On the Stratification of Multi-label Data

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas

Dept. of Informatics,
Aristotle University of Thessaloniki,
Thessaloniki 54124, Greece
{sechidis,greg,vlahavas}@csd.auth.gr

Abstract. Stratified sampling is a sampling method that takes into account the existence of disjoint groups within a population and produces samples where the proportion of these groups is maintained. In single-label classification tasks, groups are differentiated based on the value of the target variable. In multi-label learning tasks, however, where there are multiple target variables, it is not clear how stratified sampling could/should be performed. This paper investigates stratification in the multi-label data context. It considers two stratification methods for multi-label data and empirically compares them along with random sampling on a number of datasets and based on a number of evaluation criteria. The results reveal some interesting conclusions with respect to the utility of each method for particular types of multi-label datasets.

1 Introduction

Experiments are an important aspect of machine learning research [14,7]. In supervised learning, experiments typically involve a first step of distributing the examples of a dataset into two or more disjoint subsets. When training data abound, the *holdout* method is used to distribute the examples into a training and a test set, and sometimes also into a validation set. When training data are limited, *cross-validation* is used, which starts by splitting the dataset into a number of disjoint subsets of approximately equal size.

In classification tasks, the *stratified* version of these two methods is typically used, which splits a dataset so that the proportion of examples of each class in each subset is approximately equal to that in the complete dataset. Stratification has been found to improve upon standard cross-validation both in terms of bias and variance [13].

To the best of our knowledge, what stratification means for multi-label data [23] and how it can be accomplished has not been addressed in the literature. Papers conducting experiments on multi-label data use either predetermined train/test splits that come with a dataset or the random version of the holdout and cross-validation methods. Whether this version is the best that one can do in terms of variance and/or bias of estimate has not been investigated.

Furthermore, random distribution of multi-label training examples into subsets suffers from the following practical problem: it can lead to test subsets lacking even

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 145–158, 2011.

just one positive example of a rare label, which in turn causes calculation problems for a number of multi-label evaluation measures. The typical way these problems get by-passed in the literature is through complete removal of rare labels. This, however, implies that the performance of the learning systems on rare labels is unimportant, which is seldom true. As an example consider that a multi-label learner is used for probabilistic indexing of a large multimedia collection, given a small annotated sample according to a multimedia ontology. Avoiding the evaluation of the multi-label learner for rare concepts of the ontology, implies that we should not allow users to query the collection with such concepts, as the information retrieval performance level of the indexing system for these concepts would be uncertain. This limits the usefulness of the indexing system.

The above issues motivated us to investigate in this paper the concept of stratification in the context of multi-label data. Section 2 considers two interpretations of multi-label stratification. The first one is based on the distinct labelsets that are present in the dataset, while the second one considers each label independently of the rest. Section 3 proposes an algorithm for stratified sampling of multi-label data according to the second interpretation. Section 4 presents empirical results comparing the two multi-label sampling approaches as well as random sampling on several datasets in terms of a number of evaluation criteria. Results reveal some interesting relationships between the utility of each method and particular types of multi-label datasets that can help researchers and practitioners improve the robustness of their experiments. Section 5 presents the conclusions of this work and our future plans on this topic.

2 Stratifying Multi-Label Data

Stratified sampling is a sampling method that takes into account the existence of disjoint groups within a population and produces samples where the proportion of these groups is maintained. In single-label classification tasks, groups are differentiated based on the value of the target variable.

In multi-label data [23], groups could be formed based on the different *combinations of labels* (labelsets) that characterize the training examples. The number of distinct labelsets in a multi-label dataset with m examples and q labels is upper bounded by $\min(m, 2^q)$. Usually this bound equals m, because in most applications q is not very small and as a result 2^q is a very large number. Table 1 shows that, for a variety of multi-label datasets, the number of distinct labelsets is often quite large and sometimes close to the number of examples. In such cases, this *strict* interpretation of stratified sampling for multi-label data is impractical for performing k-fold cross-validation or holdout experiments, as most groups would consist of just a single example. Table 1 is actually sorted in ascending order of the ratio between distinct labelsets and number of examples and accordingly in descending order of average examples per distinct labelset. Notice that in the last two datasets, the average number of examples per labelset is 1 (rounded).

