# Quality Assessment of Multi-Label Classification for Music Data

Alicja Wieczorkowska and Piotr Synak

Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, 02-008 Warsaw, Poland
{alicja, synak}@pjwstk.edu.pl

**Abstract.** This paper investigates problems related to quality assessment in case of multi-label automatic classification of data. Various methods of assigning classes, as well as measures of assessing the quality of classification results are proposed and investigated both theoretically and in practical tests. In our experiments, audio data representing short music excerpts of various emotional contents were parameterized and then used for training and testing. Class labels represented emotions assigned to a given audio excerpt. The experiments show how various measures influence quality assessment of automatic classification of multi-label data.

## 1 Introduction

Automatic data classifiers, where a tested object is assigned to one of pre-defined classes, are broadly used worldwide and they are very useful in many applications. However, some data do not fit into this classification scheme. For instance, when listening to a piece of music from an audio database, one can feel various emotions, and when such data are classified with respect to these emotional states, multi-label classification is much more useful. In this case, each piece of music cane be labelled with various emotions associated to this music. Therefore, the authors decided to investigate multi-label classification of data, how one can produce a classifier, and how the classification quality can be tested in multi-label case.

## 2 Multi-Label Classification

Multi-label classification has already been performed in numerous applications in text mining and scene classification domains, where documents or images can be labelled with several labels describing their contents [1], [2], [5]. Such a classification requires considering additional issues, including the selection of the training model, as well as set-up of testing and evaluation of results.

The following scenarios can be used for training of a classifier using examples with multiple labels:

– *MODEL-s* - the simplest model, assuming labelling of data by using single, the most likely label,

- *MODEL-i* - in this model, all the cases with more than one label are ignored, but in such a model data for training do not even exist in fully multi-labelled data set,
- *MODEL-n* - in this case, new classes are created for each combination of labels occurring in the training sample; however, in this model the number of classes easily becomes very large, especially, if the number of labels is only limited by the number of available labels, and for such a huge number of classes the data may easily become very sparse, so some classes may have very few training samples,
- *MODEL-x* - in this model a cross-training is performed, where samples with many labels are used as positive examples (and not as negative examples) for each class corresponding to the labels.

In our experiments, we decided to follow the *MODEL-x*, since we consider it to be the most efficient model.

## 3 *K*-NN for Multi-Label Classification

In this chapter we propose a $k$-nearest neighbor ($k$-NN) classifier adapted to multi-label classification problem.

### 3.1 Classification Process

Let us consider $k$-NN classifier with variable $k$. The standard $k$-NN classifier can be modified to suit multi-label data. First of all, in order to assign a set of labels to the tested object, the histogram presenting the number of appearances of each class label in the neighborhood is calculated. Then, two versions of the algorithm assigning class labels are proposed, and then also practically tested. In the first version, the assignment of classes for the tested object is performed on the basis of the measure $p_1 \in [0, 1]$, assigned to each label from the k-neighborhood, and calculated in such a way that the histogram value is normalized by the number of all labels (including repetitions) returned by the algorithm for a given testing sample (see Figure 1).

This $p$ can be considered as a probability measure. The initial output of the classification algorithm is a set of labels with assigned probability $p_1$, and only the labels exceeding some threshold level (chosen experimentally) are yielded as a final classification result. However, the global threshold is not the best choice, since the more labels assigned to the object, the lower probabilities are obtained. Therefore, a local threshold, chosen with respect to the obtained $k$-NN distribution, may yield better results.

Of interest is also to consider another measure $p$ for assignment of class labels into the tested sample, or to consider a different threshold. There is particular reason for such consideration. Let us investigate 2 similar cases. If we have $k$-NN with $k = 6$, the tested object representing class A, and the following 6 results:

