Exploring the Pertinence of Distance Functions for Nominal Multi-label Data

Payel Sadhukhan

Exploring the pertinence of distance functions for nominal multi-label data

Payel Sadhukhan[1][0000−0001−7795−3385]

Institute for Advancing Intelligence
TCG CREST, Kolkata, India
payel0410@gmail.com

Abstract. Data with nominal features constitute a good fraction of multi-label datasets. Dealing with high-dimensional, nominal data is different from the handling of data with numeric features. The key reason being – the distance functions which work good on numeric datasets may not function optimally (without returning the true separations of the points) in a nominal feature space. We have further observed that, in a multi-label dataset, an imbalance exists in the distribution nominal features which further aggravates the learning. In this work, we focus to find the suitability of four different distance functions –*euclidean, hamming, jaccard and kulsinski* in a binary-nominal context. Additionally, we also propose and explore an ensemble of two classifiers where one classifier is modelled using jaccard distance and the other is modelled on kulsinski distance. An empirical study involving five binary-nominal datasets, four evaluation metrics and three multi-label classifiers is used to evaluate the pertinence of each distance function and the ensemble. We find that the proposed ensemble gives the best outcome across all but one case.

Keywords: nominal features · multi-label · distance functions · ensemble · jaccard · kulsinski

1 Introduction

Distance function is one of the key aspects for learning any dataset. A distance function should be able to capture the true separations of the feature vectors of a dataset. The quantitative and qualitative structures of a dataset has to be kept into account while choosing a distance function for the same. In most cases, we are biased towards using the euclidean distance, possibly due to its familiarity in our daily lives. But, it may not be the best choice always. To obtain a fruitful learning, we need to look at aspects like dimensionality of features, density, nature of the features and sparsity of the features before choosing the distance function. Multi-label datasets are obtained from several real-world domain. Quite a number of these domain deals with features which are nominal in nature, namely text [1], medical [18] and object detection [20]. Besides the abundance of binary (0 or 1) nominal features, the presence of multiple-way nominal features

(with more than two possible feature values) is also seen. In order to accomplish a proper learning from these data, it is necessary to have a distance function which will aptly capture the dissimilarities and similarities of the nominal feature vectors. In this work, we are particularly interested to explore the pertinence of different distance functions in context of multi-label datasets with binary nominal features. We denote a multi-label as $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{Y}_i), i = 1, 2, \ldots, n\}$ and the label set cardinality as $\mathcal{L}$. Here, $\mathbf{x}_i$ is the i^{th} instance and $\mathcal{Y}_i = \{y_{i1}, y_{i2}, \ldots, y_{i\mathcal{L}}\}$ is the corresponding label assignment for $\mathbf{x}_i$. y_{ik} indicates the k^{th} label membership for the i^{th} instance. If $y_{ik} = 1$ signifies that the k^{th} label is relevant (positive) for $\mathbf{x}_i$ while $y_{ik} = 0$ denotes that the k^{th} label is irrelevant (negative) for $\mathbf{x}_i$. Let the feature space be d-dimensional and $\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$ denotes an instance. In a nominal, multi-label dataset, $x_{ij} = 0$ or $1\ \forall i, j$. It is further seen that for each feature, the 0 count often outnumbers the 1 count. Sometimes, we may notice the opposite also, 1 count outnumbers the 0 count. To be precise, there is often a feature-imbalance which gives rise to a sparsity in the feature vector matrix.

We could find one work which explores the suitability of different distance functions for generic multi-label data (numeric as well as nominal) [6]. We have found the usage of hamming distance function to capture the dissimilarities of the labels (but not features) in a number of works [16]. Apart from a couple of works, we could not really find any significant work which has discussed the occurrence and consequences of these distinctive aspects of nominal multi-label feature vectors [10]. There remains a lot to be done to efficiently tackle the nominal features of the multi-label datasets which is the focus of our work. In this work, we analyze the suitability of each distance function and the proposed ensemble in the nominal, multi-label context. We run three sets of classifiers —i] Binary Relevance (BR) classifier with k-nearest neighbor classifier as the base classifier, ii] Classifier Chain Scheme with k-nearest neighbor classifier as the base classifier and iii] Rakel-D with k-nearest neighbor classifier — to test the efficacy of each function in being able to rightfully deal with the nominal feature vectors. Our goal is to study the intrinsic effect of the distance functions. Hence, we have chosen k-nearest neighbor classifier as the base classifier (as it does not involve any other parameters other than separation of the points) in all three cases. We discuss the extant works in the field of multi-label learning in the next section.