We further consider a more relaxed interpretation of stratified sampling for multi-label data, which sets as a goal the maintenance of the distribution of positive and negative examples of each label. This interpretation views each

Table 1. A variety of multi-label datasets and their statistics: number of labels, examples, distinct labelsets and distinct labelsets per example, along with the minimum, average and maximum number of examples per labelset and label

dataset	labels	examples	label sets	$\frac{\text{labelsets}}{\text{examples}}$	examples per labelset			examples per label		
					min	avg	max	min	avg	max
Scene [1]	6	2407	15	0.01	1	160	405	364	431	533
Emotions [21]	6	593	27	0.05	1	22	81	148	185	264
TMC2007 [20]	22	28596	1341	0.05	1	21	2486	441	2805	16173
Genbase [6]	27	662	32	0.05	1	21	170	1	31	171
Yeast [9]	14	2417	198	0.08	1	12	237	34	731	1816
Medical[1]	45	978	94	0.10	1	10	155	1	27	266
Mediamill [19]	101	43907	6555	0.15	1	7	2363	31	1902	33869
Bookmarks [12]	208	87856	18716	0.21	1	5	6087	300	857	6772
Bibtex [12]	159	7395	2856	0.39	1	3	471	51	112	1042
Enron[2]	53	1702	753	0.44	1	2	163	1	108	913
Corel5k [8]	374	5000	3175	0.64	1	2	55	1	47	1120
ImageCLEF2010 [16]	93	8000	7366	0.92	1	1	32	12	1038	7484
Delicious [22]	983	16105	15806	0.98	1	1	19	21	312	6495

label independently. However, note that we cannot simply apply stratification independently for each label, as this would lead to different disjoint subsets of the data for each label. Such datasets are unsuitable for evaluating multi-label learning algorithms, with the exception of the simple binary relevance approach. Even this approach, however, could only be evaluated using measures that can be calculated using independent computations for each label, such as Hamming loss and macro-averaged precision, recall and F_1.

Achieving this kind of stratification when setting up k-fold cross-validation or holdout experiments on multi-label data is meaningful, because most labels in multi-label domains are characterized by class imbalance [11,3]. The last three columns of Table 1 show the minimum, average and maximum number of examples per label for each dataset. They give an impression of the imbalance ratios found in multi-label domains.

Achieving this kind of stratification is expected to be beneficial, in two directions. Firstly, based on past studies of single-label data, it is expected to improve upon random distribution in terms of estimate bias and variance [13]. Secondly, it will lower the chance of producing subsets with zero positive examples for one or more labels. Such subsets raise issues in the calculation of certain commonly used multi-label evaluation measures, such as the macro-averaged versions of recall, F_1, area under the receiver operating characteristic curve (AUC) and average precision[3], a popular metric in multimedia information retrieval [15].

[1] `http://www.computationalmedicine.org/challenge/index.php`

[2] `http://bailando.sims.berkeley.edu/enron_email.html`

[3] The macro-averaged version of average precision is more commonly called mean average precision (MAP) in information retrieval.

Consider for example the contingency table depicted in Fig. 1, which concerns the predictions for a label. In the case where the test set has none positive examples of this label, then $fn = tp = 0$. Given that recall is defined as $tp/(tp + fn)$, the value of recall for this label is undefined (0/0). If the model is correct and doesn't predict this label for any of the test examples, then $fp = 0$, rendering the value of precision for this label undefined too (0/0), since precision is defined as $tp/(tp + fp)$. F_1 is the harmonic mean of precision and recall, which by definition is rendered undefined when one of precision and recall is undefined. AUC is also undefined, because it depends on the true positive rate, which is equivalent to recall. Average precision considers a ranking of the positively predicted examples of a label based on some confidence value. It is the average of tp precisions, $Precision_i$, $i = 1 \dots tp$, where $Precision_i$ is the precision computed for the positively predicted examples ranked higher or equally to the ith true positive example in this ranking. Since $tp = 0$, average precision is also undefined. Macro-averaging means taking the average of a measure across all labels. If a measure is undefined for one of the labels, its average across all labels is also undefined.

		predicted	
		negative	positive
actual	negative	tn	fp
	positive	fn	tp

Fig. 1. Contingency table concerning the predictions for a label

3 Iterative Stratification

We here propose an algorithm for achieving the relaxed version of multi-label stratification that we discussed in Sect. 2. The pseudo-code is given in Algorithm 1. The input to the algorithm is a multi-label data set, D, annotated with a set of labels $L = \{\lambda_1, ..., \lambda_q\}$, a desired number of subsets k and a desired proportion of examples in each subset, $r_1, \dots r_k$. For example, if we would like to use the algorithm for performing 10-fold CV, then k should be 10 and $r_1 = \dots = r_k$ should be 1/10.

The algorithm starts by calculating the desired number of examples, c_j, at each subset, S_j, by multiplying the number of examples, $|D|$, with the desired proportion for this subset r_j (lines 1-3). It then calculates the desired number of examples of each label λ_i at each subset S_j, c_j^i, by multiplying the number of examples annotated with that label, $|D^i|$, with the desired proportion for this subset r_j (lines 5-9). Note that both c_j and c_j^i will most often be decimal numbers, but this does not affect the proper functioning of the algorithm.

