ReliefF-ML: An Extension of ReliefF Algorithm to Multi-label Learning

Oscar Gabriel Reyes Pupo[1], Carlos Morell[2], and Sebastián Ventura Soto[3]

[1] University of Holguín, Cuba
oreyesp@facinf.uho.edu.cu
[2] Universidad Central "Marta Abreu" de Las Villas, Cuba
cmorellp@uclv.edu.cu
[3] University of Córdoba, Spain
sventura@uco.es

Abstract. In the last years, the learning from multi-label data has attracted significant attention from a lot of researchers, motivated from an increasing number of modern applications that contain this type of data. Several methods have been proposed for solving this problem, however how to make feature weighting on multi-label data is still lacking in the literature. In multi-label data, each data point can be attributed to multiple labels simultaneously, thus a major difficulty lies in the determinations of the features useful for all multi-label concepts. In this paper, a new method for feature weighting in multi-label learning area is presented, based on the principles of the well-known ReliefF algorithm. The experimental stage shows the effectiveness of the proposal.

Keywords: multi-label learning, feature weighting, ReliefF algorithm.

1 Introduction

The multi-label problems have been actively studied in the last years. This is because it has been found that in many applications the multi-label data is a more natural and appropriate form of problem formulation and representation. Particular examples of such applications include text categorization [1], emotions evoked by music [2] and semantic annotation of images [3]. In all of these applications an instance space is typically represented by hundreds or thousands of features, therefore commonly there are features more relevant than others, and this situation affect the effectiveness of the machine learning algorithms.

Several supervised learning methods have been proposed to multi-label classification, however feature weighting and selection methods on multi-label data are less researched problems. How to make feature weighting on multi-label data is still lacking in the literature, furthermore multi-label feature weighting is still a challenging problem.

In this work, a filter-based feature weighting method called ReliefF-ML is proposed. ReliefF-ML is based on the principles of the well-known ReliefF algorithm [4]. Some properties of ReliefF-ML method are that it can be applied to

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part II, LNCS 8259, pp. 528–535, 2013.

both continuous and discrete problems, it includes interaction among features, and take into account the label dependences.

Due to the fact that lazy learning algorithms use a similarity or distance function based in feature space, these types of algorithms can be easily used to prove the effectiveness of feature weighting methods [5]. In this work, the approach ReliefF-ML was used as a feature weighting, not as a multi-label feature selection method; therefore the comparison with the existent multi-label feature selection methods in the literature was not carried out.

To evaluate the performance of ReliefF-ML, the accuracy of 3 multi-label lazy ranking algorithms using the feature weights provided by ReliefF-ML on 11 multi-label datasets from several fields were compared, showing the effectiveness of the proposal for multi-label problems.

This paper is organized as follows. In section 2, a formal definition of the multi-label learning task and related works on feature weighting methods to multi-label data is presented. In section 3, the ReliefF-ML approach is described. The experimental set up is described in section 4. An analysis of the experiment results appears in section 5. Finally, in section 6 the conclusion of this work are presented.

2 Background

2.1 Multi-label Learning

The multi-label learning is concerned with learning from examples, where each example is associated with multiple labels. In multi-label learning there can be distinguished two types of tasks: multi-label classification (MLC) and label ranking (LR). In the case of MLC, the goal is to construct a predictive model that will provide a list of relevant labels for a given test instance. On the other hand, the goal in LR is to construct a predictive model that will provide an ordering of the labels according to their relevance for a given test instance. The generalization of these two problems has been called multi-label ranking (MLR). [6]. In general, a multi-label dataset can be defined as follows:

-A feature space $\mathcal{F}$ that consists of tuples of values of primitive data types (discrete or continuos) $\forall x_i \in \mathcal{F}$, $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where D is the number of descriptive attributes. x_i is the vector of features values for the instance i, where x_{if} represents the value of f-th attribute for the instance i.

-A label space $\mathcal{L}$ with a cardinality equal to Q, where Q is the number of labels in the dataset.

