to see an improvement in the performance of the proposed MLSL method.

We assess the performances of the proposed MLSL method and the benchmarks based on the following commonly used multilabel performance metrics: Hamming loss, accuracy, precision, recall, F-measure, and subset accuracy. Definitions and more details of these performance measures can be found in [Herrera et al. 2016]. Ten 10-fold cross-validation was used when computing the performance metrics, in addition to the 10-fold cross-validation algorithm applied to choosing the optimal weights in (3-2).

4.1. ***Data.*** We selected three open-source data sets for our real data analysis, namely *emotions* [Trohidis et al. 2011], *birds* [Briggs et al. 2013], and *scene* [Boutell et al. 2004]. Our choice of these data sets is a result of three considerations. First, we focused our attention on data sets that are accessible online and well-known to the field of multilabel classification, so that researchers and practitioners in this area can easily reference our results in comparison to existing literature as well as future research. Second, we chose data sets from diverse real-world applications, with each data set initially collected to answer a different research question. Third, to the best of our efforts, we included data sets that have distinct multilabel characteristics.

Some details of the three data sets are as follows: the *emotions* data set models the relationship between 593 song clips and six kinds of emotions each song clip may evoke; the *birds* data set focuses on identifying which bird species (out of 19) are present in each of the 645 audio clips recorded in forests; the *scene* data set associates each of the 2407 photographs by one or more of the six scenery labels that the photo may capture.

We present some characteristic metrics of the three data sets in Table 1. Among these statistics, cardinality, density, and highest label frequency represent label distribution; diversity, maximum imbalance ratio (MaxIR), mean imbalance ratio (MeanIR), and score of concurrence among imbalanced labels (SCUMBLE)

	emotions	*birds*	*scene*
#instances	593	645	2407
#labels	6	19	6
#attributes	78	279	300
cardinality	1.87	1.01	1.07
density	0.31	0.05	0.18
highest label frequency	81	194	405
diversity	4	73	3
MaxIR	1.78	17.17	1.46
MeanIR	1.58	5.41	1.25
SCUMBLE	0.01	0.03	0.00

Table 1. Characteristic metrics for datasets *emotions*, *birds* and *scene*.

reveal the degree of imbalance in the data, and a larger value in these measures indicates a more imbalanced label structure and higher difficulty level for the task of classification.

From Table 1 we can see that *scene* is the largest data set with 2407 instances and 300 attributes, but it is the least imbalanced data set. In contrast, *birds*, a smaller data set than *scene*, is much more imbalanced than either *scene* or *emotions*. Compared to *scene* and *birds*, the *emotions* data set is the smallest data set and is partially balanced, with all of its imbalance measures, including MaxIR, MeanIR and SCUMBLE, falling between those of *scene* and *birds*.

4.2. *Results.* In multilabel classification, one often considers performance metrics such as Hamming loss, accuracy, precision, F-measure, recall, and subset accuracy [Herrera et al. 2016]. In particular, F-measure is a trade-off between precision and recall. In Tables 2–7 we summarize the results of these measures after fitting our proposed MLSL model and the benchmark models based on each of the three real data sets. We considered two ways of selecting the base learners in the proposed algorithm. In the first case, we only chose simple binary classifiers for the binary relevance (BR) method. There are four such benchmark models under consideration, i.e., logistic regression (BR-GLM), linear discriminant analysis (BR-LDA), k-nearest neighbor (BR-kNN), and support vector machines (BR-SVM). In the second scenario, in addition to the previously listed simple base learners we also included two machine learning binary classifiers when fitting the binary relevance model, i.e., random forest (BR-RF) and gradient decent (BR-GD). These more powerful benchmark methods are anticipated to yield more accurate multilabel classification results at the expense of higher computational cost. We are interested in investigating how our proposed MLSL method works with or without more complex base learners from different aspects. The R package "SuperLearner" [Polley et al.

	binary relevance (BR)				MLSL			
	GLM	LDA	kNN	SVM	NNLS	NN	CC	AUC
Hamming loss	0.215	0.208	0.273	0.181	0.180	0.179	**0.178**	0.181
accuracy	0.785	0.792	0.727	0.819	0.820	0.821	**0.822**	0.819
F-measure	0.674	0.686	0.546	0.725	0.725	0.726	**0.727**	0.722
precision	0.678	0.699	0.573	0.763	0.766	0.766	**0.767**	0.761
recall	0.673	0.675	0.523	0.691	0.690	0.691	**0.692**	0.689
subset accuracy	0.236	0.259	0.183	**0.323**	0.315	0.317	0.320	0.310
computation (min)	0.002	0.001	0.000	0.012	0.158	0.158	0.158	0.421