The algorithm is iterative (lines 10-33). It examines one label in each iteration, the one with the fewest remaining examples, denoted l (lines 13-14). The motivation for this greedy key point of the algorithm, is the following: if rare labels are not examined in priority, then they may be distributed in an undesired way, and this cannot be repaired subsequently. On the other hand with frequent

Algorithm 1. IterativeStratification(D,n, $r_1 \dots r_n$)

Input: A set of instances, D, annotated with a set of labels $L = \{\lambda_1, ..., \lambda_q\}$, desired number of subsets k, desired proportion of examples in each subset, $r_1, \dots r_k$ (e.g. in 10-fold CV $k = 10, r_j = 0.1, j = 1 \dots 10$)

Output: Disjoint subsets $S_1, \dots S_k$ of D

```
// Calculate the desired number of examples at each subset
for j ← 1 to k do
    c_j ← |D|r_j
// Calculate the desired number of examples of each label at each subset
for i ← 1 to |L| do
    // Find the examples of each label in the initial set
    D^i ← {(x, Y) ∈ D : λ_i ∈ Y}
    for j ← 1 to k do
        c_j^i ← |D^i|r_j
while |D| > 0 do
    // Find the label with the fewest (but at least one) remaining examples,
    // breaking ties randomly
    D^i ← {(x, Y) ∈ D : λ_i ∈ Y}
    l ← arg min_i (|D^i|) ⋂ {i : D^i ≠ ∅}
    foreach (x, Y) ∈ D^l do
        // Find the subset(s) with the largest number of desired examples for this
        // label, breaking ties by considering the largest number of desired examples,
        // breaking further ties randomly
        M ← arg max_{j=1...k} (c_j^l)
        if |M| = 1 then
            m ∈ M
        else
            M' ← arg max_{j∈M} (c_j)
            if |M'| = 1 then
                m ∈ M'
            else
                m ← randomElementOf(M')
        S_m ← S_m ⋃ {(x, Y)}
        D ← D \ {(x, Y)}
        // Update desired number of examples
        foreach λ_i ∈ Y do
            c_m^i ← c_m^i − 1
        c_m ← c_m − 1
return S_1, ..., S_k
```

labels, we have the chance later on to modify the current distribution towards the desired, due to the availability of more examples.

Then, for each example (x, Y) of this label, the algorithm selects an appropriate subset for distribution. The first criterion for subset selection is the current desired number of examples for this label c_j^l. The subset that maximizes it gets selected (line 19). This is also a greedy choice, since this is actually the subset whose current proportion of examples of label l deviates more from the desired one. In case of ties, then among the tying subsets, the one with the highest number of desired examples c_j get selected (line 23). This is another greedy choice, since this is actually the subset whose proportion of examples irrespectively of labels deviates more from the desired one. Further ties are broken randomly (line 27).

Once the appropriate subset, m, is selected, we add the example (x, Y) to S_m and remove it from D (lines 28-29). In the end of the iteration, we decrement the number of desired examples for each label of this example at subset m, c_m^i, as well as the total number of desired examples for subset m, c_m (lines 30-33).

The algorithm will finish as soon as the original dataset gets empty. This will normally occur after $|L|$ iterations, but it may as well occur in less, due to the examples of certain labels having already been distributed. It may also occur in more, as certain datasets (e.g. mediamill) have examples that are not annotated with any label. One may argue that such examples don't carry any information, but in fact they do carry negative information for each label. These examples are distributed so as to balance the desired number of examples at each subset. This special case of the algorithm is not shown in the pseudocode of Algorithm 1 in order to keep it as legible as possible.

4 Experiments

4.1 Setup

We compare three techniques for sampling without replacement from a multi-label dataset: a) *random* sampling (R), b) stratified sampling based on distinct *labelsets* (L), as discussed in Sect. 2, and c) the *iterative* stratification technique (I), as presented in Sect. 3.

We experiment on the 13 multi-label datasets that are presented in Table 1. We have already commented on certain statistical properties of these datasets in Sect. 2. All of them, apart from ImageCLEF2010, are available for download from the web site of the Mulan library for multi-label learning[4] where their original source is also given. ImageCLEF2010 refers to the visual data released to participants in the photo annotation task of the 2010 edition of the ImageCLEF benchmark [16]. Feature extraction was performed using dense sampling with the SIFT descriptor, followed by codebook construction using k-means clustering with k=4096.

Following a typical machine learning experimental evaluation scenario, we perform 10-fold cross-validation experiments on datasets with up to 15k examples and holdout experiments (2/3 for training and 1/3 for testing) for larger

[4] `http://mulan.sourceforge.net`

datasets. Both types of experiments are repeated 5 times with different random orderings of the training examples. The results in the following sections are averages over these 5 runs.