1. {A},

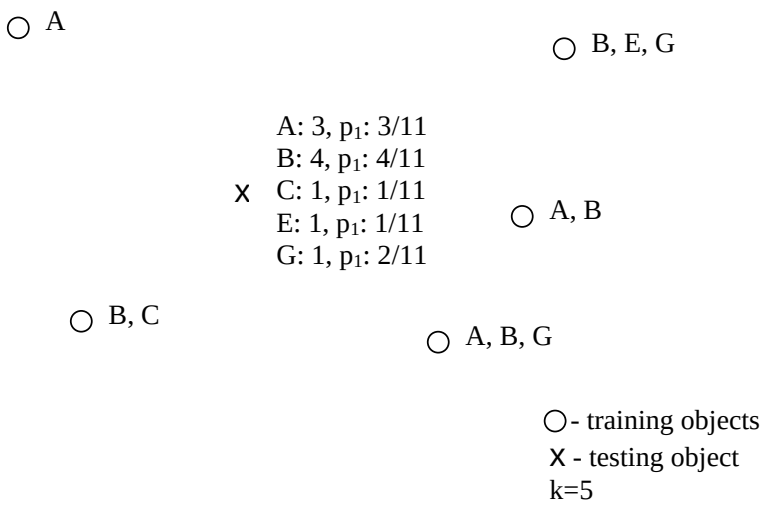


**Fig. 1.** Assignment of the measure $p_1$ for each class in the multi-class $k$-NN

2. {A},
3. {A},
4. {A},
5. {A},
6. {A},

then we obtain $p_1$=1 for the class A and perfect classification. But on the other hand, if we have $k$-NN with $k = 6$, the tested object representing classes {A, B, C, D, E} and the following 6 results:

1. {A, B, C, D, E},
2. {A, B, C, D, E},
3. {A, B, C, D, E},
4. {A, B, C, D, E},
5. {A, B, C, D, E},
6. {A, B, C, D, E},

then the probability $p_1$ assigned to each class is equal to 1/5. Therefore, $k$-NN again yields perfect match of class labels. However, if the threshold is high, then all these class labels can be rejected. This is why the authors decided to introduce a different measure, $p_2$, calculated in such a way that the histogram value for each class was normalized by $k$, i.e. the number of neighbors (see Figure 2).

In this measure, all class labels are considered separately, and higher values of the measure are obtained. Therefore, the algorithm is not sensitive to the number of labels in $k$-neighborhood.

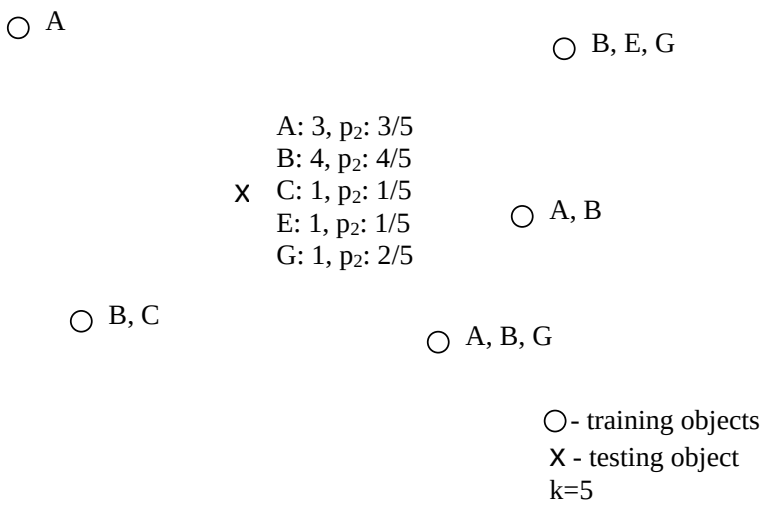


**Fig. 2.** Assignment of the measure $p_2$ for each class in the multi-class $k$-NN

### 3.2 Testing: Classification Accuracy Measurements

In order to assess the correctness of the obtained classification results, the measure of accuracy has to be defined.

Let $X$ - set of labels for a tested object, found by the $k$-NN algorithm. Let $Y$ - set of original labels for this object. Then, the classification accuracy $acc$ for this object is defined as

$$acc = \frac{card(X \cap Y)}{card(X \cup Y)}.$$

For instance, if $X$={A,B,C,E} - labels found by $k$-NN, and $Y$={A,B,D} - original labels, then $acc = 2/5$. The final measure was averaged across the entire test set.

Of course, the definition of the measure determines the obtained accuracy level, so if another measure is defined, worse or better accuracy can be obtained, and better result may only mean that the measure is more strict, or not so strict with respect to the classification errors.

When defining the accuracy measure, it cannot depend on the method how the labels are chosen by the algorithm, so any of the probabilities (or neighbors) cannot be taken into account. The measure of the accuracy may only depend on the original labels and the labels assigned by the algorithm.

