TISMIR

Rodríguez-Algarra, F., et al. (2019). Characterising Confounding Effects in Music Classification Experiments through Interventions. *Transactions of the International Society for Music Information Retrieval,* 2(1), pp. 52–66. DOI: https://doi.org/10.5334/tismir.24

RESEARCH ARTICLE

Characterising Confounding Effects in Music Classification Experiments through Interventions

Francisco Rodríguez-Algarra[*], Bob L. Sturm[†] and Simon Dixon[*]

We address the problem of confounding in the design of music classification experiments, that is, the inability to distinguish the effects of multiple potential influencing variables in the measurements. Confounding affects the validity of conclusions at many levels, and so must be properly accounted for. We propose a procedure for characterising effects of confounding in the results of music classification experiments by creating regulated test conditions through interventions in the experimental pipeline, including a novel resampling strategy. We demonstrate this procedure on the *GTZAN* genre collection, which is known to give rise to confounding effects.

Keywords: Music Classification; Evaluation; Experimental Design; Confounding; Resampling

1. Introduction

Classification experiments are arguably the most widespread tool for the evaluation of Music Information Retrieval (MIR) systems and methods (Sturm, 2014b; Urbano et al., 2013). Lack of proper control in such experiments leads to conclusions of questionable validity, yielding results that may fail to generalise beyond the experiment (Drummond, 2006). This hampers progress by obfuscating which research paths are worth pursuing, and demands revising conventional experimental practices (Sturm, 2016a). We propose and illustrate a procedure for assessing how failing to control for particular sources of information in evaluation collections affects experimental results.

The partitioning of collections into training and testing materials affects validity in classification experiments, as the MIR community has long acknowledged. For instance, the presence of the same artists or albums in both training and testing recordings artificially inflates performance estimates; this is known as artist or album effects, respectively (Pampalk et al., 2005; Flexer and Schnitzer, 2010). Performance can also decrease if one tests using a separate collection (Bogdanov et al., 2016) or manipulates recordings in presumably irrelevant ways (Sturm, 2014a; Rodríguez-Algarra et al., 2016).

Pampalk et al. (2005) introduced artist "filters" to counteract artist effects in music similarity experiments. Their approach, which we call "filtered partitioning", creates training and testing collections[1] not sharing a level of the factor one aims to control (e.g., artist information). This provides a single "regulated" testing condition (all testing instances follow a particular rule), alleviating the impact of the replication of that factor on performance estimates. Comparing regulated results from filtered partitioning with those from a conventional random partitioning enables assessing the impact of leaving a factor unregulated. Using this approach, studies (e.g., Flexer (2007); Sturm (2014b)) show not only that unregulated collections might bias performance estimates, but also that the magnitude of such bias varies across feature representations and learning algorithms.

A major limitation of filtered partitioning for assessing the effect of leaving collections unregulated is that the regulated training and testing collections it creates likely contain different instances than those included in their unregulated counterparts. No single trained system is exposed to both regulated and unregulated testing conditions, which impedes disentangling the effects of training and testing. Moreover, as Marques et al. (2011) note, the makeup of some collections constrains how many disjoint regulated partitions one can create (e.g., the number of cross-validation folds cannot exceed the number of artists per class). This may conflate the effect of the particular instances — their "difficulty" — with that of the (lack of) regulation.

Apart from altering the collection partitioning strategy, manipulating the raw data can also create regulated evaluation conditions (Sturm, 2014a, 2016b). This avoids the aforementioned limitations as instances in both conditions match, but cannot regulate all factors (e.g., artists). Previous studies combine filtered partitioning with manipulations to control multiple factors simultaneously (Rodríguez-Algarra et al., 2016), but suffer from the aforementioned limitations of filtered partitioning.

[*] Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, UK

[†] KTH Royal Institute of Technology, Stockholm, SE

Corresponding author: Francisco Rodríguez-Algarra (f.rodriguezalgarra@qmul.ac.uk)

In this article, we describe both partitioning and manipulation approaches as alternative, but complementary, types of interventions in the experimental pipeline. These interventions create regulated evaluation conditions that can be used to characterise how the outcomes of music classification experiments are affected by "confounding", a validity threat we examine in Sec. 2. We then introduce in Sec. 3 a procedure for combining multiple interventions that overcomes the limitations of filtered partitioning, including a novel resampling strategy aimed at gauging confounding effects. We focus on the effects of particular sources of confounding information on test results, as this is paramount for MIREX[2] and similar evaluation exchanges, but the approach could be extended to assess effects in training. We illustrate our approach in Sec. 4 by analysing two known confounders in the *GTZAN* music genre collection (Tzanetakis and Cook, 2002): artist replication and infrasonic content.[3] This could be adapted to other domains with minimal adjustments. We finally discuss in Sec. 5 the main limitations and broader implications of our work, and provide concluding remarks in Sec. 6.

2. Confounding in Classification Experiments

Classification experiments dominate evaluation in both pure and applied machine learning research (Flach, 2012; Alpaydin, 2014). A classification experiment essentially involves measuring how well a prediction system, or family of systems, reproduces the annotations of a collection, which acts as a proxy for success in some real-world problem (Hernández-Orallo, 2016). The diagram in **Fig. 1** represents a simplified pipeline of a music classification experiment, introducing notation used later in this article.

Any empirical study is subject to diverse validity threats that challenge the veracity and generality of its outcomes (Shadish et al., 2002; Trochim and Donnelly, 2007). Among these, confounding is particularly relevant as it leads to invalid conclusions about causal relationships (Pearl, 2009). Two variables potentially influencing measurements are confounded if the experimental design cannot disentangle their effects (Cobb, 1998). Many experimental and quasi-experimental designs thus alleviate confounding by controlling extraneous variables other than the target of the study – explicitly setting or accounting for their values in the different experimental conditions – to avoid them impacting the measurements (Montgomery, 2013; Shadish et al., 2002).

Simple experimental design choices overcome the most obvious risks of confounding in classification experiments (Langley, 1988). For instance, if one measures the performance of multiple systems each on different instances, the influence of such systems – the outcome of interest – becomes confounded with the selection of instances – an extraneous variable. This is easily avoided by comparing measurements on the same instances, a standard evaluation practice.

Subtler forms of confounding affecting the conclusions of classification experiments are receiving increasing attention in the applied machine learning literature (e.g., Chen and Asch (2017); Charalambous and Bharath (2016)). In particular, information not intrinsically linked with the problem of interest might incidentally relate with the annotations of evaluation collections, providing alternative means for systems to predict annotations in classification experiments. Causes of this phenomenon include selection bias (e.g., Mendelson et al. (2017)) and leakage (Kaufman et al., 2011), which induce confounding by conflating success in addressing the target problem – the outcome of interest – with the exploitation of auxiliary information – an extraneous influence (Sturm, 2016a). In this article, we focus on identifying and analysing the effects of these forms of confounding information.

If a collection is used in the evaluation of diverse problems and use cases, each case implicitly determines which content is potentially confounding. For instance, tempo information in a collection may be legitimate for identifying dance style, as the speed of a piece influences which dance moves are feasible, but not for identifying rhythmic patterns, as these should be invariant to reasonable variations in speed (Dixon et al., 2004; Sturm, 2014a). Artists tend to compose or perform music pieces of one or a few genres, yet artist properties are not essential to those genres (Flexer and Schnitzer, 2010). If one's sole

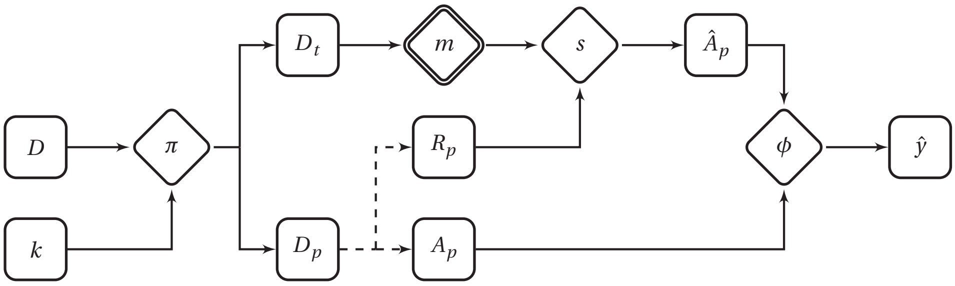

Figure 1: Pipeline of a single iteration k of a classification experiment evaluating a system construction method m (combination of feature extraction and learning algorithm) on a music collection D. Square-shaped nodes represent data structures; diamond-shape nodes represent processes. A double border indicates a treatment factor with fixed level. Solid lines indicate information flow; dashed lines join components of the same data structure. π is a data assignment/partitioning function. D_t is the training collection; D_p is the testing collection, with R_p the raw data (e.g., recordings) and A_p the corresponding annotations. (R_t and A_t omitted for simplicity.) s is the trained system, $\hat{A}_p$ the predicted annotations, ϕ the performance metric function, and $\hat{y}$ an estimate of the theoretical performance y – i.e., given the true distribution.

aim is to attach genre tags to a fixed set of recordings, artist information will likely help; if one aims to assess whether a system captures the defining characteristics of music genres instead, then artist-specific content is extraneous. Other properties, however, such as the infrasonic content present in *GTZAN* (Tzanetakis and Cook, 2002; Rodríguez-Algarra et al., 2016), are unlikely to be legitimately informative for any real problem.

The main risk of confounding in classification experiments is that conclusions fail to generalise. Systems might not succeed when deployed if they rely on information about a potential confounder being present, as there is no guarantee that the observed association will remain outside the experimental setting. The MIR community has adopted evaluation practices to counter this pitfall. The aforementioned filtered partitioning approach yields performance estimates free of the influence of the regulated potential confounder (Pampalk et al., 2005). Others suggest leveraging data augmentation to avoid confounding information influencing the training process (Charalambous and Bharath, 2016; Stowell, 2017). This synthetically generates combinations of background information and target categories that force systems to learn general concepts rather than incidental correlations.

