Supervised Feature Space Reduction for Multi-Label Nearest Neighbors

Wissam Siblini, Réda Alami, Frank Meyer, Pascale Kuntz

Supervised Feature Space Reduction for Multi-Label Nearest Neighbors

Wissam Siblini[1,2], Reda Alami[1], Frank Meyer[1], and Pascale Kuntz[2]

[1] Orange Labs, Av. Pierre Marzin - 22300 Lannion, France
firstname.lastname@orange.com,
[2] Laboratoire d'Informatique de Nantes Atlantique - Site Polytech Nantes - 44300 Nantes Cedex, France
firstname.lastname@univ-nantes.fr

Abstract. With the ability to process many real-world problems, multi-label classification has received a large attention in recent years and the instance-based ML-kNN classifier is today considered as one of the most efficient. But it is sensitive to noisy and redundant features and its performances decrease with increasing data dimensionality. To overcome these problems, dimensionality reduction is an alternative but current methods optimize reduction objectives which ignore the impact on the ML-kNN classification. We here propose ML-ARP, a novel dimensionality reduction algorithm which, using a variable neighborhood search metaheuristic, learns a linear projection of the feature space which specifically optimizes the ML-kNN classification loss. Numerical comparisons have confirmed that ML-ARP outperforms ML-kNN without data processing and four standard multi-label dimensionality reduction algorithms.

Keywords: multi-label classification · k-nearest neighbors · dimensionality reduction

1 Introduction

In the traditional single-label classification paradigm, the objective is to associate each instance to one label only. However, in various real-world applications (e.g. music annotation, image categorization, text mining), objects are intrinsically describable with multiple labels. Consequently, multi-label classification has received a large attention in recent years and many algorithms have been proposed [19, 10, 14]. Among them, the multi-label adaptation of the well-known k-nearest neighbor algorithm (ML-kNN [18]) is probably one of the most successful. Based on the maximum a posteriori principle, ML-kNN operates instance-based learning. Numerical comparisons with many model-based methods have confirmed the high quality of its results.

However, instance-based algorithms such as ML-kNN have two major shortcomings [2]. First, as they rely on a distance function, they are very sensitive to noisy, redundant and irrelevant features. Second, they encounter the explosion of their computational complexity when dealing with high-dimensional data where

numerous instances are described by numerous variables. In practice these serious issues are brought to the fore today with the expansion of online labeling services which produce massive raw data of varying quality.

By appearing as a promising lever for these problems, dimensionality reduction encounters a renewed interest. Roughly speaking, the reduction approaches used in multi-label classification can be divided into two families: (i) the unsupervised methods that reduce the feature space independently of any label information [1] and (ii) supervised methods that benefit from the labeling information with an objective that is either independent [12, 13, 20] or dependent on the classifier [7, 9]. The last type of method seems more promising as the final objective is to optimize the classification quality. However, the joint problem between classification and dimensionality reduction is generally set in the form of a multi-objective optimization which is hard to solve even heuristically.

In this article, we skirt the explicit multi-objective formulation with a novel linear reduction method for optimizing the ML-kNN classification performances. Our approach, called Multi-Label Adaptative Random Projection (ML-ARP), initializes a random linear projection and iteratively adapts it with a reduced variable neighborhood search in order to increase the ML-kNN performances on the projected feature space. Numerical comparisons on twelve classical multi-label datasets have confirmed that, while reducing the dimensionality of data and the neighborhood search complexity up to 90 percent, ML-ARP is not only better on average than ML-kNN without data processing but it also outperforms a simple random projection technique and four standard multi-label dimensionality reduction algorithms from the literature (Principal Component Analysis [1], Canonical Correlation Analysis [13], Multi-label Dimensionality reduction via Dependence Maximization [20] and the Orthonormal version of Partial Least Squares [12]).

The remainder of this paper is organized as follows. Section 2 reviews previous approaches for multi-label dimensionality redution. We describe our new algorithm ML-ARP in Section 3 and present the experimental comparisons in Section 4.

2 Multi-Label Feature Space Dimensionality Reduction

We here present the two main families of dimensionality reduction methods: the unsupervised methods which do not take label information into account and the supervised methods which use it to guide the reduction.