On one hand, one may want to consider separately precision and recall of the classifier, because the collective measure always depends on how false positives and omissions influence the final classification accuracy measure. Therefore, in the simplest way one can consider:

- precision=(number of labels correctly recognized by the algorithm)/(number of all classes assigned by the algorithm)

– recall=(number of labels correctly recognized by the algorithm)/(number of all classes actually labelling this object)

For instance, if X={A, B, C} - actual (original) labels of the object, Y={A, B} - labels found by the algorithm, then we have an omission. Therefore, although the precision for Y is equal to 1, the recall is equal to 2/3. If the original labels are X={A, B, C, D, E}, and the labels found by the algorithm are Y={A, B, C, D, E, F}, then we have a false positive and the precision is equal to 5/6, although the recall is equal to 1.

On the other hand, the total error for the entire test set is useful for assessing the quality of the classification. Therefore, the measure *acc* was used in this research, since it estimates accuracy of the entire classification process.

One can also use the most strict, binary measure, when 0 accuracy is assigned if the recognition result differs for a given object, and 1 when the labels found by the algorithm are identical with the original labelling. However, this would be unjust measure, since it seems to be too strict.

## 4 Binary Classification

Apart from classifying data in multi-class case, one can also consider binary classification, where each class is recognized against the rest of the classes, for each class separately. Any classifier can be applied for this purpose. In the training phase for a given class, all objects which are labelled with this class (and possibly with other classes) are considered as positive examples, whereas objects do not containing this class label are considered as negative examples. The weights can be also assigned to classification rules, to set up priorities of using them in the testing phase. For instance, rules against a given class can be assigned lower weight (or even discarded).

For such a binary classification, even for single-label classification the results can be much higher, since in this case the recognition accuracy mainly depends on the number of objects representing each class, and how big is the class in proportion to the entire data set. Generally, if the data set consists of numerous classes, then even the classifier that always votes against the tested class has the accuracy level equal to 1-(number of objects in the testes class)/(number of objects in the entire data set). Therefore, high recognition accuracy is easy to obtain in this case, but it does not imply good discernment power of the classifier. Instead, one should rather consider general recognition. Since one is usually interested in checking how the results reflect the classification abilities of the algorithm and descriptors, one should focus on the issue how various improvements influence final classification quality, i.e. if it has been improved or deteriorated. However, the binary recognition evokes a few interesting issues on how to assign class labels and estimate classification results.

For example, in case of multi-label classification one must consider the following situation. If the object X={A, B, D} is used in training for the class A (against the other classes), how to assess the accuracy, if the algorithm recognizes

it as B, or as D? One of the possibilities is to introduce additional recognition class, neutral, and assign it in this case, i.e. consider the result neither correct nor incorrect (but then we risk "neutral" assessment even for a perfect classifier, for instance showing {A, B, D} in this case); if other classes are assigned by the algorithm, the accuracy should be equal to zero. Of course, the same object labelled by the algorithm as {A, B, D} should be assessed as correctly classified. So, if one performs the experiment for the class A, then yielded labelling that includes this class should be considered correct. However, all class labels not included in the original labelling of the test object (for instance G in this case) should lower the final accuracy measure. Also, if the classifier yields X={A, B} for this case, the accuracy measure should be lower. The final measure should consider punishment for wrong labels assigned to the tested object. Also, the accuracy estimation of the algorithm should consider the fact, that the object labelled {A, B} is different than the object labelled {A}. For instance, if the object {A, B, C, D, E, F} is tested, it is difficult to decide which class it should actually represent; if the test for the A class against the rest is performed, then this particular object may be considered as representing other classes to higher extent than it represents class A. On the other hand, if the classifier assigns {A, G, H, I, J, K, L, M} to this object, then it is still considered correct, even if it is almost totally wrong. One can even consider each combination of labels as a separate super-label, but this approach would lead to numerous classes with very few representants, not of much use for experiments.

Generally, the measure should take into account the length of original labelling of the object and the length of the labelling found by the classification algorithm (and possibly the number of available classes). In standard setting, the weights for both omissions and false positives should be the same. One can also be interested in observation which classes are most difficult for the classifier, i.e. for which classes the classifier is most frequently wrong, and with which classes the objects are mistaken.

## 5 Experiments

The theoretical issues investigations in the previous sections has been practically tested on the data representing audio sequences, labelled with emotions they evoke when human subjects listen to them.

### 5.1 Data

The data used in experiments represented 875 audio excerpts from musical recordings. Each audio fragment was 30 second long, recorded stereo in .mp3 format and then converted to .au format, with 44100 Hz sampling frequency and 16-bit resolution. Spectral analysis was performed on 32768 samples long analyzing frame, taken from the left channel, with Hanning window applied. The spectral components up to 12 kHz and no more than 100 partials were taken into account when calculating the feature vector, describing various aspects of

long-time spectrum. The following 29 parameters constitute the feature vector [7], [8]:

- $Frequency$: dominating fundamental frequency
- $Level$: maximal level of sound
- $Tristimulus1, 2, 3$: Tristimulus parameters for $Frequency$, describing strength of the fundamental, the middle partials (no. 2, 3, and 4) and high partials in the spectrum [6]:
- $EvenHarm$ and $OddHarm$: Contents of even and odd harmonics in the spectrum,
- $Brightness$: brightness of sound - gravity center of the spectrum,
- $Irregularity$: irregularity of spectrum, defined as [3]

$$Irregularity = \log\left(20\sum_{k=2}^{N-1}\left|\log\frac{A_k}{\sqrt[3]{A_{k-1}A_kA_{k+1}}}\right|\right) \tag{1}$$

- $Frequency1, Ratio1, ..., 9$: parameters describing 10 most prominent peaks in the spectrum. The lowest frequency within this set is chosen as $Frequency1$, and proportions of other frequencies to the lowest one are denoted as $Ratio1, ..., 9$
- $Amplitude1, Ratio1, ..., 9$: the amplitude of $Frequency1$ in decibel scale, and differences in decibels between peaks corresponding to $Ratio1, ..., 9$ and $Amplitude1$.

The data represent various emotions, perceived by the listener when listening to a given excerpt. The data were labelled by musicians, using 13 emotional classes, as follows [4]:

1. frustrated,
2. bluesy, melancholy,
3. longing, pathetic,
4. cheerful, gay, happy,
5. dark, depressing,
6. delicate, graceful,
7. dramatic, emphatic,
8. dreamy, leisurely,
9. agitated, exciting, enthusiastic,
10. fanciful, light,
11. mysterious, spooky,
12. passionate,
13. sacred, spiritual.

Each tuple was initially labelled with a single emotion, and the labelled again, with no limitation to the number of assigned labels. Number of objects assigned to each class is presented in Table 1.

Grouping of data was later redefined; the classes were combined and 6 super-classes were created, as follows [4]:

**Table 1.** Number of objects representing emotion classes in 875-element multi-labelled database of 30-second audio samples, for grouping into 13 classes

| **Class** | **No. of objects** | **Class** | **No. of objects** |
|---|---|---|---|
| Agitated, exciting, enthusiastic | 304 | Fanciful, light | 317 |
| Bluesy, melancholy | 214 | Frustrated | 62 |
| Cheerful, gay, happy | 62 | Longing, pathetic | 147 |
| Dark, depressing | 41 | Mysterious, spooky | 100 |
| Delicate, graceful | 226 | Passionate | 106 |
| Dramatic, emphatic | 128 | Sacred, spiritual | 23 |
| Dreamy, leisurely | 151 | | |

1. frustrated, agitated, exciting, enthusiastic, dramatic, emphatic,
2. bluesy, melancholy, dark, depressing,
3. longing, pathetic, passionate,
4. cheerful, gay, happy, fanciful, light,
5. delicate, graceful, dreamy, leisurely,
6. mysterious, spooky, sacred, spiritual.

Number of classes assigned to each of 6 super-classes is shown in Table 2.

**Table 2.** Number of objects representing emotion classes in 875-element multi-labelled database of 30-second audio samples, for grouping into 6 super-classes

| **Class** | **No. of objects** |
|---|---|
| Agitated, exciting, enthusiastic<br>Dramatic, emphatic<br>Frustrated | 494 |
| Bluesy, melancholy<br>Dark, depressing | 255 |
| Cheerful, gay, happy<br>Fanciful, light | 379 |
| Delicate, graceful<br>Dreamy, leisurely | 377 |
| Longing, pathetic<br>Passionate | 253 |
| Mysterious, spooky<br>Sacred, spiritual | 123 |

### 5.2 Results

The experiments started with single-label data. For 13 classes the obtained accuracy was 20.12%, for $k = 13$. For 6 super-classes, 37.47% correctness was obtained, also for $k = 13$. Therefore, the results were low. However, subjective tests conducted with human experts (experienced musicians) also gave low results, since compatibility of classification among experts was at the level of 24-33%.