As a homage to Clever Hans (Pfungst et al., 1911), some MIR publications refer to systems exploiting confounding information as "horses" (Sturm, 2014a, 2016b). To assess whether a system is indeed a "horse", one might test on a completely separate collection than the one used for training (Bogdanov et al., 2016). This, however, does not reveal the source of discrepancy. Others propose to illuminate the behaviour of trained systems through interpretable explanations of predictions (Mishra et al., 2017) or interventions in the experimental pipeline (Rodríguez-Algarra et al., 2016). We extend the latter approach to gauge how confounding impacts the outcomes of classification experiments.

3. Characterising Confounding Effects

We propose a simple procedure that uses interventions to characterise the effects of confounders in performance measurements from classification experiments, overcoming the limitations of filtered partitioning via a novel resampling strategy. We here focus on the effects in testing, but the procedure could be easily adapted to assess the effects in the training of systems, or a factorial combination of both.

3.1. Interventions on the Experimental Pipeline

In empirical studies, an intervention is the act of explicitly fixing a factor to one of its levels (Pearl, 2009). A conventional music classification experiment involves intervening on the system creation method, as **Fig. 1** represents with a double-bordered node. This specifies evaluation conditions to compare, each with different feature extraction and/or learning algorithms, yielding estimates of differences in performance. Apart from such conventional intervention, one might also intervene on other steps of the pipeline to create further evaluation conditions. These may reveal information unavailable otherwise, such as the impact of a potential confounder.

Consider the train/test pipeline of a classification experiment, with training and testing materials drawn from a collection D. Let z be a potential confounder. If z correlates with the classes in some way within D, legitimately or not, then such correlation should appear in both training and testing instances unless a regulation is introduced, making z available for both training and prediction. Interventions regulating z thus impede its availability in such steps by breaking its correlation with the classes.

A classification experiment pipeline offers many opportunities for intervening. One might intervene on training or prediction, altering methods and systems to avoid relying on z. For instance, knowing which dimensions of the feature representations capture information related with z, one might regulate by removing or masking such dimensions in the feature extractor. This is the case in the tempo-invariant features of Dixon et al. (2004). Previous studies, however, often intervene on the creation of training and testing materials, through either "instance assignment" or "data manipulation" interventions.

Instance Assignment interventions regulate π, the criterion for assigning instances to either training or testing, taking z into account. These interventions thus require knowledge of z, i.e., the value that z takes for each instance. Properties such as artist, album, file format, or recording device are suitable for this approach.

Filtered partitioning belongs to this category, with the intervention involving an assignment function $\pi'(D)$ that creates D'_t and D'_p both containing different instances than their unregulated counterparts. Other strategies may distinguish between regulated and unregulated conditions only for testing, using the exact same training materials in both (i.e., $\pi'(D) = (D_t, D'_p)$). This enables isolating the potential effect of z in the evaluation of fixed systems. If one aims to estimate the impact of z in system construction instead, a suitable intervention might fix the testing collection and create regulated and unregulated conditions distinguished only in the selection of training instances (i.e., $\pi'(D) = (D'_t, D_p)$).

Data Manipulation interventions alter the raw data (e.g., audio recordings) in a way that preserves their membership to a class, but modifies the correlation between z and the classes. Manipulations such as pitch-preserving time-stretching (Sturm, 2016b) and high-pass filtering (Rodríguez-Algarra et al., 2016) have been used to this end. These interventions do not require instance-level knowledge of z, and they permit comparing predictions on the same instances (manipulated and not). Nevertheless, they require identifying and implementing suitable manipulations.

Similar to instance assignment interventions, data manipulation interventions may create regulated conditions in different ways. Given a class-preserving manipulation, one might transform instances in both D_t and D_p in the same way, thus obtaining a pair of regulated collections (D'_t, D'_p). This, however, may not break correlations if the manipulation is deterministic, failing to regulate z. It is more appropriate to keep either D_t or D_p unaltered and manipulate the other.

These types of interventions are complementary, as they affect different steps of the experimental pipeline, but it is feasible to stack various interventions affecting the same step (e.g., time-stretching and filtering recordings). They might be integrated into the experiment using a factorial design (Montgomery, 2013), where each intervention creates an additional treatment factor with at least two levels: regulated and unregulated. Comparing measurements under combinations of such levels reveals the marginal and joint impact of the interventions, illuminating the effects of the potential confounders.

3.2. Analysing Confounding with Interventions

To date, interventions on the experimental pipeline have been used to reveal whether a potential confounder affects the evaluation of particular methods or systems. Given an annotated music collection D, we now describe the steps we propose to extend this approach to assess how such a potential confounder impacts evaluations conducted on D over multiple methods, and how several potential confounders interact.

a) Identify potential confounders

As a prerequisite of the analysis, one should determine which potential confounders are worth considering for the collection and problem at hand. This may come from exploratory analyses of collections, published systems and/or domain knowledge.

b) Design interventions

For each identified potential confounder z, one should specify at least one suitable intervention to distinguish regulated and unregulated evaluation conditions with respect to z. The adequate type of intervention depends on the nature of z.

c) Create train/test materials

Let D_t be a training collection drawn from D, and D_p and D'_p a pair of testing collections associated with D_t that differ only in whether they regulate a potential confounder z. In particular, D_p is drawn from D (usually $D \backslash D_t$), and D'_p comes from an intervention on the experimental pipeline. For instance, D'_p might be a pruned version of D_p with instances whose value of z appears in D_t removed, or the result of a manipulation on the recordings in D_p for regulating z. If the analysis considers J interventions simultaneously, then one creates (at least) 2^J testing collections associated with D_t, one for each combination of regulation condition.

To avoid the performance estimates being confounded with the selection of instances, it is advisable to create multiple training collections through a resampling strategy (Weihs et al., 2017). In this case, one would draw K training collections $D_{t,k}$ and derive the testing collections associated with each as above. Conventional resampling strategies, however, cannot ensure testing collections from instance assignment interventions fulfil the intended regulation. The strategy we propose later in Sec. 3.3 addresses this issue.

d) Select methods

Characterising the impact of a potential confounder z requires a wide range of performance estimates. One may then train multiple systems on each $D_{t,k}$ using diverse combinations of feature extraction and learning algorithms. We denote the total number of combined methods as M. These methods should cover a broad spectrum of modelling approaches and expected performance values. Optimisation is not essential if the goal is to gauge how different approaches behave when exposed to particular perturbations on the data and not to maximise performance, but plays an important role if this procedure is integrated into real evaluations.

e) Obtain performance estimates

For each trained system s_n, $1 \leq n \leq K \cdot M$, one can then compute figures of merit (e.g., accuracy, mean recall) in the corresponding testing collections. For simplicity, we call $\hat{y}$ and $\hat{y}'$ the generic unregulated and regulated performance estimates, respectively.

f) Relate regulated and unregulated estimates

As $\hat{y}$ and $\hat{y}'$ differ only in their regulation of z, one assumes any observed difference reflects an effect of z. Given enough $(\hat{y}, \hat{y}')$ pairs, one might estimate the expected relationship between regulated and unregulated measurements $\hat{y}' \sim f(\hat{y})$.[4] Fitting a model of $f(\hat{y})$ from data pairs $(\hat{y}, \hat{y}')$ describes the *confounding effect* of z in evaluations on D. This reflects how a potential confounder tends to affect performance estimates of trained systems evaluated in the collection. For simplicity, we may use a linear model, such as

$$\hat{y}' \sim f(\hat{y}) = \alpha \cdot \hat{y} + \kappa \tag{1}$$

though other relationships (e.g., quadratic, exponential) could be preferable. If $\alpha \approx 1$ and $|\kappa| \gg 0$, we say the confounding effect of z is mostly additive (i.e., the relationship between $\hat{y}$ and $\hat{y}'$ appears as a fixed effect); if $\alpha \not\approx 1$ and $\kappa \approx 0$, we say it is mostly multiplicative (i.e., a gain). To estimate κ in the former case, one could average performance differences between conditions per iteration. Denote $\hat{y}_{m,k}$ the performance of a system trained with $D_{t,k}$ using method m measured on a test collection $D_{p,k}$, and $\hat{y}'_{m,k}$ the measurement on the associated regulated test collection $D'_{p,k}$, then:

$$\hat{\kappa} = \frac{\Sigma_{k=1}^{K} \Sigma_{\forall m} (\hat{y}_{m,k} - \hat{y}'_{m,k})}{K \cdot M} \tag{2}$$

with K and M defined as above.

In the general case, $\hat{y}$ and $\hat{y}'$ will not keep a simple relationship over all observations. Different system-construction methods can exploit a potential confounder differently, and the effect might also differ across classes. One may thus analyse the data marginally to identify clearly distinct behaviours.

If the analysis involves multiple interventions, comparing marginal and joint measurements can elucidate whether the different confounders (or approaches to the same confounder) interact. Let $\hat{y}$ be the performance estimated in the original testing collection, $\hat{y}'_1$ and $\hat{y}'_2$ the performances in testing collections from two different interventions, and $\hat{y}'_{1,2}$ the performance on a testing collection subjected to both interventions. Apart from relating $\hat{y}$ with both $\hat{y}'_1$ and $\hat{y}'_2$ to analyse the effects of each confounder separately, one might compare the sum of those two marginal effects with the difference between $\hat{y}$ and $\hat{y}'_{1,2}$. Let Δ_A be the "accumulated" variation, defined as:

$$\Delta_A = (\hat{y} - \hat{y}'_1) + (\hat{y} - \hat{y}'_2) \tag{3}$$

and Δ_R be the "real" variation:

$$\Delta_R = (\hat{y} - \hat{y}'_{1,2}). \tag{4}$$

The difference Δ_R–Δ_A indicates whether the two confounding effects under study reinforce each other, do not interact, or overlap. This can be generalised to higher-order interactions if more interventions coexist.

3.3. Regulated Bootstrap Resampling

The procedure above requires multiple distinct train/test pairs. Various resampling strategies address this, but none is entirely suitable for instance assignment interventions. In particular, the fixed size of the partitions in *k*-fold cross-validation (*k*CV) impedes adjusting to imbalances in the presence of the potential confounder *z*. Bootstrap sampling (Efron, 1977), drawing $|D|$ training instances with replacement from the whole collection *D*, overcomes this issue. Sampling with replacement is often preferred in the statistical learning literature (Hastie et al., 2009; Hothorn et al., 2005), as it enhances the statistical properties of the generated samples over *k*CV, such as reducing the variance of the derived estimates (Efron, 1983; Efron and Tibshirani, 1997). Nevertheless, training collections generated with bootstrap sampling may not permit suitable regulations if, e.g., too many instances in $D_p = D \backslash D_t$ have values of *z* also in D_t.