2 Related Works

The study of multi-label learning has gained momentum in the past few decades. The primary reason is the availability of this class of data from several real-world domain. Accordingly, the need for a dedicated learning of such datasets was established. The sincere efforts of the research community has resulted in the formulation of diversified techniques which can be principally classified into i] Problem transformation (PT) approaches and ii] Algorithm Adaptation (AA) approaches [5]. In PT approaches, a certain number of base classifiers are used

to tackle the multi-label classification. A base classifier can be a traditional classifier in its innate form or a tweaked form which can facilitate multi-label learning better. PT approaches can be further classifier into – i) *first-order*, ii) *second-order* or iii) *higher-order* approaches on the basis of the number of labels that are considered together to train the models. In First order approach, a classifier is learned for a label independently of all other labels [18]. In second order methods, the pair-wise correlations of labels are explored by packing the learning of two labels in one classifier [21]. The cumulative learning of all pairs of labels are taken together to ddliver the final output. Capturing the association of three or more labels is the primary focus of higher order approaches [12]. A number of diversified techniques have facilitated higher order label associations through interesting schemes including classifier chains [3], RAkEL [19], random graph ensembles [2] and IBLR-ML+. In a recent work by [11], zero-shot learning is used to facilitate the learning in images. In AA approaches, an existing learning scheme is transformed to learn the multi-label datasets. Several approaches have been made and techniques like k-nearest neighborhood [7], bayesian learning [22], neural network [8] to name a few. In addition to the above described class of methods, researchers have also resorted to the methods of data transformation via feature extraction and selection [17] for enhanced learning of multi-label datasets. Class-imbalance is an important characteristic of multi-label datasets and in recent years a number of schemes have been developed to address this issue via data preprocessing [9], label-correlation [23], cost-sensitive learning [14] and Helinger forests [4]. However, we could not find any specific study on nominal multi-label datasets. In all the works (barring a few) [16] that we have described in the previous paragraphs of this section, we have seen that nominal multi-label datasets are treated in the same way as that of the numeric multi-label datasets. This work seems to be the first one which is carrying out a study focused on this aspect. We explain the motivation of our work using a toy example in the next section.

3 A toy example and the motivation of this work

Let us consider three pairs of nominal features a) $\mathbf{x}_1$, $\mathbf{x}_2$, b) $\mathbf{y}_1$ and $\mathbf{y}_2$ and c) $\mathbf{z}_1$ and $\mathbf{z}_2$. All six feature points are in the binary domain.

- Case 1: $\mathbf{x}_1 = \{0, 1, 1, 0\}$ and $\mathbf{x}_2 = \{1, 0, 1, 0\}$. We can see that $\mathbf{x}_1$ and $\mathbf{x}_2$ varies at exactly two positions and if we were to compute their Euclidean distances, the value would be $\sqrt{2}$.
- Case 2: $\mathbf{y}_1 = \{0, 1, 1, 0, 0, 0, 0\}$ and $\mathbf{y}_2 = \{1, 0, 1, 0, 0, 0, 0\}$. A simple inspection like in the previous case will tell us that $\mathbf{y}_1$ and $\mathbf{y}_2$ also vary at exactly two locations and their euclidean distance is $\sqrt{2}$.
- Case 3: $\mathbf{z}_1 = \{0, 1, 1, 0, 1, 1, 1\}$ and $\mathbf{z}_2 = \{1, 0, 1, 0, 1, 1, 1\}$. A similar computation like the previous two cases will tell us that $\mathbf{z}_1$ and $\mathbf{z}_2$ also vary at exactly two locations and their euclidean distance is $\sqrt{2}$.

Feature vector length of a vector is the number of components present in it. Percentage of positive features of a vector denotes the fraction of feature

Vectors	Dimensionality of features	Percentage of positive features
$\mathbf{x}_1, \mathbf{x}_2$	4	0.500
$\mathbf{y}_1, \mathbf{y}_2$	7	0.286
$\mathbf{z}_1, \mathbf{z}_2$	7	0.714

components which are positive among all the components. In the following table, we show the values of these two parameters of the pair of feature vectors.