Next, experiments for multi-labelled data were performed. Firstly, the probability measure $p_1$ was used, with the threshold value equal to 0.15 yielding the best results. For 13 classes, 27.1% correctness for $k = 13$ was obtained, and for 6 super-classes, 38.62% correctness for $k = 15$, using the accuracy measure *acc*.

For the probability measure $p_2$, $k$=15 yielded the best results. The accuracy 28.05% was obtained for 13 classes and the threshold value equal to 0.28. For 6 super-classes, accuracy equal to 38.98% was obtained for the threshold value equal to 0.32.

In case of binary classification, even for single-label classification results were much higher, as illustrated in Figure 3. Like previously, the $k$-NN algorithm was applied in the experiments. In case of such a classification the recognition accuracy mainly depends on the number of objects representing each class (and how big is the class in proportion to the entire data set). For the smallest class, the results have even reached 95%.

| Class | No. of objects | Correctness |
|---|---|---|
| 1. happy, fanciful | 74 | 95.97% |
| 2. graceful, dreamy | 91 | 89.77% |
| 3. pathetic, passionate | 195 | 71.72% |
| 4. dramatic, agitated, frustrated | 327 | 64.02% |
| 5. sacred, spooky | 88 | 89.88% |
| 6. dark, bluesy, | 97 | 88.80% |

**Fig. 3.** Results of binary classification of emotions for 6 super-classes

The results presented in Figure 3, although tempting, are meaningless, since they can be easily obtained without any classifier, only by pointing always against the tested class (as mentioned in Section 4.

In case of the specific data used in the described experiments, one must consider yet another problem. Namely, the emotions in a 30-second music piece may change, even a few times within this time, so probably more detailed labelling (considering changes of emotions in time) should be performed on these data before further experiments.

For all the reasons mentioned above, the authors did not perform further experiments with binary classification, neither for single nor multi-labelling.

## 6 Summary

In this paper, the authors investigated theoretical problems related to multi-label classification of data, and tested them practically on database of multi-labelled tuples, where each datum could have been labelled by any number of 6 or 13 classes, representing emotions perceived by human subjects when listening to the parameterized music pieces. The result of this paper is a methodology for multi-label classification of data, illustrated with practical experiments using data representing audio files. Although the accuracy of this classification was not high, it still outperformed single-label accuracy, and was comparable with human performance.

## 7 Acknowledgements

This research was supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland, by the National Science Foundation under grant IIS-0414815 and by the Research Center at the Polish-Japanese Institute of Information Technology, Warsaw, Poland.

The authors would like to express thanks to Doctor Rory A. Lewis from the University of North Carolina at Charlotte for elaborating the initial audio database for research purposes, and to Doctor Zbigniew W. Raś from the UNCC for numerous discussions regarding binary multi-label classification.

## References

1. Boutell, M., Shen, X., Luo, J., Brown, C.: Multi-label Semantic Scene Classification. Technical Report, Dept. of Computer Science, U. Rochester (2003)
2. Clare, A., King, R.D.: Knowledge Discovery in Multi-label Phenotype Data. Lecture Notes in Computer Science **2168** (2001) 42–53
3. Fujinaga, I., McMillan, K.: Realtime recognition of orchestral instruments. Proceedings of the International Computer Music Conference, (2000) 141–143.
4. Li, T., Ogihara, M.: Detecting emotion in music. 4th International Conference on Music Information Retrieval ISMIR, Washington, D.C., and Baltimore, MD. (2003) Available at http://ismir2003.ismir.net/papers/Li.PDF
5. McCallum, A.: Multi-label Text Classification with a Mixture Model Trained by EM. AAAI'99 Workshop on Text Learning (1999)
6. Pollard, H. F., Jansson, E. V.: A Tristimulus Method for the Specification of Musical Timbre. Acustica **51** (1982) 162–171
7. Synak, P. and Wieczorkowska, A.: Some Issues on Detecting Emotions in Music. In: D. Slezak, J. Yao, J. F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 10th International Conference, RSFDGrC 2005, Regina, Canada, August/September 2005, Proceedings, Part II. LNAI 3642, Springer, (2005) 314–322
8. Wieczorkowska, A., Synak, P., Lewis, R., Ras, Z. W.: Creating Reliable Database for Experiments on Extracting Emotions from Music. In: M. A. Klopotek, S. Wierzchon, K. Trojanowski (Eds.), Intelligent Information Processing and Web Mining. Proceedings of the International IIS: IIPWM'05 Conference held in Gdansk, Poland, June 13-16, 2005. Advances in Soft Computing, Springer (2005) 395–402