To address these issues, we propose *regulated bootstrap*, a multi-phase resampling strategy expressed in **Alg. 1**. The algorithm takes as input a collection *D* (sequence of instances, each a tuple $(r, a, z)_i$ of data element r_i, class annotation a_i from the set *A*, and attribute z_i from the set *Z*) and the desired number of recordings per class n_r. It first attempts to create a pair (D_t, D_p) using stratified bootstrap – sampling with replacement from each class separately. If this cannot derive a regulated testing collection D'_p of size n_r, it then proceeds to a partially-curated approach. This may be repeated an arbitrary number of times. The output of each sampling run can then be used to generate a D'_p through pruning: removing all instances in D_p with *z* also in D_t. Although the pruned instances do not appear in D'_p, they cannot be added to D_t as they remain in D_p. Supplementary material S1 describes a simple illustrative example of regulated bootstrap.

Algorithm 1: Regulated Bootstrap resampling strategy, given a collection *D* and a threshold $n_r \in \mathbb{N}$.

```
RegulatedBootstrap($D$, $n_r$):
- Initialise: $D_t \leftarrow (\varnothing)$, $D_p \leftarrow (\varnothing)$
- For each $\mathrm{a} \in A$:
    0. Define $D_{\mathrm{a}}$ as the instances in $D$ with $a_i = \mathrm{a}$;
    1. Phase 1: Stratified Bootstrap Sampling
        (a) Create $d_t$ by uniformly sampling with replacement $|D_{\mathrm{a}}|$ instances from $D_{\mathrm{a}}$;
        (b) Create $d_p \leftarrow D_{\mathrm{a}} \backslash d_t$;
    2. Phase 2: Size Verification
        (a) Define $Z_t$ as the union of all $z_i$ in $d_t$;
        (b) Create $d'_p$ by selecting all instances $(r, a, z)_i$ in $d_p$ with $z_i$ not in $Z_t$;
        (c) If $|d'_p| < n_r$, proceed to Phase 3, as it lacks enough regulated instances; otherwise, go to Phase 4;
    3. Phase 3: Curated Sampling
        (a) Define $Z_{\mathrm{a}}$ as the union of all $z_i$ in $D_{\mathrm{a}}$;
        (b) Initialise a hold-out collection $d_h \leftarrow (\varnothing)$;
        (c) Randomly select a $\mathrm{z} \in Z_{\mathrm{a}}$, and remove it from $Z_{\mathrm{a}}$;
        (d) Define $d_{\mathrm{z}}$ as the instances in $D_{\mathrm{a}}$ with $\mathrm{z} \in z_i$;[5]
        (e) Append $d_{\mathrm{z}}$ to $d_h$: $d_h \leftarrow d_h \frown d_{\mathrm{z}}$;
        (f) If $|d_h| < n_r$, go to (3c), as $d_h$ still lacks enough instances;
        (g) Create $d_t$ by uniformly sampling with replacement $|D_{\mathrm{a}}|$ instances from $D_{\mathrm{a}} \backslash d_h$;
        (h) Create $d_p \leftarrow D_{\mathrm{a}} \backslash d_t$;
        (i) Go to Phase 2 to check size requirements;
    4. Phase 4: Concatenation
        (a) Append $d_t$ to $D_t$: $D_t \leftarrow D_t \frown d_t$;
        (b) Append $d_p$ to $D_p$: $D_p \leftarrow D_p \frown d_p$;
- Return: train/test pair $(D_t, D_p)$
```

Some aspects of the algorithm deserve clarification. First, it does not immediately accept the pair generated after Step (3h), as instances might relate with more than one value of *z* (e.g., a song might be a collaboration between two artists), making different d_{z} overlap. In that case, the number of unique elements of d_h might fall short of the specified minimum, requiring multiple attempts until finally succeeding. Second, the algorithm does not impose any restriction regarding the same value of *z* appearing across different classes to avoid benefiting systems exploiting *z*. Finally, the sampling is performed at instance level to favour scalability of the algorithm, allowing in the future regulations over multiple *z* simultaneously.

Although class-wise computations ensure stratification in the training collections, the associated testing collections will likely be imbalanced and of different size across iterations. Moreover, pruning causes regulated and unregulated testing collections to differ in size. If these issues raise reliability concerns, it might prove useful to randomly prune test collections under both conditions to a fixed size per class, such as n_r or a larger value if suitable. The choice of n_r depends on the context, but aiming at a number of regulated instances at least equal to the size of a fold in 10CV might be a good rule of thumb, both overcoming these issues and avoiding sample size concerns. In case n_r is too high and it becomes impossible to create D'_p, it is trivial to include an exit condition in the algorithm. Along with collecting instance-level information about z, if missing, only the choice of n_r requires human involvement in this otherwise automated resampling strategy.

4. Application to GTZAN

We now illustrate the analysis procedure proposed in Sec. 3, applying it to investigate the confounding effects of artist replication and infrasonic content in classification experiments involving the *GTZAN* music genre collection (Tzanetakis and Cook, 2002). The presence of multiple known confounders that can be regulated using different intervention types makes this collection ideal to showcase the factorial analysis approach we propose. The code is available online.[6]

4.1. Data and Machine Learning Methods

4.1.1. About the GTZAN Collection

GTZAN (Tzanetakis and Cook, 2002) is the most widely used public collection for music genre recognition. It contains 100 30-second music recordings of each of 10 categories: `blues`, `classical`, `country`, `disco`, `hiphop`, `jazz`, `metal`, `pop`, `reggae`, and `rock`. *GTZAN* has been used in the evaluation of over a hundred published studies (Sturm, 2014b), and remains a benchmark collection in recent publications (e.g., Choi et al. (2017)).

Sturm (2014b) provides a thorough analysis of the contents of *GTZAN*, reporting repetitions, distortions and mislabellings, highlighting the replication of artists in many classes. At the moment of writing, all but 23 of the 1000 recordings in *GTZAN* have been identified. (An updated index is included with the code.) **Fig. 2** summarises the artist distribution for each class in *GTZAN*, assuming all artists from still unidentified excerpts are unique. Queen is the only artist known to appear across classes in the collection (`rock` and `metal`). `blues` remains the class with highest artist replication, with all but one artist appearing in more than 10 excerpts. In `reggae`, a single artist (Bob Marley) appears in more than a third of the excerpts. This complicates creating conventional artist filters.

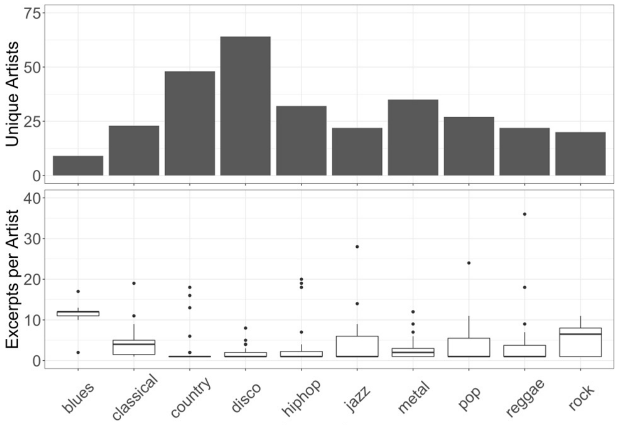

Figure 2: Artist distribution across classes in *GTZAN*, showing the number of unique artists (Top) and the quartiles of the number of excerpts per artist (Bottom) in each class. Dots indicate outliers.

Rodríguez-Algarra et al. (2016) highlight a further issue in *GTZAN*. Some recordings contain acoustic information at frequencies below 20 Hz associated with genre annotations, although it is not yet clear which. Such infrasonic information is arguably extraneous for the problem of genre recognition.

4.1.2. Evaluation Conditions

We draw training and testing instances from *GTZAN* using the regulated bootstrap resampling strategy described in Sec. 3.3, regulating over artist metadata. In particular, we draw $K = 40$ pairs with $n_r = 10$. This ensures that at least 10 recordings per *GTZAN* class in each testing collection feature no artist that appears in its corresponding training collection. **Table 1** includes estimates of the proportion of train/test samples that require curated sampling to achieve this.

Fig. 3 shows the distribution of the number of unique excerpts per class across iterations. Although all training collections contain exactly 100 excerpts per class, some of them are repeated. The expected number of unique instances in a bootstrap sample drawn from 100 elements is 63.2 (Efron and Tibshirani, 1997), approximately what **Fig. 3** (Top) shows for the training collections despite the curation. The size of the testing collections (with and without pruning) matches their number of unique excerpts, as they contain no duplicates. **Fig. 3** (Top) also shows that training collections generally include more unique excerpts than their corresponding testing collections. Some outliers appear in `reggae` due to the large proportion of Bob Marley recordings. **Fig. 3** (Bottom) highlights the expected decrease in artist variety after pruning. As suggested by **Fig. 2**, `blues` suffers from the lowest variety in all collections.

We also manipulate every recording in *GTZAN* similarly to the audio filtering intervention by Rodríguez-Algarra et al. (2016). We design a high-pass IIR filterbank, with stop-band frequency at 19 Hz, passband frequency at 20 Hz, 60 dB attenuation in the stop-band, and maximum 1 dB ripple allowed in the pass-band. Combining which recordings are included in the collections with their audio filtering status defines six distinct evaluation conditions for each iteration. We refer to these conditions as `train`, `test`, and `pr. test`, appending "`(filt.)`" to their name (e.g., `train (filt.)`) if the recordings have been high-pass filtered.

Table 1: Estimated proportion of train/test samples requiring curated sampling for each *GTZAN* class if drawn using Alg. 1 to regulate over artists, from 100,000 simulations with $n_r = 10$.