Euclidean distance returned $\sqrt{2}$ value in each of the three cases as the vector components varied at exactly two positions in each. And this very popular distance function (widely used in multi-label learning) seems to be thoroughly unaware of two basic properties of the binary-nominal, multi-label datasets. In two of three cases (Case 1 vs Case 2, Case 3), the feature vector lengths varied, but the number of differences were same. Euclidean metric, being unaware of this aspect, gave us the same distance. But we can presume that, given a shorter vector length in Case 1, the difference would be much more significant in Case 1 than that of Case 2 and Case 3. When we inspect Case 2 and Case 3 (where the feature vector lengths are same), there is a significant difference in the percentage of positive features between the two cases. Given that, the implication of the two-vector difference is more in Case 2 than that of Case 3. We need to explore some more distance functions or metrics which will have a higher suitability to capture the true dissimilarities and similarities of the multi-label, binary-nominal feature vectors (in the two given contexts). We will use this example in later parts of this example, wherever we require it.

4 Exploration

We explore the pertinence of each of four distance functions in three contexts. In our first study, we use a state-of-the-art multi-label learner, Classifier Chain [13] where a k-nearest neighborhood classifier is used as the base classifier. We run four different instances of it. A different distance function (described in the previous section) is used in each of these four instances. In a similar fashion, in our second study, we employ Binary Relevance classifier [15] with k-nearest neighborhood classifier as the base classifier. In our third study, we perform the experiments on another state-of-the art approach, RAKEL-d (without overlap) [19] with k-nearest neighborhood classifier as the base classifier.

Classifier Chain (CC) is a higher order approach of multi-label learning where the label correlations are also taken into account. On the contrary, Binary Releavance (BR) classifier is a first-order learning approach, which is unaware of the learning of the remaining labels. The working principle of RAKEL-d is to learn a set of multiclass classifiers obtained through powersets of non-overlapping subsets of labels). Classifier Chain incorporates label correlation by packing the input space of succeeding labels with the outputs of the preceding ones. The competence of these three methods is well established in multi-label domain [24]. The modus operandi of each is significantly different from each other. We select these three distinct and non-overlapping approaches to find the suitability

of the distance functions diversified multi-label contexts. We have used k-nearest neighborhood classifier (as a base classifier) in all three cases as its working protocol is based on distance between the points only. In the following paragraph, we briefly describe our thoughts for selecting these four distance functions.

- *Euclidean distance*: The most widely used distance function and is indicative of the disagreements of the feature vector components. It is used as a baseline.
- *Hamming distance*: It uses a scaling with respect to the number of features. Hence it is aware of the disagreements of the features values as well as the total number of features.
- *Jaccard distance*: It gives zero weightage the zero matches and ignores them in distance computation. It can be employed to tackle the sparsity of the features (class-imbalance of features). Jaccard distance is particularly helpful, when the number of zero matches are considerably high.
- *Kulsinski distance*: This distance function is also useful in tackling class-imbalance of feature, specially when the number of 1's outnumbers the number of zeros. The all-one matches are ignored in kulsinski distance computation.

4.1 The proposed ensemble

We construct an ensemble of classifiers where we consider two classifiers – i]$Classifier_j$ *-jaccard* distance is used for modelling the classifier and prediction of the test points and ii] $Classifier_k$*-kulsinski* distance is used for modelling the classifier and prediction of the test points. We integrate the predictions and scores from these two classifiers to obtain the final classification results.
We compute the pairwise distances of the points using jaccard and kulsinski separately and compute the overall variance in the distances (for each function). If the variance obtained (using a specific distance function) is high, we can say that the function is able to capture the dispersion (as well as separation) of the data in a given space. Hence, for a dataset we are motivated to assign more weightage to the classifier which is trained using a distance function that returned more variance. We assign weights to the predictions from $Classifier_j$ and $Classifier_k$ in accordance with the variances of *jaccard* distances values and *kulsinski* distance values respectively. We integrate the two of them to obtain the final prediction from the ensemble. Let D_j and D_k denote the population mean of the pair-wise distances for a given set of points with respect to jaccard and kulsinski functions respectively. Let $d_j(\cdot,\cdot)$ and $d_k(\cdot,\cdot)$ be the jaccard and kulsinski distance functions. Let $\mathbf{x}_1,\ \mathbf{x}_2,\ldots,\mathbf{x}_n$ be the points in the dataset.

