A Coupled k-Nearest Neighbor Algorithm for Multi-label Classification

Sonu Kumar , Galgotias University Greater Noida

Abstract

ML-kNN is a well-known algorithm for multi-label classification. Multi-label classification has more frequently used in recent years. Although it is more useful in some cases, ML-kNN has major issues due to the fact that it is a binary relevance classifier which only takes one label every time. In this paper, we proposed a lazy learning approaches to classify an unseen instance on the basis of its k nearest neighbors to solve the multi-label classification problem . We collect different real-word data sets from various domains for the experiment. By introducing the coupled similarity between class labels, the proposed method utilize the correlations between class labels, which overcomes the shortcoming of ML-kNN. Experiments on standard data sets show that our proposed Coupled Multi-Label k Nearest Neighbor algorithm (CML-kNN) reachs heigher performance than some existing multi-label classification algorithms. We believe that nearly utilizing k-nearest neighbors is useful to solve the multi-label problem.

.

1.Introduction

Traditional single-label classification refers an object to only one class, from a set of Q disjoint classes. Multi-label classification is the task of refering an instance simultaneously to one or multiple classes. Multi-label classification tasks are boundless in real-world problems. For example, in text categorization, each document may belong to different predefinedtopics; inbioinformatics, one protein may have many effects on a cell when prognosticating its functional classes. Such tasks are usually designated as multi-label classification problems. Typical examples for multi-label problems are functional genomics ,text categorization, scene clasification ,image classfication .There are different types of Multilabel classification.

These methods are divided into two categories:

(i)Transformation Methods : Problem transformation methods first transform the multi-label learning functions into multiple single-label learning functions which are then handle by the standard single-label learning algorithms.

(ii) Algorithm Adaptation Methods: Problem algorithm adaptation method, which modifies existing single-label learning algorithms in order to expend its ability to use multi-label data, such as ML-kNN , IBLR , BSVM , and BP-MLL .

In this paper, and in the same motive, we present a generalization of the ML−KNN based approach to multi-label classification approach (CML-kNN for short) based on non-iidness [5]. The major presentation of this paper is summarized as follows: - We developed a multi-label learning algorithm that based on lazy learning and its inner relationship between labels.

We develop a new coupled label similarity for multi-label kNN algorithm. The coupled label similarity will include similar kind of neighbors in the process to overcome the problem of missing neighbors with certain label.

We spreaded the concept of the nearest neighbor in multi-label classification with coupled label similarity. Based on this

extended nearest neighbors, we introduce a new frequency array strategy.

2 . MLkNN

MLKNN, the multi-label lazy learning approach, is based on the traditional KNN algorithm and the maximum a posteriori (MAP) principle .The rationale for the approach is that an instance's labels depend on the number of neighbors that have identical labels. Given to a instance x with an unknown label set

M (x) ⊆ L, MLKNN first identifies the k nearest neighbors in the training data and counts the number of neighbors belonging to each class (i.e. a variable Z from 0 to k). Then the maximum a posteriori principle is used to establish the label set for the test instance. The posterior probability of λi ∈ L is given by

$$P(\lambda i \in M(x)|Z = z) = , \quad (1)$$

where Z is the number of neighbors accompanying to each class ($0 \leq Z \leq k$). Then, for each label $\lambda_i \in L$, the algorithm builds a classifier h_i rule $h_i(x) =$

(2)

$h_i(x) = 1$ means λ_i is real label set to x, while 0 means it does not . The earlier and likelihood probabilities in Equation 1 are estimated from the training data set in advance. ML-kNN has two acquiring merits from both lazy learning and MAP principle. However, MLKNN is actually a binary relevance learner because it enrolls a single classifier h_i for each label independently. In other words, it does not consider the correlations between labels. The algorithm is generally criticized because of this drawback.

3 Methodology

3.1 Problem Statement

We formally define the multi-label classification problem as this: Let X denotes the space of instances and Y = {l1,...,ln} denotes the whole label set where |Y| = n. T = {(x1,M(x1)),...,(x_m,M(x_m))} (|T| = m) is the multi-label training data set, whose instances are drawn identically and independently from an unknown distribution D. Each instance x ∈ X is associated with a label set M(x) ∈ Y . The goal of our multi-label classification is to get a classifier h : X → Y that maps a feature vector to a set of labels, while optimizing some specific evaluation metrics.