4.2 Distribution of Labels and Examples

This section compares the three different sampling techniques in terms of a number of statistical properties of the produced subsets. The notation used here, follows that of Sect. 3. In particular, we consider a set of instances, D, annotated with a set of labels, $L = \{\lambda_1, ..., \lambda_q\}$, a desired number, k, of disjoints subsets of D, $S_1, \dots S_k$, and a desired proportion of examples in each of these subsets, $r_1, \dots r_k$. The desired number of examples at each subset S_j is denoted c_j and is equal to $|D|r_j$. The subsets of D and S_j that contain positive examples of label λ_i are denoted D^i and S_j^i respectively.

The *Labels Distribution* (LD) measure, evaluates the extent to which the distribution of positive and negative examples of each label in each subset, follows the distribution of that label in the whole dataset. For each label λ_i, the measure computes the absolute difference between the ratio of positive to negative examples in each subset S_j with the ratio of positive to negative examples in the whole dataset D, and then averages the results across all labels. Formally:

$$LD = \frac{1}{q}\sum_{i=1}^{q}\left(\frac{1}{k}\sum_{j=1}^{k}\left|\frac{\left|S_j^i\right|}{|S_j|-\left|S_j^i\right|}-\frac{\left|D^i\right|}{|D|-|D^i|}\right|\right)$$

The *Examples Distribution* (ED) measure evaluates the extend to which the number of examples of each subset S_j deviates from the desired number of examples of that subset. Formally:

$$ED = \frac{1}{k}\sum_{j=1}^{k}||S_j| - c_j|$$

For the cross-validation experiments we further compute two additional measures that quantify the problem of producing subsets with zero positive examples: a) The number of folds that contain at least one label with zero positive examples (FZ), and b) the number of fold-label pairs with zero positive examples (FLZ).

Table 2 presents the afore-mentioned statistical properties (ED, LD, FZ, FLZ) for the produced subsets in each of the 13 datasets. The best result for each dataset and measure is underlined. The second column of the table presents the ratio of labelsets to examples in each dataset to assist in the interpretation of the results that follows.

We first observe that iterative stratification achieves the best performance in terms of LD in all datasets apart from *Scene*, *Yeast* and *TMC2007*, where the labelsets-based method is better. This shows that the proposed algorithm is generally better than the others in maintaining the ratio of positive to negative examples of each label in each subset.

We further notice that the difference in LD between iterative stratification and the labelsets-based method grows with the ratio of labelsets over examples (2nd column of Table 2). Indeed, when this ratio is small (e.g. ≤ 0.1), the LD of the labelset-based method is close to that of iterative stratification, while when it is large (e.g. ≥ 0.39), it is close to that of random sampling. This behavior is reasonable, since as we discussed in Sect. 2, the larger this ratio is, the more impractical the stratification according to labelsets becomes, as each labelset annotates a very small number of examples (e.g. one or two). This also justifies the fact that the labelsets-based method managed to overcome iterative stratification in terms of LD in *Scene*, *Yeast* and *TMC2007*, as these datasets are characterized by a small ratio of labelsets over examples.

In terms of ED, the labelsets-based and the random sampling methods achieve the best performance in all datasets, while iterative stratification is much worse, with the exception of *Mediamill*. The subsets produced by these methods pay particular attention to the desired number of examples. Iterative stratification on the other hand, trades-off the requirement for constructing subsets with specified number of examples in favor of maintaining the class imbalance ratio of each label. The exception of *Mediamill* is justified from the fact that it contains a number of examples with no positive labels, which are distributed by our algorithm so as to balance the desired number of examples in each subset, as discussed in the last paragraph of Sect. 3.

Finally we observe that iterative stratification produces the smallest value for FZ and FLZ in all datasets. In the *Bibtex* and *ImageCLEF2010* datasets in particular, only iterative stratification leads to subsets with positive examples for all folds. This means that only iterative stratification allows the calculation of the multi-label evaluation measures that were mentioned in Sect. 2. All methods fail to produce subsets with positive examples for all labels in the datasets *Corel5k*, *Enron*, *Medical* and *Genbase*, which contain labels characterized by absolute rarity [11] (notice in Table 1 that the minimum number of examples per label in these datasets is just one). All methods produce subsets with at least one positive example for all labels in the *scene* and *emotions* datasets, where the minimum number of examples per label is large.

4.3 Variance of Estimates

This section examines how the variance of the 10-fold cross-validation estimates for six different multi-label evaluation measures is affected by the different sampling methods. Table 3 shows the six measures, categorized according to the required type of output from a multi-label model (two representative measures from each category). The experiments are based on the 9 out of 13 datasets, where cross-validation was applied.

