Efficient Two Stage Voting Architecture for Pairwise Multi-label Classification

Gjorgji Madjarov, Dejan Gjorgjevikj, and Tomche Delev

Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Rugjer Boshkovikj bb, 1000 Skopje, R. of Macedonia
`{madzarovg,dejan,tdelev}@feit.ukim.edu.mk`

Abstract. A common approach for solving multi-label classification problems using problem-transformation methods and dichotomizing classifiers is the pair-wise decomposition strategy. One of the problems with this approach is the need for querying a quadratic number of binary classifiers for making a prediction that can be quite time consuming especially in classification problems with large number of labels. To tackle this problem we propose a two stage voting architecture (TSVA) for efficient pair-wise multiclass voting to the multi-label setting, which is closely related to the calibrated label ranking method. Four different real-world datasets (enron, yeast, scene and emotions) were used to evaluate the performance of the TSVA. The performance of this architecture was compared with the calibrated label ranking method with majority voting strategy and the quick weighted voting algorithm (QWeighted) for pair-wise multi-label classification. The results from the experiments suggest that the TSVA significantly outperforms the concurrent algorithms in term of testing speed while keeping comparable or offering better prediction performance.

Keywords: Multi-label, classification, calibration, ranking.

1 Introduction

Traditional single-label classification is concerned with learning from set of examples that are associated with a single label λ_i from a finite set of disjoint labels $L = \{\lambda_1, \lambda_2, ..., \lambda_Q\}$, $Q > 1$. If $Q = 2$, then the learning problem is called a binary classification problem, while if $Q > 2$, then it is called a multi-class classification problem. On the other hand, multi-label classification is concerned with learning from a set of examples $S = \{(x_1, Y_1), (x_2, Y_2), ..., (x_p, Y_p)\}$ ($x_i \in X$, X denote the domain of examples) where each of the examples is associated with a set of labels $Y_i \subseteq L$.

Many classifiers were originally developed for solving binary decision problems, and their extensions to multi-class and multi-label problems are not straight-forward. Because of that, a common approach to address the multi-label classification problem is utilizing class binarization methods, i.e. decomposition of the problem into several binary subproblems that can then be solved using a binary base learner. The simplest strategy in the multi-label setting is the one-against-all strategy also referred to as the binary relevance method. It addresses the multi-label classification problem by learning one classifier (model) M_k $(1 \leq k \leq Q)$ for each class, using all the examples labeled with that class as positive examples and all other (remaining) examples as negative examples. At query time, each binary classifier predicts whether its class is relevant for the query example or not, resulting in a set of relevant labels.

Another approach for solving the multi-label classification problem using binary classisifers is pair-wise classification or round robin classification [1][2]. Its basic idea is to use $Q * (Q-1)/2$ classifiers covering all pairs of labels. Each classifier is trained using the samples of the first label as positive examples and the samples of the second label as negative examples. To combine these classifiers, the pair-wise classification method naturally adopts the majority voting algorithm. Given a test instance, each classifier delivers a prediction for one of the two labels. This prediction is decoded into a vote for one of the labels. After the evaluation of all $Q * (Q-1)/2$ classifiers the labels are ordered according to their sum of votes. To predict only the relevant classes for each instance a label ranking algorithm is used. Label ranking studies the problem of learning a mapping from set of instances to rankings over a finite number of predefined labels. It can be considered as a natural generalization of conventional classification, where only a single label (the top-label) is requested instead of a ranking of all labels.

Brinker et al. [3] propose a conceptually new technique for extending the common pair-wise learning approach to the multi-label scenario named calibrated label ranking. The key idea of calibrated label ranking is to introduce an artificial (calibration) label λ_0, which represents the split-point between relevant and irrelevant labels. The calibration label λ_0 is assumed to be preferred over all irrelevant labels, but all relevant labels are preferred over it. At prediction time (when majority voting strategy is usually used), one will get a ranking over $Q+1$ labels (the Q original labels plus the calibration label). The calibrated label ranking is considered a combination of both multi-label classification and ranking.

