Efficient Voting Prediction for Pairwise Multilabel Classification (resubmission)*

Eneldo Loza Mencía, Sang-Hyeun Park and Johannes Fürnkranz
TU-Darmstadt - Knowledge Engineering Group
Hochschulstr. 10 - Darmstadt - Germany

Abstract

The pairwise approach to multilabel classification reduces the problem to learning and aggregating preference predictions among the possible labels. A key problem is the need to query a quadratic number of preferences for making a prediction. To solve this problem, we extend the recently proposed *QWeighted* algorithm for efficient pairwise multiclass voting to the multilabel setting, and evaluate the adapted algorithm on several real-world datasets. We achieve an average-case reduction of classifier evaluations from n^2 to $n+dn\log n$, where n is the total number of labels and d is the average number of labels, which is typically quite small in real-world datasets.

1 Introduction

Multilabel classification refers to the task of learning a function that maps instances $\bar{\mathrm{x}} \in \mathcal{X}$ to label subsets $R_{\bar{\mathrm{x}}} \subset \mathcal{L}$, where $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ is a finite set of predefined labels, typically with a small to moderate number of alternatives. Thus, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance. The predominant approach to multilabel classification is *binary relevance learning* (BR), where one classifier is learned for each class, in contrast to pairwise learning, where one classifier is learned for each pair of classes.

While it has been shown that the complexity for training an ensemble of pairwise classifiers is comparable to the complexity of training a BR ensemble [Fürnkranz, 2002; Loza Mencía and Fürnkranz, 2008b], it remained the problem that a quadratic number of classifiers has to be evaluated to produce a prediction. Our first attempts in efficient multilabel pairwise classification lead to the algorithm MLPP, which uses the fast perceptron algorithm as base classifier. With this algorithm, we successfully tackled the large Reuters-RCV1 text classification benchmark, despite the quadratic number of base classifiers [Loza Mencía and Fürnkranz, 2008b]. Although we were able to beat the competing fast MMP algorithm [Crammer and Singer, 2003] in terms of ranking performance and were competitive in training time, the costs for testing were not satisfactory. Park and Fürnkranz [2007] recently introduced a method named *QWeighted* for multiclass problems that intelligently selects only the base classifiers that are actually necessary to predict the top class. This reduced the evaluations needed from $n(n-1)/2$ to only $n\log n$ in practice, which is near the n evaluations processed by BR.

In this paper we introduce a novel algorithm which adapts the *QWeighted* method to the MLPP algorithm. In a nutshell, the adaption works as follows: instead of stopping when the top class is determined, we repeatedly apply *QWeighted* to the remaining classes until the final label set is predicted. In order to determine at which position to stop, we use the calibrated label ranking technique [Fürnkranz *et al.*, 2008], which introduces an artificial label for indicating the boundary between relevant and irrelevant classes. We evaluated this technique on a selection of multilabel datasets that vary in terms of problem domain, number of classes and label density. The results demonstrate that our modification allows the pairwise technique to process such data in comparable time to the one-per-class approaches while producing more accurate predictions.

2 Multilabel Pairwise Perceptrons

In the pairwise binarization method, one classifier is trained for each pair of classes, i.e., a problem with n different classes is decomposed into $\frac{n(n-1)}{2}$ smaller subproblems. For each pair of classes (λ_u, λ_v), only examples belonging to either λ_u or λ_v are used to train the corresponding classifier $o_{u,v}$. In the multilabel case, an example is added to the training set for classifier $o_{u,v}$ if λ_u is a relevant class and λ_v is an irrelevant class or vice versa, i.e., $(\lambda_u, \lambda_v) \in R\times I \cup I\times R$ with $I = \mathcal{L}\backslash R$ as negative labelset. The pairwise binarization method is often regarded as superior to binary relevance because it profits from simpler decision boundaries in the subproblems [Fürnkranz, 2002; Hsu and Lin, 2002; Loza Mencía and Fürnkranz, 2008b]. This allows us to use the simple but fast (linear, one-layer) perceptron algorithm as a base classifier, so that we denote the algorithm as *Multilabel Pairwise Perceptrons (MLPP)* [Loza Mencía and Fürnkranz, 2008b]. The predictions of the base classifiers $o_{u,v}$ may then be interpreted as *preference statements* that predict for a given example which of the two labels λ_u or λ_v is preferred. In order to convert these binary preferences into a class ranking, we use a simple voting strategy known as *max-wins*, which interprets each binary preference as a vote for the preferred class. Classes are then ranked according to the number of received votes. Ties are broken randomly in our case.

To convert the resulting ranking of labels into a multilabel prediction, we use the *calibrated label ranking* approach [Fürnkranz *et al.*, 2008]. This technique avoids the

*This manuscript is a resubmission and was already published in the Proceedings of the 17^{th} European Symposium on Artificial Neural Networks, Bruges, 22 – 24 April 2009.