`blues`	`classical`	`country`	`disco`	`hiphop`	`jazz`	`metal`	`pop`	`reggae`	`rock`
99.99%	98.62%	1.93%	0.11%	79.19%	99.22%	71.40%	98.31%	98.11%	99.70%

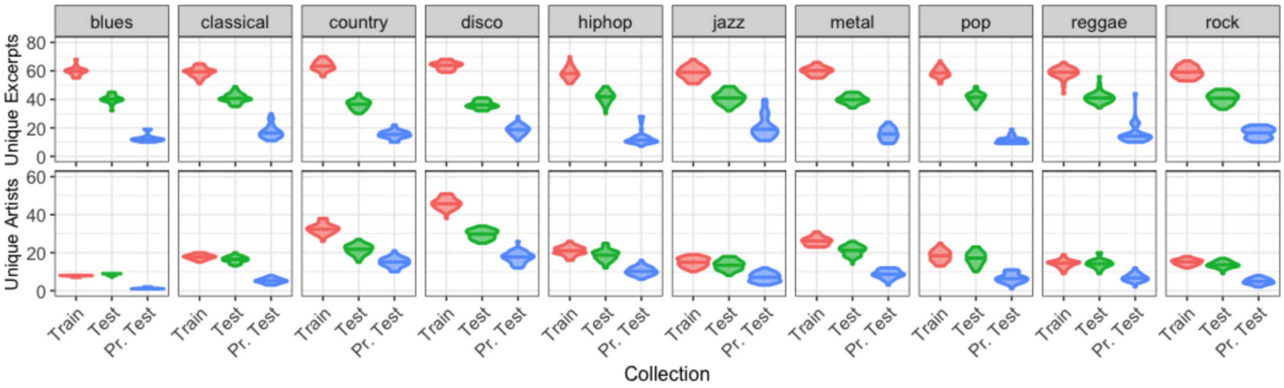

Figure 3: Distribution of the number of unique excerpts (Top) and artists (Bottom) per class in the training and testing collections sampled from *GTZAN* using bootstrap regulated over artists.

4.1.3. Feature Extraction and Learning Algorithms

We train prediction systems using multiple combinations of feature representations and learning algorithms. The learning algorithms we employ cover a wide range of supervised learning approaches, from parametric to non-parametric. In particular, we use `scikit-learn`[7] implementations of: Naive Bayes (NB), 1- and 5-Nearest Neighbours (`1-NN` and `5-NN`), Decision Trees with and without AdaBoost (`ABDT` and `DT`), Random Forests (`RF`), Support Vector Machines (`SVM`), and Multi-layer Perceptrons (`MLP`). In order to gauge how confounding affects measurements, we need a variety of modelling approaches whose performance on *GTZAN* spans the axis, including at its lower end, and not necessarily the best-performing. We thus use out-of-the-box implementations and avoid hyperparameter tuning, which allows us to increase the number of methods and iterations considered. Therefore, the reported performances should not be taken as representative of the potential of each method.

We select multiple feature representations, focusing on different aspects of the audio signals, from two sources: the `essentia` music extractor (Bogdanov et al., 2013) and the scattering-based audio features by Andén and Mallat (2014). We group the features extracted from `essentia` into 8 disjoint sets: `Rhythm`, `Tonal`, `Tim+Dyn` (i.e., timbre plus dynamics), `MFCC`, `GFCC`, `Barkbands`, `Melbands`, and `Erbbands`, referred to jointly as `non-scattering` features hereinafter. Regarding the scattering-based features, we compute Mel-scaled (`Mel Sc.`), first-layer (`1-L Sc.`), and joint first- and second-layer time-scattering features (`1&2-L Sc.`). Unlike `non-scattering` features, these express frame-level information, so we add excerpt-level summary statistics of first-layer time-scattering features (`Des.1-L Sc.`).

4.2. Instance Assignment: Artist Information

We first compare measurements obtained in `test` and `pr. test` to assess the effect of artist replication. Other than size, these conditions differ only in whether their artist content is regulated. We train systems using every combination of the selected feature extractors and learning algorithms on each of the K training collections drawn, yielding 40 × 12 × 8 = 3840 distinct systems. **Fig. 4** shows performance statistics across iterations, using mean recall as metric to compensate for class imbalances derived from the resampling strategy employed. We see systematically lower performance in `pr. test` than the others, agreeing with results in Sturm (2014b).

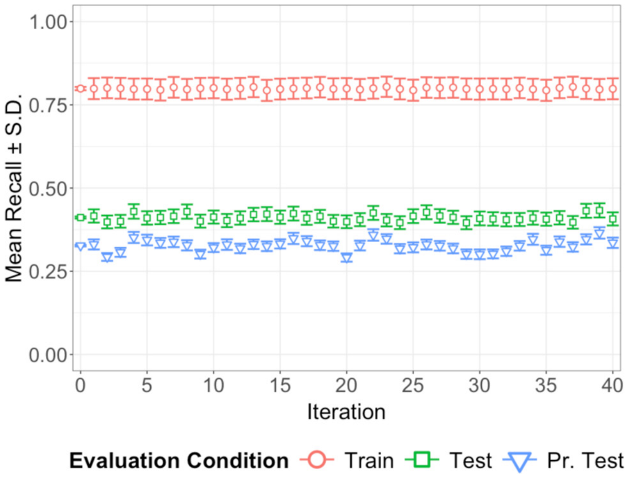

Figure 4: Mean recall (± standard deviation) in `train`, `test`, and `pr. test` for each regulated bootstrap iteration over all combinations of feature extraction and learning algorithms on original *GTZAN* recordings. Position 0 represents the mean recall over all iterations.

Only 12.8% of all measurements in `pr. test` are greater or equal than their counterpart in `test`. From 100 simulations using randomly generated subsets of `test` with identical class sizes as in `pr. test`, we find that figure to be on average 53.7% (±2.3) without the regulation. Moreover, 15.6% (±0.5) of measurements in `pr. test` are greater than or equal to their counterpart in the simulations, compared to an average of 54.4% (±2.3) between simulations (see Supplementary Material S3). This suggests performance differences arise due to the regulation and not size.

An estimate of κ according to Eq. (2) yields a decrease in mean recall of approximately $\hat{\kappa} \approx 0.085$ (8.5 percentage points). A closer look at the measurements reveals the naivety of this approach. **Fig. 5** shows that, despite consistently lower results in `pr. test` than `test`, the distribution of the performance metrics varies widely when marginalised over class, feature set or learning

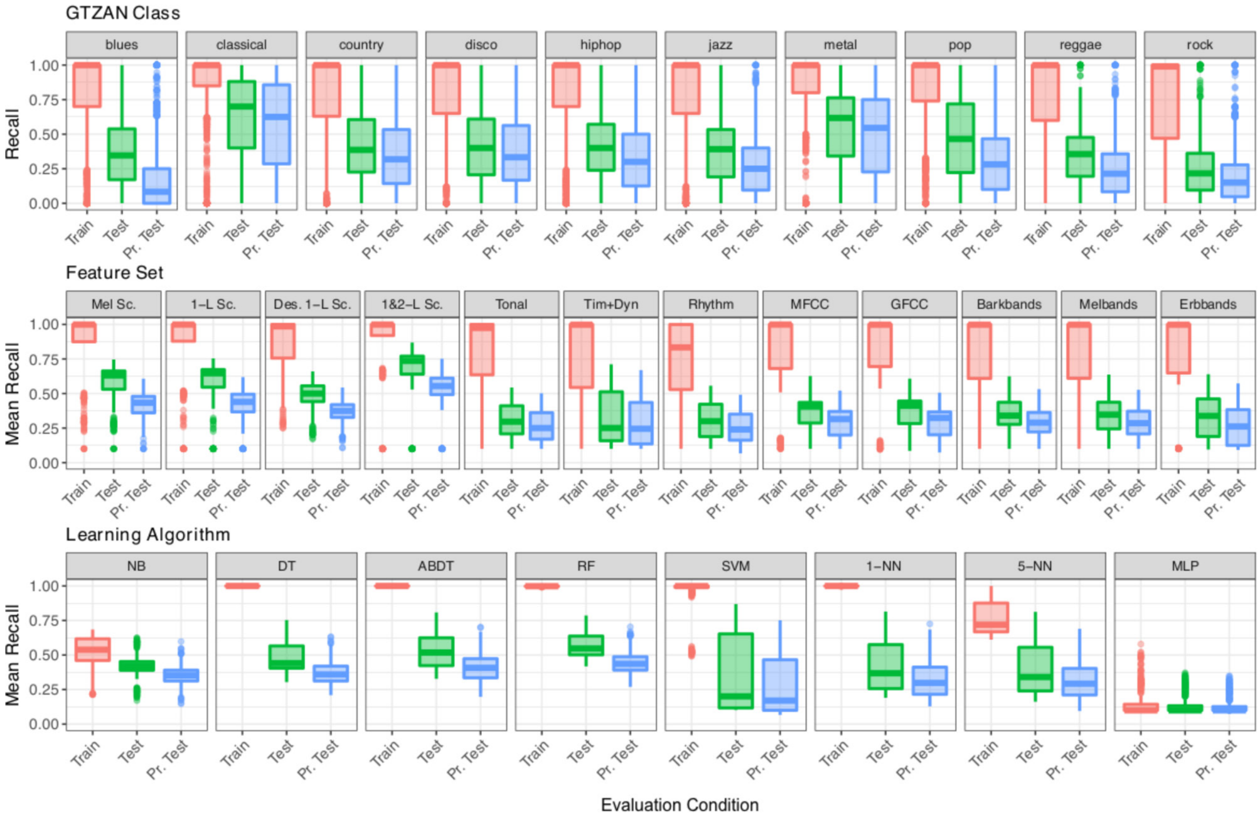

Figure 5: Quartiles of (mean) recall distribution obtained in train, test, and pr. test, marginalised over *GTZAN* class (Top), feature set (Middle), and learning algorithm (Bottom).

algorithm. This suggests the confounding effect of artist replication in *GTZAN* does not impact performance measurements as an additive fixed effect, i.e., that there exist interactions between that metric and the classes, features, and learning algorithms.