Besides the majority voting that is usually used strategy in the prediction phase of the calibrated label ranking algorithm, Park et al. [4] propose another more effective voting algorithm named Quick Weighted Voting algorithm (QWeighted). QWeighted computes the class with the highest accumulated voting mass avoiding the evaluation of all possible pair-wise classifiers. It exploits the fact that during a voting procedure some classes can be excluded from the set of possible top rank classes early in the process when it becomes clear that even if they reach the maximal voting mass in the remaining evaluations they can no longer exceed the current maximum. Pair-wise classifiers are selected depending on a voting loss value, which is the number of votes that a class has not received. The voting loss starts with a value of zero and increases monotonically with the number of performed preference evaluations. The class with the current minimal loss is the top candidate for the top rank class. If all preferences involving this class have been evaluated (and it still has the lowest loss), it can be concluded that no other class can achieve a better ranking. Thus, the QWeighted algorithm always focuses on classes with low voting loss. An adaptation of QWeighted to multi-label classification (QWeightedML) [5] is to repeat the process while all relevant labels are not determined i.e. until the returned class is the artificial label λ_0, which means that all remaining classes will be considered to be irrelevant.

In this paper we propose an efficient Two Stage Voting Architecture (TSVA) that modifies the majority voting algorithm for calibrated label ranking technique [6]. We have evaluated the performance of this architecture on a selection of multi-label datasets that vary in terms of problem domain and number of labels. The results demonstrate that our modification outperforms the majority voting algorithm for pair-wise multi-label classification and the QWeightedML [5] algorithm in terms of testing speed, while keeping comparable prediction results.

For the readers' convenience, in Section 2 we will briefly introduce notations and evaluation metrics used in multi-label learning. The Two Stage Voting Architecture is explained in Section 3. The experimental results that compare the performance of the proposed TSVA with concurrent methods are presented in Section 4. Section 5 gives a conclusion.

2 Preliminaries

Let X denote the domain of instances and let $L = \{\lambda_1, \lambda_2, ..., \lambda_Q\}$ be the finite set of labels. Given a training set $S = \{(x_1, Y_1), (x_2, Y_2), ..., (x_p, Y_p)\}$ $(x_i \in X, Y_i \subseteq L)$, the goal of the learning system is to output a multi-label classifier $h : X \rightarrow 2^L$ which optimizes some specific evaluation metric. In most cases however, instead of outputting a multi-label classifier, the learning system will produce a real-valued function of the form $f : X \times L \rightarrow R$. It is supposed that, given an instance x_i and its associated label set Y_i, a successful learning system will tend to output larger values for labels in Y_i than those not in Y_i, i.e. $f(x_i, y_1) > f(x_i, y_2)$ for any $y_1 \in Y_i$ and $y_2 \notin Y_i$. The real-valued function $f(\bullet, \bullet)$ can be transformed to a ranking function $rank_f(\bullet, \bullet)$, which maps the outputs of $f(x_i, y)$ for any $y \in L$ to $\{\lambda_1, \lambda_2, ..., \lambda_Q\}$ such that if $f(x_i, y_1) > f(x_i, y_2)$ then $rank_f(x_i, y_1) < rank_f(x_i, y_2)$. Note that the corresponding multi-label classifier $h(\bullet)$ can also be derived from the function $f(\bullet, \bullet) : h(x_i) = \{y | f(x_i, y) > t(x_i); y \in L\}$, where $t(\bullet)$ is a threshold function which is usually set to be the zero constant function. Performance evaluation of multi-label learning system is different from that of classical single-label learning system. The following multi-label evaluation metrics proposed in [7] are used in this paper:

(1) Hamming loss: evaluates how many times an instance-label pair is misclassified, i.e. a label not belonging to the instance is predicted or a label belonging to the instance is not predicted. The performance is perfect when $hloss_S(h) = 0$. The smaller the value of $hloss_S(h)$, the better the performance. This metric is given by

$$hloss_S(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{Q}\left|h(x_i)\Delta Y_i\right| \tag{1}$$

where Δ stands for the symmetric difference between two sets and Q is the total number of possible class labels. Note that when $|Y_i| = 1$ for all instances, a multi-label system reduces to multi-class single-label one and the hamming loss becomes $2/Q$ times the usual classification error.