Require: example $\bar{\mathrm{x}}$; classifiers $\{o_{u,v} \mid u < v, \lambda_u, \lambda_v \in \mathcal{L}\}$; $l_0, \ldots, l_n = 0$

while λ_{top} not determined **do**
  $\lambda_a \leftarrow \operatorname{argmin}_{\lambda_i \in \mathcal{L}} l_i$ ▷ select top candidate class
  $\lambda_b \leftarrow \operatorname{argmin}_{\lambda_j \in \mathcal{L} \setminus \{\lambda_a\}} l_i$ **and** $o_{a,b}$ not yet evaluated ▷ select second
  if no λ_b exists **then**
    $\lambda_{top} \leftarrow \lambda_a$ ▷ top rank class determined
  else ▷ evaluate classifier
    $v_{ab} \leftarrow o_{a,b}(\bar{\mathrm{x}})$ ▷ one vote for λ_a ($v_{ab} = 1$) or λ_b ($v_{ab} = 0$)
    $l_a \leftarrow l_a + (1 - v_{ab})$ ▷ update voting loss for λ_a
    $l_b \leftarrow l_b + v_{ab}$ ▷ update voting loss for λ_b

Figure 1: Pseudocode of the *QWeighted* algorithm (multiclass classification).

dataset	n	#instances	# attributes	∅ label-set size d	density $\frac{d}{n}$	distinct
scene	6	2407	86732	1.074	17.9 %	15
yeast	14	2417	10712	4.237	30.3 %	198
r21578	120	11367	10000	1.258	1.0 %	533
rcv1-v2	101	804414	25000	2.880	2.9 %	1028
eurlex_sj	201	19596	5000	2.210	1.1 %	2540
eurlex_dc	412	19596	5000	1.292	0.3 %	1648

Table 1: Statistics of datasets.

need for learning a threshold function for separating relevant from irrelevant labels, which is often performed as a post-processing phase after computing a ranking of all possible classes. The key idea is to introduce an artificial *calibration label* λ_0, which represents the split-point between relevant and irrelevant labels. Thus, it is assumed to be preferred over all irrelevant labels, but all relevant labels are preferred over λ_0. As it turns out, the resulting n additional binary classifiers $\{\, o_{i,0} \,|\, i = 1 \ldots n\}$ are identical to the classifiers that are trained by the binary relevance approach. Thus, each classifier $o_{i,0}$ is trained in a one-against-all fashion by using the whole dataset with $\{\, \bar{\mathrm{x}} \,|\, \lambda_i \in R_{\bar{\mathrm{x}}}\} \subseteq \mathcal{X}$ as positive examples and $\{\, \bar{\mathrm{x}} \,|\, \lambda_i \in I_{\bar{\mathrm{x}}}\} \subseteq \mathcal{X}$ as negative examples. At prediction time, we will thus get a ranking over $n+1$ labels (the n original labels plus the calibration label). We denote the MLPP algorithm adapted in order to support the calibration technique as CMLPP.

3 Quick Weighted Voting for Multilabel Classification

As already mentioned, the quadratic number of base classifiers does not seem to be a serious drawback for training MLPP and also CMLPP. However, at prediction time it is still necessary to evaluate a quadratic number of base classifiers.

QWeighted algorithm: For the multiclass case, the simple but effective voting strategy can be performed efficiently with the Quick Weighted Voting algorithm (*QWeighted*), which is shown in Figure 1 [Park and Fürnkranz, 2007]. This algorithm computes the class with the highest accumulated voting mass without evaluating all pairwise perceptrons. It exploits the fact that during a voting procedure some classes can be excluded from the set of possible top rank classes early on, because even if they reach the maximal voting mass in the remaining evaluations they can no longer exceed the current maximum. For example, if class λ_a has received more than $n-j$ votes and class λ_b has lost j binary votings, it is impossible for λ_b to achieve a higher total voting mass than λ_a. Thus further evaluations with λ_b can be safely ignored for the comparison of these two classes. Pairwise classifiers will be selected depending on a *voting loss* value, which is the number of votes that a class has *not* received. More precisely, the voting loss l_i of a class λ_i is defined as $l_i := p_i - v_i$, where p_i is the number of evaluated incident classifiers of λ_i and v_i is the current number of votes for λ_i. Obviously, 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), we can conclude that no other class can achieve a better ranking. Thus, the *QWeighted* algorithm always focuses on classes with low voting loss.

QCMLPP1 algorithm: A simple adaptation of *QWeighted* to multilabel classification is to repeat the process. We can compute the top class λ_{top} using *QWeighted*, remove this class from $\mathcal{L}$ and repeat this step, until the returned class is the artificial label λ_0, which means that all remaining classes will be considered to be irrelevant. Of course, the information about which pairwise perceptrons have been evaluated and their results are carried through the iterations so that no pairwise perceptron is evaluated more than once. As we have to repeat this process until λ_0 is ranked as the top label, we know that the number of votes for the artificial label has to be computed at some point. So, in hope for a better starting distribution of votes, all incident classifiers $o_{i,0}$ respectively $\bar{\mathrm{w}}_{i,0}$ of the artificial label are evaluated explicitly before iterating *QWeighted*.

QCMLPP2 algorithm: However, QCMLPP1 still performs unnecessary computations, because it neglects the fact that for multilabel classification the information that a particular class is ranked *above* the calibrated label is sufficient, and we do not need to know *by which amount*. Thus, we can further improve the algorithm by predicting the current top ranked class λ_t as relevant as soon as it has accumulated more votes than λ_0. The class λ_t is then not removed from the set of labels (as in QCMLPP1), because its incident classifiers $o_{t,j}$ may be still be needed for computing the votes for other classes. However, it can henceforth no longer be selected as a new top rank candidate.

Complexity: It is easy to see that the number of base classifier evaluations for the multilabel adaptations of *QWeighted* is bounded from above by $n + d \cdot C_{\mathrm{QW}}$,

dataset	n	BR	CMLPP	QCMLPP1	QCMLPP2	$n \log n$	$n + dn \log n$
scene	6	6	21	11.51 *(54.8%)*	11.46 *(54.6%)*	10.75	17.50
yeast	14	14	105	67.57 *(64.4%)*	64.99 *(61.9%)*	36.94	170.65
rcv1-v2	103	103	5356	485.23 *(9.06%)*	456.23 *(8.52%)*	477.38	1649.70
r21578	120	120	7260	378.45 *(5.21%)*	325.94 *(4.49%)*	574.50	843.87
eurlex_sj	201	201	20301	1144.2 *(5.64%)*	825.07 *(4.06%)*	1065.96	2556.78
eurlex_dc	412	412	85078	2610.76 *(3.07%)*	1288.22 *(1.51%)*	2480.66	3612.05

Table 2: Computational costs at prediction in average number of predictions per instance. The italic values next to the two multilabel adaptations of *QWeighted* show the ratio of predictions to CMLPP.

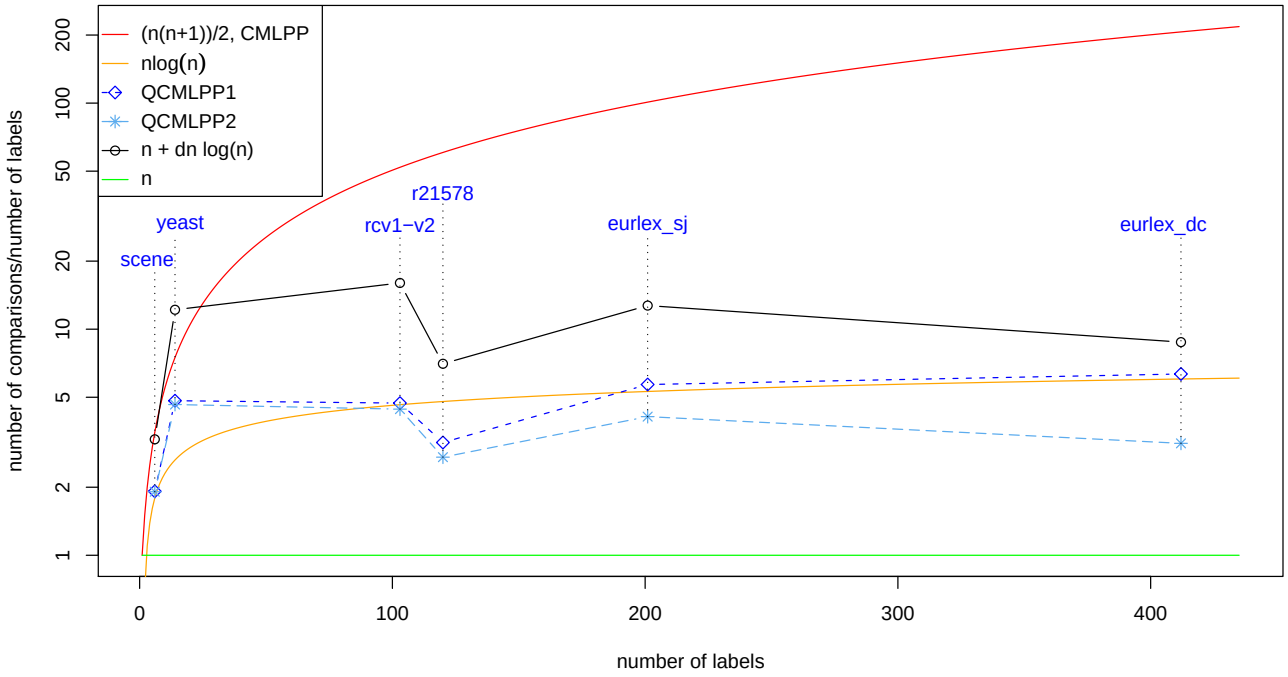

Figure 2: Prediction complexity of QCMLPP: number of comparisons needed in dependency of the number of classes n for different multilabel problems.

since we always evaluate the n classifiers involving the calibrated class, and have to do one iteration of *QWeighted* for each of the (on average) d relevant labels. Assuming that *QWeighted* on average needs $C_{QW} = n \log n$ base classifier evaluations as suggested in [Park and Fürnkranz, 2007], we can expect an average number of $n + dn \log n$ classifier evaluations for the QCMLPP variants, as compared to the $\approx n^2$ evaluations for the regular CMLPP [Fürnkranz *et al.*, 2008]. Thus, the effectiveness of the adaption to the multilabel case crucially depends on the average number d of relevant labels. We can expect a high reduction of pairwise comparisons if d is small compared to n, which holds for most real-world multilabel datasets.

4 Evaluation

Table 1 shows the multilabel datasets we used for our experiments. As the QCMLPP algorithms do not change the predictions of the CMLPP algorithm, and the superiority of the latter has already been established in other publications [Loza Mencía and Fürnkranz, 2008a,b; Fürnkranz *et al.*, 2008], we will here focus only on the computational costs. Descriptions of these datasets and results on the predictive performance may also be found in the long version of this paper [Loza Menía *et al.*, 2008]. Table 2 depicts the gained reduction of prediction complexity in terms of the average number of base classifier evaluations. In addition, we also report the ratios of classifier evaluations for the two QCMLPP variants over the CMLPP algorithm.

We can observe a clear improvement when using the *QWeighted* approach. Except for the *scene* and *yeast* datasets, both variants of the QCMLPP use less than a tenth of the classifier evaluations for CMLPP. We also add the values of $n \log n$ and $n + dn \log n$ for the corresponding datasets, which allow us to confirm that the number of classifier evaluations is smaller than the previously estimated upper bound of $n + dn \log n$ for all considered datasets. Figure 2 visualizes the above results and allows again a comparison to different complexity values such as n, $n \log n$ and n^2. Though the figure may indicate that a reduction of classifier evaluations to $n \log n$ is still achievable for multilabel classification, especially for QCMLPP2, we interpret the results more cautiously and only conclude that $n + dn \log n$ can be expected in practice.

5 Conclusions

The main disadvantage of the pairwise approach in multilabel classification was, until now, the quadratic number of base classifiers needed and hence the increased computational costs for computing the label ranking that is used for partitioning the labels in relevant and irrelevant labels. The presented QCMLPP approach is able to significantly reduce these costs by stopping the computation of the la-

bel ranking when the bipartite separation is already determined. Though not analytically proven, our empirical results show that the number of base classifier evaluations is bounded from above by $n + dn \log n$, in comparison to the evaluation of n in the case of binary relevance ranking and n^2 for the unmodified pairwise approach.

The key remaining bottleneck is that we still need to store a quadratic number of base classifiers, because each of them may be relevant for some example. We are currently investigating alternative voting schemes that use a static allocation of base classifiers, so that some of them are not needed at all. In contrast to the approach presented here, such algorithms may only approximate the label that is predicted by the regular pairwise classifier.

Acknowledgements

This work was supported by the EC 6th framework project *ALIS (Automated Legal Information System)* and by the *German Science Foundation (DFG)*.

References

Koby Crammer and Yoram Singer. A Family of Additive Online Algorithms for Category Ranking. *Journal of Machine Learning Research*, 3(6):1025–1058, 2003.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

Johannes Fürnkranz. Round Robin Classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

Chih-Wei Hsu and Chih-Jen Lin. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

Eneldo Loza Mencía and Johannes Fürnkranz. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Disocvery in Databases (ECML-PKDD-2008), Part II*, pages 50–65, Antwerp, Belgium, 2008.

Eneldo Loza Mencía and Johannes Fürnkranz. Pairwise learning of multilabel classifications with perceptrons. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IJCNN 08)*, pages 2900–2907, Hong Kong, 2008.

Eneldo Loza Menía, Sang-Hyeun Park, and Johannes Fürnkranz. Advances in efficient pairwise multilabel classification. Technical Report TUD-KE-2008-06, TU Darmstadt, Knowledge Engineering Group, 2008.

Sang-Hyeun Park and Johannes Fürnkranz. Efficient pairwise classification. In *Proceedings of 18th European Conference on Machine Learning (ECML-07)*, pages 658–665, Warsaw, Poland, 2007.