$$D_j = \frac{1}{\binom{n}{2}} \sum_{j=1,\ i\neq j}^{n} \sum_{i=1}^{n} d_j(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

Similarly, we obtain D_k.

$$D_k = \frac{1}{\binom{n}{2}} \sum_{j=1,\ i\neq j}^{n} \sum_{i=1}^{n} d_k(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

We compute the weights w_j and w_k for $Classifier_j$ and $Classifier_k$ from D_j and D_k. We may note that $w_j + w_k = 1$.

$$w_j = \frac{D_j}{D_k + D_j} \tag{3}$$

$$w_k = \frac{D_k}{D_k + D_j} \tag{4}$$

Let y_{ij}^l and y_{ik}^l be the predictions for instance $\mathbf{x}_i$ for some label l from $Classifier_j$ and $Classifier_k$ respectively. Let $scores_{ij}^l$ and $scores_{ik}^l$ be the prediction probabilities of $\mathbf{x}_i$ with respect to the positive class of l from $Classifier_j$ and $Classifier_k$ respectively. Let y_i^l and $scores_i^l$ the final prediction and prediction probability with respect to the positive class of label l for instance $\mathbf{x}_i$.

$$y_i^l = \begin{cases} 1, & \text{if}(y_{ij}^l \times w_j + y_{ik}^l \times w_k) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$scores_i^l = scores_{ij}^l \times w_l + scores_{ik}^l \times w_k \tag{6}$$

We empirically evaluate the suitability of the distance functions and the proposed ensemble by evaluating their performance scores in an experimental study, which is described in the next section.

5 Experimental Study

The empirical study is devoted to evaluate the effectiveness of different distance functions for binary-nominal, multi-label datasets. This section describes the datasets, experimental setup and evaluating metrics involved in the study. We have taken five binary-nominal, multi-label datasets in our study and their statistics are given in Table 1. The datasets are obtained from MULAN[1] and MEKA[2]. In Table 1, *instances*, *inputs* and *labels* indicate the total number of instances, features, and the number of labels respectively. *Type* indicates if the input space is numeric or nominal. *Cardinality* calculates the average number of labels per instance, and Cardinality scaled with respect to the number of labels is reported as *Density*. All the datasets have binary, nominal features. Two multi-label classifiers, Binary Relevance classifier and Classifier Chain are used in our experimental study. In each case, the base classifer is k-nearest neighbor classifier with $k = 5$. We have used *four* metrics *micro* F_1, *macro* F_1, *hamming loss* and *average precision* to evaluate the performance of the classifiers.

We discuss the results and analysis of our experiment in the next section.

[1] http://mulan.sourceforge.net/datasets-mlc.html
[2] http://http://meka.sourceforge.net/

Table 1. Description of datasets

Dataset	Instances	Inputs	Labels	Type	Cardinality	Density
medical	978	144	14	nominal	1.075	0.077
enron	1702	50	24	nominal	3.113	0.130
llog	1460	100	18	nominal	0.851	0.047
corel5k	5000	499	44	nominal	2.241	0.050
slashdot	3782	53	14	nominal	1.134	0.081