-A set of instances (examples) $E = \{(x_i, y_i) | x_i \in \mathcal{F}, y_i \subseteq \mathcal{L}, 1 \leq i \leq N\}$, where N is the number of instances and y_i is the set of relevant labels for the instance i. A label l is relevant for an instance i if the instance belongs to the class l, a label l is irrelevant for an instance i otherwise.

2.2 Related Works

The feature weighting process is a more general method than the feature selection task, in which the features are multiplied by a weight value proportional to the

ability of the feature to distinguish pattern classes, whereas the feature selection problem assigns a weight restricted to the binary values 0 or 1 to a feature.

ReliefF [4] is a classical method for feature estimation. ReliefF is able to deal with incomplete and noisy data and can be used for evaluating the feature quality in multi-class problems. Commonly the ReliefF algorithm is used as a feature selection method, however it is a feature weighting method. The feature weighting is an important component of any lazy learning scheme. ReliefF was tested as feature weighting method in [5] and was found to be very useful to improve the performance of lazy algorithms.

In [7] was proposed a feature weighting method that learns a similarity metric to improve the performance of multi-label ranking lazy algorithms. The search process of the best weight vector was performed using a genetic algorithm (GA). This method can be very expensive in complex multi-label datasets.

An approximation of ReliefF algorithm to multi-label data was presented in [8]. The authors decompose the multi-label problem into a set of pairwise multi-label 2-class problems. The algorithm excludes those examples that fall into *Hits* and *Misses* neighbors at the same time. The authors expose that the occurrence of these cases is very small, and therefore the exclusion of these instances will not affect the results significantly. However, this reasoning was done because the two specific datasets used in the experiment present this characteristic. In multi-label datasets a very high number of examples can fall into Hit and *Misses* neighbors at the same time, therefore excluding these examples can affect the results significantly.

In [9] other adaptation of ReliefF algorihtm to multi-label data was presented. It uses the standard ReliefF for single-label, where is measured the contribution of each feature according to each label. Afterwards, the average of the score of each feature across all labels is considered, and features with an averaged score greater than a threshold are selected. This approach use the Binary Relevance [10] approach to decompose the multi-label problem into several binary classification problems, therefore it does not consider label correlations.

3 The ReliefF-ML Algorithm

The biggest problem for the multi-label feature weighting process is that an instance is assigned to multiple labels simultaneously, therefore nearest $Hits$ and $Misses$ cannot be used in a strict sense as in classic ReliefF algorithm. Given a multi-label dataset, the prior probability of a label l is computed as follows:

$$P_l = \frac{C_l + s}{N + 2s} \tag{1}$$

, where C_l is the number of instances in the dataset that belong to label l and s is the smoothing parameter controlling the strength of uniform prior ($s = 1$ yields the Laplace smoothing).

Given the instances i and j, the distance between the sets of labels of i and j is calculated by the Hamming Distance (see equation 2). The distance $d_{\mathcal{L}}$ represents a measure of how much differ the sets of labels of two instances.

$$d_{\mathcal{L}}(i,j) = \frac{\mid y_i \triangle y_j \mid}{Q} \tag{2}$$

ReliefF-ML uses the HEOM distance(Heterogeneous Euclidean Overlap Metric) [11](equation 3) to retrieve the k-nearest neighbors of an instance i according to the feature space.

$$d_{\mathcal{F}}(i,j) = \sqrt{\sum_{\forall f \in \mathcal{F}} \delta(x_{if}, x_{jf})^2} \tag{3}$$

$$\delta(x_{if}, x_{jf}) = \begin{cases} 1 & discrete,\ x_{if} \neq x_{jf} \\ 0 & discrete,\ x_{if} = x_{jf} \\ \frac{|x_{if} - x_{jf}|}{max(f) - min(f)} & continuous \end{cases} \tag{4}$$