Two different multi-label classification algorithms are used for performance evaluation: The popular *binary relevance* (BR) approach, which learns a single independent binary model for each label and the *calibrated label ranking* (CLR) method [10], which learns pairwise binary models, one for each pair of labels. Similarly to iterative stratification, BR treats each label independently of the

Table 2. Statistical properties of the produced subsets by a) random sampling, b) labelsets-based stratification, and c) iterative stratification: Labels Distribution (LD), Examples Distribution (ED), folds that contain at least one label with zero positive examples (FZ), and number of fold-label pairs with zero positive examples (FLZ).

dataset	$\frac{\text{labelsets}}{\text{examples}}$	stratification	ED	LD	FZ	FLZ
Scene	0.01	Random	0.42	0.0267	0 out of 10	0 out of 60
		Labelsets	0.42	0.0038	0 out of 10	0 out of 60
		Iterative	2.77	0.0043	0 out of 10	0 out of 60
Emotions	0.05	Random	0.42	0.0973	0 out of 10	0 out of 60
		Labelsets	0.42	0.0316	0 out of 10	0 out of 60
		Iterative	1.80	0.0273	0 out of 10	0 out of 60
Genbase	0.05	Random	0.32	0.0205	10 out of 10	90 out of 270
		Labelsets	0.32	0.0078	10 out of 10	77 out of 270
		Iterative	0.45	0.0055	10 out of 10	74 out of 270
TMC2007	0.05	Random	0.00	0.00250	—	
		Labelsets	0.00	0.00046		
		Iterative	27.4	0.00052		
Yeast	0.08	Random	0.42	0.0862	1 out of 10	1 out of 140
		Labelsets	0.42	0.0273	0 out of 10	0 out of 140
		Iterative	3.53	0.0342	0 out of 10	0 out of 140
Medical	0.10	Random	0.32	0.0110	10 out of 10	203 out of 450
		Labelsets	0.32	0.0059	10 out of 10	179 out of 450
		Iterative	1.47	0.0039	10 out of 10	173 out of 450
Mediamill	0.15	Random	0.33	0.00140	—	
		Labelsets	0.33	0.00056		
		Iterative	0.33	0.00002		
Bookmarks	0.21	Random	0.67	0.00026	—	
		Labelsets	0.67	0.00016		
		Iterative	71.20	0.00002		
Bibtex	0.39	Random	0.50	0.0033	1 out of 10	1 out of 1590
		Labelsets	0.50	0.0027	1 out of 10	1 out of 1590
		Iterative	7.08	0.0006	0 out of 10	0 out of 1590
Enron	0.44	Random	0.32	0.0165	10 out of 10	95 out of 530
		Labelsets	0.32	0.0132	10 out of 10	88 out of 530
		Iterative	2.96	0.0050	10 out of 10	47 out of 530
Corel5k	0.64	Random	0.00	0.0026	10 out of 10	1140 out of 3740
		Labelsets	0.00	0.0023	10 out of 10	1118 out of 3740
		Iterative	4.20	0.0010	10 out of 10	788 out of 3740
ImageCLEF2010	0.92	Random	0.00	0.0324	4 out of 10	4 out of 930
		Labelsets	0.00	0.0265	4 out of 10	4 out of 930
		Iterative	4.48	0.0069	0 out of 10	0 out of 930
Delicious	0.98	Random	0.67	0.00084	—	
		Labelsets	0.67	0.00084		
		Iterative	52.47	0.00034		

Table 3. Six multi-label evaluation measures categorized according to the required type of output from a multi-label model

Measure	Type of Output
Hamming Loss	Bipartition
Subset Accuracy	Bipartition
Coverage	Ranking
Ranking Loss	Ranking
Mean Average Precision	Probabilities
Micro-averaged AUC	Probabilities

rest. Similarly to the labelsets-based stratification, CLR considers label combinations, though only combinations of pairs of labels. Both BR and CLR are instantiated using *random forests* [2] as the binary classification algorithm underneath. We selected this particular algorithm, because it is fast and usually highly accurate without the need of careful tuning.

Following the recommendations in [5], we will discuss the results based on the average ranking of the three different stratification methods. The method that achieves the lowest standard deviation for a particular measure in a particular dataset is given a rank of 1, the next one a rank of 2 and the method with the largest standard deviation is given a rank of 3.

Table 4 shows the mean and standard deviation of the 10-fold cross-validation estimates for the six different measures on the 9 different datasets using BR, along with the average ranks: a) across datasets with small ratio of labelsets over examples (≤ 0.1), b) across datasets with large ratio of labelsets over examples (≥ 0.39), and c) across all datasets.

Looking at the last row of the table, we first notice that random sampling has the worst total average rank in all measures, as its estimates have the highest standard deviation in almost all cases. Iterative stratification has an equal or better overall rank compared to the labelsets-based method, apart from the case of Mean Average Precision. However, these ranks are computed based only on the two datasets where none of the measures was undefined. As already noted, iterative stratification manages to output an estimate in two datasets more than the labelsets-based method and three datasets more than random sampling.

We then look at the average ranks for the upper and lower part of the table that differ in terms of the labelsets over examples ratio. We notice that in the upper part of the table, the labelsets-based method exhibits better rank in all measures, apart from ranking loss. On the other hand, in the lower part of the table, iterative stratification is better than the other methods for all measures. This reinforces the conclusion of the previous section, where we found that the labelsets-based method is more suited to datasets with small ratio of labelsets over examples.