6 Results and analysis

We have randomly partitioned each dataset into a training set and a test set. We have performed our experiment in the Hold-out settings where i] the training set and the test set are mutually exclusive and exhaustive ii] the training set and the test set comprises of 50% data instances. The training set is used to model the classifier and the test set is used for prediction. We conduct three studies – each one is devoted to a particular classifier (Binary Relevance, Classifier Chain and Rakel-d). Five sets of outputs are obtained in each study, where the first four sets are dedicated to the use of four distance functions (*euclidean*, *hamming*, *jaccard* and *kulsinski*)and the fifth set corresponds to the output from the *ensemble*. The process is repeated 10 times in each case for each dataset. The mean values obtained on *micro* F_1, *macro* F_1, *hamming loss* and *average precision* are reported in Tables 2-4. Firstly, we analyze the outcomes of the four distance metrics. The scores obtained from Tables 2-4 are in congruence with each other. All the tables indicate that *jaccard* distance function (which is focused on handling the sparsity of features and ignores the zero-zero matchings) is most effective for learning binary-nominal, multi-label datasets. In 49 out of 60 cases ($> 80\%$), the use of *jaccard* distance has given the best results among the four distance metrics *excluding the ensemble*. In most of the cases (27 out of 60), the use of *jaccard* has given more than 50% improvement in performance over *euclidean* and *hamming* distances. In the remaining 11 cases (excluding the ensemble), the best scores are obtained with the use of *kulsinski* distance function which is aware of the one-one abundances (thereby ignores them in distance computation). It is worth noting that the difference in performance between *jaccard* and *kulsinski* is not as high as that of the previous case (*jaccard* versus *euclidean* and *hamming*). *kulsinski* distance function is also focused on handling sparsity of the features (where positive-positive feature matchings outnumber the remaining combination of features). The use of *kulsinski* distance function has given the least perfect scores in case of *Llog* dataset. Overall, *jaccard* distance function has served as the most consistent distance function for learning the binary-nominal multi-label datasets. Usually, in a binary-nominal, multi-label dataset, zeros are in abundance (compared to ones) as features which results in higher number of zero-zero matchings. The zero-zero matchings (ZE) are ignored in computation of *jaccard* distance between two points and that is the likely explanation of its efficiency in binary-nominal, multi-label context. Secondly, we make a comparative study of the multi-label performance delivered by the *four* distance functions and the *ensemble*. The proposed *ensemble* has achieved the best scores in 58 out of 60 cases ($> 96\%$). The only two remain-

ing best cases are obtained by the use of *jaccard* distance on *Enron* dataset in *Classifier Chain* classifier. The ensemble comprises of a *jaccard* distance based classifier and a *kulsinski* distance based classifier. Each one caters to a specific component of feature imbalance – i] zero-zero matchings (ZEs) are ignored in the computation of the *jaccard* distance between the points (thereby taking care of the zero abundance), while ii] one-one matchings are ignored in the computation of the *kulsinski* distance between the points (which takes care of the one-one abundance). The complementary nature of the ensemble components contributes to the betterment of the multi-label performance across all datasets.

Table 2. Micro F_1 and Macro F_1 results for Binary Relevance based on k-nearest neighbor classifier. On *hamming loss*, a lower score is desirable, but on *micro* F_1*, macro* F_1 *and average precision* a higher score is desirable.

	Micro F_1 ↑					Macro F_1 ↑				
Datasets	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble
Enron	0.3495	0.3514	0.5494	0.3897	**0.5176**	0.1973	0.1972	0.2891	0.1789	**0.2615**
Medical	0.6474	0.6402	0.6942	0.6722	**0.7129**	0.4556	0.4650	0.5346	0.5299	**0.5873**
Slashdot	0.1845	0.1774	0.4264	0.4235	**0.4504**	0.0524	0.0522	0.3119	0.3244	**0.3356**
Llog	0.1017	0.1012	0.1266	0.0421	**0.1774**	0.0582	0.0570	**0.0949**	0.0398	**0.1264**
Corel5k	0.0651	0.0646	0.1519	0.1509	**0.1774**	0.0397	0.0406	0.1232	0.1202	**0.1268**
	Hamming Loss ↓					*Average Precision* ↑				
Datasets	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble
Enron	0.3598	0.3606	0.2349	0.3780	**0.1880**	0.2487	0.2500	0.2500	0.2098	**0.3247**
Medical	0.1060	0.1034	0.0906	0.0917	**0.0778**	0.5788	0.5773	0.6174	0.6053	**0.6434**
Slashdot	0.6570	0.6481	0.3203	0.3227	**0.2983**	0.1365	0.1377	0.3279	0.3303	**0.3458**
Llog	0.4670	0.4672	0.3914	0.4525	**0.3129**	0.1043	0.1049	0.1373	0.0903	**0.1654**
Corel5k	0.6149	0.6152	0.4818	0.4834	**0.4789**	0.0923	0.0921	0.0987	0.1230	**0.1421**

Table 3. Micro F_1,Macro F_1, *hamming loss* and *averaage precision* results for Classifier Chain based on k-nearest neighbor classifier. On *hamming loss*, a lower score is desirable, but on *micro* F_1*, macro* F_1 *and average precision* a higher score is desirable.