For each relevant and irrelevant label of an instance i a group of k-nearest neighbors is defined. Therefore, the following groups of *Hits* (H_i^l) and *Misses* (M_i^l) respect to an instance i are defined:
-H_i^l: k-nearest neighbors that have the relevant label l of i as relevant label
-M_i^l: k-nearest neighbors that have the irrelevant label l of i as relevant label

Based in the defined groups of *Hits* and *Misses* the following "probability" was defined, it is modelled with the distance between the sets of labels of two learning instances.

$$P_{G_i^l} = \frac{\sum_{\forall j \in G_i^l} d_{\mathcal{L}}(i,j)}{k} \tag{5}$$

, where:
-$P_{H_i^l}$: is the probability that two nearest instances that share the label l as relevant, belong to different set of labels.
-$P_{M_i^l}$: is the probability that two nearest instances belong to different set of labels, where i has the label l as irrelevant and the k-nearest neighbors have the label l as relevant.

In ReliefF-ML the dependence among labels is taken into account through the calculus of $P_{H_i^l}$ and $P_{M_i^l}$ for each relevant and irrelevant label respectively of a sampling instance. Each feature weight reflects its ability to distinguish class labels, thus a high weight indicates that there is differentiation in this attribute among instances with very different sets of labels and has similar values for instances with similar sets of labels otherwise. The weight updating of an attribute f uses the equation (6).

$$W_f = W_f - \sum_{l \in y_i} \left(\frac{P_l}{\sum_{q \in y_i} P_q} \frac{1 - P_{H_i^l}}{1 + P_{H_i^l}} \sum_{j \in H_i^l} \frac{\delta(x_{if}, x_{jf})}{mk}\right) + \sum_{l \notin y_i} \left(\frac{P_l}{\sum_{q \notin y_i} P_q} P_{M_i^l} \sum_{j \in M_i^l} \frac{\delta(x_{if}, x_{jf})}{mk}\right) \tag{6}$$

The contributions of each relevant and irrelevant label are weighted by the factors $\frac{P_l}{\sum_{q \in y_i} P_q}$, $\frac{1-P_{H_i^l}}{1+P_{H_i^l}}$ and $\frac{P_l}{\sum_{q \notin y_i} P_q}$, $P_{M_i^l}$ respectively.

Algorithm 1. Pseudocode of ReliefF-ML algorithm

Input: E: learning multi-label instances, m: sampling parameter, k: number of nearest neighbors to retrieve
Output: weight vector W
1: **for** each $l \in \mathcal{L}$ **do** Calculate P_l **end for**;
2: **for** each $f \in \mathcal{F}$ **do** Set $W_f = 0$ **end for**;
3: **for** $n = 1$ to m **do**
4: Pick randomly an instance i from E
5: **for** each relevant label $l \in y_i$ **do**
6: Get k-nearest *Hits* H_i^l
7: Calculate $P_{H_i^l}$
8: **end for**
9: **for** each irrelevant label $l \notin y_i$ **do**
10: Get k-nearest *Misses* M_i^l
11: Calculate $P_{M_i^l}$
12: **end for**
13: **for** each attribute $f \in \mathcal{F}$ **do**
14: Calculate W_f by expression (6)
15: **end for**
16:**end for**
17:Scale the weights in the range [0..1]

ReliefF-ML picks randomly a predefined number of instances (m) from the E set to estimate the feature weights. It uses the whole training set to retrieve the k nearest neighbors of a selected instance. To fix the number of instances to be selected to estimate the feature weights the following rules were used:
1.***if*** $(|E| \leq 5000)$ ***then*** $(m{=}0.1 \times |E|)$
2.***if*** $(|E| > 5000\, and \mid E \mid \leq 10000)$ ***then*** $(m{=}0.05 \times |E|)$
3.***if*** $(|E| > 10000)$ ***then*** $(m{=}0.01 \times |E|)$

4 Experimental Section

In [12] a lazy algorithm named ML-kNN was proposed, it uses the maximum a posteriori principle (MAP) in order to determine the set of labels of a query instance. DML-kNN [13] can be considered a generalization of the ML-kNN based approach where the dependencies among labels are considered. MLC-WkNN appears in [14], the author constructs a weighted kNN version for multi-label learning according to the Bayesian theorem.