3.2.Coupled Label Similarity

It is very easy for numerical data to compute the distance or similarity, since the existing metrics such as and Euclidean distance and Manhattan distance are mainly built for numeric variables, but the labels are categorise data. The main issue is to denote the similarity between them. As we all know, matching and frequency [1] are the most common ways to measure the similarity of categorical data. Accordingly, two main similarity measures are defined:

$$\text{Sim Overlap}(u_i,u_j) = , \quad (3)$$

The overlap similarity between two categorical values is to assign 0 if they different otherwise 1 if they are identical . The similarity between them will be proportional to the number of features in which they match for two multilabel categorical data points. The Frequency Based Cosine Similarity between two vectors U_i and U_j is defined as

$$\text{Sim Cosine}(U_i,U_j) = , \quad (4)$$

Frequency based measures, they take the several categorical values but with the same incident times as the same. Hence, the Overlap measure and Frequency Based measure are too simplistic by just giving the equal importance to matches and mismatches(not match).The co-occurrence information in categorical data reflects the interaction between features and can be used to define what makes two categorical values more or less similar.How ever, such co-occurrence information hasn't been incorporated into the existing similarity metrics.

We introduce an Intra-Coupling Label Similarity (IaCLS) and an Inter-Coupling Label Similarity (IeCLS) of two label values from two different labels for the inner relationship between categorical labels

Definition 1. Given a multi-label training data set D and two different labels $\mathbf{l_i}$ and $\mathbf{l_j}$(i != j), the label value is u^x_i, u^y_j respectively. The Intra-Coupling Label Similarity (IaCLS) between label values u^x_i and u^y_j of label $\mathbf{l_i}$ and $\mathbf{l_j}$ is denoteded as:

$\delta^{intra}(u^x_i, u^y_j) =$, (5)

where $RF(u^x_i)$ and $RF(u^y_j)$ are the occurrence frequency of label value u^x_i and u^y_j in label li and lj, respectively. The Intra-coupling Label Similarity returns the interaction of two different label values in the label space. The two values are more closer when these have higher similarities. Thus, Equation (5) is intended to capture the label value similarity in terms of incident times by taking into account the frequencies of categories. Besides, since $1 \leq RF(u^x_i), RF(u^y_j) \leq s$, then $\delta^{intra} \in [1/3, s/(s+2)]$.In contrast to the Intra-Coupling, we also define an Inter-Coupling Label Similarity below to capture the interaction of two different label values according to the co-occurrence of some value (or discretized value group) from feature spaces.

Definition 2. Given a training multi-label data set D and two different labels li and lj (i != j), the label value is u^x_i, u^y_j respectively. u^x_i and u^y_j are defined to be Inter-Coupling related if there exists at least one pair value (u^{zx}_p) or (u^{zy}_p) that occurs in feature az and labels of instance Up. TheInter-Coupling Label Similarity (IeCLS) between label values u^x_i and u^y_j according to feature value u^z_p of feature a_z is formalized as:

$$\delta^{intra}(u^x_i, u^y_j \mid u^z_p) = , \quad (6)$$

where $F(u^{zx}_p)$ and $F(u^{zy}_p)$ are the co-occurrence frequency count function for value pair u^{zx}_p or u^{zy}_p, and $RF(u^x_i)$ and $RF(u^y_i)$ is the occurrence frequency of related class label. u^z_p is the value in categorical feature a_z or the discretized value group in numerical feature a_z..

Accordingly, we have $\delta^{Ie} \in [0,1]$. The Inter-Coupling Label Similarity restore the interaction or relationship of two label values from label space but based on the connection to some other functions.

Definition 3. By taking into account both the Intra-Coupling and the Inter-Coupling, the Coupled Label Similarity (CLS) between two label values u^x_i and u^y_i is defined as:

$$CLS(u^x, u^y) = \delta^{Intra}(u^x, u^y) \cdot \quad (7)$$

where u^x_i and u^y_j are the label values of label l_i and l_j, respectively. δ^{intra} and δ^{inter} are the intra-coupling label similarity (Eq. 5) and inter-coupling label similarity (Eq. 6), respectively.

The n is the number of attributes and u_k denotes the values in the kth feature a_k.