	Micro F_1 ↑					Macro F_1 ↑				
Datasets	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble
Enron	0.3327	0.3351	**0.5462**	0.4047	0.5132	0.1753	0.1765	**0.2896**	0.1784	0.2608
Medical	0.6610	0.6612	0.6974	0.6778	**0.7097**	0.4771	0.4763	0.5681	0.5277	**0.5828**
Slashdot	0.1756	0.1757	0.4406	0.4304	**0.4558**	0.0521	0.0527	0.3389	0.3392	**0.3474**
Llog	0.1065	0.1036	0.1333	0.0444	**0.1732**	0.0535	0.0536	0.1090	0.0437	**0.1193**
Corel5k	0.0702	0.0721	0.1783	0.1745	**0.1792**	0.0464	0.0501	0.1452	0.1383	**0.1484**
	Hamming Loss ↓					Average Precision ↑				
Datasets	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble
Enron	0.3830	0.3806	0.2451	0.3740	**0.1929**	0.2451	0.2450	0.3019	0.2052	**0.3262**
Medical	0.1242	0.1264	0.1085	0.1227	**0.0829**	0.5609	0.5570	0.5947	0.5539	**0.6456**
Slashdot	0.6559	0.6586	0.3194	0.3223	**0.2949**	0.1276	0.1260	0.3311	0.3332	**0.3503**
Llog	0.4588	0.4585	0.3853	0.4501	**0.3039**	0.1015	0.1014	0.1328	0.0904	**0.1616**
Corel5k	0.6438	0.6425	0.4892	0.4909	**0.4821**	0.0885	0.0897	0.1403	0.1388	**0.1436**

7 Conclusion

In this work, we have explored the use of different distance functions in binary-nominal context (features) of multi-label datasets. Four different distance functions have been used to address the specific characteristics of such datasets, namely – feature imbalance, zero-zero abundance and one-one abundance. The

Table 4. Micro F_1, Macro F_1, *average precision* and *haming loss* results for RAKEL-D classifier based on k-nearest neighbor classifier. On *hamming loss*, a lower score is desirable, but on *micro* F_1*, macro* F_1 *and average precision* a higher score is desirable.

	Micro F_1 ↑					Macro F_1 ↑				
Datasets	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble
Enron	0.6505	0.6494	0.6852	0.6740	**0.7011**	0.4339	0.4503	0.5336	0.5193	**0.5821**
Medical	0.6413	0.6396	0.6865	0.6889	**0.7093**	0.4321	0.4512	0.5318	0.5674	**0.5884**
Slashdot	0.1778	0.1811	0.4331	0.4332	**0.4589**	0.0501	0.0530	0.3144	0.3255	**0.3447**
Llog	0.1083	0.1065	0.1553	0.0746	**0.1863**	0.0548	0.0544	0.1113	0.0535	**0.1329**
Corel5k	0.0718	0.0671	0.1596	0.1537	**0.1715**	0.0418	0.0416	0.1226	0.1191	**0.1301**
	Hamming Loss ↓					Average Precision ↑				
Datasets	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble	Euclidean	Hamming	Jaccard	Kulsinski	Ensemble
Enron	0.1236	0.1261	0.1000	0.1110	**0.0760**	0.5729	0.5570	0.5972	0.5821	**0.6518**
Medical	0.1173	0.1201	0.0942	0.0912	**0.0678**	0.5593	0.5459	0.5923	0.5894	**0.6565**
Slashdot	0.6561	0.6446	0.3219	0.3196	**0.2990**	0.1391	0.1384	0.3284	0.3345	**0.3472**
Llog	0.4684	0.4670	0.3939	0.4506	**0.3113**	0.1015	0.1024	0.1360	0.0911	**0.1631**
Corel5k	0.6188	0.6160	0.4830	0.4839	**0.4801**	0.0926	0.0942	0.1427	0.1406	**0.1438**

outcomes of the study indicate that the feature imbalances do play some role in aggravating the performance of such datasets. The use of *jaccard* and *kulsinski* distance function helps in tackling the feature imbalances. The simulation of the classifiers using these two distance functions is shown to improve the macro F_1 and micro F_1 scores over the others. Our study indicates *jaccard* distance is most effective among the chosen distance function in learning the binary-nominal, multi-label datasets. The results from the empirical study further establishes that the use an ensemble of classifiers – one of which is modelled on *jaccard* and the other is modelled on *kulsinski* improves the learning of such datasets to a significant extent (over the four chosen distance functions). In our future work, we would like to carry out this investigation in nominal, multi-label features with more than two values and also mixed feature space containing both nominal and numeric features.