While hamming loss is based on the multi-label classifier $h(\bullet)$, the other four metrics are defined based on the real-valued function $f(\bullet, \bullet)$ that takes into account the ranking quality of different labels for each instance:

(2) One-error: evaluates how many times the top-ranked label is not in the set of proper labels of the instance. The performance is perfect when $one - error_S(f) = 0$. The smaller the value of $one - error_S(f)$, the better the performance. This evaluation metric is given by:

$$one - error_S(f) = \frac{1}{p}\sum_{i=1}^{p}\left[\!\left[\left[\arg\max_{y\in Y} f(x_i, y)\right] \notin Y_i\right]\!\right] \tag{2}$$

where for any predicate π, $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise. Note that, for single-label classification problems, the one-error is identical to ordinary classification error.

(3) Coverage: evaluates how far, on the average we need to go down the list of ranked labels in order to cover all the proper labels of the instance. The smaller the value of $coverage_S(f)$, the better the performance.

$$coverage_S(f) = \frac{1}{p}\sum_{i=1}^{p}\max_{y\in Y_i} rank_f(x_i, y) - 1 \tag{3}$$

(4) Ranking loss: evaluates the average fraction of label pairs that are reversely ordered for the particular instance given by:

$$rloss_S(f) = \frac{1}{p}\sum_{i=1}^{p}\frac{|D_i|}{|Y_i|\left|\bar{Y_i}\right|} \tag{4}$$

where $D_i = \left\{f(y_1, y_2)|f(x_i, y_1) \leq f(x_i, y_2), (y_1, y_2) \in Y_i \times \bar{Y_i}\right\}$, while $\bar{Y}$ denotes the complementary set of Y in L. The smaller the value of $rloss_S(f)$, the better the performance, so the performance is perfect when $rloss_S(f) = 0$.

(5) average precision: evaluates the average fraction of labels ranked above a particular label $y \in Y$ that actually are in Y. The performance is perfect when $avgprec_S(f) = 1$; the bigger the value of $avgprec_S(f)$, the better the performance. This metric is given by:

$$avgprec_S(f) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|L_i|}{rank_f(x_i, y)} \tag{5}$$

where $L_i = \{y'|rank_f(x_i, y') \leq rank_f(x_i, y), y' \in Y_i\}$.

Note that in the rest of this paper, the performances of the multi-label learning algorithms are evaluated based on the five metrics explained above.

3 Two Stage Voting Architecture (TSVA)

Conventional pair-wise approach learns a model M_{ij} for all combinations of labels λ_i and λ_j with $1 \leq i < j \leq Q$. This way $Q * (Q-1)/2$ different pair-wise models are learned. Each pearwise model M_{ij} is learned with the examples labelled with label λ_i as positive examples and the examples labelled with λ_j as negative examples. The main disadvantage of this approach is that in the prediction phase a quadratic number of base classifiers (models) have to be consulted for each test example.

Further, as a result of introducing the artificial calibration label λ_0 in the calibrated label ranking algorithm, the number of the base classifiers is increased by Q i.e. additional set of Q binary preference models M_{0k} $(1 \leq k \leq Q)$ is learned. The models M_{0k} that are learned by a pair-wise approach to calibrated ranking, and the models M_k that are learned by conventional binary relevance are equivalent. At prediction time (when standard majority voting algorithm is usually used) each test instance needs to consult all the models (classifiers) in order to rank the labels by their order of preference. This results in slower testing, especially when the number of the labels in the problem is big.

In this paper we propose an efficient two stage voting architecture which modifies the majority voting algorithm for the calibrated label ranking technique. It reduces the number of base classifiers that are needed to be consulted in order to make a final prediction for a given test instance. The number of base classifiers that are trained by the calibrated label ranking algorithm and the TSVA in the learning process is equivalent.

The proposed (TSV) architecture is organized in two layers. In the first layer of the architecture Q classifiers are located, while in the second layer of the architecture the rest $Q * (Q-1)/2$ classifiers are located. All of the classifiers in the first layer are the binary relevance models M_{0k}, while in the second layer of the architecture the pair-wise models M_{ij} are located. Each model M_{0k} from the first layer is connected with $Q-1$ models M_{ij} from the second layer, where $k = i$ or $k = j$ $(1 \leq i \leq Q-1, i+1 \leq j \leq Q)$. An example of TSVA for solving four-class multi-label classification problems is shown on Fig. 1.