As far as the measures are concerned, we notice that iterative stratification is particularly well suited to ranking loss, independently of the ratio of labelsets over examples. This may seem strange at first sight, as ranking loss is a measure

Table 4. Mean and standard deviation of six multi-label evaluation measures (columns 3 to 8) computed using 10-fold cross validation, the binary relevance algorithm and the three different sampling methods: (R)andom, (L)abelsets, and (I)terative. The first 5 rows correspond to datasets with small ratio of labelsets over examples (≤ 0.1), followed by the average rank of each method. The next 4 rows correspond to datasets with large ratio of labelsets over examples (≥ 0.39), followed by the average rank of each method. The last line presents the average rank for all 9 datasets.

dataset	str.	Hamming Loss	Subset Accuracy	Coverage	Ranking Loss	Mean Average Precision	Micro-averaged AUC
Scene	R	0.0806±0.0078	0.5938±0.0333	0.3542±0.0406	0.0543±0.0070	0.8695±0.0177	0.9612± 0.0064
	L	0.0801±0.0059	0.5959±0.0279	0.3557±0.0421	0.0545±0.0082	0.8696±0.0163	0.9616± 0.0055
	I	0.0805±0.0060	0.5947±0.0261	0.3573±0.0454	0.0549±0.0069	0.8699± 0.0149	0.9613±0.0058
Emotions	R	0.1809±0.0193	0.3247±0.0570	1.6528±0.1424	0.1397±0.0269	0.7568±0.0378	0.8777±0.0197
	L	0.1792±0.0170	0.3299±0.0434	1.6394±0.1221	0.1367±0.0223	0.7603±0.0340	0.8804±0.0193
	I	0.1786±0.0175	0.3270±0.0553	1.6453±0.1308	0.1380±0.0265	0.7616±0.0409	0.8787±0.0222
Genbase	R	0.0024±0.0013	0.9444±0.0295	0.4077±0.2308	0.0030±0.0043	NaN±NaN	0.9952±0.0075
	L	0.0024±0.0012	0.9444±0.0267	0.3995±0.1968	0.0027±0.0038	NaN±NaN	0.9957±0.0063
	I	0.0024±0.0011	0.9438±0.0232	0.3878±0.1808	0.0025±0.0032	NaN±NaN	0.9962±0.0054
Yeast	R	0.1892±0.0070	0.1746±0.0196	6.1383±0.1853	0.1584±0.0108	NaN±NaN	0.8533±0.0093
	L	0.1884±0.0045	0.1762±0.0146	6.1236±0.1082	0.1576±0.0063	0.5358±0.0160	0.8543±0.0062
	I	0.1887±0.0051	0.1757±0.0185	6.1247±0.1219	0.1578±0.0076	0.5429±0.0198	0.8539±0.0071
Medical	R	0.0153±0.0014	0.4531±0.0413	1.5570±0.4584	0.0224±0.0071	NaN±NaN	0.9789±0.0072
	L	0.0151±0.0012	0.4616±0.0351	1.5022±0.3972	0.0217±0.0071	NaN±NaN	0.9798±0.0062
	I	0.0151±0.0014	0.4557±0.0400	1.4497±0.3715	0.0209±0.0069	NaN±NaN	0.9803±0.0058
Average Rank (≤ 0.1)	R	2.9	3	2.6	2.7	2.5	2.8
	L	1.2	1.4	1.6	1.9	1.5	1.4
	I	1.9	1.6	1.8	1.4	2	1.8
Bibtex	R	0.0308±0.0029	0.1015±0.0101	44.9221±1.5618	0.2130±0.0083	NaN±NaN	0.7780±0.0066
	L	0.0315±0.0021	0.1025±0.0064	45.0660±1.0094	0.2140±0.0066	NaN± NaN	0.7682±0.0056
	I	0.0313±0.0017	0.1029±0.0079	44.6686±1.0397	0.2181±0.0067	0.3505±0.0100	0.7691±0.0054
Enron	R	0.0475±0.0020	0.1229±0.0179	12.7126±1.0364	0.0820±0.0084	NaN±NaN	0.9138±0.0068
	L	0.0474±0.0021	0.1245±0.0202	12.6388±0.8306	0.0810±0.0070	NaN±NaN	0.9148±0.0074
	I	0.0474±0.0018	0.1213±0.0197	12.4571±0.6189	0.0797±0.0062	NaN±NaN	0.9165±0.0055
Corel5k	R	0.0094±0.0001	0.0032±0.0023	217.6020±5.5919	0.2717±0.0084	NaN±NaN	0.7821±0.0061
	L	0.0094±0.0001	0.0022±0.0020	217.1086±4.7339	0.2708±0.0086	NaN±NaN	0.7827±0.0060
	I	0.0094±0.0001	0.0026±0.0022	217.4484±4.0884	0.2701±0.0058	NaN±NaN	0.7834±0.0044
Image CLEF2010	R	0.0996±0.0013	0.0003±0.0006	60.5913±0.8638	0.1391±0.0025	NaN±NaN	0.8591±0.0023
	L	0.0997±0.0013	0.0005±0.0007	60.6276±0.8242	0.1392±0.0022	NaN±NaN	0.8589±0.0021
	I	0.0997±0.0008	0.0001±0.0004	60.8236±0.6342	0.1394±0.0021	0.2338±0.0048	0.8588±0.0019
Average Rank (≥ 0.39)	R	2.4	2.3	3.0	2.8		2.8
	L	2.4	2.0	1.8	2.0	-	2.3
	I	1.3	1.8	1.3	1.3		1.0
Average Rank	R	2.7	2.7	2.8	2.7	2.5	2.8
	L	1.7	1.7	1.7	1.9	1.5	1.8
	I	1.6	1.7	1.6	1.3	2.0	1.4