2.1 Unsupervised Dimensionality Reduction

The unsupervised methods can themselves be organized into two classes: methods based on random projection and methods based on feature information. The first type has been investigated in multi-label classification to reduce both label [16] and feature space [11]. In the classical context, it is known to be the fastest way to reduce the dimensionality and the Johnson-Lindenstrauss lemma

[6] has proved that it accurately preserves the pairwise l_2 distances between the instances in the projected space. However, the result quality declines with the reduced space dimensionality.

The second type usually tends to reduce the feature space while keeping a maximum of its structural information (e.g. feature covariance or co-occurence). It has a long history dating back to the inception of data analysis. The most popular method still remains the Principal Component Analysis [1] but several variants have been proposed [4]. However, these approaches do not consider some useful information contained in the links between the features and the labels.

2.2 Supervised Dimensionality Reduction

Supervised approaches guide the reduction with constraints or label information. The reduction can be done independently or dependently of the classifier criterion.

The most prevalent methods ignore the classifier objective and usually aim at strengthening the link between the projected features and the labels (e.g. with a dependence or covariance criterion). In the multi-label context, among the most popular are the Canonical Correlation Analysis (CCA) [13, 8], the Partial Least Square (PLS) [3] and the Multi-label Dimensionality reduction via Dependence Maximization (MDDM) [20]. CCA seeks the directions in both label and feature spaces which maximize the correlations between each other. PLS seeks the directions in the feature space that maximize the covariance with the label space. A variant of PLS (Orthonormal PLS [12]) introduces orthogonality constraints between the computed directions. MDDM computes a projection of the feature space that minimizes the Hilbert-Schmidt independence criterion between the projected data features and the labels. In studies previously published, all these approaches have been applied at a pre-processing stage before the ML-kNN classifier. The experimental results are promising but, by only optimizing their own criteria (covariance, dependence and co-occurrence), these methods can degrade the performances of the classifier.

Recent researches have confirmed that the best dimensionality reduction method can vary with the choice of the classifier [20]. These results stimulate the development of approaches which integrate a coupling between dimensionality reduction and classification in a global optimization problem. They usually resort to an SVM classifier [9] or a large margin classifier [7]. However, in both cases the optimization process tries to combine explicitly two different objectives. In [7] the expressed loss function is a sum of two reconstruction errors: dimensionality reduction and classification. In [9] the combination of the two formulations leads to a two-parameter optimization problem where each parameter is computed alternatively. This multi-objective strategy may converge to a poor quality solution for the classifier. Moreover, these previous approaches do not consider a coupling with the ML-kNN classifier which is, with its intrinsic multi-label nature, its powerful classification rules and its potential for an online adaptation, the center of our attention in this study.

To overcome these limits, we here propose a novel approach where the projection of the reduced space is the unique parameter and the optimization of the ML-kNN performance is the unique objective.

3 Description of the ML-ARP Algorithm

In the following we consider a d_x - dimensional feature space $\mathcal{X}$ and a d_y - dimensional label space $\mathcal{Y}$. Each instance (x_i, y_i) is represented by a feature vector x_i and its associated binary label vector y_i where the j^{th} component $(y_i)^j$ is equal to 1 if the instance is described by the j^{th} label and 0 otherwise. In the learning scenario, data are partitioned in two sets: the training set $\mathcal{L} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid \ i \in \{1, ..., N_{\mathcal{L}}\}\}$ of cardinality $N_{\mathcal{L}}$ used to train the model and the testing set $\mathcal{T} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid i \in \{1, ..., N_{\mathcal{T}}\}\}$ of cardinality $N_{\mathcal{T}}$ used to compute the performances of the model.

3.1 The Algorithm ML-kNN

Let us recall that ML-kNN [18] combines the principle of the k-nearest neighbor algorithm with a powerful multi-label decision rule. More precisely, for a given feature vector $x \in \mathcal{X}$ it first determines its neighborhood in $\mathcal{L}$ using the Euclidean l_2 distance. Next, it predicts a real-valued output $\widehat{y_{int}} \in \mathbb{R}^{d_y}$ by summing the labels of the k-nearest neighbors. Then, it converts its prediction into classification with a maximum a posteriori rule; this rule benefits from the labeling pattern embodied in the instance neighborhood. This operation requires a training phase where two quantities are computed for each label l: (i) the prior probability of the presence (resp. the absence) of the label l which is its frequency (resp. the complementary) in $\mathcal{L}$ and (ii) the likelihood in $\mathcal{L}$ that an instance associated with the label l has exactly j neighbors with the label l, for j in $\{0, ..., k\}$. With these two pieces of information and $\widehat{y_{int}}$, ML-kNN determines the posterior probability for the presence/absence of each label with a Bayes rule. If the presence probability is higher than the absence probability, the label is set to 1 in the final predicted label vector $\widehat{y} \in \mathcal{Y}$.