At prediction time, each model M_{0k} of the first layer of the architecture tries to determine the relevant labels for the corresponding test example. Each model M_{0k} gives the probability (the output value of model M_{0k} is convert to probability) that the test example is associated with the label λ_k. If that probability is appropriately small (under some threshold), we can conclude that the artificial calibration label λ_0 is preferred over the label λ_k i.e. the label λ_k belongs to the set of irrelevant labels. In such case, one can conclude that for the test example, the pair-wise models of the second layer M_{ij} where $i = k$ or $j = k$, need not be consulted, because the binary relevance model M_{0k} from the first layer has already made a decision that the label λ_k belongs to the set of irrelevant labels.

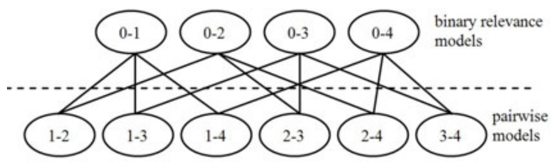

Fig. 1. TSV Architecture

For each test example for which it is known that the label λ_k belongs to the set of irrelevant labels, the number of models that should be consulted decreases for $Q-1$.

In order to make a decision which labels belong to the set of irrelevant labels i.e. which pair-wise models M_{ij} from the second layer do not have to be consulted a threshold t $(0 \leq t \leq 1)$ is introduced.

According to the previously mentioned, in TSVA every test instance first consults all binary relevance models M_{0k} of the first layer of the architecture. If the corresponding model M_{0k} $(1 \leq k \leq Q)$ response with a probability that is above the threshold t, the test instance is then forwarded only to the models M_{ij} of the second layer of the architecture that are associated to the model M_{0k}. The pair-wise model M_{ij} from the second layer is connected to the binary relevance models M_{0i} and M_{0j}. This does not mean that the model M_{ij} has to be consulted twice, if the prediction probabilities of the models M_{0i} and M_{0j} are both above the threshold t. Instead the model M_{ij} is consulted only once and its prediction is decoded into a vote for one of the labels λ_i or λ_j. If the prediction of one of the models M_{0i} and M_{0j} results with probability under the threshold t, the corresponding model M_{ij} is not consulted and the vote from this model goes to the label which binary relevance model prediction probability is above the threshold t.

By increasing the value of the threshold, the number of consulted pair-wise models decreases. If $t = 1$ the test instance is not forwarded to the second layer of the architecture and the TSVA becomes binary relevance method. On the other hand, if $t = 0$, all pair-wise models of the second layer are consulted and the TSVA becomes calibrated label ranking method with majority voting.

4 Experimental Results

In this section, we present the results of our experiments with several multi-label classification problems. The performance was measured on the problem of recognition of text, music, image and gene function.

Here, the performance of the TSV architecture is compared with the calibrated label ranking method with majority voting strategy for pair-wise multi-label classification (CLR-S) and the QWeightedML algorithm [5].

The training and testing of the TSVA was performed using a custom developed application that uses the MULAN library [8] for the machine learning framework Weka [9]. The LIBSVM library [10] utilizing the SVMs with radial basis kernel

Table 1. Datasets

	scene	**yeast**	**enron**	**emotions**
Domain	image	biology	text	music
Training Instances	1211	1500	1123	391
Test Instances	1159	917	579	202
Features	294	103	1001	72
Labels	6	14	53	6

Table 2. The evaluation of each method for every dataset

	t	**Evaluation Metric**	**CLR-S**	**QWeightedML**	**TSVA**
enron	0.03	**Hamming Loss**	0.0476	0.0481	0.0501
		One-error	0.2297	0.2262	0.2193
		Coverage	11.5198	20.3333	14.4317
		Ranking Loss	0.0756	0.1516	0.0969
		Avg. Precision	0.7018	0.6543	0.6970
		Testing time (s)	605.06	174.31	147.57
emotions	0.25	**Hamming Loss**	0.2566	0.2623	0.2590
		One-error	0.3812	0.3762	0.3663
		Coverage	2.4059	2.8465	2.3960
		Ranking Loss	0.2646	0.3381	0.2612
		Avg. Precision	0.7215	0.6795	0.7242
		Testing time (s)	2.56	1.67	1.34
yeast	0.15	**Hamming Loss**	0.1903	0.1909	0.1906
		One-error	0.2334	0.2301	0.2300
		Coverage	6.2758	8.6215	6.7633
		Ranking Loss	0.1632	0.2934	0.1805
		Avg. Precision	0.7685	0.7003	0.7641
		Testing time (s)	104.34	60.39	54.65
scene	0.02	**Hamming Loss**	0.0963	0.0956	0.0946
		One-error	0.2349	0.2349	0.2366
		Coverage	0.4883	0.7073	0.4974
		Ranking Loss	0.0779	0.1190	0.0799
		Avg. Precision	0.8600	0.8400	0.8598
		Testing time (s)	66.15	40.32	35.73