Table 2. Results for the *emotions* data set using four base learners. Here and in the following tables the column headers "NN" and "CC" stand for "NNlogik" and "CC_nloglik".

	binary relevance (BR)						MLSL			
	GLM	LDA	kNN	SVM	RF	GD	NNLS	NN	CC	AUC
Hamming loss	0.215	0.208	0.273	0.181	0.178	0.200	0.178	**0.176**	0.177	0.178
accuracy	0.785	0.792	0.727	0.819	0.822	0.800	0.822	**0.824**	0.823	0.822
F-measure	0.674	0.686	0.546	0.725	0.731	0.688	0.732	**0.738**	0.734	0.729
precision	0.678	0.699	0.573	0.763	0.761	0.704	0.769	**0.774**	0.772	0.769
recall	0.673	0.675	0.523	0.691	0.704	0.674	0.699	**0.706**	0.701	0.694
subset accuracy	0.236	0.259	0.183	0.323	**0.331**	0.280	0.317	0.321	0.317	0.315
computation (min)	0.002	0.001	0.001	0.013	0.165	0.191	3.666	3.667	3.667	4.145

Table 3. Results for the *emotions* data set using six base learners.

	binary relevance (BR)				MLSL			
	GLM	LDA	kNN	SVM	NNLS	NN	CC	AUC
Hamming loss	0.128	0.082	0.057	0.043	0.042	0.042	**0.041**	0.056
accuracy	0.872	0.918	0.943	0.957	0.958	0.958	**0.959**	0.944
F-measure	0.327	0.463	0.259	0.698	0.682	0.691	**0.701**	0.516
precision	0.234	0.386	0.354	0.747	0.736	0.752	**0.763**	0.505
recall	0.552	0.588	0.206	**0.659**	0.639	0.643	0.652	0.534
subset accuracy	0.319	0.427	0.468	0.523	0.532	0.536	**0.538**	0.485
computation (min)	0.127	0.030	0.006	0.107	2.765	2.768	2.767	3.139

Table 4. Results for the *birds* data set using four base learners.

2018] was used in the implementation process of MLSL to realize Step 4 of the proposed algorithm. In Tables 2–7 we highlighted the value of performance metric, up to three decimal places, corresponding to the best performance given each criterion. In addition, we also reported the computation time in minutes for fitting each model. Recall that 10-fold cross-validation (CV) was used when determining the coefficients of MLSL in (3-2). So, MLSL essentially fit each base learner ten times, which is reflected in the total running time of MLSL.

The numerical results suggest that the proposed MLSL algorithm is quite competitive compared to the benchmarks based on almost all performance metrics.

	binary relevance (BR)						MLSL			
	GLM	LDA	kNN	SVM	RF	GD	NNLS	NN	CC	AUC
Hamming loss	0.128	0.082	0.057	0.043	0.040	0.042	0.039	**0.038**	**0.038**	0.039
accuracy	0.872	0.918	0.943	0.957	0.960	0.958	0.961	**0.962**	**0.962**	0.961
F-measure	0.327	0.463	0.259	0.698	0.711	0.627	0.713	**0.718**	**0.718**	0.715
precision	0.234	0.386	0.354	0.747	**0.893**	0.763	0.864	0.854	0.853	0.851
recall	0.552	0.588	0.206	**0.659**	0.594	0.535	0.609	0.622	0.622	0.620
subset accuracy	0.319	0.427	0.468	0.523	0.530	0.525	0.547	0.554	**0.555**	0.548
computation (min)	0.118	0.027	0.005	0.097	1.138	1.256	26.03	26.04	26.03	26.94

Table 5. Results for the *birds* data set using six base learners.