In the original ML-kNN, a Laplace smoothing is optionnally applied to prevent events which do not occur in the training set $\mathcal{L}$ from having a likelihood or a prior probability equal to zero. In our experiments, without any prior knowledge about the data, we prefer to avoid using this smoothing. Moreover, as ML-kNN is here applied for each method of our benchmark, the smoothing parameter would only affect absolute performances and not relative comparisons.

3.2 ML-ARP: Multi-Label Adaptative Random Projection

Our objective is to build a projection which explicitly optimizes the ML-kNN performances Θ in the reduced feature space (of dimensionality r). This is likely to correct the two previously-cited shortcomings by (i) implicitely filter the features that are irrelevant for classification and (ii) reducing the complexity of

distance evaluation from d_x operations to r operations. The performances are here measured with the Hamming Loss (HL) which is a global reconstruction error widely used in the multi-label context. The minimization problem over the objective $\Theta(P)$ is then defined by:

$$\min_{P \in \mathbb{R}^{d_x \times r}} \Theta(P) = \min_{P \in \mathbb{R}^{d_x \times r}} \sum_{i=1}^{N_{\mathcal{L}}} HL(y_i, \widehat{y_i} = \text{ML-}k\text{NN}(\mathcal{L}^P, x_i)) \tag{1}$$

where ML-kNN$(\mathcal{L}^P, x_i)$ denotes the prediction for x_i of ML-kNN applied in the P-projected training set.

As the variation of ML-kNN$(\mathcal{L}^P, x_i)$ in function of P is hard to express, standard optimization approaches are impracticable and we resort to a Reduced Variable Neighorhood Search (RVNS) heuristic [17] to compute a solution to the problem (1). Our implementation of the RVNS changes the projection parameter P iteratively and randomly and selects the changes which improve the objective Θ. More precisely, the different steps of the algorithm are the following:

1. Initialize P with a random projection drawn from a zero-mean, unit-variance Gaussian distribution.
2. Make a slight modification of the solution P into a new solution P' using a speed matrix ΔP: $P' = P + \Delta P$.
3. Evaluate the loss $\Theta(P')$ of the new parameter P'.
4. If $\Theta(P')$ is lower than $\Theta(P)$, then consider P' as the new current solution; otherwise keep P.
5. If the new solution is P', repeat the steps 2., 3. and 4 with the same speed matrix ΔP; otherwise, repeat these steps with a new sparse speed matrix (The speed matrix is chosen to be sparse so that only a few parameters are changed at each RVNS iteration) generated with the following process: Randomly select a mutation rate α in $[0, 1]$. Then, for each term of the matrix ΔP, run a coin toss with a probability of α. If the result is negative, the term is set to 0; otherwise, the term is randomly generated from a zero-mean Gaussian distribution.

The process stops after a fixed number of iterations or a maximum computation time. Let us remark that the conditions of the Johnson-Lindenstrauss lemma are not valid here: by selecting specific modifications, the ML-ARP algorithm produces a final solution P which is no longer a random projection. Consequently, the initial distances in the original space $\mathcal{X}$ are not preserved; they are modified in order to improve the ML-kNN performances.

Let us remark that a non linear reducing mapping could also be a candidate. However, without any further information on the search spaces, we have here favoured the simplest choice of a linear mapping. The non linearity is indirectly tackled by the combination of the mapping with the non linear classifier ML-kNN.

4 Experiments

We first describe the experimental protocol and then present the comparisons obtained with six different approaches on twelve data sets of various sizes.

4.1 Experimental Settings

Datasets We have conducted our experimental comparisons on twelve real-world datasets from various domains: music annotation (Emotions), image annotation (Scene, Corel5k), video (Mediamill), text mining (Enron, Bibtex, Delicious, Bookmarks, Reuters) and medical mining (Yeast). Their main statistical properties are described in Table 1 and we refer to Mulan [15] for details.

