Music genre recognition with risk and rejection

Sturm, Bob L.

Published in:
International Conference on Multimedia and Expo

Publication date:
2013

Document Version
Early version, also known as pre-print

Link to publication from Aalborg University

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
? You may not further distribute the material or use it for any profit-making activity or commercial gain
? You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.
ABSTRACT

We explore risk and rejection for music genre recognition (MGR) within the minimum risk framework of Bayesian classification. In this way, we attempt to give an MGR system knowledge that some misclassifications are worse than others, and that deferring classification to an expert may be a better option than forcing a label under high uncertainty. Our experiments show this approach to have some success with respect to reducing false positives and negatives.

Index Terms— Music genre recognition, machine learning, Bayesian classification

1. INTRODUCTION

The problem of making a machine recognize kinds of cultural content has been explored in various mediums, from identifying literary authors [1], to categorizing “tweets” [2], automatically writing music reviews [3], characterizing painting styles [4], recognizing spoken dialects [5], and classifying the genre and mood of music [6]. A large amount of work addresses the specific problem of music genre recognition (MGR) [7]; and a recent review [6] appears to show significant progress has been made.

Our recent work [8, 9], however, challenges the claim that this progress is due to any real increase of the capacity of MGR systems to recognize genre. In [8], we take two systems exhibiting some of the highest classification accuracies reported for a particular benchmark dataset, and submit them to three tests. In the first, we show each system persistsently makes misclassifications that are very poor with respect to musicological principles — explored further in [9]. For instance, one system persistently misclassifies as Metal, “Mamma Mia” by ABBA. In the second test, we show how each system can be tricked into classifying as different genres the same piece of music just by minor filtering of the signal. Finally, we show humans are unable to recognize the genres used by music excerpts composed by each system to be highly representative of the genres in which they have been trained. Whether or not an MGR system is really recognizing genre, these findings motivate the question: can we modify an MGR system such that it avoids making very poor misclassifications? In this paper, we explore this idea within the minimum risk framework offered by Bayesian classification [10].

Surprisingly little research in MGR explores the idea that some kinds of misclassification are worse than others. A few works, e.g., [11–19], argue that MGR systems should be evaluated in light of the specific confusions humans make. For instance, confusing metal for rock music is “better” than confusing metal for classical music. Weighting more heavily errors of the latter kind might better reflect the usability of an MGR system. This “discounted” approach to evaluation appears, for instance, in the 2005, 2007, and 2009 editions of the MGR challenge of the Music Information Retrieval Evaluation eXchange (MIREX) [20]. Though we do not find work directly implementing such knowledge in an MGR system, rather than in its evaluation, there is some work in music autotagging employing risk, e.g., that of Lo et al. [21]. That work, however, estimates a cost from a dataset, whereas we define it according to what a user deems offensive. Other areas, such as automatic speech recognition, have also applied risk minimization, e.g., [22].

Another aspect little explored in MGR are systems able to defer classification. For instance, when asked whether a piece of music is “Blues” or “Disco,” acceptable answers include “don’t know,” and “something other” [11,12,15,16,23]. Nearly all MGR systems so far proposed are designed to choose only from those genres in which they are trained, but we find four exceptions. Dannenberg et al. [24] describe how they can reduce the false positives committed by their style-recognition system with a threshold on the minimum difference in distances between class means to observations. Pye [25] describes, but provides no details about, forming a “garbage model” by augmenting his dataset with songs outside of his six selected genres. Akin to this, Harb and Chen [26] also state they use a “garbage model,” but do not discuss what music they use to define that class, and how it affects their system. Finally, the system of McKay [13] labels an excerpt “unknown” if the largest weighted sum of class-specific “scores” from component classifiers (neural networks) is too low, or not high enough with respect to other classes.
2. CLASSIFICATION WITH RISK AND REJECTION

Consider an $M$-dimensional real observation $x$ from a sample space of $K$ classes. Bayesian classification [10] seeks the class with the minimum expected risk $R(k|x)$:

$$
\hat{k} = \arg \min_{k \in \mathcal{K}} R(k|x) = \arg \min_{k \in \mathcal{K}} \sum_{m=1}^{K} \ell_{km} P(m|x)
$$

where $e_k$ is the $k$th standard basis vector, and we define the loss matrix and vector of posteriors

$$
L := \begin{bmatrix}
\ell_{11} & \ell_{12} & \cdots & \ell_{1K} \\
\ell_{21} & \ell_{22} & \cdots & \ell_{2K} \\
\vdots & \vdots & \ddots & \vdots \\
\ell_{K1} & \ell_{K2} & \cdots & \ell_{KK}
\end{bmatrix},
\quad p(x) := \begin{bmatrix}
P(1|x) \\
P(2|x) \\
\vdots \\
P(K|x)
\end{bmatrix}
$$

where the set $\mathcal{K} := \{1, \ldots, K\}$ indexes the classes. The loss $\ell_{km}$ is user-defined to encapsulate the loss associated with choosing class $k$ for $x$ when it is actually from $m$. $P(m|x)$ is the posterior of class $m$ given the observation $x$. We assume that all priors $P(m)$ are non-zero, and all classes are disjoint. For MGR, the last assumption means our model restricts each piece of music to be of only one class. This is obviously artificial [27], but provides a starting point. Furthermore, out of 467 works in MGR, only ten do not use this model [7].