were used for solving the partial binary classification problems. Usually, the most important criterion when evaluating a classifier is its prediction performance, but very often the testing time of the classifier can be equally important. In our experiments, four different multi-label classification problems were addressed by each classifying methods. The recognition performance and the testing time were recorded for every method. The problems considered in the experiments include scene [11] (scene), gene function [12] (yeast), text [13](enron) and music [14] (emotions) classification.

The complete description of the datasets (domain, number of training and test instances, number of features, number of labels) is shown in Table 1.

In all classification problems the classifiers were trained using all available training samples of the sets and were evaluated by recognizing all the test samples from the corresponding set. Table 2 gives the performance of each method applied on each of the datasets. The first column of the table describes the datasets. The second column shows the values of the threshold t for each dataset separately, for which the presented results of TSVA are obtained.

The value of the threshold t for each dataset was determined by 5-fold cross validation using only the samples of the training set in order to achieve maximum benefit in terms of prediction results on testing speed.

Table 2 clearly shows that among the three tested approaches TSVA offers best performance in terms of testing speed. The results show that for the four treated classification problems TSVA is 2 to 4 times faster than calibrated label ranking algorithm with majority voting and 10% to 15% faster than the QWeightedML method. It can also be noticed that TSVA offers better performance than QWeightedML method in all evaluation metrics, while showing comparable performance to calibrated label ranking algorithm with majority voting. The dependence of the predictive performances for different values of the threshold t ($0 \leq t \leq 1$) are shown on Fig. 2. Fig. 3 shows the testing time

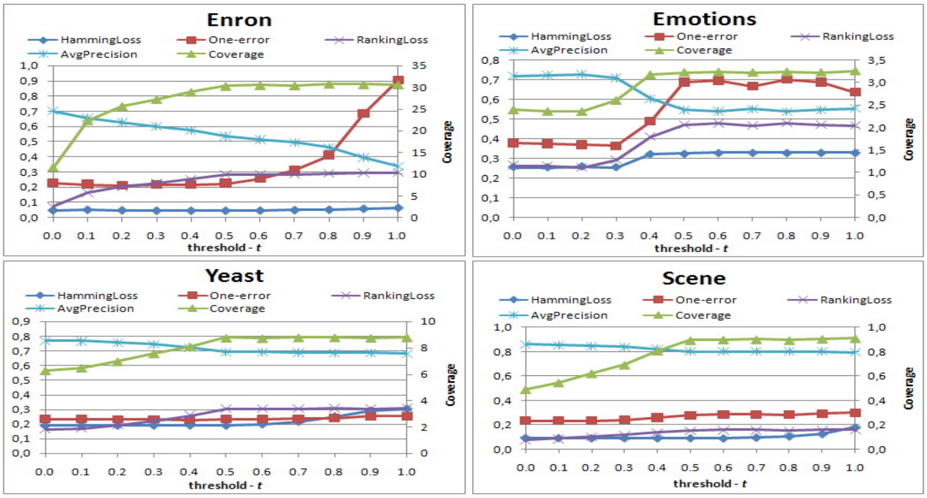

Fig. 2. Predictive performance of TSVA as a function of the threshold t ($0 \leq t \leq 1$) for each dataset