	binary relevance (BR)				MLSL			
	GLM	LDA	kNN	SVM	NNLS	NN	CC	AUC
Hamming loss	0.135	0.112	0.093	0.074	0.073	0.073	0.073	**0.074**
accuracy	0.865	0.888	0.907	0.926	**0.927**	**0.927**	**0.927**	0.926
F-measure	0.720	0.780	0.773	0.862	**0.865**	0.862	0.862	0.861
precision	0.677	0.760	0.778	0.861	**0.868**	0.864	0.864	0.865
recall	0.769	0.802	0.769	**0.864**	0.862	0.860	0.860	0.857
subset accuracy	0.443	0.507	0.645	0.664	0.671	**0.672**	**0.672**	0.668
computation (min)	0.104	0.029	0.028	0.225	3.751	3.754	3.753	4.130

Table 6. Results for the *scene* data set using four base learners.

	binary relevance (BR)						MLSL			
	GLM	LDA	kNN	SVM	RF	GD	NNLS	NN	CC	AUC
Hamming loss	0.135	0.112	0.093	0.074	0.083	0.074	**0.069**	**0.069**	**0.069**	0.070
accuracy	0.865	0.888	0.907	0.926	0.917	0.926	**0.931**	**0.931**	**0.931**	0.930
F-measure	0.720	0.780	0.773	0.862	**0.903**	0.874	0.885	0.884	0.884	0.889
precision	0.677	0.760	0.778	0.861	**0.914**	0.871	0.890	0.890	0.890	0.897
recall	0.769	0.802	0.769	0.864	**0.892**	0.877	0.880	0.879	0.879	0.882
subset accuracy	0.443	0.507	0.645	0.664	0.578	0.656	0.676	**0.678**	0.677	0.670
computation (min)	0.098	0.027	0.025	0.198	2.843	2.490	55.35	55.35	55.35	55.95

Table 7. Results for the *scene* data set using six base learners.

Through the binary relevance approach, a given multilabel data set is converted to multiple binary data sets, revealing differences among labels. As a result, each individual classification method, regardless of its complexity, is unlikely to perform well on all of the transformed single-label data sets. By introducing an ensemble classifier that incorporates a diverse group of base learners, the MLSL method increases the chance for each instance to be predicted correctly via the ensemble of different base learners [van der Laan et al. 2007]. When including two more complex base learners, i.e., BR-RF and BR-GD, the performance of MLSL (see Tables 3, 5, and 7) showed further improvement compared to the results based on four simple learners. In theory, one would always expect better performance of MLSL if a larger number of base learners are considered.

Our numerical investigations also revealed that the computation time for finding the optimal weights in (3-2) is negligible compared to the cost of fitting each base learner. As a result, the running time of MLSL is driven by the complexity of each base learner, the total number of base learners, and the number of folds in the cross-validation algorithm. Since 10-fold CV was used in finding the optimal coefficients in (3-2), the computation time of MLSL is approximately equal to ten times the total time for fitting all base learners. (The running time of MLSL would be roughly cut by half, if one uses 5-fold CV, instead of 10-fold CV.) As shown in the tables, the total running time increased significantly when we introduced complex base learners (BR-RF and BR-GD) to the algorithm. In practice, there is a trade-off between performance and computational cost. It is common to consider a few to a dozen base learners for real data implementation.

5. Future work

As noted in Section 2, one of the drawbacks of the binary-relevance multilabel classification method is its lack of consideration of label correlations. Since our proposed multilabel super learner is built upon the one-vs-all binary relevance algorithm, it suffers from the same shortcoming. In practice, there are a few existing methods to account for label correlations in multilabel classification. For our future work, we are interested in exploring the possibility of implementing classifier chain [Read et al. 2011], an algorithm that takes into consideration label correlations, to MLSL. We anticipate that combining classifier chain with MLSL would further improve the performance of multilabel classification.

Acknowledgments

We would like to thank the two anonymous reviewers for their constructive feedback that helped improve our original manuscript significantly.

References

[Boutell et al. 2004] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification", *Pattern Recog.* **37**:9 (2004), 1757–1771.

[Breiman 1996] L. Breiman, "Stacked regressions", *Mach. Learn.* **24**:1 (1996), 49–64. Zbl

[Briggs et al. 2013] F. Briggs, Y. Huang, R. Raich, K. Eftaxias, Z. Lei, W. Cukierski, S. F. Hadley, A. Hadley, M. Betts, X. Z. Fern, J. Irvine, L. Neal, A. Thomas, G. Fodor, G. Tsoumakas, H. W. Ng, T. N. T. Nguyen, H. Huttunen, P. Ruusuvuori, T. Manninen, A. Diment, T. Virtanen, J. Marzat, J. Defretin, D. Callender, C. Hurlburt, K. Larrey, and M. Milakov, "The 9th annual MLSP competition: new methods for acoustic classification of multiple simultaneous bird species in a noisy environment", art. id. 34 in *IEEE International Workshop on Machine Learning for Signal Processing* (Southampton, UK, 2013), IEEE, Piscataway, NJ, 2013.