For each *GTZAN* class, we see clear differences in the distributions of recall. The largest difference by far occurs in blues, with an average drop of 19 percentage points—a relative decrease of more than 53%. This behaviour might be expected, as blues is the class in *GTZAN* with the least artist variety. Similarly, the average recall in reggae drops 9.7 percentage points (almost 30% relative decrease), which may relate to one artist dominating the class. The relative decrease in pop is even higher (32.4%), and might arise from duplicate recordings in that class (Sturm, 2014b).

At the other end of the spectrum, we find metal, classical and disco suffer average relative decreases in recall below 10% (7.7%, 8.1%, and 9.6%, respectively). **Fig. 2** shows disco is the class in *GTZAN* with largest artist variety. Despite having less than half the number of unique artists, however, metal and classical not only suffer the smallest relative average decrease, but also yield the highest average in both test and pr. test. This suggests these classes are so different from others in *GTZAN* that they are distinguished even without artist-specific information.

Marginalising over feature extraction method, **Fig. 5** shows systems using scattering-based features tend to obtain higher performances than non-scattering, both in test and pr. test. The variance in frame-level approaches is substantially lower than for those computing whole-excerpt summaries, even in train. Overall, differences in mean recall between test and pr. test are highest in both Mel Sc. and 1-L Sc. features, with a decrease of approximately 15.8 percentage points in both—a decrease of 27.7% from test. The lowest drop, both in absolute and relative terms, occurs in Tim+Dyn (4 percentage points, 12% decrease from test).

Marginalising over learning algorithm also reveals clear differences in performance distribution. Systems constructed using the suboptimal MLP architecture tend to perform close to the random baseline of 0.1 mean recall. For every single learning algorithm, including MLP, performance decreases between train and test, and between test and pr. test. Apart from MLP, NB is the only other algorithm that shows an average relative difference in mean recall between test and pr. test below 20%. It is also the algorithm that seems to suffer the least from overfitting. Despite a far lower performance in train, NB systems perform on average equivalently to 1-NN systems in test, and slightly superior in pr. test, with substantially lower variance in both cases. Systems from all other algorithms decrease on average around 20.5% to 23.5% between test and pr. test, with DT having the largest drop.

Fig. 6 relates the performance trained systems achieve in test with that in pr. test, both individually (left) and grouped by feature representation and learning algorithm (right). A linear fit gives a slope $\hat{\alpha} = 0.712 \pm 0.003$ and an intercept $\hat{\kappa} = 0.034 \pm 0.001$ ($R^2 = 0.929$). The slope is thus lower than the case of no confounding, represented with

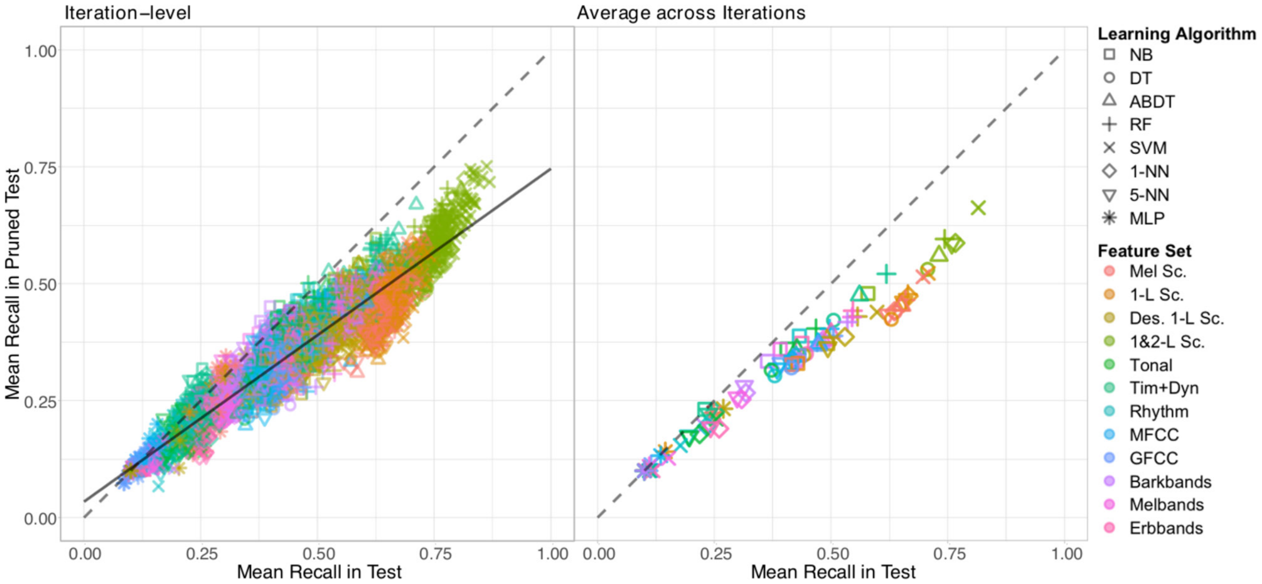

Figure 6: Relationship between mean recall in `test` and `pr. test` obtained by systems constructed with different combinations of feature representations and learning algorithms on training collections sampled from *GTZAN* with bootstrap regulated over artists, represented both as individual values for each system (Left) and averages across iterations (Right). The dashed line indicates the case of equal mean recall in `test` and `pr. test`; the solid line indicates the linear regression model fitting the data as in Eq. (1).

a dashed line in **Fig. 6**. This suggests regulating by artist in *GTZAN* attenuates the estimated performance to around 70% of its unregulated value. This equates to considering the confounding effect of artist replication in *GTZAN* as a gain in mean recall of approximately $1/0.712 \approx 1.4$.

The data points at the higher end of performance measurements in **Fig. 6** deviate from the estimated regression line. This may suggest using more complex models, but exponential and polynomial models up to third degree do not substantially improve the fit. A model including both third degree polynomial and exponential terms increases R^2 to 0.932, but at the cost of hard to interpret coefficients and the risk of overfitting.

4.3. Data Manipulation: Infrasonic Content

The analysis by Rodríguez-Algarra et al. (2016) suggests that infrasonic content in *GTZAN* recordings affects performance estimates of scattering-based `SVM` systems. We here include non-scattering feature representations and a wider range of learning algorithms to gauge the extent of this effect. We compare performance measurements from the same systems in Sec. 4.2 in `test` and `test (filt.)`, which differ exclusively in sub-20 Hz content. Overall, the average decrease in mean recall between these two conditions calculated as in Eq. (2) is $\hat{\kappa} \approx 0.098$, slightly larger than the one we observe for artist replication.

Fig. 7 shows the observed performances, marginalised by *GTZAN* class, feature representation, and learning algorithm. The figure includes measurements on the training recordings and their filtered equivalents, revealing that performance estimates decrease between `train` and `train (filt.)` across system-construction methods and classes. Overall, the average decrease in mean recall between these two conditions is of 28 percentage points. Regardless of whether they exploit class-specific patterns of infrasonic content to predict annotations in unseen instances, systems trained in *GTZAN* seem to often rely on such content (or related information, such as the overall energy level) to identify recordings previously seen during training and predict their class.

The *GTZAN* class with largest relative average decrease in recall between `test` and `test (filt.)` is `jazz`, with 37.2%, followed by `pop`, the largest drop in absolute terms, and `blues`, with 34.9% and 33.7%, respectively. The smallest decrease by far occurs in `hiphop`, with an average 5.5% relative decrease. The closest classes are `reggae` and `classical`, both with over 16.5% relative decrease on average. Some might speculate these reductions in performance originate from removing information legitimately characteristic of some music genres, such as sub-bass kick drums in Hip-Hop recordings. Seeing how measurements in *GTZAN's* `hiphop` are barely affected by the intervention compared to other classes that should not present any pattern at those frequencies (such as `jazz`), seems to disprove this explanation.

Marginal analysis of measurements by feature representation reveals two clearly distinct behaviours, and suggests models such as Eq. (1) might not apply in this case. The mean recall of scattering-based systems decreases on average between 41% (`1&2-L Sc.`) and 57% (`1-L Sc.`) when comparing `test` and `test (filt.)`. On the other hand, no average decrease of `non-scattering` features exceeds 4%, one order of magnitude lower. This brings the average performance of all scattering-based systems except those using `1&2-L Sc.` to the bottom of the list in `test (filt.)`, despite appearing substantially more successful than any `non-scattering` feature set in `test`. Feature representations such as `MFCC` discard infrasonic information, with all filters centered at frequencies above the human hearing threshold (Davis and Mermelstein, 1980). Scattering-based features, even those supposedly Mel-scaled, have multiple filters centered below 20 Hz (Rodríguez-Algarra et al., 2016). **Fig. 8** shows the distinct behaviour of each group, where measurements from systems using `non-scattering` feature representations follow quite closely the ideal behaviour indicated by the dashed line, whereas those from scattering-based systems tend to create clusters away from that line.

Among the considered learning algorithms, `SVM` is the one with largest drop in performance between `test` and `test (filt.)` – an average decrease of 42.6% in mean recall. Other than `MLP`, `NB` is the algorithm that suffers the lowest average decrease (10.5%), with the remaining algorithms decreasing between 16.7% and 31.7% mean recall on average.