computed across all labels for a given test example. However, it is also true that good ranking loss for BR depends on good probability estimates for each label, which in turn is affected by the distribution of positive and negative examples for each label.

Table 5 shows the mean and standard deviation of the 10-fold cross-validation estimates for the six different measures using CLR on 5 datasets only, those with less than 50 labels, as the quadratic space complexity of CLR resulted into memory shortage problems during our experiments with datasets having more than 50 labels. The last row shows the average rank of the three stratification methods across these datasets.

We here notice that random sampling again has the worst average rank, while the labelsets-based method is better than iterative stratification, even in terms of ranking loss. In this experiment, all datasets have a small ratio of labelesets over examples (≤ 0.1), so according to what we have seen till now, the behavior that we notice is partly expected.

Table 5. Mean and standard deviation of six multi-label evaluation measures (columns 3 to 8) computed using 10-fold cross validation, the calibrated label ranking (CLR) algorithm and the three different sampling methods: (R)andom, (L)abelsets, and (I)terative. The last row shows the average rank of the three stratification methods across the datasets.

dataset	str.	Hamming Loss	Subset Accuracy	Coverage	Ranking Loss	Mean Average Precision	Micro-averaged AUC
Scene	R	0.0807±0.0073	0.5899±0.0329	0.3943±0.0498	0.0624±0.0085	0.8246±0.0242	0.9423±0.0082
	L	0.0802±0.0051	0.5918±0.0237	0.3884±0.0358	0.0613± 0.0070	0.8137±0.0218	0.9427±0.0062
	I	0.0808±0.0061	0.5898±0.0267	0.3891±0.0534	0.0612±0.0082	0.8191±0.0232	0.9423±0.0083
Emotions	R	0.1803±0.0196	0.3264±0.0575	1.6522±0.1311	0.1400±0.0255	0.7240±0.0500	0.8583±0.0225
	L	0.1795±0.0169	0.3272±0.0384	1.6354±0.1260	0.1365± 0.0221	0.7230±0.0371	0.8603±0.0202
	I	0.1782±0.0171	0.3278±0.0553	1.6457±0.1261	0.1382±0.0237	0.7405±0.0432	0.8596±0.0221
Genbase	R	0.0024±0.0013	0.9444±0.0295	0.4743±0.2597	0.0043±0.0049	NaN±NaN	0.9907±0.0083
	L	0.0025±0.0012	0.9432±0.0270	0.4651±0.2040	0.0041±0.0041	NaN±NaN	0.9913±0.0064
	I	0.0024±0.0012	0.9438±0.0238	0.4879±0.1925	0.0045±0.0037	NaN±NaN	0.9898±0.0063
Yeast	R	0.1888±0.0071	0.1756±0.0183	6.0975±0.1887	0.1580±0.0109	NaN±NaN	0.8339±0.0095
	L	0.1883±0.0045	0.1793±0.0143	6.0852±0.1050	0.1570± 0.0062	0.4830±0.0134	0.8346±0.0058
	I	0.1883±0.0052	0.1791±0.0198	6.0895±0.1109	0.1575±0.0069	0.5670±0.0254	0.8344± 0.0059
Medical	R	0.0154±0.0014	0.4497±0.0402	2.2879±0.5601	0.0337±0.0075	NaN±NaN	0.9610±0.0100
	L	0.0150±0.0012	0.4612±0.0359	2.1629±0.3853	0.0319±0.0069	NaN±NaN	0.9631±0.0073
	I	0.0152±0.0013	0.4532±0.0398	2.1774±0.2709	0.0320±0.0052	NaN±NaN	0.9629±0.0052
Average Rank	R	3.0	2.8	2.8	3.0	3.0	2.8
	L	1.1	1.2	1.4	1.4	1.0	1.4
	I	1.9	2.0	1.8	1.6	2.0	1.8

However, if we compare the rankings in Table 5 with the rankings in the upper part of Table 4, which contains exactly the same datasets, we notice that for CLR the benefits of the labesets-based method are larger. We attribute this to the fact that contrary to BR, CLR does consider combinations between pairs of labels, and contrary to iterative stratification, the labelsets-based method distributes examples according to label combinations.

It is also interesting to notice that the measure where iterative stratification exhibits the best performance is again ranking loss, as in the case of BR.

5 Conclusions and Future Work

This paper studied the concept of stratified sampling in a multi-label data context. It presented two different approaches for multi-label stratification and empirically investigated their performance in comparison to random sampling on several datasets and in terms of several criteria.

The main conclusion of this work can be summarized as follows:

- Labelsets-based stratification achieves low variance of performance estimates for datasets where the ratio of distinct labelsets over examples is small, irrespectively of the learning algorithm. It also works particularly well for the calibrated label ranking algorithm. This could be generalizable to other algorithms that take into account label combinations.
- Iterative stratification approach achieves low variance of performance estimates for datasets where the ratio of the distinct labelsets to the number of examples is large. This was observed when the binary relevance approach was used, but could be generalizable to other algorithms, especially those learning a binary model for each label in one of their steps [18,4]. Furthermore,

iterative stratification works particularly well for estimating the ranking loss, independently of algorithm and dataset type. Finally, iterative stratification produces the smallest number of folds and fold-label pairs with zero positive examples and it manages to maintain the ratio of positive to negative examples of each label in each subset.
- Random sampling is consistently worse than the other two methods and should be avoided, contrary to the typical multi-label experimental setup found in the literature.

In this paper we mainly focused on the application of stratified sampling to experimental machine learning, in particular producing subsets for cross-validation and holdout experiments. Apart from the purpose of estimating performance, cross-validation and holdout are also widely used for hyper-parameter selection, model selection and overfitting avoidance (e.g. reduced error pruning of decision trees/rules). The points of this paper are relevant for all these applications of stratified sampling in learning from multi-label data. For example, the stratified sampling approaches discussed in this paper could be used for reduced error pruning of multi-label decision trees [24], for down-sampling without replacement in the ensembles of pruned sets approach [17] and for deciding when to stop the training of a multi-label neural network [25].

In the future, we plan to investigate the construction of a hybrid algorithm that will combine the benefits of both the iterative and the labelsets-based stratification, in order to have a single solution that will work well for any type of dataset, classification algorithm and evaluation measure.

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)
2. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
3. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
4. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning 76(2-3), 211–225 (2009)
5. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
6. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 448–456. Springer, Heidelberg (2005)
7. Drummond, C.: Machine learning as an experimental science (revisited). In: 2006 AAAI Workshop on Evaluation Methods for Machine Learning, pp. 1–5 (2006)
8. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
9. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems, vol. 14 (2002)

10. Fürnkranz, J., Hüllermeier, E., Mencia, E.L., Brinker, K.: Multilabel classification via calibrated label ranking. Machine Learning 73(2), 133–153 (2008)
11. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284 (2009)
12. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proceedings of the ECML/PKDD 2008 Discovery Challenge, Antwerp, Belgium (2008)
13. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI, pp. 1137–1145 (1995)
14. Langley, P.: Machine learning as an experimental science. Machine Learning 3, 5–8 (1988)
15. Nowak, S., Lukashevich, H., Dunker, P., Rüger, S.: Performance measures for multilabel evaluation: a case study in the area of image classification. In: Proceedings of the International Conference on Multimedia Information Retrieval, MIR 2010, pp. 35–44. ACM, New York (2010)
16. Nowak, S., Huiskes, M.: New strategies for image annotation: Overview of the photo annotation task at imageclef 2010. Working Notes of CLEF 2010 (2010)
17. Read, J., Pfahringer, B., Holmes, G.: Multi-label classification using ensembles of pruned sets. In: Proc. 8th IEEE International Conference on Data Mining (ICDM 2008), pp. 995–1000 (2008)
18. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Proc. 20th European Conference on Machine Learning (ECML 2009), pp. 254–269 (2009)
19. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: MULTIMEDIA 2006: Proceedings of the 14th Annual ACM International Conference on Multimedia, pp. 421–430. ACM, New York (2006)
20. Srivastava, A., Zane-Ulman, B.: Discovering recurring anomalies in text reports regarding complex space systems. In: Proc. 2005 IEEE Aerospace Conference, pp. 3853–3862 (2005)
21. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA (2008)
22. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD 2008), pp. 30–44 (2008)
23. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn., ch. 34, pp. 667–685. Springer, Heidelberg (2010)
24. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73(2), 185–214 (2008)
25. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)