[Charte et al. 2015] F. Charte, A. J. Rivera, M. del Jesus, and F. Herrera, "QUINTA: a question tagging assistant to improve the answering ratio in electronic forums", pp. 255–260 in *IEEE International*

Conference on Computer as a Tool (*EUROCON*) (Salamanca, Spain, 2015), edited by J. Haase et al., IEEE, Piscataway, NJ, 2015.

[Cheng and Hüllermeier 2009] W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for multilabel classification", *Mach. Learn.* **76**:2-3 (2009), 211–225.

[Cortes and Vapnik 1995] C. Cortes and V. Vapnik, "Support-vector networks", *Mach. Learn.* **20**:3 (1995), 273–297. Zbl

[Cover and Hart 1967] T. Cover and P. Hart, "Nearest neighbor pattern classification", *IEEE Trans. Info. Theory* **13**:1 (1967), 21–27. Zbl

[Cox 1958] D. R. Cox, "The regression analysis of binary sequences", *J. Roy. Stat. Soc. Ser. B* **20**:2 (1958), 215–242. Zbl

[Crammer et al. 2007] K. Crammer, M. Dredze, K. Ganchev, P. P. Talukdar, and S. Carroll, "Automatic code assignment to medical text", pp. 129–136 in *Proceedings of the workshop on BioNLP 2007: biological, translational, and clinical language processing* (Prague, 2007), Assoc. Comput. Linguistics, Stroudsburg, PA, 2007.

[Diplaris et al. 2005] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, "Protein classification with multiple algorithms", pp. 448–456 in *Advances in informatics* (Volas, Greece, 2005), edited by P. Bozanis and E. N. Houstis, Lecture Notes in Comput. Sci. **3746**, Springer, 2005.

[Duygulu et al. 2002] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: learning a lexicon for a fixed image vocabulary", pp. 97–112 in *Computer vision: ECCV 2002* (Copenhagen, 2002), edited by A. Heyden et al., Lecture Notes in Comput. Sci. **2353**, Springer, 2002. Zbl

[Elisseeff and Weston 2001] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification", pp. 681–687 in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic* (Vancouver, 2001), edited by T. G. Dietterich et al., MIT Press, Cambridge, MA, 2001.

[Fawcett 2006] T. Fawcett, "An introduction to ROC analysis", *Pattern Recog. Lett.* **27**:8 (2006), 861–874.

[Freund et al. 1997] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth, "Using and combining predictors that specialize", pp. 334–343 in *Proceedings of the 29th annual ACM Symposium on Theory of Computing* (El Paso, TX, 1997), ACM, New York, 1997. Zbl

[Fürnkranz et al. 2008] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking", *Mach. Learn.* **73**:2 (2008), 133–153.

[Gibaja and Ventura 2015] E. Gibaja and S. Ventura, "A tutorial on multilabel learning", *ACM Comput. Surv.* **47**:3 (2015), art. id. 52.

[Hansen 1998] J. V. Hansen, "Combining predictors: some old methods and a new method", pp. 14–16 in *Proceedings of the Joint Conference on Information Sciences (JCIS '98), II* (Research Triangle Park, NC, 1998), edited by P. P. Wang, Assoc. Intelligent Machinery, Durham, NC, 1998.

[Herrera et al. 2016] F. Herrera, F. Charte, A. J. Rivera, and M. J. del Jesus, *Multilabel classification: problem analysis, metrics and techniques*, Springer, 2016. MR

[Hüllermeier et al. 2008] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences", *Artificial Intelligence* **172**:16-17 (2008), 1897–1916. MR Zbl

[Katakis et al. 2008] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion", pp. 75–84 in *Proceedings of the ECML/PKDD Discovery Challenge* (Antwerp, 2008), ECML/PKDD, 2008.