Table 1. An Example of Multi-label Data

Instances	Label1	Label2	Label3	Label4
*v*1	*l*1			*l*4
*v*2			*l*3	*l*4
*v*3	*l*1		*l*3	
*v*4		*l*2	*l*3	
*v*5		*l*2	*l*3	*l*4

The Coupled Label Similarity defined in Eq. (7) reflects the interaction or similarity of two different labels. The two labels to be similar when its have higher the CLS . In Table 1, for example, $CLS(l_1,l_4)=0.33$, $CLS(l_1,l_3)=0.25$, so in the data set, an instance with label l_4 is more similar or close to instances with label l_1 than those instances with label l_3 do. The label pair (l_1,l_3) is closer to each other than the label pair (l_1, l_4). For Table 1, we got the coupled label similarity array which showed in Table 2.

Table 2. CLS Array

	Label1	**Label2**	**Label3**	**Label4**
Label1	1.0	0	0.25	0.33
Label2	0	1.0	0.50	0.33
Label3	0.25	0.50	1.0	0.50
Label4	0.33	0.33	0.50	1.0

3.3 Extended Nearest Neighbors

We present our extended nearest neighbors, based on the Coupled Label Similarity. Based on the similarity between labels, We can transfer a label set into a set with only a certain label based on the similarity between labels, it also means that a multi-label instance can be spreaded to a set of single-label. If we specify a basic label l_b, then any instance can be changed into a set with only one label l_b.For example, inTable1 ,instance v_5 has a label set of **Error! Bookmark not defined.**, then according to the label similarity array Table 2, it can be changed into , if we choose label l_2 as the basic label. We can then call the original multi-label instance v_5 equal a single-label instance with a label of

Table 3. Extended Nearest Neighbors

instance	Extended Neighbors	To Label
$v5$	$0 \cdot l1 + 0.25 \cdot l1 + 0.33 \cdot l1$	$l1$
$v5$	$1 \cdot l2 + 0.5 \cdot l2 + 0.33 \cdot l2$	$l2$
$v5$	$0.5 \cdot l3 + 1 \cdot l3 + 0.5 \cdot l3$	$l3$
$v5$	$0.33 \cdot l4 + 0.5 \cdot l4 + 1 \cdot l4$	$l4$

If v_5 is the neighbor of some instance, when we consider the label l_2, the instance v_5 can be presented as an instance which contains 1+0.5+0 .33 = 1.83 label l_2, and vice versa, instance u_5 also presents there are (1-1)+(1-0.5)+(1-0.33)=1.17 , instances which not contain the label l_2, and there will have (1.83+1.17 = 3 =|L(v_5)|. This is the main idea when we finding extended nearest neighbors.

Coupled ML-kNN

For the invisible instance x, lets M(x) denotes the set of its k nearest neighbors identified in data set D. For the j-th class label, CML-kNN chooses to evaluate the following statistics:

$C_j = Round(\ )$; (8)

Where L_i N (x), and δ_{L*} j denotes the sum of the CLS values of the i-th neighbor's label set to the j-th label l_j, Li is the label set of the i-th neighbor and *Round*() is the rounding function.

Cj is a rounding number which records all the CLS value of all x's neighbors to label lj. Let Ej be the event that x has label lj ,and P(Ej|Cj) denotess the posterior probability that Ej holds under the condition that x has exactly Cj neighbors with label lj . P(¬Ej|Cj) denotes the posterior probability that Hj doesn't hold under the same condition. MAP rule specify that, the predicted label set is resoluted by deciding whether P(Ej|Cj) is greater than P(¬Ej|Cj) or not:

Y={lj|>1,1≤j≤q} (9)

According to the Bayes Theory we have:

= (10)

Here, P(Ej) andP(¬Ej) showss the prior probability that Ej holds and doesn't hold, P(Cj|Ej) shows the likelihood that x has exactly Cj neighbors with label lj when Ej holds, and (P(Cj|¬Ej)) shows the likelihood that x has exactly Cj neighbors with label lj when Ej doesn't hold.