Table 1. Description of the twelve datasets: application domain, training set cardinality ($N_{\mathcal{L}}$), testing set cardinality ($N_{\mathcal{T}}$), feature space dimensionality (d_x), label space dimensionality (d_x), label space density (r_y).

	Domain	# instances	$N_{\mathcal{L}}$	$N_{\mathcal{T}}$	d_x	d_y	r_y
Yeast	genetic	2417	2173	244	103	14	0.3
Emotions	audio	593	533	60	72	6	0.31
Mediamill	video	43907	39516	4391	120	101	0.043
Scene	images	2407	2166	241	294	6	0.18
Corel5k	images	5000	4500	500	499	374	0.0094
Delicious	text(tags)	16105	14495	1610	500	983	0.019
Enron	text	1702	1531	171	1001	53	0.064
Genbase	biology	662	595	67	1186	27	0.05
Medical	health	978	880	98	1449	45	0.0027
Bibtex	text	7395	6656	739	1836	159	0.015
Bookmarks	text	87856	79070	8786	2150	208	0.0098
Reuters	text	6000	5400	600	47229	101	0.026

Algorithms The new algorithm ML-ARP has been compared to four other dimensionality reduction approaches from the state-of-the-art (PCA [1], CCA [13], MDDM [20], OPLS [12]) coupled to ML-kNN. We have added two other comparisons which play the role of yardsticks: one with a normalized random projection (RP) drawn from a zero-mean, unit-variance Gaussian distribution and another with the original ML-kNN classifier without dimensionality reduction. In our experiments, the dimensionality r of the reduced feature space is of the same order of magnitude as those classically used in the literature: 128 or 64 if the dimensionality of the original feature space is smaller than 128. The higher the reduced space dimensionality r, the more expressive the projection. Fixing the same value for every method therefore allows an equal comparison. The chosen baseline systematically predicts the labels frequencies computed on $\mathcal{L}$, for any $x \in \mathcal{T}$. The real values of the frequency vector are binarized with a threshold

of 0.5. As we here restrict ourselves to the comparison of the different approach performances, we have not explored the impact of the neighbor number. We have followed the recommandation of [18] and fixed k = 5. As well as for the smoothing parameter, changing k would mostly affect absolute performances. The maximal computation time was fixed to two hours to meet our operational constraints.

Quality evaluation To evaluate the performances of the algorithms on each dataset, we have performed a 10-fold evaluation and computed the mean performance and standard deviation of 11 different measures [19, 10, 14] evaluating ranking performances, classification accuracies and global reconstruction errors: Ranking Loss, One Error, Coverage, Jaccard Loss, Hamming Loss, Accuracy, Recall, Precision, Subset Accuracy, Average Precision, F1-Score.

Further analysis with statistical tests on Hamming Loss have been carried out to evaluate the significative differences and similarities between the algorithms. Using the R *scmamp* package [5], we have applied the Friedman test with $\alpha = 0.1$ (90% confidence) and completed it with the Nemenyi post-hoc test.

4.2 Results

The results obtained with the Hamming Loss for the different approaches are summarized in Table 2. Firstly, they show that ML-ARP outperforms the other dimensionality reduction approaches (MDDM, PCA, OPLS, CCA) for three datasets (Yeast, Emotions, Delicious) and that it is very close to the best values for the other datasets. Secondly, they suggest that ML-ARP is better than ML-kNN, RP and the baseline but the statistical significance of the dominance is only confirmed against the baseline by the Nemenyi test (Figure 1). Thirdly, the performances of the original ML-kNN are always improved by at least one dimensionality reduction approach. But, for some datasets, the independent dimensionality reduction may lead to degraded results (e.g. MDDM for Emotions and CCA for Scene). MDDM, CCA, OPLS, PCA are not applied on some datasets (N/A values) either because their complexity (spatial and temporal) is too high or because they require an inversion of a non invertible matrix.

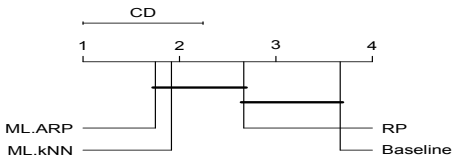

Fig. 1. Results of the Nemenyi test for ML-ARP, RP, ML-kNN and the baseline on all the datasets.