Fig. 8 separates measurements from systems using `Des. 1-L Sc.` because the clusters they form suggest

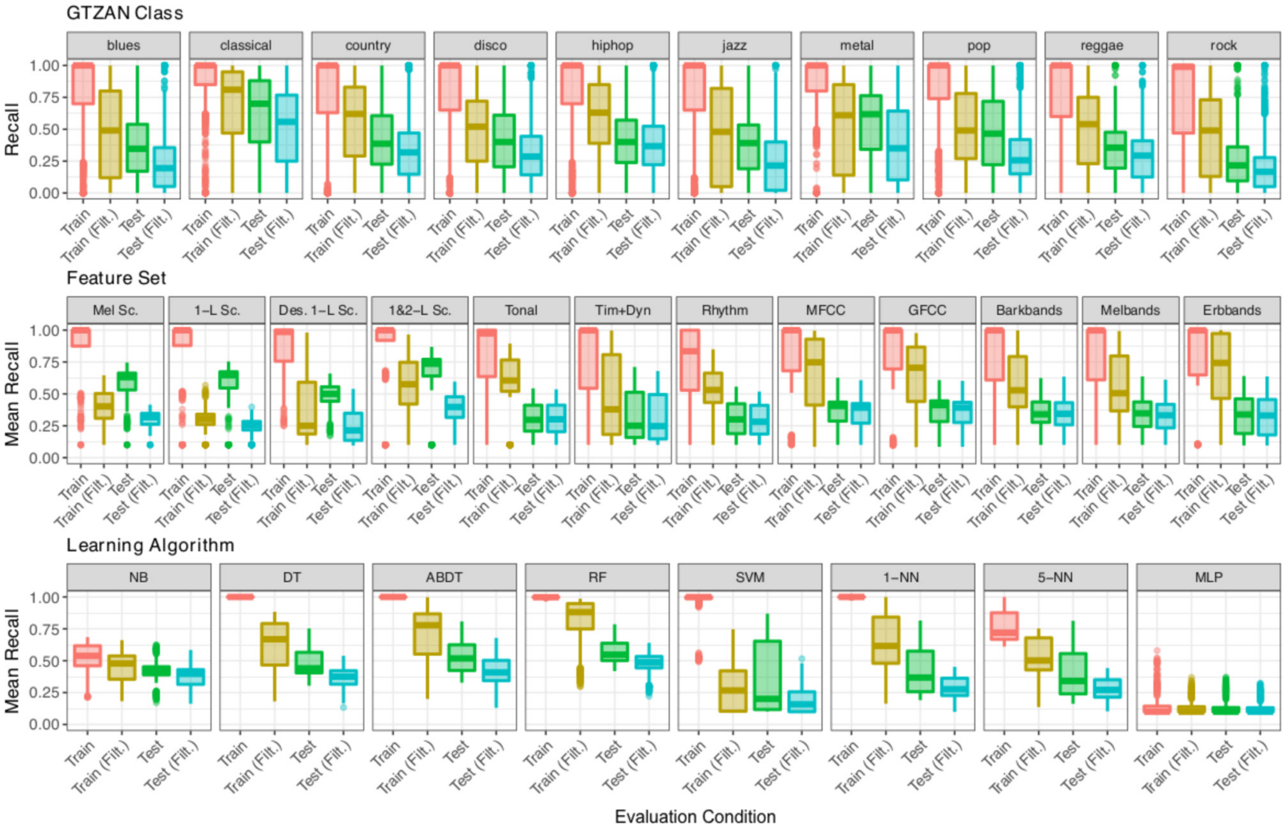

Figure 7: Quartiles of (mean) recall distribution obtained in `train`, `train (filt.)`, `test`, and `test (filt.)`, marginalised over *GTZAN* class (Top), feature set (Middle), and learning algorithm (Bottom). Note that the colours in this figure not matching those in Figs. 3, 4 and 5 correspond to different evaluation conditions.

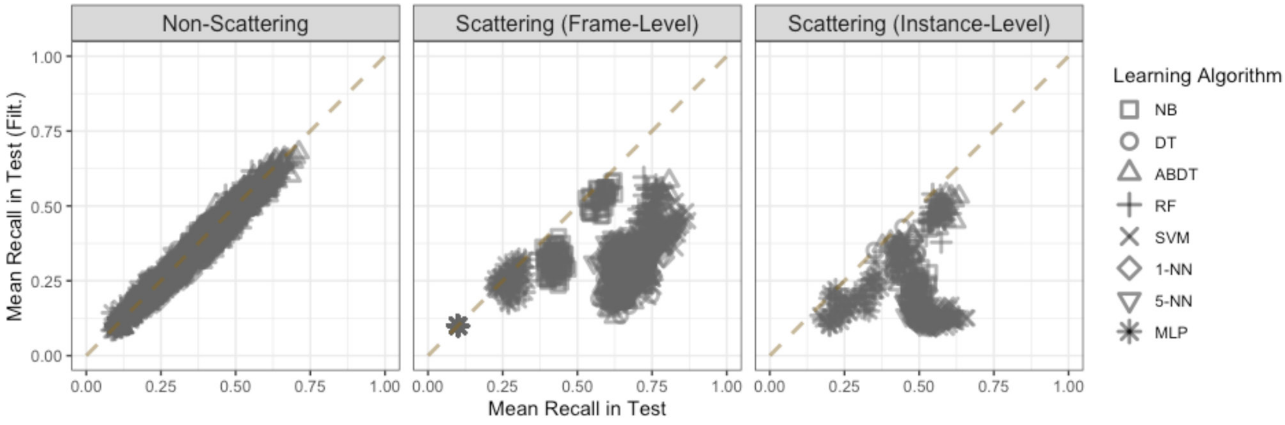

Figure 8: Relationship between mean recall in `test` and `test (filt.)` obtained by systems constructed with different combinations of feature representations and learning algorithms using training collections sampled from GTZAN with bootstrap regulated over artists, grouped by the source of feature set. Non-Scattering features are extracted with `essentia`. Instance-level scattering features correspond to `Des. 1-L Sc.`; the rest are frame-level. The dashed line indicates the case of equal mean recall in `test` and `test (filt.)`.

interactions with learning algorithms different from frame-level scattering systems. Leaving MLP systems aside, the clusters close to the dashed line in the middle panel only contain measurements from NB systems. Their average decrease in mean recall is of 9 percentage points, corresponding to a 19% drop. NB systems with Des. 1-L Sc. feature representations, however, suffer an average 52% decrease. Conversely, the clusters closer to the ideal case for Des. 1-L Sc. systems correspond to algorithms of a similar kind: DT, ABDT, and RF. The average drop in performance for these algorithms is between 15% and 25% with Des. 1-L Sc. feature representations, but DT is the algorithm with the largest drop for the rest of the scattering-based representations, with an average 61.5% decrease in mean recall; ABDT follows with 55.8% decrease.

4.4. Factorial Integration of Interventions

The separate analyses above highlight the particularities of each confounding effect. We now conduct both interventions simultaneously in a factorial way: we expose each trained system to all evaluation conditions. In particular, pr. test (filt.) contains the same instances as pr. test but high-pass filtered.

Fig. 9 summarises the performance distributions in test and pr. test, both under original and filtered audio conditions, marginalised by *GTZAN* class, feature representation and learning algorithm. We see the distribution in pr. test (filt.) is centered around lower values than those on any other evaluation condition for scattering-based representations. Systems using non-scattering only suffer drops when regulating over artist, but not due to high-pass filtering.

Combining multiple interventions allows us to analyse interactions between confounders. Using the notation in Sec. 3.2, let $\hat{y}$, $\hat{y}'_1$, $\hat{y}'_2$, and $\hat{y}'_{1,2}$ be the mean recall a system obtains in test, pr. test, test (filt.), and pr. test (filt.), respectively. Let Δ_A be the "accumu-lated" variation of mean recall, defined as in Eq. (3), and Δ_R be the "real" variation, defined as in Eq. (4). **Fig. 10** shows the distribution of Δ_R–Δ_A, grouped by

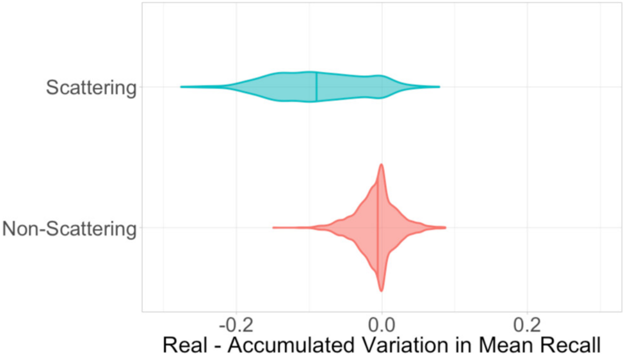

Figure 10: Distribution of differences between the real variation Δ_R and the accumulated variation Δ_A in mean recall for artist and infrasonic regulation interventions in *GTZAN*, grouped by the source of feature set.

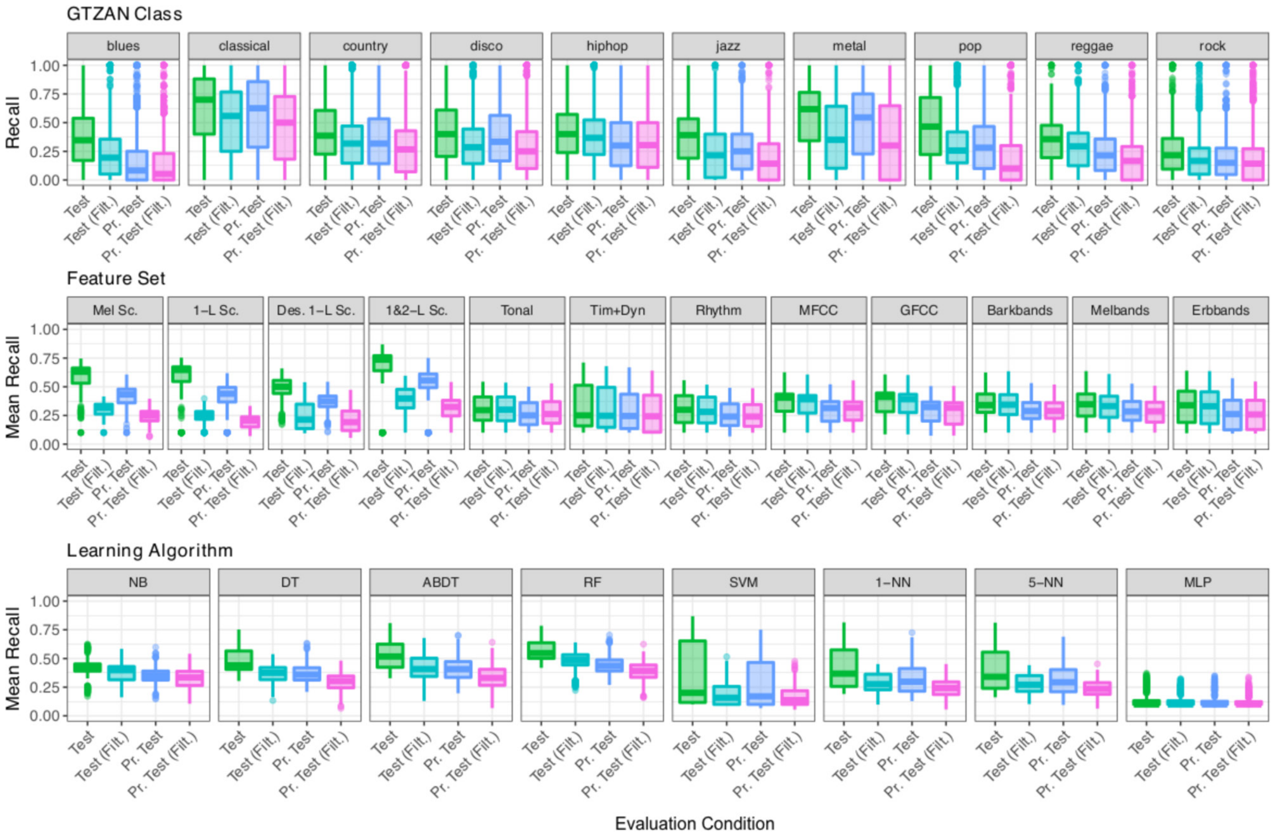

Figure 9: Quartiles of (mean) recall distribution obtained in test, test (filt.), pr. test, and pr. test (filt.), marginalised over *GTZAN* class (Top), feature set (Middle), and learning algorithm (Bottom). Note that the colours in this figure not matching those in Figs. 3, 4, 5 and 7 correspond to different evaluation conditions.