We integrated our coupled label similarity into the process, When we count the prior probabilities:

P(Ej); (11)

P(¬Ej) = 1−P(Ej);

where (1 ≤ j ≤ m) and n is the records number in training set, and s is a smoothing parameter controlling the effect of uniform prior on the estimation which generally takes the value of 1 (resulting in Laplace smoothing). Our CMLkNN maintains two frequency arrays αj and βj, same as ML-kNN, for the j-th class label lj,. , the frequency arrays will contain k×m+1 elements,

as our method reflectrs the other labels which have a similarity to a specific label

αj[r]= ; (12)

βj[r]=;

Where (0 ≤ r ≤ k ×m). We take an instance with δL∗ i|j ≥ 0.5 as an instance which does have label j and we take an instance with δL∗ i|j < 0.5 as an instance which doesn't have label j. Therefore, αj[r] counts the sum of CLS values to label j of training examples which have label lj and have exactly r neighbors with label lj, while βj[r] counts the CLS to label j of training examples which don't have label lj and have exactly r neighbors with label lj. Later, the likelihood can be analysed based on elements in αj and βj:

P(Cj|Ej)=; (13)

P(Cj|¬Ej)= ;

(1≤ j ≤ m,0≤ Cj ≤ k×m)

Again,by combing the prior probabilities (Eq.11) and the likelihoods(Eq.13) into Eq.(10),we will get the prognosticate label set in Eq.(9).

Algorithm 1. Coupled ML-kNN Algorithm

Input: An unlabeled instance xt and a labeled dataset
T{(x1,L(x1)),...,(xm,L(xm))}, where|T| = n and |L| = m

Output: The label set L(xt) of instance xt
1: Calculate the CLS array A(L) according to Eq.(7);
2: for i =1to n do;
3: Identify the k nearest neighbors N(xi) for xi
4: end for
5: for j =1to m do
6: Calculate P(Ej) andP(¬Ej) according to Eq.(11)
7: Maintain the label-coupled frequency arrays αj,βj using Eq.(12)
8: end for
9: Identify the k nearest neighbors N(xt) forxt
10: for j =1to m do
11: Calculate the statistic Cj according to Eq.(8)
12: end for
13: Return the label set L(xt) of instance xt according to Eq.(9)

Yeast	2417	103	14	4.237	0.303	198	0.082	n
medical	978	1449	45	1.245	0.028	94	0.096	c
Enron	1702	1001	53	3.378	0.064	753	0.442	c
Scene	2407	294	6	1.074	0.179	15	0.006	n
Bibtex	7395	1836	159	2.402	0.015	2856	0.386	c
Genbase	662	1185	27	1.252	0.046	32	0.048	c
emotions	593	72	6	1.869	0.311	27	0.046	n

4 Experiments and Evaluation

4.1 Experiment Data

A total of eight mostly used multi-label data sets are tested for experiments in this study, and the statistics of the data sets are shown in Table 4. Given a multi-label data set $M = \{(xi, L\ i) | 1 \leq i \leq q\}$, we use |S|, F(S), f(S), La(S), to represent the number of instances, feature type, number of features, number of total labels respectively. In addition, different multi-label statistics are also shown in the Table. The Label cardinality (LC(S)) measures the average number of labels per example; the Label density (LD(S)) normalizes LC(S) by the number of possible labels; the Distinct label sets (DL(S)) counts the number of distinct label combinations appeared in the data set; the Proportion of distinct label sets (PDL(S)) which normalizes DL(S) by the number of instances. As shown in Table 4, eight data sets are included and are ordered by Label density LD(S).

4.2 Experiment Setup

In our experiments,we compare the accomplishment of our proposed CML-kNN with that some mostly used multi-label classification algorithms: BR-KNN, ML-kNN, BSVM and IBLR. All nearest neighbor based algorithms are calculated by the size of the neighborhood k. We repeat the experiments with k =5 ,7,9 respectively (odd number for voting), and use the Euclidean metric as the distance function when calculating the nearest neighbors. The BR-kNN as the basic algorithm to compare with.As for as BSVM, models are learned via the cross-training strategy .We perform 10-inclose cross-validation three(3) times on all the above data sets.

4.3 Evaluation Criteria

Multi-label classification needs several metrics than whose used in single-label classification. To many section have been proposed for evaluating the performance of multi-label classification algorithms . In this paper, we use three popular evaluation section for multi -label classification: the One Error, the Hamming Loss and the Average Precision.