Table 2. Hamming Loss performances (with N/A for unavailable values)

		ML-ARP	Baseline	ML-kNN	RP	MDDM	PCA	OPLS	CCA
Yeast	$r = 64$	**0.191**	0.232	0.195	0.202	0.227	0.194	0.203	0.204
Emotions	$r = 64$	**0.226**	0.313	0.262	0.261	0.31	0.262	0.256	0.252
Scene	$r = 128$	0.091	0.179	**0.088**	0.108	0.089	0.097	0.166	0.162
Enron	$r = 128$	0.05	0.062	0.051	0.053	**0.049**	**0.049**	0.067	0.064
Genbase	$r = 128$	0.047	0.047	0.047	0.047	0.047	0.047	0.047	N/A
Corel5k	$r = 128$	0.009	0.009	0.009	0.009	0.009	0.009	0.009	N/A
Delicious	$r = 128$	**0.014**	0.019	**0.014**	0.020	0.018	0.018	N/A	N/A
Medical	$r = 128$	0.017	0.028	0.015	0.018	**0.013**	0.015	N/A	N/A
Bibtex	$r = 128$	0.014	0.015	0.014	0.014	**0.012**	0.013	N/A	N/A
Mediamill	$r = 64$	0.027	0.035	0.027	0.028	N/A	N/A	**0.025**	N/A
Bookmarks	$r = 128$	0.008	0.009	0.008	0.008	N/A	N/A	N/A	N/A
Rcv1	$r = 4000$	0.026	0.028	0.026	0.026	N/A	N/A	N/A	N/A

Table 3. Ranks regarding all performance measures on four datasets (Emotions, Scene, Enron and Yeast).

	ML.ARP	Baseline	ML-kNN	RP	MDDM	PCA	OPLS	CCA
Accuracy	**2.00**	8.00	3.25	4.75	4.00	3.00	6.5	4.5
Average Precision	**2.00**	7.75	3.375	5.00	4.25	2.875	6.125	4.625
Coverage	3.25	6.75	**2.375**	4.375	4.00	3.00	6.5	5.75
F1	**2.5**	8.00	3.00	5.25	4.00	2.625	6.125	4.5
Hamming Loss	**2.00**	7.5	3.375	4.5	4.375	3.25	5.75	5.25
Jaccard Loss	**2.5**	8.00	3.00	4.75	4.00	2.75	6.5	4.5
One Error	2.75	8.00	**2.625**	5.00	5.00	**2.625**	5.75	5.25
Precision	3.375	5.75	4.375	4.625	3.75	**3.125**	6.00	5.00
Ranking Loss	3.00	8.00	**2.75**	4.25	4.00	**2.75**	6.5	4.75
Recall	**2.625**	8.00	3.875	5.25	4.00	3.00	5.5	3.75
Subset Accuracy	**2.00**	8.00	3.25	4.25	4.00	3.5	6.00	5.00
Mean	**2.55**	7.52	3.2	4.73	4.13	2.95	6.11	4.81
Global rank	**1.64**	7.91	2.64	5.41	4.05	1.91	7.09	5.36

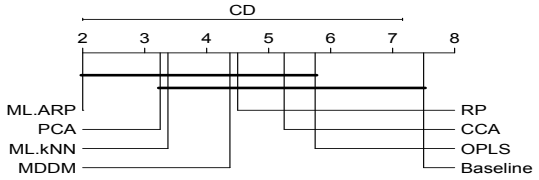

Fig. 2. Results of the Nemenyi test for all the algorithms on four datasets (Yeast, Emotions, Scene, Enron).

