Multi-Label Learning Approaches for Music Instrument Recognition

Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, and Ioannis Vlahavas

Department of Informatics, Aristotle University of Thessaloniki, 54124 Greece
espyromi@csd.auth.gr, greg@csd.auth.gr, vlahavas@csd.auth.gr

Abstract. This paper presents the two winning approaches that we developed for the instrument recognition track of the ISMIS 2011 contest on Music Information. The solution that ranked first was based on the Binary Relevance approach and built a separate model for each instrument on a selected subset of the available training data. Moreover, a new ranking approach was utilized to produce an ordering of the instruments according to their degree of relevance to a given track. The solution that ranked second was based on the idea of constraining the number of pairs that were being predicted. It applied a transformation to the original dataset and utilized a variety of post-processing filters based on domain knowledge and exploratory analysis of the evaluation set. Both solutions were developed using the Mulan open-source software for multi-label learning.

1 Introduction

With the explosion of multimedia Internet services, such as YouTube and Last.fm, vast amounts of multimedia data are becoming available for exchange between Internet users. Efficiently browsing and searching in such enormous digital databases requires effective indexing structures. However, content-based indexing of multimedia objects such as music tracks is commonly based on manual annotations, since automatic understanding and categorization is still too difficult for computers. Here we focus on the challenging problem of recognizing pairs of instruments playing together in a music track.

While the automatic recognition of a single instrument is fairly easy, when more than one instruments play at the same time in a music track, the task becomes much more complex. The goal of the competition was to build a model based on training data concerning both single instruments and instrument mixtures in order to recognize pairs of instruments. The learning task can be viewed as a special case of multi-label classification [5], where the output of the classifier must be a set of exactly two labels.

Multi-label classification extends traditional single-label classification in domains with overlapping class labels (i.e. where instances can be associated with more than one labels simultaneously). More formally, in multi-label classification $\mathcal{X} = \mathbb{R}^M$ denotes the input attribute space. An *instance* $\boldsymbol{x} \in \mathcal{X}$ can

be represented as an M-vector $\boldsymbol{x} = [x_1, \ldots, x_M]$. The set of L possible labels $Y = \{1, \ldots, L\}$ for a particular instance is represented by an L-vector $\mathbf{y} = [y_1, \ldots, y_L] = \{0, 1\}^L$ where $y_j = 1$ iff the jth label is relevant ($y_j = 0$ otherwise). A multi-label classification algorithm accepts as input a set of multi-label training examples $(\boldsymbol{x}, \mathbf{y})$ and induces a model that predicts a set of relevant labels for unknown test instances. In the domain of the contest, music instruments represent the overlapping class labels and each test instance is associated with exactly two of them.

Both solutions presented in this paper are based on the simple Binary Relevance (BR) approach for multi-label classification. BR learns L binary classifiers, one for each different label in Y. It transforms the original data set into L datasets that contain all examples of the original dataset, labeled positive if the original example was annotated with y_j and negative otherwise. For the classification of a new instance, BR outputs the union of the labels y_j that are positively predicted by the L classifiers.

The quality of predictions were evaluated by the contest organizers as follows:

- If no recognized instrument matched the actual ones, the score was 0
- If only one instrument was correctly recognized, the score was 0.5
- If both instruments matched the target ones, the score was 1.0

The final score of a solution was the average of its scores across all test instances.

The rest of the paper is organized as follows. Section 2 presents our exploratory analysis of the datasets. Sections 3 and 4 describe in detail the two solutions that we developed. Finally, Section 5 concludes this paper.

2 The Data

2.1 The Training Sets

The training set consisted of two datasets: one containing data of single instruments and one containing data of mixtures of instrument pairs. A first challenge of the contest, was that these two datasets were significantly heterogeneous.

The single instrument data comprised 114914 recordings of 19 different instruments. The instrument pairs data comprised just 5422 mixtures of 21 different instruments. In total there were 32 distinct instruments, just 8 of which appeared in both datasets. Table 1 presents the number and percentage of examples from each instrument in each of the two training datasets as well as in their union.

The mixtures dataset contained the following 12 different pairs of 21 instruments: (SopranoSaxophone, TenorTrombone), (AltoSaxophone, TenorTrombone), (TenorSaxophone, Tuba), (TenorSaxophone, B-FlatTrumpet), (BaritoneSaxophone, CTrumpet), (BassSaxophone, CTrumpet), (AcousticBass, Piano), (B-flatclarinet, Viola), (Cello, Oboe), (ElectricGuitar, Marimba), (Accordian, DoubleBass), (Vibraphone, Violin).

It is interesting to notice that the pairs dataset contained instruments that can be considered as *kinds* of instruments in the single instruments dataset.

Instrument	Single Examples	%	Pairs Examples	%	Total Examples	%	Validation Examples	%	Test Examples	%
SynthBass	918	0.7	0	0	918	1	595	4.0	635	4.3
EnglishHorn	1672	1.4	0	0	1672	1	0	0.0	0	0.0
Frenchhorn	2482	2.1	0	0	2482	2	364	2.4	425	2.9
Piccolo	2874	2.5	0	0	2874	2	0	0.0	0	0.0
Saxophone	4388	3.8	0	0	4388	3	0	0.0	0	0.0
Trombone	4503	3.9	0	0	4503	4	0	0.0	0	0.0
Bassoon	5763	5.0	0	0	5763	5	0	0.0	0	0.0
Flute	7408	6.4	0	0	7408	6	0	0.0	0	0.0
Clarinet	9492	8.2	0	0	9492	8	0	0.0	0	0.0
Trumpet	11152	9.7	0	0	11152	9	0	0.0	0	0.0
Guitar	34723	30.2	0	0	34723	28	0	0.0	0	0.0
Vibraphone	0	0	249	4.6	249	0	622	4.2	594	4.1
SopranoSaxophone	0	0	329	6.1	329	0	586	4.0	542	3.7
AltoSaxophone	0	0	337	6.2	337	0	405	2.7	388	2.6
B-FlatTrumpet	0	0	412	4	412	0	0	0.0	0	0.0
AcousticBass	0	0	417	4	417	0	0	0.0	0	0.0
BassSaxophone	0	0	503	9.2	503	0	340	2.4	335	2.4
BaritoneSaxophone	0	0	530	9.8	530	0	346	2.3	296	2.0
B-flatclarinet	0	0	567	10.5	567	0	3528	24.1	3661	25.0
ElectricGuitar	0	0	590	10.9	590	0	4149	28.3	4222	28.8
Marimba	0	0	590	10.9	590	0	455	3.1	377	2.6
TenorTrombone	0	0	666	12.2	666	1	1293	8.8	1248	8.5
TenorSaxophone	0	0	934	17.2	934	1	2137	14.6	2160	14.7
CTrumpet	0	0	1033	19.0	1033	1	2155	14.7	2048	14.0
Oboe	1643	1.4	332	6.1	1975	2	3184	21.7	3247	22.1
Accordian	1460	1.2	634	11.7	2094	2	2466	16.8	2427	16.6
Viola	3006	2.6	567	10.4	3573	3	762	5.2	815	5.6
Tuba	3463	3.0	522	9.6	3985	3	0	0	0	0.0
DoubleBass	3849	3.3	634	11.7	4483	4	1384	9.4	1338	9.1
Violin	5010	4.4	249	4.6	5259	4	3528	22.2	3237	22.1
Cello	4964	4.3	332	6.1	5296	4	698	4.7	694	4.7
Piano	6144	5.3	417	7.7	6561	5	595	4.0	635	4.3

Table 1. Instrument distribution in the training sets and the test set.

SopranoSaxophone, AltoSaxophone, TenorSaxophone, BaritoneSaxophone and BassSaxophone are kinds of Saxophone, CTrumpet and B-FlatTrumpet are kinds of Trumpet, TenorTrombone is a kind of Trombone, B-FlatClarinet is a kind of Clarinet and ElectricGuitar is a kind of Guitar. These relations complicate the learning problem in some ways. Firstly, examples of the specialized class (e.g. TenorTrombone) could be semantically considered as examples of the general class (e.g. Trombone). It may be difficult to distinguish between such parent-child pairs of classes. Secondly, different kinds of the same instrument could be difficult to distinguish (e.g. is one of the instruments a soprano or an alto saxophone?).

It is also interesting to notice a special property of the *ElectricGuitar* and *Marimba* instruments. It is the only pair of instruments that satisfies both of the following conditions: a) none of the instruments of the pair appears in the single instruments dataset, b) none of the instruments of the pair appears together with another instrument in the mixtures dataset. This means that out of the 32 instruments, these particular two instruments are the only ones to have exactly the same positive and negative training examples. It would therefore be impossible for any classifier to distinguish them.

2.2 The Test Set

Besides the heterogeneity of the training sets, the following statements about the synthesis of the test set brought additional complexity to the learning task:

- Test and training sets contain different pairs of instruments (i.e. the pairs from the training set do not occur in the test set).
- Not all instruments from the training data must also occur in the test part.
- There may be some instruments from the test set that only appear in the single instruments part of the training set.

In order to have a clearer idea about the synthesis of the test set and for other reasons which will be explained in the analysis of the respective solutions, we queried the evaluation system for the frequency of each instrument in the test set by submitting a prediction containing the same instrument for all test instances. Table 1 (Column 4) contains the percentage of each label measured in the validation set (35% of the test data) along with a projection of the expected number of examples in the full test set. Column 5 of Table 1 contains the actual percentage and number of examples of each label in the full test set.

By examining Table 1, we reach to the following conclusions:

- Only 20 out of the 32 instruments appear in the test set.
- The mixtures training set contained 18 of the 20 instruments of the test set plus 3 additional instruments.
- The single instruments training set contained 9 of the 20 instruments of the test set plus 10 additional instruments.
- There is a great discrepancy between the distribution of the labels in the training and the test data.

2.3 Features

Each track in both the training and the test data was described by 120 precomputed attributes capturing various sound properties:

- Flatness coefficients: BandsCoef1-33, bandsCoefSum
- MFCC coefficients: MFCC1-13
- Harmonic peaks: HamoPk1-28
- Spectrum projection coefficients: Prj1-33, prjmin, prjmax, prjsum, prjdis, prjstd
- Other acoustic spectral features: SpecCentroid, specSpread, energy, log spectral centroid, log spectral spread, flux, rolloff, zerocrossing
- Temporal features: LogAttackTime, temporalCentroid

The single instruments set was described by the following additional five attributes:

- Frameid - Each frame is 40ms long signal
- Note - Pitch information
- Playmethod - One schema of musical instrument classification according to the way they are played
- Class1,class2 - Another schema of musical instrument classification according to Hornbostel-saches

3 Investigation of Multi-Label Learning Methods

A first important issue was to determine which multi-label learning method was the most appropriate one for our particular problem. We compared the performance of various state-of-the-art multi-label methods that were available in our Mulan open-source software for multi-label learning[1], such as ECC [3], CLR [2] and RAKEL [6] along with baseline methods such as the Binary Relevance (BR) approach. In this first set of experiments the union of the two datasets (single and mixtures) was used as the training set and the performance of the methods was evaluated directly on the test set. The reason was that the training data was substantially different from the test data (see Section 2) and the results of a comparison on the training data could be misleading.

The results of a comparison using various binary base classifiers revealed that state-of-the-art multi-label methods had little or no benefit in comparison with the simple BR approach, especially when BR was coupled with ensemble-based binary classifiers such as Random Forest [1]. The results were not surprising since the main advantage of advanced multi-label learning methods over the BR approach is their ability to capture and exploit correlations between labels. In our case, learning the correlations which appear in the training set was not expected to be useful since these correlations are not repeated in the test set.

4 The Solution that Ranked First

4.1 Engineering the Input

While in our initial set of experiments we used the union of the given training sets, we were also interested in measuring the performance of the methods given either only the mixture or only the single-instrument examples as training data. The results showed that using only the mixture examples for training was far better than using only the single-instrument examples, and was even better than using all the available training examples. We gave two possible explanations for this outcome:

- Learning from pairs of instruments is better when the task is to predict pairs of instruments (even though the pairs appearing in the test set are different).
- The distribution of the labels in the mixtures dataset matches better to that of the test set.

The findings regarding the nature of the test set, presented in Subsection 2.2, were quite revealing. By using only the single-instruments set for training, we could predict only 9 of the 20 instruments which appear in the test set, compared to 18 when using the mixtures set. However, it was still difficult to determine why using the mixtures set alone was better than combining all the data since, in the latter case, all the relevant instruments were present in the training set. To make things more clear we performed a new set of experiments.

[1] mulan.sourceforge.net

We first removed the training data corresponding to the 12 instruments which were not present in the test set and then created the following training sets: a) One that contained both mixture and single-instrument examples for the instruments appearing in the test set. b) One that contained only mixture examples for the 18 out of 20 instruments and single-instrument examples for the 2 remaining instruments of the test set. c) One that contained only single-instrument examples for the 9 out of 20 instruments and mixture examples for the rest 11 instruments of the test set. The best results were obtained using the second training set, and verified that learning from mixtures is better when one wants to recognize mixtures of instruments. Note that adding single-instrument examples for the 2 instruments which had no examples in the mixtures set, slightly improved the performance of using only examples of mixtures. This revealed that using single-instrument data can be beneficial in the case that no mixture data is available. The set used to train the winning method comprised of the union of the 5422 mixture examples and the 340 single-instrument examples of SynthBass and Frenchhorn. All the given feature attributes describing the mixture examples were used, while we ignored the 5 additional attributes of the single-instruments set since they were not present in the test set.

4.2 Base Classifier

A problem arising from the use of the one-versus-rest or BR approach for multi-label classification is that most of the labels have much more negative than positive examples. Class imbalance is known to negatively affect the performance of classifiers by biasing their focus towards the accurate prediction of the majority class. This often results in poor accuracy for the minority class, which is the class of interest in our case. For this reason, special attention was paid on selecting a classification scheme that is able to tackle this problem.

To deal with class imbalance we extended the original Random Forest (RF) [1] algorithm. RF creates an ensemble of unpruned decision trees where each tree is built on a bootstrap sample of the training set. Random feature selection is used in the tree induction process. To predict the class of an unknown object the predictions of the individual trees are aggregated. RF has been proven to have superior accuracy among current classification algorithms, however, it is susceptible on imbalanced learning situations. Our idea is based on combining RF with Asymmetric Bagging [4]. Instead of taking a bootstrap sample from the whole training set, bootstrapping is executed only on the examples of the majority (negative) class. The Asymmetric Bagging Random Forest (ABRF) algorithm is given below:

1. Take a sample with replacement from the negative examples with size equal to the number of positive examples. Use all the positive examples and the negative bootstrap sample to form the new training set.
2. Train the original RF algorithm with the desired number of trees on the new training set.
3. Repeat the two steps above for the desired number of times. Aggregate the predictions of all the individual *random trees* and make the final prediction.

Building a forest of 10 random trees on each one of 10 balanced training sets yielded the best evaluation results.

4.3 Informed Ranking

The output produced for each label by an ABRF classifier can be used either as a hard classification (the decision of the majority) or transformed into a confidence score of the label being true by dividing the number of random trees that voted for the label with the total number of random trees. In a typical multi-label classification problem (where the number of relevant labels for each test instance is unknown) we would either use the first approach to select the relevant labels for each test instance, or apply a decision threshold to the confidence scores in order to transform them into hard classifications. In the domain of the contest though, we a priori knew that exactly two instruments are playing on each track, thus we followed a different approach. We focused on producing an accurate ranking of the labels according to their relevance to each test instance and selected the two top-ranked labels. Instead of directly using the confidence scores to produce a ranking of the labels, we developed a novel ranking approach which takes into account the prior probability distribution of the labels. Our approach is as follows:

1. Use the trained classifiers to generate confidence scores for all test instances.
2. Sort the list of confidence scores given for each label.
3. Given a test instance, find its rank in the sorted list of confidences for each label. These ranks are indicative of the relevance of the instance to each label.
4. Normalize the ranks produced from step 3 by dividing them with the estimated (based on their prior probabilities) number of relevant instances for each label in the test set and select the n labels with the lowest normalized rank.

We explain the effect of normalization with an example: Assume that we have 100 test instances and an instance x_i is ranked 30th for label1 and label2 and 40th for label3. We further know that only one label is relevant for x_i and that the prior probabilities of the labels are $P(label1) = P(label2) = 0.25$ and $P(label3) = 0.5$. By normalizing the ranks we get 30/25 for label1 and label2 and 40/50 for label3. Thus, we would select label3 for x_i although label1 and label2 have a lower absolute rank. This is rational since based on the priors we expect that label1 and label2 will have only 25 relevant instances and x_i's rank for these labels was 30. In the context of the contest, we had the chance to use the frequencies of the labels in the validation set to estimate the number of relevant instances in the full test set. In a real-world situation, the prior probabilities of the labels in the training set could be used for this purpose.

4.4 Engineering the Output

As a final step, a post-processing filter was applied which disallowed instrument pairs that were present in the training set. In such cases, the second-ranked label

was substituted by the next label which would not produce a label pair of the training set when combined with the first-ranked label. This substitution was based on the assumption that the classifier is more confident for the first-ranked label. The information for this filter was given in the description of the task by the contest organizers (see Section 2).

5 The Solution that Ranked Second

The mixtures dataset consists of 5422 examples, yet the number of distinct instrument pairs it contains is just 12. This observation, led us to the hypothesis that the test set, which consists of 14663 instances, might also contain a small number of instrument pairs. However, the number of distinct instrument pairs predicted by our early attempts on the problem was quite large. This led to the core idea of this solution: constraining the number of pairs that were being predicted.

5.1 Engineering the Input

A first step was to join the two training datasets into a single one. The extra features of the single-instruments dataset were deleted in this process, while the label space of the datasets was expanded to cover the union of all labels. The union of the examples of the two datasets was then considered.

We then adopted the following transformation of this dataset. We considered a new label space consisting of all pairs of instruments. The labels of this new label space had a positive value, whenever one of the labels in the original space, i.e. one of the instruments, had a positive value. In other words, the new label space applied an OR operator on all pairs of the original labels space. Figure 1 exemplifies this process with just three instruments, respecting the pairs that appear in the training set.

Cello	Oboe	Piano		Cello OR Oboe	Cello OR Piano	Oboe OR Piano
true	true	false		true	true	true
true	false	false	⇒	true	true	false
false	true	false		true	false	true
false	false	true		false	true	true

Fig. 1. Transformation of the data to a new label space.

This quite strange transformation was motivated from the fact that the task required us to predict pairs of instruments, but didn't provide us with examples of mixtures of these pairs. The transformation allowed the direct modeling of all pairs, using as examples either available mixtures, or available examples of one of the two instruments.

5.2 Learning

We applied the binary relevance approach on the transformed dataset. Each of the binary models was trained using the random forest algorithm [1] with 200 trees, after random sub-sampling so as to have at most a 10:1 ratio between the negative and positive class. Given a test instance, the output of this approach was a ranking of the labels (pairs of instruments) according to relevance to each of the test instances, based on the probability estimates of the random forest algorithm.

5.3 Engineering the Output

As already mentioned in the beginning of this section, the key point of this solution was constraining the instrument pairs given in the output. This was achieved via a variety of filters operating at a post-processing step after the learning step has ranked all possible pairs of instruments.

A first simple post-processing filter disallowed instrument pairs that were present in the training set. The information for this filter was given in the description of the task by the contest organizers (see Section 2). A second filter disallowed instrument pairs, where at least one of the instruments was absent from the evaluation set, as discovered from our exploratory analysis of the evaluation set (see Section 2).

Then a number of filters were applied, one for each instrument that was present in the evaluation set, which disallowed pairs of this instrument with other instruments based on two main information sources:

- Domain knowledge, which was sought in the Internet, as our musical literacy was rather limited for this task. The Vienna Symphonic Library[2] was a good source of knowledge for combinations of instruments that make sense. We also issued Google queries for pairs of instruments and considered the number of returned documents as evidence supporting the common appearance of these instruments in a music track. Sites with free music pieces for instruments were also consulted.
- The projected instrument distribution in the test set based on the evaluation set (see Section 2). Instruments with predicted distribution much higher than the projected one, hinted us that pairs containing them should be candidates for removal from the allowed set of instrument pairs. On the other hand, instruments with predicted distribution much lower than the projected one, hinted us that perhaps we have wrongly disallowed pairs containing them.

Constructing this last set of filters was a time-consuming iterative process, involving several submissions of results for evaluation feedback, that in the end led to allowing just 20 instrument pairs. After the test set was released, we found out that it actually contained 24 instrument pairs, 13 of which were within the allowed 20 by our approach. The remaining 11 were disallowed by our approach, which further allowed 7 pairs that were not present in the test set.

[2] http://www.vsl.co.at/

Instrument pairs were examined in the order of relevance to a test instance as output by the learning algorithm, until a pair that was not disallowed by the filters was reached. This was the final output of the post-processing algorithm for that test instance.

We also included another post-processing step that led to slight improvements. This step took into account the parent-child relationships of instruments that were discussed in Section 2 and performed the following replacements of instruments in a predicted pair, prior to passing this pair from the filters: Clarinet was replaced by B-FlatClarinet, Trumpet by CTrumpet, Guitar by ElectricGuitar and Trombone by TenorTrombone.

6 Conclusions

Our motivation for participating in the instrument recognition track of the ISMIS 2011 Contest on Music Information Retrieval was to explore the potential of multi-label learning methods [5].

One interesting conclusion was that in multi-label learning problems, like the one of this contest, where modeling label correlations is not useful, combining simple multi-label learning techniques, such as Binary Relevance with strong single-label learning techniques, such as Random Forest, can lead to better performance compared to state-of-the-art multi-label learning techniques. Another interesting conclusion derived from the solution that ranked first was that it is better to use only mixture examples when pairs of instruments need to be recognized.

An interesting direction for the next year's contest would be the generalization of the task to the recognition of an arbitrary number of instruments playing together.

References

1. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (October 2001)
2. Fürnkranz, J., Hüllermeier, E., Mencia, E.L., Brinker, K.: Multilabel classification via calibrated label ranking. Machine Learning (2008)
3. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Proceedings of ECML PKDD '09. pp. 254–269 (2009)
4. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1088–1099 (2006)
5. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, chap. 34, pp. 667–685. Springer, 2nd edn. (2010)
6. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multi-label classification. IEEE Transactions on Knowledge and Data Engineering (2011)