To prove the effectiveness of the proposal, each lazy algorithm using the weights reached by ReliefF-ML were tested, and then the results were compared with the original methods. The modified algorithms were named ML-kNN-WF, DML-kNN-WF and MLC-WkNN-WF to differentiate them from the original

methods. In the adapted lazy algorithms the function used originally to retrieve the k-nearest neighbors was replaced by the Weighted HEOM distance version, which takes into account the feature weights.

ReliefF-ML and the lazy algorithms were implemented on MULAN [15], that is a Java library which contains several methods for multi-label learning. For each possible combination of algorithms and datasets a stratified 10-fold cross validation strategy was used. For each fold in the training phase, ReliefF-ML finds the weight vector by picking randomly the sampling instances from the training set. The lazy learning algorithms use the weight vector in the distance functions to retrieve the k nearest neighbors of an instance. The best value for the parameter k used by ReliefF-ML and the lazy algorithms on each dataset was determined. As for comparison between the originals and adapted methods, the Wilcoxon signed ranks test was used as proposed in [16].

The algorithms were tested with 11 multi-label datasets from different domains. Selection was made in order to understand the behaviour of our approach in datasets with diverse characteristics. All datasets are available for download at the web page http://mlkd.csd.auth.gr/multilabel.html. In order to verify the effectiveness of the proposal, 4 evaluation measures that have been suggested for MLR problems in [10] were used. The Hamming Loss (H_L) reports how many times on average, the relevance of an example to a class label is incorrectly predicted. Accuracy (A_{cc}) returns the proportion of the predicted correct labels to the total number (predicted and actual) of labels for that instance, over all instances. One Error (O_E) measures how many times the top ranked predicted label is not in the set of true labels of the instances. Ranking Loss (R_L) evaluates the average proportion of label pairs that are incorrectly ordered for an instance.

5 Results and Dicussion

The performance of the ReliefF-ML was evaluated through comparisons of the algorithms ML-kNN, DML-kNN and MLC-WkNN, and their respective extensions ML-kNN-WF, DML-kNN-WF and MLC-WkNN-WF. In all cases the best results are highlighted in bold typeface in the tables. Tables 1 to 4 show the results of H_L, A_{cc}, O_E and R_L on the 3 selected algorithms.

The results shows that the adapted algorithms perform better than the original algorithms in almost all datasets with the 4 measures used in the experiment. Table 5 shows Wilcoxon's signed rank test; it summarizes the positive (R^+) and negative (R^-) ranks, ties and if the hypothesis is rejected (R) or not (NR) with a significance α equals to 0.01.

The evidences suggest that ML-kNN-WF, DML-kNN-WF and MLC-WkNN-WF are statistically better than the original algorithms in all the measures used. The results obtained show that the proposed approach is robust, it does well in datasets with different characteristics. Furthermore, the proposed method to multi-label feature weighting improves the performance of multi-label lazy learning algorithms.

Table 1. H_L results

Dataset	ML-kNN -	ML-kNN WF	DML-kNN -	DML-kNN WF	MLC-WkNN -	MLC-WkNN WF
Emotions	0.1963	**0.1812**	0.1965	**0.1840**	0.1884	**0.1800**
Yeast	0.1925	**0.1915**	0.1924	**0.1910**	0.1935	**0.1915**
Scene	0.0868	**0.0865**	0.0872	**0.0859**	0.0846	**0.0840**
Cal500	0.1387	**0.1382**	0.1377	**0.1373**	**0.1472**	**0.1472**
Genbase	0.0043	**0.0036**	0.0046	**0.0043**	0.0012	**0.0009**
Medical	0.0151	**0.0136**	0.0157	**0.0145**	0.0146	**0.0137**
Enron	0.0526	**0.0525**	0.0520	**0.0518**	0.0558	**0.0557**
TMC2007-500	0.0649	**0.0620**	0.0646	**0.0620**	0.0380	**0.0366**
Mediamill	0.0281	**0.0279**	0.0282	**0.0280**	0.0246	**0.0245**
Corel5k	**0.0094**	**0.0094**	**0.0094**	**0.0094**	**0.0096**	**0.0096**
Corel16k	**0.0175**	**0.0175**	**0.0175**	**0.0175**	0.0181	**0.0180**

Table 2. A_{cc} results

ML-kNN -	ML-kNN WF	DML-kNN -	DML-kNN WF	MLC-WkNN -	MLC-WkNN WF
0.5344	**0.5645**	0.5352	**0.5645**	0.5518	**0.5789**
0.5201	0.5188	**0.5196**	**0.5196**	0.5268	**0.5359**
0.6665	**0.6784**	0.6665	**0.6800**	**0.6879**	0.6878
0.1954	**0.1998**	0.1914	**0.1959**	0.2216	**0.2217**
0.9499	**0.9618**	0.9453	**0.9501**	0.9894	**0.9895**
0.5828	**0.6412**	0.5288	**0.5858**	0.5815	**0.6198**
0.3032	**0.3046**	0.2978	**0.3025**	0.3162	**0.3168**
0.5296	**0.5567**	0.5285	**0.5559**	0.7264	**0.7351**
0.4727	**0.4728**	**0.4700**	0.4691	0.5517	**0.5521**
0.0148	**0.0170**	0.0026	**0.0039**	0.0344	**0.0378**
0.0076	**0.0083**	0.0043	**0.0053**	0.0339	**0.0360**

Table 3. O_E results

Dataset	ML-kNN -	ML-kNN WF	DML-kNN -	DML-kNN WF	MLC-WkNN -	MLC-WkNN WF
Emotions	0.2680	**0.2296**	0.2646	**0.2300**	0.2462	**0.2385**
Yeast	0.2272	**0.2150**	0.2263	**0.2162**	0.2325	**0.2271**
Scene	**0.2244**	0.2255	**0.2252**	0.2294	0.2285	**0.2232**
Cal500	0.1168	**0.1147**	**0.1147**	**0.1147**	0.2264	**0.1920**
Genbase	0.0151	**0.0084**	0.0166	**0.0085**	0.0030	**0.0022**
Medical	0.2239	**0.1975**	0.2393	**0.2042**	0.2198	**0.1949**
Enron	0.3111	**0.3100**	0.3093	**0.3012**	**0.3732**	0.3782
TMC2007-500	0.2313	**0.2131**	0.2315	**0.2020**	0.1412	**0.1352**
Mediamill	0.1554	**0.1486**	0.1536	**0.1521**	0.1321	**0.1312**
Corel5k	0.7288	**0.7170**	0.7314	**0.7248**	0.7824	**0.7640**
Corel16k	0.7396	**0.7320**	0.7401	**0.7301**	0.7760	**0.7660**

Table 4. R_L results

ML-kNN -	ML-kNN WF	DML-kNN -	DML-kNN WF	MLC-WkNN -	MLC-WkNN WF
0.1596	**0.1500**	0.1558	**0.1484**	0.1641	**0.1565**
0.1658	**0.1630**	0.1646	**0.1631**	0.1739	**0.1726**
0.0801	**0.0801**	0.0777	**0.0770**	0.0834	**0.0819**
0.1812	**0.1807**	0.1992	**0.1787**	0.2482	**0.2473**
0.0071	**0.0063**	0.0070	**0.0059**	0.0038	**0.0037**
0.0363	**0.0341**	0.0353	**0.0322**	0.0438	**0.0427**
0.0898	**0.0898**	0.0894	**0.0892**	**0.1857**	**0.1857**
0.0584	**0.0520**	0.0563	**0.0498**	0.0510	**0.0490**
0.0369	**0.0363**	**0.0360**	**0.0360**	**0.0608**	0.0613
0.1300	**0.1292**	0.1306	**0.1302**	0.4731	**0.4656**
0.1641	**0.1635**	0.1647	**0.1642**	0.3086	**0.3060**

Table 5. Wilcoxon's signed rank test

Measures	R^+	R^-	$Ties$	$p-value$	$Hypothesis$
ML-kNN-FW vs ML-kNN					
H_L	0	9	2	0.008	R
A_{cc}	1	10	0	0.008	R
O_E	1	10	0	0.005	R
R_L	0	9	2	0.008	R
DML-kNN-FW vs DML-kNN					
H_L	0	9	2	0.008	R
A_{cc}	1	9	1	0.007	R
O_E	1	9	1	0.009	R
R_L	0	10	1	0.005	R
MLC-WkNN-FW vs MLC-WkNN					
H_L	0	9	2	0.007	R
A_{cc}	1	10	0	0.006	R
O_E	1	10	0	0.008	R
R_L	1	9	1	0.009	R

6 Conclusions

The attention given to the study of feature weighting methods in multi-label learning has been negligible. In this paper, a filter feature weighting method called ReliefF-ML to deal with multi-label problems was proposed. The proposed method has significant advantages; it is a preprocessing step that is completely independent of the choice of particular multi-label algorithm. Also, it uses the given representation of the original datasets (handles multi-label data directly), it learns a single set of weights that are employed globally over the entire instance space, it takes into account the label correlations in the estimation of feature weights and does not employ domain specific knowledge to set feature weights. The algorithm ReliefF-ML is a generalization of the classic ReliefF algorithm.

The experiments aimed to measure the performance of multi-label lazy algorithms in conjunction with the proposed method for feature weighting. Results from the statistical tests show that the proposed method has significant advantages, which indicate that the approach is robust for MLR problems.

References

1. McCallum, A.: Multi-label text classification with a mixture model trained by EM. In: Working Notes of the AAAI-99 Workshop on Text Learning (1999)
2. Li, T., Ogihara, M.: Detecting emotion in music. In: Proceedings of the International Symposium on Music Information Retrieval (2003)
3. Yang, S., Kim, S., Ro, Y.: Semantic home photo categorization. IEEE Transactions on Circuits and Systems for Video Technology 17, 324–335 (2007)
4. Kononenko, I.: Estimating attributes: Analysis and extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
5. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review 11, 273–314 (1997)
6. Brinker, K., Furnkranz, J., Hullermeier, E.: A unified model for multilabel classification and ranking. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), pp. 489–493 (2006)
7. Reyes, O., Morell, C., Ventura, S.: Learning similarity metric to improve the performance of lazy multi-label ranking algorithms. In: IEEE Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 246–251 (2012)
8. Kong, D., Ding, C., Huang, H., Zhao, H.: Multi-label ReliefF and F-statistic Feature Selections for Image Annotation. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 2352–2359 (2012)
9. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: Filter approach feature selection methods to support multi-label learning based on reliefF and information gain. In: Barros, L.N., Finger, M., Pozo, A.T., Gimenénez-Lugo, G.A., Castilho, M. (eds.) SBIA 2012. LNCS, vol. 7589, pp. 72–81. Springer, Heidelberg (2012)
10. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 667–686. Springer (2010)
11. Wilson, D., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research (JAIR) 6, 1–34 (1997)
12. Zhang, M.L., Zhou, Z.H.: ML-kNN: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)
13. Younes, Z., Abdallah, F., Denceux, T.: Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies. In: Proceedings of the 16th Eropean Signal Processing Conference, Lausanne, Switzerland (2008)
14. Xu, J.: Multi-label weighted k-nearest neighbor classifier with adaptive weight estimation. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 79–88. Springer, Heidelberg (2011)
15. Tsoumakas, G., Spyromitros-Xioufi, E., Vilcek, J., Vlahavas, I.: MULAN: A java library for multi-label learning. Journal of Machine Learning Research 12, 2411–2414 (2011)
16. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)