[Klimt and Yang 2004] B. Klimt and Y. Yang, "The Enron corpus: a new dataset for email classification research", pp. 217–226 in *Machine learning: ECML 2004* (Pisa, 2004), edited by J.-F. Boulicaut et al., Lecture Notes in Comput. Sci. **3201**, Springer, 2004. Zbl

[van der Laan and Dudoit 2003] M. J. van der Laan and S. Dudoit, "Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: finite sample oracle inequalities and examples", working paper 130, Univ. California, Berkeley, Division of Biostatistics, 2003, available at https://biostats.bepress.com/ucbbiostat/paper130/.

[van der Laan et al. 2006] M. J. van der Laan, S. Dudoit, and A. W. van der Vaart, "The cross-validated adaptive epsilon-net estimator", *Statist. Decisions* **24**:3 (2006), 373–395. MR Zbl

[van der Laan et al. 2007] M. J. van der Laan, E. C. Polley, and A. E. Hubbard, "Super Learner", *Stat. Appl. Genet. Mol. Biol.* **6**:1 (2007), art. id. 25. MR Zbl

[Lewis et al. 2004] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: a new benchmark collection for text categorization research", *J. Mach. Learn. Res.* **5** (2004), 361–397.

[McCallum 1999] A. K. McCallum, "Multi-label text classification with a mixture model trained by EM", preprint, 1999, available at https://tinyurl.com/akmccall.

[Metz 1978] C. E. Metz, "Basic principles of ROC analysis", *Semin. Nuclear Medicine* **8**:4 (1978), 283–298.

[Polley et al. 2018] E. Polley, E. LeDell, C. Kennedy, S. Lendle, and M. van der Laan, "SuperLearner", 2018, available at https://github.com/ecpolley/SuperLearner. R package.

[Read et al. 2011] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification", *Mach. Learn.* **85**:3 (2011), 333–359. MR

[Sobol-Shikler and Robinson 2010] T. Sobol-Shikler and P. Robinson, "Classification of complex information: inference of co-occurring affective states from their expressions in speech", *IEEE Trans. Pattern Anal. Mach. Intell.* **32**:7 (2010), 1284–1297.

[Sriram et al. 2010] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in Twitter to improve information filtering", pp. 841–842 in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Geneva, 2010), edited by H.-H. Chen et al., ACM, New York, 2010.

[Swets 1973] J. A. Swets, "The relative operating characteristic in psychology", *Science* **182**:4116 (1973), 990–1000.

[Trohidis et al. 2011] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music by emotion", *EURASIP J. Audio Speech Music Proc.* **2011** (2011), art. id. 4.

[Tsoumakas and Katakis 2007] G. Tsoumakas and I. Katakis, "Multi-label classification: an overview", *Int. J. Data Warehousing Mining* **3**:3 (2007), 1–13.

[Turnbull et al. 2008] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects", *IEEE Trans. Audio Speech Lang. Process.* **16**:2 (2008), 467–476.

[Wolpert 1992] D. H. Wolpert, "Stacked generalization", *Neural Netw.* **5**:2 (1992), 241–259.

[Zhang and Zhou 2007] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning", *Pattern Recog.* **40**:7 (2007), 2038–2048. Zbl

[Zhang and Zhou 2014] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms", *IEEE Trans. Knowledge Data Eng.* **26**:8 (2014), 1819–1837.

Received: 2018-10-25 Revised: 2019-03-18 Accepted: 2019-03-30

ywu3@wellesley.edu *Department of Mathematics, Wellesley College, Wellesley, MA, United States*

qwang@wellesley.edu *Department of Mathematics, Wellesley College, Wellesley, MA, United States*

involve

msp.org/involve

INVOLVE YOUR STUDENTS IN RESEARCH

Involve showcases and encourages high-quality mathematical research involving students from all academic levels. The editorial board consists of mathematical scientists committed to nurturing student participation in research. Bridging the gap between the extremes of purely undergraduate research journals and mainstream research journals, *Involve* provides a venue to mathematicians wishing to encourage the creative involvement of students.

See inside back cover or msp.org/involve for submission instructions. The subscription price for 2019 is US $195/year for the electronic version, and $260/year (+$35, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues and changes of subscriber address should be sent to MSP.

Involve (ISSN 1944-4184 electronic, 1944-4176 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

Involve peer review and production are managed by EditFlow® from Mathematical Sciences Publishers.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

involve
2019 vol. 12 no. 6