origin of feature set. We see the difference is centered around 0 for systems using `non-scattering` feature representations, as the overall confounding effect in those systems originates mainly from artist replication. On the other hand, we see the difference tends to be negative for systems using scattering-based feature representations. This suggests the two confounding effects overlap for those systems, which stands to reason as the recording conditions of excerpts from the same artist are likely to be similar.

Confounders not only impact the magnitude of performance estimates, as we saw before, but also alter their ranking. For instance, **Fig. 11** shows that, for systems trained using `1&2-L Sc.`, `NB` goes from the lowest (ignoring `MLP`) to the highest position depending on whether one applies a data manipulation intervention; Sturm (2014b) reaches the same conclusion. Similar interactions arise in other methods (see Supplementary Material S3).

Kendall's τ provides estimates of concordance between rankings, with 1 meaning exact match, −1 completely reversed match, and 0 non-correlation (Kendall, 1938). The value of τ between `test` and `pr. test` is fairly high (0.91), which aligns with our interpretation that artist information biases performance estimates in a similar way across methods (i.e., without substantially altering their ordering). τ decreases between `test` and `test (filt.)` (0.52) and between `test` and `pr. test (filt.)` (0.45), reflecting the fact that infrasonic content affects ranking in higher degree.

5. Discussion

Our procedure for characterising confounding effects in music classification experiments facilitates understanding how particular confounders impact evaluation outcomes. It extends well-established practices in MIR, such as filtered partitioning, overcoming their limitations. In particular, our approach enables integrating multiple types of interventions, targeted to the same or distinct potential confounders but not necessarily multiple interventions of the same type. Introducing a suitable resampling strategy, such as the regulated bootstrap we describe, is key to this integration. This provides a distribution of regulated/unregulated measurement pairs instead of single sample comparisons, such as those found in previous studies (e.g., Rodríguez-Algarra et al. (2016)). It also enables disentangling the effects of confounding between training and prediction.

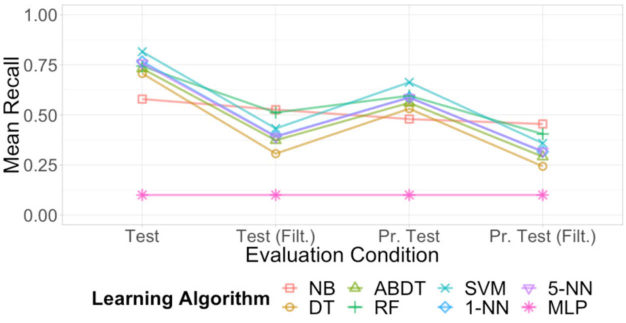

Figure 11: Interaction between learning algorithm and evaluation condition in average mean recall for systems constructed using training collections sampled from *GTZAN* with bootstrap regulated over artists and `1&2-L Sc.` feature representations.

The example application using *GTZAN* showcases the benefits of our procedure. The factorial structure across runs of the experiment enables both marginal and joint analyses, revealing distinct behaviours when systems are exposed to each potential confounder, as well as their interactions. These observations, however, are subject to the caveats we discuss next.

Systems in our case study underperform due to the lack of hyperparameter tuning. We deliberately prioritise variety over optimisation to gather performance estimates of different magnitude and susceptibility to confounding. The evidently unsuitable `MLP` architecture chosen is a clear example of this, allowing us to obtain measurements close to the random baseline that could still be affected by the regulations. Alternatives to achieve measurements in the lower end, such as random or systematic classifiers, would by definition remain unaffected regardless of the condition. Tuning model hyperparameters, while relevant in real benchmarking studies, would likely concentrate performances at the high end of the axis, thus hampering the intended illustration of the proposed methodology. Further studies could incorporate optimisations as additional treatment conditions in the experimental design to illuminate how tuning impacts the susceptibility of systems to confounding effects.

Our analysis suggests the confounding effect of artist replication in *GTZAN* appears multiplicative rather than additive. This might seem obvious knowing that the performance metric used is bounded between 0 and 1. As Carterette (2012) mentions, additive effects could easily make predicted values exceed those boundaries. In fact, current proposals for modelling measurements from classification experiments (e.g., Alpaydin (2014); Eugster (2011)) assume additive effects for all components of the experiment, ignoring the boundary problem. This motivates revising those models, potentially using *logit* transformations to convert multiplicative effects into unbounded additive components, although it might be unnecessary if one's only concern is the ranking between systems.

The clear divergence between the proposed linear model and the observations of the highest end of performance measurements in **Fig. 6** might require collecting further data, either from not yet considered methods or through the optimisation of existing ones. That divergence, however, illuminates a substantial difference in slope between observations using a particular feature representation and the overall trend. This seems to reflect Simpson's paradox (Simpson, 1951; Pearl, 2014), in which behaviour per group diverges from, or even completely reverses, the aggregated pattern. Together with the clusters suggested in **Fig. 8** for the case of infrasonic content, this highlights the need to study interactions between learning algorithms and feature representations under various potentially confounding environments.

A general limitation of our method regards its scope, as it neither illuminates previously unknown confounders nor prevents confounding from affecting performance estimates. It is actually impossible to guarantee confounding does not appear at all, as there might be a plethora of yet unknown potential confounders still affecting observations to some extent. Devoted exploratory analyses informed by both domain knowledge and system analysis are necessary to uncover further potential confounders before assessing their impact using intervention-type approaches. This enables one to design or improve system-construction methods accounting for that risk and devise train/test mechanisms that prevent them from appearing. To this end, it is of paramount importance for MIR researchers to devote efforts to expose such potential confounders and assess their effects.

The current study does not consider all possible effects of confounding, focusing on characterising its effects on evaluation results, but leaving aside other equally relevant research questions for the moment. In particular, by introducing and comparing new conditions only at the prediction stage, we ignore the effects of confounding on the training of systems. This might be easily addressed for data manipulation interventions by adding training conditions with manipulated recordings, thus multiplying the number of models to consider and experimental conditions to analyse. In the case of instance assignment interventions, however, it would require modifying the regulated bootstrap resampling strategy to enable creating regulated and unregulated collections for both training and testing simultaneously. This is a promising research path for future work.

Some may argue that the curation process inherent to regulated bootstrap resampling introduces biases in the performance estimates, and thus in the comparisons between conditions, bringing into question the validity of the extracted conclusions. However, this process increases control over the measurements, not unlike the stratification performed in conventional classification experiments, as well as blocking in statistical design of experiments (Montgomery, 2013). In particular, stratification preserves the distribution of annotations present in the original collection, thus facilitating performance estimates within the collection that approximate what systems would have achieved had they used the whole collection, but does not account for the likely imbalances that real life data could have. This favours internal over external validity, a methodological trade-off often encouraged to create experimental conditions that differ only in the factor under study and to warrant against external factors affecting the conclusions (Shadish et al., 2002).

The size of the testing collections generated might also cause concern, as there is no guarantee that the original class balance remains and, by definition, the number of instances decreases after pruning. The use of mean recall as performance metric should compensate for imbalances, and, in the case study we conduct, the differences in performance between collections of the same iteration clearly exceed the differences across iterations. This suggests unequal size should not affect our conclusions. As mentioned before, in the general case, one might want to introduce a further control step that forces all original and pruned testing collections, and all classes within those collections, to match in size, such as randomly selecting a fixed number of instances. This might also alleviate the likely lack of independence between instances from the curation involved in their sampling. Due to the infeasibility of pure random sampling from the whole population, evaluation collections are often constructed through convenience sampling (Urbano et al., 2013), hampering independence in the first place. Curation thus does not necessarily affect conclusions in this regard.

The analysis approach we describe and exemplify in this article can be applied to a wider range of collections, machine learning methods and potential confounders than the ones we show here. Published studies and evaluation exchanges, such as MIREX, could incorporate similarly extended pipelines to assess the susceptibility of proposed systems to a set of interventions. Domains other than music would also benefit from similar analysis approaches. Despite its caveats, the insights obtained through this kind of analysis should help building more robust systems and obtaining performance estimates that generalise to deployment scenarios.

6. Conclusion

In this article, we explored the nature of confounding in music classification experiments and described a procedure for assessing its impact in the evaluation of MIR systems and methods. We used interventions in the experimental pipeline and proposed a novel resampling strategy that introduces regulations on a conventional bootstrap sampling. Using our approach, we analysed the effects of artist replication and infrasonic content in *GTZAN* on performance estimates of a range of feature extraction methods and learning algorithms. We found the effects of artist replication appear to be multiplicative, while those from infrasonic content depend on the system-construction method employed. We also showed that these two potential confounders appear to partially overlap, and their effect might alter the ranking of different solutions with respect to their average performance. Further improvements of the approach could include introducing evaluation conditions through interventions on the training data, controlling the testing collection size, and analysing the effect of optimisation. We hope that future MIR research will focus not only on maximising performance estimates, but also on developing and assessing solutions with regards to their susceptibility to confounding effects.