4.4 Experiment Results

The experiment results are shown in Table 5 ,Table 6,Table 7. For each evaluation criterion, "↑" indicates "the bigger the better" while "↓" indicates "the smaller the better", And the rank of the algorithms is denoted to numbers in parentheses among the five compared algorithms. The result tables indicate that CML-kNN and BSVM outperforms other algorithms significantly, which implies that exploiting the frequency of neighbors' label is effective, and especially for our CML-kNN, the improvement is significant compared to BR-kNN, that means incorporating the label relationship will greatly improve the BR strategy. Meanwhile, ML-kNN, IBLR and BR -kNN do not perform as well compared to the other algorithms. This implies that only exploiting the exact neighbor information is not sufficient, and the similar neighbor (correlations between labels) should also be considered.

Table 4. Experiment Data Sets

Data Set	|S|	f(S)	La(S)	LC(S)	LD(S)	DL(S	PDL(S)	F(S
Image	2000	294	5	1.236	0.247	20	0.010	n

Table 5. Experiment Result2 - One Error↓

	CML-*k*NN	BR-*k*NN	ML-*k*NN	IBLR	BSVM
image)	0.267(1)	0.601(5)	0.319(3)	0.432(4)	0.314(2)
yeast	0.222(1)	0.235(4)	0.228(2)	0.237(5)	0.232(3)
medical	0.158(2)	0.327(4)	0.252(3)	0.414(5)	0.151(1)

enron	0.308(3)	0.237(1)	0.313(4)	0.469(5)	0.245(2)
scene	0.197(1)	0.821(5)	0.219(2)	0.235(3)	0.251(4)
bibtex	0.376(1)	0.631(5)	0.589(3)	0.576(2)	0.599(4)
genbase	0.008(2)	0.012(5)	0.009(3)	0.011(4)	0.002(1)
emotions	0.244(1)	0.318(5)	0.263(3)	0.279(4)	0.253(2)
AvgRank	**(1.50)**	4.25	2.88	4.00	2.38

Table 6. Experiment Result1 - Hamming Loss↓

	CML-*k*NN	BR-*k*NN	ML-*k*NN	IBLR	BSVM
image	0.157(1)	0.189(5)	0.172(2)	0.182(4)	0.176(3)
yeast	0.194(1)	0.205(5)	0.195(2)	0.198(3)	0.199(4)
medical	0.013(1)	0.019(4)	0.016(3)	0.026(5)	0.013(1)
enron	0.061(4)	0.052(2)	0.052(2)	0.064(5)	0.047(1)
scene	0.078(1)	0.152(5)	0.084(2)	0.089(3)	0.104(4)
bibtex	0.013(1)	0.016(4)	0.014(2)	0.016(4)	0.015(3)
genbase	0.003(2)	0.004(3)	0.005(4)	0.005(4)	0.001(1)
emotions	0.189(1)	0.219(5)	0.194(2)	0.201(4)	0.199(3)
AvgRank	**(1.50)**	4.13	2.38	4.00	2.50

Table 7. Experiment Result3 - Average Precision↑

	CML-*k*NN	BR-*k*NN	ML-*k*NN	IBLR	BSVM
image	0.824(1)	0.601(5)	0.792(3)	0.761(4)	0.796(2)
yeast	0.769(1)	0.596(5)	0.765(2)	0.759(3)	...
medical	0.876(1)	0.782(4)	0.806(3)	0.686(5)	0.871(2)
enron	0.591(3)	0.435(5)	0.626(2)	0.564(4)	0.702(1)
scene	0.885(1)	0.651(5)	0.869(2)	0.862(3)	0.849(4)
bibtex	0.567(1)	0.329(5)	0.351(4)	0.476(3)	0.531(2)
genbase	0.994(3)	0.992(4)	0.989(5)	0.994(2)	0.998(1)
emotions	0.819(1)	0.595(5)	0.799(3)	0.798(4)	0.807(2)
AvgRank	**(1.50)**	4.75	3.00	3.50	2.25

Overall, our proposed CML-kNN outperforms all the assimilated methods on all three measures. The average ranking of our method on these data sets using three different metrics is the first one, with (1.50, 1.50, 1.50) respectively, while the second best algorithm, BSVM, only performs (2.50, 2.38, 2.25). The BR-kNN performs the worst, which only attains (4.13,4.25,4.75). It is worth noting that although our proposed method runs the best on average, it does not mean that it is suitable for all kinds of data. For example, when used on data set "enron" and "genbase", the result is not as good as on other data sets. Sometimes it even got a worse result than BR-kNN. For example, when used on "enron" and evaluated by the Hamming Loss, our supposed CML-kNN only achieved a 4th rank(0.061), while BR-kNN can get a second well result(0.052). The reason is because of the weak or loose connection between different labels in those data sets, and our extended neighbors may introduce more noisy information than useful information. But in terms of average performance, our method performs the best (the first rank).