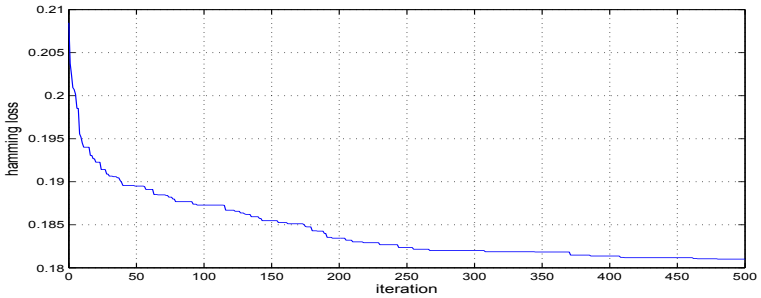

Fig. 3. Evolution of Hamming loss training error for ML-ARP on Emotions (mean curve for 10 runs)

For the four datasets where all the algorithms have been applied (Emotions, Scene, Enron and Yeast) ML-ARP obtains the highest mean rank for a majority of performance measures (Table 3). For the global reconstruction error measures (Hamming Loss, Jaccard Loss, Accuracy), it is always the best. For some ranking sensitive measures (Coverage, One Error, Precision, Ranking Loss), it is slightly surpassed by ML-kNN and PCA with very close performances but the differences are not statistically significant (Figure 2). Moreover, if a closer examination of the convergence time goes beyond the objective of this paper, we have observed that, on average, ML-ARP optimization converges fast enough after several hundreds of iterations (Figure 3).

5 Conclusions and Future Works

Whatever the dataset, it has been observed that there exists a reduced space for which ML-kNN performances are improved or maintained. Thus, dimensionality reduction approaches not only have the advantage of reducing the number of features and speeding up the neighborhood search but also have the potential of improving the ML-kNN classification. However, in practice, classical reduction approaches have obtained poor performances for some datasets and have deteriorated the classification on average because their independent objective does not guarantee an effective neighborhood for ML-kNN.

In contrast, ML-ARP presents two advantages. From a statistical point of view, it is more stable than the other methods: as a wrapper designed to specifically target the ML-kNN objective, it presents the most regular performances and the best mean rank when facing a wide variety of problems. From a technological point of view, it is easily implementable, anytime and more scalable.

To accelerate the algorithm in the big data scenario we plan in the next future to explore a sampling strategy (random, clustering, condension) and a GPU implementation for the nearest neighbor search.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2(4), 433–459 (2010)
2. Bellet, A., Habrard, A., Sebban, M.: A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709 (2013)
3. Bishop, C.M.: Pattern recognition. Machine Learning (2006)
4. Burges, C.J.: Geometric methods for feature extraction and dimensional reduction-a guided tour. In: Data mining and knowledge discovery handbook, pp. 53–82. Springer (2009)
5. Calvo, B., Santafe, G.: scmamp: Statistical comparison of multiple algorithms in multiple problems. The R Journal Accepted for publication (2015)
6. Dasgupta, S., Gupta, A.: An elementary proof of a theorem of johnson and lindenstrauss. Random Structures & Algorithms 22(1), 60–65 (2003)
7. Guo, Y., Schuurmans, D.: Semi-supervised multi-label classification. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 355–370. Springer (2012)
8. Hotelling, H.: Relations between two sets of variates. Biometrika 28(3/4), 321–377 (1936)
9. Ji, S., Ye, J.: Linear dimensionality reduction for multi-label classification. In: IJCAI. vol. 9, pp. 1077–1082. Citeseer (2009)
10. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recognition 45(9), 3084–3104 (2012)
11. Ran, R., Oh, H.: Adaptive sparse random projections for wireless sensor networks with energy harvesting constraints. EURASIP Journal on Wireless Communications and Networking 2015(1), 113 (2015)
12. Rosipal, R., Krämer, N.: Overview and recent advances in partial least squares. In: Subspace, latent structure and feature selection, pp. 34–51. Springer (2006)
13. Sun, L., Ji, S., Ye, J.: Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(1), 194–200 (2011)
14. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006)
15. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. Journal of Machine Learning Research 12(Jul), 2411–2414 (2011)
16. Wan, S., Mak, M.W., Kung, S.Y.: Sparse regressions for predicting and interpreting subcellular localization of multi-label proteins. BMC bioinformatics 17(1), 97 (2016)
17. Xiao, Y., Kaku, I., Zhao, Q., Zhang, R.: A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems. European Journal of Operational Research 214(2), 223–231 (2011)
18. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern recognition 40(7), 2038–2048 (2007)
19. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering 26(8), 1819–1837 (2014)
20. Zhang, Y., Zhou, Z.H.: Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data (TKDD) 4(3), 14 (2010)