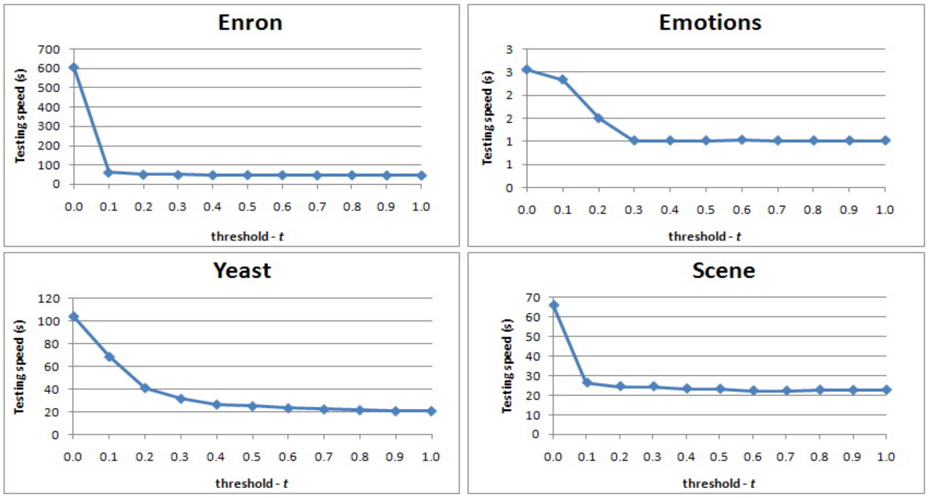

Fig. 3. Testing time of TSVA as a function of the threshold t ($0 \leq t \leq 1$) for each dataset measured in seconds

for the four classification problems as a function of the selected threshold t. It can be noticed that for small values of the threshold t (0.0 - 0.2) the predictive performance of TSVA changes moderately, but the testing time decreases for more than 40%. The reduction of the testing time of the TSVA over the CLR-S becomes even more notable as the number of labes in the treated classification problem increases. The experiments showed that for the enron dataset with quite big number of labels (53) the testing time of TSVA is four times shorter comparing to the calibrated label ranking algorithm.

5 Conclusion

A two stage voting architecture (TSVA) for efficient pair-wise multiclass voting to the multi-label setting was presented. The performance of this architecture was compared with the calibrated label ranking method with majority voting strategy for pair-wise multi-label classification and the QWeightedML algorithm on four different real-world datasets (enron, yeast, scene and emotions). The results show that the TSVA significantly outperforms the calibrated label ranking method with majority voting and the QWeightedML algorithm in term of testing speed while keeping comparable or offering better prediction performance. TSVA was 2 to 4 times faster than calibrated label ranking algorithm with majority voting and 10% to 15% faster than the QWeightedML method. TSVA is expected to show even bigger advantage when addressing classification problems with large number of labels.

References

1. Furnkranz, J.: Round robin classification. Journal of Machine Learning Research 2(5), 721–747 (2002)
2. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multiclass classification by pairwise coupling. Journal of Machine Learning Research 5(8), 975–1005 (2004)
3. Brinker, K., Furnkranz, J., Hullermeier, E.: A unified model for multilabel classification and ranking. In: 17th European Conference on Artificial Intelligence, Riva Del Garda, Italy, pp. 489–493 (2006)
4. Park, S.H., Furnkranz, J.: Efficient pairwise classification. In: 18th European Conference on Machine Learning, Warsaw, Poland, pp. 658–665 (2007)
5. Loza Mencia, E., Park, S.H., Furnkranz, J.: Efficient voting prediction for pairwise multi-label classification. Neurocomputing 73, 1164–1176 (2010)
6. Furnkranz, J., Hullermeier, E., Loza Mencia, E., Brinker, K.: Multi-label classification via calibrated label ranking. Machine Learning 73(2), 133–153 (2008)
7. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning 39(2), 135–168 (2000)
8. `http://mulan.sourceforge.net/`
9. `http://www.cs.waikato.ac.nz/ml/weka/`
10. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
11. Boutell, M.R., Luo, J., Xipeng, S., Brown, C.: Learning multi-labelscene classiffication. Pattern Recognition 37(9), 1757–1771 (2004)
12. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. Advances in Neural Information Processing Systems 14 (2001)
13. `http://bailando.sims.berkeley.edu/enron_email.html`
14. Trohidis, K., Tsoumakas, G., Vlahavas, I.: Multilabel classification of music into emotions. In: International Conference on Music Information Retrieval, Philadelphia, PA, USA, pp. 320–330 (2008)