5. Conclusions and Future Work

ML-kNN acquires a single classifier hi for each label li independently, so it is actually a binary relevance classifier. In other words, it does not reflect the correlations between different labels. The algorithm is often anatomized for this drawback. In this paper, we introduced a coupled label similarity, which examines the innerrelationship between several labels in multi-label classification according to their natural co-occupance. This similarity returns the distance of the different labels. Furthermore, by integrating this similarity into the multi-label kNN algorithm, we overcome the ML-kNN's shortcoming and improved the performance. Evaluated over three commonly-used multi-label data sets and in terms of Hamming Loss, One Error and Average Precision, the proposed method outperforms ML-KNN, BR-kNN, IBLR and even BSVM. This result shows that our assumedd coupled label similarity is proper for multi-label learning problems and can work more effectively than other methods.

References

1. [Boriah, S., Chandola, V. and Kumar, V., 2008, April. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM international conference on data mining* (pp. 243-254). Society for Industrial and Applied Mathematics.]
2. [Zhang, M.L. and Zhou, Z.H., 2007. ML-KNN: A lazy learning approach to multi-

label learning. *Pattern recognition*, *40*(7), pp.2038-2048.]

3. [Boutell, M.R., Luo, J., Shen, X. and Brown, C.M., 2004. Learning multi-label scene classification. *Pattern recognition*, *37*(9), pp.1757-1771.]
4. [Zhang, M.L. and Zhou, Z.H., 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, *18*(10), pp.1338-1351.]
5. [Brinker, K. and Hüllermeier, E., 2007, January. Case-Based Multilabel Ranking. In *IJCAI* (pp. 702-707).]
6. [Cao, L., 2015. Coupling learning of complex interactions. *Information Processing & Management*, *51*(2), pp.167-186.]
7. [Wieczorkowska, A., Synak, P. and Raś, Z.W., 2006. Multi-label classification of emotions in music. In *Intelligent Information Processing and Web Mining* (pp. 307-315). Springer, Berlin, Heidelberg.]
8. [Wang, C., Cao, L., Wang, M., Li, J., Wei, W. and Ou, Y., 2011, October. Coupled nominal similarity in unsupervised learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management* (pp. 973-978).]
9. [Cao, L., 2014. Non-iidness learning in behavioral and social data. *The Computer Journal*, *57*(9), pp.1358-1370.]
10. [Cao, L., Ou, Y. and Philip, S.Y., 2011. Coupledbehavioranalysiswith applications. *IEEE Transactions on Knowledge and Data Engineering*, *24*(8), pp.1378-1392.]
11. [Cheng, W. and Hüllermeier, E., 2009. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, *76*(2-3), pp.211-225.]
12. [Liu, C., Cao, L. and Philip, S.Y., 2014, July. Coupled fuzzy k-nearest neighbors classification of imbalanced non-IID categorical data. In *2014 international joint conference on neural networks (IJCNN)* (pp. 1122-1129). IEEE.]
13. [Read, J., Pfahringer, B., Holmes, G. and Frank, E., 2011. Classifier chains for multi-label classification. *Machine learning*, *85*(3), p.333.]
14. [Schapire, R.E. and Singer, Y., 2000. BoosTexter: A boosting-based system for text categorization. *Machine learning*, *39*(2-3), pp.135-168.]
15. [Spyromitros, E., Tsoumakas, G. and Vlahavas, I., 2008, October. An empirical study of lazy multilabel classification algorithms. In *Hellenic conference on artificial intelligence* (pp. 401-406). Springer, Berlin, Heidelberg.]
16. [Tsoumakas, G. and Katakis, I., 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, *3*(3), pp.1-13.]
17. [Vembu, S. and Gärtner, T., 2010. Label ranking algorithms: A survey. In *Preference learning* (pp. 45-64). Springer, Berlin, Heidelberg.]