References

1. Chen, W., Yan, J., Zhang, B., Chen, Z., Yang, Q.: Document transformation for multi-label feature selection in text categorization. In: Seventh IEEE International Conference on Data Mining (ICDM 2007). pp. 451–456. IEEE (2007)
2. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning **76**(2-3), 211–225 (2009)
3. Cheng, W., Hüllermeier, E., Dembczynski, K.J.: Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 279–286 (2010)
4. Daniels, Z., Metaxas, D.: Addressing imbalance in multi-label classification using structured hellinger forests. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
5. Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **4**(6), 411–444 (2014)
6. Gjorgjioski, V., Kocev, D., Džeroski, S.: Comparison of distances for multi-label classification with pcts. In: Proceedings of the Slovenian KDD Conference on Data Mining and Data Warehouses (SiKDD'11). vol. 8 (2011)

7. Kanj, S., Abdallah, F., Denoeux, T., Tout, K.: Editing training data for multi-label classification with the k-nearest neighbor rule. Pattern Analysis and Applications **19**(1), 145–161 (2016)
8. Kurata, G., Xiang, B., Zhou, B.: Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 521–526 (2016)
9. Liu, B., Tsoumakas, G.: Synthetic oversampling of multi-label data based on local label distribution. arXiv preprint arXiv:1905.00609 (2019)
10. Moyano, J.M., Gibaja, E.L., Cios, K.J., Ventura, S.: Review of ensembles of multi-label classifiers: Models, experimental study and prospects. Information Fusion **44**, 33 – 45 (2018)
11. Narayan, S., Gupta, A., Khan, S., Khan, F.S., Shao, L., Shah, M.: Discriminative region-based multi-label zero-shot learning. arXiv preprint arXiv:2108.09301 (2021)
12. Nazmi, S., Yan, X., Homaifar, A., Doucette, E.: Evolving multi-label classification rules by exploiting high-order label correlations. Neurocomputing **417**, 176–186 (2020)
13. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine learning **85**(3), 333 (2011)
14. Sadhukhan, P., Palit, S.: Reverse-nearest neighborhood based oversampling for imbalanced, multi-label datasets. Pattern Recognition Letters **125**, 813 – 820 (2019)
15. Sorower, M.S.: A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis (2010)
16. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: Relieff for multi-label feature selection. In: 2013 Brazilian Conference on Intelligent Systems. pp. 6–11. IEEE (2013)
17. Spolaôr, N., Monard, M.C., Tsoumakas, G., Lee, H.D.: A systematic review of multi-label feature selection and a new method based on label construction. Neurocomputing **180**, 3–15 (2016)
18. Tanaka, E.A., Nozawa, S.R., Macedo, A.A., Baranauskas, J.A.: A multi-label approach using binary relevance and decision trees applied to functional genomics. Journal of Biomedical Informatics **54**, 85–95 (2015)
19. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering **23**(7), 1079–1089 (July 2011)
20. Wang, D., Zhang, S.: Unsupervised person re-identification via multi-label classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10981–10990 (2020)
21. Weng, W., Lin, Y., Wu, S., Li, Y., Kang, Y.: Multi-label learning based on label-specific features and local pairwise label correlation. Neurocomputing **273**, 385–394 (2018)
22. Yan, X., Wu, Q., Sheng, V.S.: A double weighted naive bayes with niching cultural algorithm for multi-label classification. International Journal of Pattern Recognition and Artificial Intelligence **30**(06), 1650013 (2016)
23. Zhang, M.L., Li, Y.K., Yang, H., Liu, X.Y.: Towards class-imbalance aware multi-label learning. IEEE Transactions on Cybernetics (2020)
24. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering **26**(8), 1819–1837 (2013)