2.1. Uniform loss

If we believe all misclassifications are equally bad, then we can define $\ell_{km} := 1 - \delta_{k-m}$ where $\delta_p = 1$ for $p = 0$, and zero otherwise. This is also called the “zero-one loss” function, which makes $L = 11^T - I$, where $I$ is a vector of $K$ ones, and $I$ is the identity matrix of appropriate size. In this case, (1) becomes the maximum a posteriori (MAP) classifier [10]:

$$
\hat{k} = \arg \max_{k \in \mathcal{K}} e_k^T (11^T - I)p(x) = \arg \max_{k \in \mathcal{K}} e_k^T [1 - p(x)] = \arg \max_{k \in \mathcal{K}} P(k|x).
$$

MAP classification in MGR has been used in, e.g., [28–32]. If all classes are equally likely, the MAP classifier becomes the maximum likelihood (ML) classifier [10]:

$$
\hat{k} = \arg \max_{k \in \mathcal{K}} P(x|k).
$$

ML classification in MGR has been used in [24,33–43].

2.2. Non-uniform loss

Consider in the sample space a $c \in \mathcal{K}$ for which it is imperative the system has high precision (most observations it labels $c$ are from $c$) and recall (it mislabels few observations from $c$). We are not concerned with other misclassifications. Thus, define the “persnickety-aphetic” loss function with loss $l$

$$
\ell_{km} := (1 - \delta_{m-k}) \left[ 1 + (l - 1)(\delta_{k-c} + \delta_{m-c}) \right].
$$

The loss matrix $L$ has zeros on its diagonal, $K - 1$ elements in row $c$ of value $l$, and $K - 1$ elements in column $c$ of value $l$, and ones everywhere else. We now analyze the role of $l$.

The risk in (1) with this loss function becomes

$$
R(k|x) = e_k^T \left[ (11^T - I)p(x) + (l - 1)P(c|x)1 \right] = 1 - P(k|x) + (l - 1)P(c|x) - (l - 1)[2P(c|x) - 1]e_c
$$

which makes the selection criterion become

$$
\hat{k} = \arg \max_{k \in \mathcal{K}} P(k|x) - (l - 1)[P(\neg c|x) - P(c|x)]\delta_{k-c}
$$

with $P(\neg c|x) := 1 - P(c|x)$. The classifier selects $c$ when

$$
P(c|x) - (l - 1)[P(\neg c|x) - P(c|x)] > \max_{k \in \mathcal{K} \setminus c} P(k|x).
$$

If $l = 1$, this reduces to (3). For $l > 1/2$, if

$$
P(c|x) \leq \frac{l - 1}{2l - 1}
$$

then the classifier will never select $c$ since the left hand side of (8) is then negative. The same thing occurs for $l < 1/2$ if

$$
P(c|x) \geq \frac{l - 1}{2l - 1}.
$$

In the limit as $l \to \infty$, the classifier will not select $c$ as long as $P(c|x) \leq 1/2$; and in the limit as $l \to -\infty$, the classifier will not select $c$ as long as $P(c|x) \geq 1/2$. We see by substitution that for $l = 1/2$, if $\max_{k \in \mathcal{K} \setminus c} P(k|x) < 0.5$, then the classifier selects $c$. 
2.3. Rejection

We can enable this system to reject classification when

\[
\min_{k \in \mathcal{K}} R(k|x) > R_{\text{max}}
\]

for a maximum risk \(R_{\text{max}}\). For the “persnickety-apathetic” loss function (5), the system rejects classification if

\[
\min_{k \in \mathcal{K}} \left[ 1 - P(k|x) + (l - 1)P(\neg k|x) \right]
\]

\[
- (l - 1)[2P(\neg k|x) - 1] \delta_{k - c} > R_{\text{max}}
\]

If, for a given \(l\), the class with smallest risk in (6) is \(c\), and \(lP(\neg c|x) > R_{\text{max}}\), then the classifier rejects classification. If the least-risk class in (6) is \(k^* \neq c\), and \(P(\neg k^*|x) > R_{\text{max}} - (l - 1)P(c|x)\), then the classifier will reject classification. Hence, we see that in order to make the system capable of rejecting classification, we must define \(R_{\text{max}} < l\).

3. SIMULATIONS

We now test the impact of loss and rejection for an MGR system. As an example scenario, consider we have many hours of radio recordings, and wish to estimate the extent to which music that sounds “classical” appears. The amount of data we have is such that it prohibits manual listening and labeling. The success criteria of a useful MGR system include: high precision, i.e., that it is correct in most of what it identifies as “classical-sounding”; high recall, i.e., that very few “classical-sounding” excerpts are mislabeled; and that it produces a set of rejected classifications that is either mostly “classical-sounding” or mostly “not classical-sounding”, i.e., manual listening and labeling of this set is not prohibitive.

3.1. Method

We use the following experimental design. We train a classifier in six genre categories using features extracted from all disjoint 27.9 s (5 \(\times\) 217 samples at 22.05 kHz sampling rate) excerpts from the 729 audio files of the ISMIR2004 training dataset. Table 1 lists the six categories, the number of files, and the number of excerpts from each. Then, using the validation set of ISMIR2004 [44], we estimate the best loss and rejection threshold \(R_{\text{max}}\) with respect to minimizing

\[
f(l, R) = [1 - \text{tpr}(l, R)]^2 + [\text{fpr}(l, R)]^2
\]

\[
+ \frac{\beta}{N} \frac{1}{2} - \frac{1}{2} \cos[2\pi \text{pur}(l, R)]
\]

where \(\text{tpr}(l, R)\) is the true positive rate, \(\text{fpr}(l, R)\) is the false positive rate, \(\beta X/N\) is the weighted ratio of rejections \(X\) to the total number of excerpts \(N\), and \(\text{pur}(l, R)\) is the proportion of the rejected classifications that are labeled “classical.” For our problem, we want \(\text{tpr}(l, R) \approx 1\), \(\text{fpr}(l, R) \approx 0\), and, if \(\beta > 0\), the number of rejected classifications either to be small, or to be large and consist mostly of excerpts that sound “non-classical” (in which case we ignore them), or “classical” (in which case we add them all to the positives). In our implementation, we compute \(f(l, R)\) at several \(\{(l, R)\}\), and find where it is the smallest for each classifier. Finally, we test the tuned system using features extracted from the 1000 excerpts of the GTZAN dataset. Since GTZAN has recently been analyzed and shown to have several faults [46], we remove from the analysis 49 exact replicas, and relabel two “jazz” excerpts as “classical.” The GTZAN column in Table 1 reflects these changes.

Few papers in MGR apply training on one dataset and testing on another, e.g., [40, 47]. The danger of this is that the concepts between two datasets may not be the same, even though names of some of their classes are identical. In our simulations, we thus assume the concepts of “Classical” in ISMIR2004 and GTZAN are similar enough that this will not be a serious problem for the validity of our experiments.

To create features for each 29.7 s excerpt, we compute “scattering coefficients” — recently proposed by Anden and Mallet [48] — using the implementation in [49]. The specific settings we define are: second-order decompositions, filter q-factor 16, and maximum scale 160. Since each excerpt produces 40 feature vectors of dimension 469, we classify an excerpt by selecting the class with the least \(\text{total risk}\), i.e.,

\[
\hat{k} = \arg \min_{k \in \mathcal{K}} e_i^T L \sum_{i=1}^{40} p(x_i) \tag{14}
\]

where \(p(x_i)\) are the posteriors for the \(i\)th feature vector. If the argument above is greater than \(40R_{\text{max}}\), then we make the system reject classification.

For each class, we define \(P(x|k) = N(x; \mu_k, C_k)\), i.e., an observation of class \(k\) is distributed multivariate Gaussian with mean \(\mu_k\) and covariance \(C_k\):

\[
P(x|k) \propto |C_k|^{-1/2} e^{-\frac{1}{2} (x - \mu_k)^T C_k^{-1} (x - \mu_k)}.
\]

Table 1. Summary of training, validation, and testing datasets
To learn the parameters of each distribution, we use unbiased minimum mean-squared error estimators with the features we extract from the ISMIR2004 training data [50]. If we assume all classes are distributed with different means but the same covariance (estimated from the data of all classes) the classifier (1) is called “Linear Bayes Normal” (LBN); and if each class is distributed with different covariances (estimated from the data of each class), it is called “Quadratic Bayes Normal” (QBN) [10]. We define the priors of all classes to be the same.

### Table 2
Validation (ISMIR2004) confusion tables for LBN and QBN. Column “r” shows number of rejections in each.

#### Uniform loss, no rejection: $l = 1, R_{\text{max}} = \infty$

<table>
<thead>
<tr>
<th></th>
<th>Pred. (LBN)</th>
<th>Pred. (QBN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>1905</td>
<td>1985</td>
</tr>
<tr>
<td>-</td>
<td>195</td>
<td>293</td>
</tr>
</tbody>
</table>

#### Non-uniform loss, rejection, but no rejection purity ($\beta = 0$). LBN: $l = 2, R_{\text{max}} = 0.5$; QBN: $l = 3, R_{\text{max}} = 0.75$

<table>
<thead>
<tr>
<th></th>
<th>Pred. (LBN)</th>
<th>Pred. (QBN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>1490</td>
<td>1971</td>
</tr>
<tr>
<td>-</td>
<td>36</td>
<td>140</td>
</tr>
</tbody>
</table>

#### Non-uniform loss, rejection, and purity ($\beta = 1$). LBN: $l = 1, R_{\text{max}} = 0.75$; QBN: $l = 2, R_{\text{max}} = 0.5$

<table>
<thead>
<tr>
<th></th>
<th>Pred. (LBN)</th>
<th>Pred. (QBN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>1905</td>
<td>1971</td>
</tr>
<tr>
<td>-</td>
<td>194</td>
<td>140</td>
</tr>
</tbody>
</table>

### 3.2. Results of validation

From evaluation using the validation set of ISMIR2004, Fig. 1 shows the receiver operating characteristics (ROC) of LBN and QBN for several pairs of loss $l$ and maximum risk $R_{\text{max}}$; and Table 2 shows the confusion tables of three particular pairs. First, the open circles shows the performance of each system with uniform loss and no rejection, and Table 2(a) shows the confusion tables. We see LBN has about 100 fewer false positives than QBN, but about 80 more false negatives. The diamond in each ROC shows the best performance of each system with respect to (13) for $\beta = 0$ for non-uniform loss and rejection; and Table 2(b) shows the confusion tables. While the number of false positives and negatives for LBN decrease dramatically, it now finds 500 fewer true positives than before. QBN, however, now halves its number of false positives and negatives, and produces a set of rejections that is only 6% Classical. Finally, the star in each ROC shows the best performance of each system with respect to (13) for non-uniform loss and rejection, and rejection purity $\beta = 1$; Table 2(c) shows the confusion tables. Now we see that QBN produces 2 fewer false negatives, and has a set of rejections that is 4% Classical. The performance of LBN here essentially equals what it is for uniform loss and no rejection.

### 3.3. Results of classification

We now test each of these systems using the GTZAN dataset, the results of which are shown in Table 3. First, for the systems using uniform loss and no rejection we see in Table 3(a) that while LBN here has less than half as many false positives as QBN, it has a far higher number of false negatives. When these system consider non-uniform loss and rejection, but do not consider the purity of the rejections, Table 3(b) shows QBN and LBN both have a large decrease in the num-
Table 3. Test (GTZAN) confusion tables for LBN and QBN. Column “r” shows number of rejections in each.

<table>
<thead>
<tr>
<th>LBN</th>
<th>QBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>−</td>
</tr>
<tr>
<td>+</td>
<td>73</td>
</tr>
<tr>
<td>−</td>
<td>20</td>
</tr>
<tr>
<td>Non-uniform loss and rejection, but no purity (β = 0). LBN: l = 2, R_{max} = 0.5; QBN: l = 3, R_{max} = 0.75</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>31</td>
</tr>
<tr>
<td>−</td>
<td>2</td>
</tr>
</tbody>
</table>

4. CONCLUSION

That an artificial system misclassifies is no surprise; and to aim for one that does not misclassify certainly aims too high. When it comes to cultural content such as music genre, which escapes clear and definitive categories [27], and of which humans often disagree [11, 12, 23, 51, 52], misclassification is an inevitable part of an MGR system. One might see this as a selling point: “some people would be entertained by the predictions, especially when they were wrong” [53]. However, not all misclassifications are equal — some are worse (funnier?) than others — and little work in MGR explores this idea outside of evaluation. We have shown how such an idea can be naturally incorporated into an MGR system by using non-uniform loss and rejection and the minimum risk framework of Bayesian classification. We analyzed a particular form of the loss, and applied it within a scenario of detecting “classical-sounding” music. We could, of course, have trained the classifiers to discriminate between “classical” and “non-classical” — lumping together all excerpts in the ISMIR 2004 dataset that are not labeled “Classical” — but the point of this paper is not to solve that particular scenario. It is to investigate how loss and rejection can tune a multiclass MGR system to produce results that could be more useful in a scenario, regardless of whether or not the MGR system has any capacity to recognize the genre used by music [8, 9].

5. REFERENCES