Notes

[1] We avoid the conventional term "set" and use "collection" instead for both training and testing material to include those from sampling with replacement, such as the bootstrap (Efron, 1977), as "set" implies no repeated elements.

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME.

[3] Infrasonic content refers to vibrations below 20 Hz, the lower threshold of human hearing.

[4] The symbol ~ indicates “modelled as”, similar to R notation.

[5] z_i may contain multiple elements, e.g., collaborations between artists.

[6] https://code.soundsoftware.ac.uk/projects/confint.

[7] http://scikit-learn.org/stable/.

Additional File

The additional file for this article can be found as follows:

- **Supplementary Material.** Example of regulated bootstrap resampling and auxiliary figures. DOI: https://doi.org/10.5334/tismir.24.s1

Competing Interests

Simon Dixon is a co-Editor-in-Chief of the Transactions of the International Society for Music Information Retrieval. He was removed completely from all editorial processing. There are no other competing interests to declare.

References

Alpaydin, E. (2014). *Introduction to Machine Learning*. The MIT Press, Cambridge, MA, USA, 3rd edition.

Andén, J., & **Mallat, S.** (2014). Deep Scattering Spectrum. *IEEE Transactions on Signal Processing*, *62*(16), 4114–4128. DOI: https://doi.org/10.1109/TSP.2014.2326991

Bogdanov, D., Porter, A., Herrera, P., & **Serra, X.** (2016). Cross-Collection Evaluation for Music Classification Tasks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR'16)*, pages 379–385. New York City, NY, USA.

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., & **Serra, X.** (2013). Essentia: An Audio Analysis Library for Music Information Retrieval. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR'13)*, Curitiba, Brazil.

Carterette, B. A. (2012). Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments. *ACM Transactions on Information Systems*, *30*(1), 4:1–4:34. DOI: https://doi.org/10.1145/2094072.2094076

Charalambous, C. C., & **Bharath, A. A.** (2016). A Data Augmentation Methodology for Training Machine/Deep Learning Gait Recognition Algorithms. In *British Machine Vision Conference*. DOI: https://doi.org/10.5244/C.30.110

Chen, J. H., & **Asch, S. M.** (2017). Machine Learning and Prediction in Medicine – Beyond the Peak of Inflated Expectations. *New England Journal of Medicine*, *376*(26), 2507–2509. DOI: https://doi.org/10.1056/NEJMp1702071

Choi, K., Fazekas, G., Sandler, M., & **Cho, K.** (2017). Transfer Learning for Music Classification and Regression Tasks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR'17)*, Suzhou, China.

Cobb, G. W. (1998). *Design and Analysis of Experiments*. Springer-Verlag.

Davis, S. B., & **Mermelstein, P.** (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Audio, Speech, and Language Processing*, *28*(4), 357–366. DOI: https://doi.org/10.1109/TASSP.1980.1163420

Dixon, S., Gouyon, F., & **Widmer, G.** (2004). Towards Characterisation of Music via Rhythmic Patterns. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 509–517. Barcelona, Spain.

Drummond, C. (2006). Machine Learning as an Experimental Science (Revisited). In *Procedings of the AAAI'06 Workshop on Evaluation for Machine Learning*, Boston, MA, USA.

Efron, B. (1977). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, *7*(1), 1–26. DOI: https://doi.org/10.1214/aos/1176344552

Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, *78*(382), 316–331. DOI: https://doi.org/10.1080/01621459.1983.10477973

Efron, B., & **Tibshirani, R.** (1997). Improvements on Cross-Validation: The 632+ Bootstrap Method. *Journal of the American Statistical Association*, *92*(438), 548–560. DOI: https://doi.org/10.1080/01621459.1997.10474007

Eugster, M. J. A. (2011). *Benchmark Experiments. A Tool for Analyzing Statistical Learning Algorithms*. PhD thesis, Ludwig-Maximilians-Universität München, München, Germany.

Flach, P. (2012). *Machine Learning*. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9780511973000

Flexer, A. (2007). A Closer Look on Artist Filters for Musical Genre Classification. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*, Vienna, Austria.

Flexer, A., & **Schnitzer, D.** (2010). Effects of Album and Artist Filters in Audio Similarity Computed for Very Large Music Databases. *Computer Music Journal*, *34*(3), 20–28. DOI: https://doi.org/10.1162/COMJ_a_00004

Hastie, T., Tibshirani, R., & **Friedman, J.** (2009). *The Elements of Statistical Learning*. Springer, 2nd edition. DOI: https://doi.org/10.1007/978-0-387-84858-7

Hernández-Orallo, J. (2016). Evaluation in Artificial Intelligence: From Task-Oriented to Ability- Oriented Measurement. *Artificial Intelligence Review*, *48*(3), 397–447. DOI: https://doi.org/10.1007/s10462-016-9505-7

Hothorn, T., Leisch, F., Zeileis, A., & **Hornik, K.** (2005). The Design and Analysis of Benchmark Experiments. *Journal of Computational and Graphical Statistics*, *14*(3), 675–699. DOI: https://doi.org/10.1198/106186005X59630

Kaufman, S., Rosset, S., & **Perlich, C.** (2011). Leakage in Data Mining: Formulation, Detection, and Avoidance. In *Proceedings of the 17th ACM SIGKDD Conference* *on Knowledge Discovery and Data Mining (KDD'11)*, pages 556–563. San Diego, CA, USA. DOI: https://doi.org/10.1145/2020408.2020496

Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrica*, *30*(1–2), 81–89. DOI: https://doi.org/10.2307/2332226

Langley, P. (1988). Machine Learning as an Experimental Science. *Machine Learning*, *3*(1), 5–8. DOI: https://doi.org/10.1007/BF00115008

Marques, G., Domingues, M. A., Langlois, T., & **Gouyon, F.** (2011). Three Current Issues in Music Autotagging. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR'11)*, pages 795–800. Miami, FL, USA.

Mendelson, A. F., Zuluaga, M. A., Lorenzi, M., Hutton, B. F., & **Ourselin, S.** (2017). Selection Bias in the Reported Performances of AD Classification Pipelines. *NeuroImage: Clinical*, *14*, 400–416. DOI: https://doi.org/10.1016/j.nicl.2016.12.018

Mishra, S., Sturm, B. L., & **Dixon, S.** (2017). Local Interpretable Model-Agnostic Explanations for Music Content Analysis. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR'17)*, Suzhou, China.

Montgomery, D. C. (2013). *Design and Analysis of Experiments*. John Wiley and Sons, 8th edition.

Pampalk, E., Flexer, A., & **Widmer, G.** (2005). Improvements of Audio-Based Similarity and Genre Classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 628–633, London, UK.

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition. DOI: https://doi.org/10.1017/CBO9780511803161

Pearl, J. (2014). Comment: Understanding Simpson's Paradox. *The American Statistician*, *68*(1), 8–13. DOI: https://doi.org/10.1080/00031305.2014.876829

Pfungst, O., Stumpf, C., Rahn, C. L., & **Angell, J. R.** (1911). Clever Hans (the Horse of Mr. von Osten): A Contribution to Experimental, Animal, and Human Psychology. *Journal of Philosophy, Psychology and Scientific Methods*, *8*(24), 663–666. DOI: https://doi.org/10.2307/2012691

Rodríguez-Algarra, F., Sturm, B. L., & **Maruri-Aguilar, H.** (2016). Analysing Scattering-Based Music Classification Systems: Where's the Music? In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR'16)*, pages 344–350. New York City, NY, USA.

Shadish, W. R., Cook, T. D., & **Campbell, D. T.** (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin Company, Boston, MA, USA.

Simpson, E. H. (1951). The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society, Series B*, *13*, 238–241. DOI: https://doi.org/10.1111/j.2517-6161.1951.tb00088.x

Stowell, D. (2017). Reducing Confounding Factors in Automatic Acoustic Recognition of Individual Birds. In *Workshop on "Horses" in Applied Machine Learning (HORSE 2017)*, http://c4dm.eecs. qmul.ac.uk/horse 2017/HORSE2017_Stowell.pdf

Sturm, B. L. (2014a). A Simple Method to Determine if a Music Information Retrieval System Is a "Horse". *IEEE Transactions on Multimedia*, *16*(6), 1636–1644. DOI: https://doi.org/10.1109/TMM.2014.2330697

Sturm, B. L. (2014b). The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval. *Journal of New Music Research*, *43*(2), 147–172. DOI: https://doi.org/10.1080/09298215.2014.894533

Sturm, B. L. (2016a). Revisiting Priorities: Improving MIR Evaluation Practices. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR'16)*, New York City, NY, USA.

Sturm, B. L. (2016b). The "Horse" Inside: Seeking Causes of the Behaviours of Music Content Analysis Systems. *Computers in Entertainment, Special Issue on Musical Metacreation*, *14*(2). DOI: https://doi.org/10.1145/2967507

Trochim, W. M. K., & **Donnelly, J. P.** (2007). *The Research Methods Knowledge Base*. Atomic Dog, 3rd edition.

Tzanetakis, G., & **Cook, P.** (2002). Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, *10*(5), 293–301. DOI: https://doi.org/10.1109/TSA.2002.800560

Urbano, J., Schedl, M., & **Serra, X.** (2013). Evaluation in Music Information Retrieval. *Journal of Intelligent Information Systems*, *41*(3), 345–369. DOI: https://doi.org/10.1007/s10844-013-0249-4

Weihs, C., Jannach, D., Vatolkin, I., & **Rudolph, G.** Editors (2017). *Music Data Analysis. Foundations and Applications*. CRC Press.

How to cite this article: Rodríguez-Algarra, F., Sturm, B. L., & Dixon, S. (2019). Characterising Confounding Effects in Music Classification Experiments through Interventions. *Transactions of the International Society for Music Information Retrieval,* 2(1), pp. 52–66. DOI: https://doi.org/10.5334/tismir.24

Submitted: 19 October 2018 **Accepted:** 27 June 2019 **Published:** 21 August 2019

]u[*Transactions of the International Society for Music Information Retrieval* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS