Support Vector Machine Active Learning for Music Mood Tagging

Álvaro Sarasúa, Cyril Laurier and Perfecto Herrera

Music Technology Group, Universitat Pompeu Fabra
{alvarosarasua, cyril.laurier}@gmail.com, perfecto.herrera@upf.edu

Abstract. *Active learning* is a subfield of machine learning based on the idea that the accuracy of an algorithm can be improved with fewer training samples if it is allowed to choose the data from which it learns. We present the results for Support Vector Machine (SVM) active learning experiments for music mood tagging based on a multi-sample selection strategy that chooses samples according to their proximity to the boundary, their proximity to points in the training set and the density around them. The influence of those key active learning parameters is assessed by means of ANalysis Of Variance (ANOVA). Using these analyses we demonstrate the efficiency of active learning compared to typical full-dataset batch learning: our method allows to tag music by mood more efficiently than a regular approach, requiring fewer instances to obtain the same performance than using random sample selection methods.

Keywords: active learning, music mood detection, support vector machines

1 Introduction

Detection of moods and emotions in music is a topic of increasing interest in which many problems and issues are still to be explored. So far, most works have dealed with the so-called "basic emotions"such as happiness, sadness, anger, fear and disgust [1] [2]. This work tries to give one step towards detecting music emotions that are more specific and hard to articulate. One of the factors that make non-basic mood detection difficult is that they are usually perceived with less agreement among listeners than basic ones [1]. In such a context, user-tailored systems that learn from user perception become a must. Usually, such systems require getting a big amount of information from the user (e.g. using relevance feedback or other techniques) and his/her tastes in a process that can take too long.

In our case, we extend Laurier's method [3] for music mood classification. In such system, mood tags are assigned by means of a two-step process of feature extraction and statistical analysis. Those features are obtained for labeled songs and then the system is trained to learn which values of the extracted features define every group. A careful selection of the training examples is always required in order to maximize the generalization power of the final system.

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

Our work deals with the task of optimizing the training process by trying to speed user customization up, keeping the system as general as possible to different music genres for a specific user. We use **active learning** as it has shown good performance on many multimedia applications [4] [5] [6] but, surprisingly, it has been rarely used in Music Information Retrieval (MIR) even though its promising results [7] [8]. We perform a study on the application of active learning techniques to music mood classification extending Laurier's method with a multi-sample selection strategy based on Wang's [8]. In addition, we study the influence that all the parameters of this strategy have on the final results.

During this introduction, we review some concepts about active learning and its application to MIR. In Section 2 we explain our methodology. Results are shown and explained in Section 3. Finally, we discuss these results in Section 4.

1.1 Active Learning

Active Learning (AL) is aimed at maximizing the accuracy of a machine learning algorithm by means of allowing it to choose the data from which it learns (this usually leading to minimizing the size of the training set). In order to do so, the system may pose *queries* (unlabeled data instances) to be labeled by an *oracle*. AL is useful for problems where unlabeled data is easy to obtain, while labels are not [9]. We will deal with **uncertainty-based AL**, in which the learner queries instances for which it is least certain about how to label. The basic idea is that if, for example, the classification is binary, the instance that would be queried would be the one which probability of being positive is closest to 0.5 (total uncertainty). For more information about refinements of this technique we refer to [9].

Active Learning in MIR

Mandel et al. [7] demonstrated that AL techniques can be used for music retrieval with quite good results. Specifically, they classified songs according to their *style* and *mood*, being able to perform the same accuracy with half as many samples as without using AL to intelligently choose the training examples.

Wang *et al.* [8] propose a strategy for multi-samples selection for Support Vector Machine (SVM) AL. Their assumption is that, in music retrieval systems, it is necessary to present multiple samples (i.e. to make multiple queries) at each iteration. This is because the user could very likely lose patience after some time if just one sample is presented to him to label at each iteration. As we use the method they propose, we explain it in depth in Section 2.4.

2 Methodology

2.1 Mood tags and songs datasets

We used datasets from previous research ([3]) for *happy*, *sad*, *aggressive* and *relaxed* categories and we also created new datasets for *humorous*, *triumphant*,

mysterious and *sentimental* mood tags. These tags were chosen according to two main criteria: first, they improve the emotional representation given by the already existing ones (i.e. they do not have a close semantic relationship in the sense that they cover a broad and non-overlapping emotional landscape); second, they have a certain social relevance (judged by their presence as tags in lastfm[1]) that makes them interesting to study.

As we deal with binary classification, for each mood tag we actually need two collections of songs: one containing songs that *belong* to that category and another one with songs that *do not belong* to it. They contain 150 to 200 30-seconds MP3 files and we have tried to get a high coverage of genres and artists. They were created by collecting a high number of songs according to their mood annotation at lastfm and then validating them by 6 listeners, keeping just those songs which tag was agreed by at least 4 of them. For further details on this part of the work, please refer to [10].

2.2 Feature Extraction and Dataset Management

Descriptors of timbre [11], loudness, rhythm [12] and tonal characteristics [13] were computed and processed using MTG-internal libraries [14]. We compute statistics of these values (min, max, mean, variance). Then, we normalize descriptors and reduce dimensionality using Principal Component Analysis (PCA). The number of components that is kept is different depending on the size of the dataset ($\approx$ datasetsize/20). Finally, we compute the density around each point. This is one of the parameters that the AL strategy uses to measure the *informativeness* of each point [8]. This strategy is explained with more detail in Section 2.4.

2.3 Active Learning Experiments

For our experiments, we set a scenario in which we simulate the interaction with a user. We do so because our work tries to study a large set of combinations of parameters. Performing experiments with users would be time-intensive and attention-demanding, thus increasing the risk of getting wrong input. We follow (as [7]) these steps:

1. Randomly split the database into equal-sized training and test sets.
2. Select a random sample from the training set as the seed (the song for which the user is looking for songs with the same mood).
3. If we are in the first round, select $ITS - 1$ (Initial Train Size) samples plus the seed as the initial set for feedback. We choose them randomly. Otherwise, select EPI (Elements to add Per Iteration) samples according to the sample selection strategy (see details in 2.4).
4. According to the ground truth, automatically label the selected samples (simulating user relevance feedback). Add the labels to the labeled dataset and remove them from the training set.

[1] `http://www.last.fm`

5. Retrain the SVM model with the available dataset. Get precision and recall over the test dataset.
6. Repeat 1-6 100 times to avoid being biased by the selected seed and initial random-selected training set.

2.4 Active Learning Strategies under Study

Wang's Multi-Sample Selection Strategy

In this approach, introduced in [8], multiple samples are selected in a way that they are i) not just close to the boundary (most uncertain/informative samples), but also ii) representative of the underlying structure and iii) not redundant among them. To fulfill these three criteria, three values are calculated on every iteration for each point and different weights can be given to them: the **distance to the decision boundary**, the **distance diversity** and the **density** around the sample.

The first one is given by the own SVM classifier, which calculates the decision boundary ant tells the distance of each point to it. The diversity is calculated every time a new unlabeled sample x is selected as a candidate to be added to the current sample set S. It is defined as the minimum distance between samples in the current selected sample set S (the higher the diversity, the more scattered the set of samples are in space) and is calculated as

$$Diver(S+x) = \arg \min_{x_i, x_j \in \{S+x\}} D(x_i, x_j) \tag{1}$$

Where $D(x_i, x_j)$ is the distance between points x_i and x_j and can be calculated using the function $\Phi(x)$ that maps points into the transformed SVM space:

$$D(x_i, x_j) = \sqrt{(\Phi(x_i) - \Phi(x_j))^2} = \sqrt{\Phi(x_i)^2 + \Phi(x_j)^2 - 2 \cdot \Phi(x_i) * \Phi(x_j)} \tag{2}$$

Given that the kernel function $K(x, y)$ performs the dot product of two points in the transformed space, equation (2) can be rewritten as

$$D(x_i, x_j) = \sqrt{K(x_i, x_i) + K(x_j, x_j) - 2 \cdot K(x_i, x_j)} \tag{3}$$

The density around the sample, which is included to avoid choosing outliers as candidates, selects samples from the densest regions. An average distance $T(x)$ from a particular sample x to its 10 closest neighbors is computed off-line as

$$T(x) = \frac{D(x_{j1}, x) + \ldots D(x_{j10}, x)}{10}, x_{j1} \neq \ldots x_{j10} \tag{4}$$

Once these values are obtained, the selected point is the one that minimizes ($distance\_to\_boundary - diversity + density$) and the process is repeated as many times as samples are to be added at the current iteration.

Modified Wang's Multi-sample Selection Strategy

This strategy is proposed as an option to solve a drawback of uncertainty-based AL strategies which is even clearer when small training sets are used: reducing uncertainty may not always be the best choice. The idea is that the system should be quite sure about elements that fall far from the boundary, but if actually a newly retrieved tagged song of this kind is wrongly classified, the information it will bring to the system will be high as it will imply a big change on the decision boundary. Therefore, what we do is to perform exactly the same strategy just explained for half of the samples to be added, while the other half is actually selected from those furthest from the boundary.

3 Results

The experiment explained in Section 2.3 was performed for different sets of parameters for all the datasets. Although here we present some of the most relevant ones, please refer to [10] to find the results with all the possible considered combinations of *initial training size*, *elements per iteration*, combinations of weights for *distance to boundary*, *diversity* and *density* values and strategies (AL or random).

ANalysis Of VAriance (ANOVA) was used in order to test the influence of each of the parameters on one of the performance measures (F-measure). ANOVA looks for significant differences (i.e., unlikely to be found by chance) between means of different experimental conditions by comparing variances. The null hypothesis in the test is that the means of assumed normally distributed populations, all having the same standard deviation, are equal. In our specific case, each distribution corresponds to a certain configuration of values of one (several) parameter(s). If the null hypothesis is rejected, the value(s) of that (those) parameter(s) is considered to influence the results of F-measure.

ANOVA test were performed at different moments of the experiments for all the iterations. These tests showed that none of the parameters except the *Initial Training Size* had influence on the results for the first iteration, which is exactly what we could expect. Also, results show that the interactions between the method or combination of weights with *Initial Training Size* and *Elements Per Iteration* create significant differences in the F-measure mean. Another interesting observation is that using AL creates significant changes in the mean F-measure already by the second iteration.

Results in Table 1a confirm that there is an influence of METHOD (a variable coding random or each of the AL methods) on the results of the F-measure already in the second iteration, F (2, 115363 = 361.452), p = 000. As shown in Table 1b, the same applies for WEIGHTS (each value of WEIGHTS corresponds to a combination of the 3 weight values): it has a significant influence on the F-measure for the second iteration, F (3, 115354) = 195.085, p = 0.000. These results tell us that that changing the parameters and their combinations has an influence on the results that is not by chance.

Table 1a: Results of ANOVA test on the influence and interactions among *Initial Training Size(ITS)*, *Elements Per Iteration (EPI)* and *METHOD* on the F-measure in the first iteration.

Source	Degrees of freedom	F	p
METHOD	2	361.452	0.000
ITS	2	3959.617	0.000
EPI	2	993.995	0.000
METHOD*ITS	4	172.726	0.000
METHOD*EPI	4	60.673	0.000
ITS*EPI	4	185.998	0.000
METHOD*ITS*EPI	8	84.022	0.000
Error	115363		
Total	115390		

Table 1b: Results of ANOVA test on the influence and interactions among *Initial Training Size (ITS)*, *Elements Per Iteration (EPI)* and *WEIGHTS* on the F-measure in the second iteration. This table shows the influence of *WEIGHTS* on Wang's multi-sample selection strategy, therefore *METHOD* does not appear.

Source	Degrees of freedom	F	p
WEIGHTS	3	195.085	0.000
ITS	2	4498.762	0.000
EPI	2	1312.364	0.000
WEIGHTS*ITS	33.215	172.726	0.000
WEIGHTS*EPI	6	3.087	0.000
ITS*EPI	4	233.452	0.000
WEIGHTS*ITS*EPI	12	3.709	0.000
Error	115354		
Total	115390		

Figures 1a and 1b show average precision and recall values after 100 runs for both AL strategies and random sample selection for an initial training size of 5 samples, adding 8 elements per iteration and giving the same weight to the three parameters for the AL strategies. Precision and recall values are higher using Wang's multi-sample selection strategy (up to $\approx$ 4 percent point units higher precision than random sample selection at second iteration and $\approx$ 7 percent point units at third and fourth).

In Figures 1c and 1d each line corresponds to a different combination of weights for *distance to the boundary*, *diversity* and *density* using Wang's multi-sample selection strategy. The results show that the differences are not big, though the case in which *diversity* is given a higher weight is the one with the best performance (very close to the case in which all elements are given the same weight). The other two cases (higher weight for *distance to the boundary* or *density*) perform worse $\approx$ 5 percent point units lower precision in third iteration).

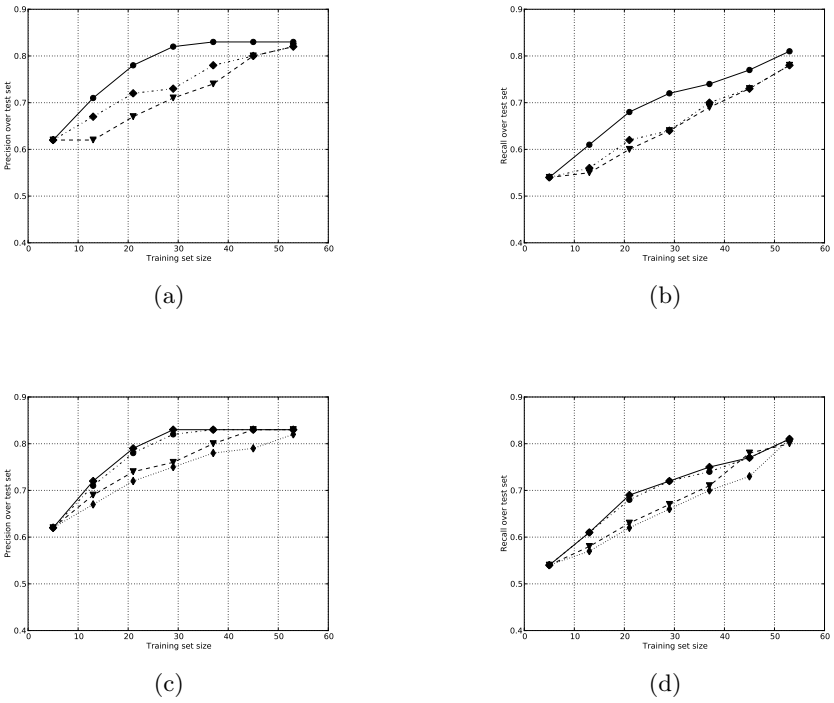

Fig. 1: (a) and (b) show precision and recall values during 7 rounds for different sample selection strategies.Wang's multi-sample selection strategy (•) converges faster faster to best performance than random sample selection (⋄) and modified Wang's strategy (▽). (c) and (d) show precision and recall values during 7 rounds for different weight combinations on Wang's multi-sample selection strategy. Best performance is achieved giving the same weight to the three parameters (•) or giving more weight to *diversity* (big ⋄). Results are worse for cases in which more weight is given to *distance to the boundary* (▽) or *density* (small ⋄).

4 Discussion and Future Work

Results of our experiments confirm those by Mandel [7] or Wang [8], in the sense that AL can help achieving a given performance on mood classification using less training instances than random sample selection. As shown in Fig. 1, the same precision can be achieved by means of AL with half instances than typical batch learning experiments. We also explored the influence of different parameters and determined that distance diversity plays a critical role for achieving good results. This is the parameter responsible for taking non-redundant samples, so one of our conclusions is that, in the presented scenario, it is more important to ensure learning about the whole distribution of points in the space rather than stressing other aspects. For example, by giving more weight to the distance to

the boundary, we may be querying for points that are too close to each other in every iteration, thus not learning about the whole dataset.

It may be interesting to explore different AL techniques. A comprehensive state-of-the-art review on AL can be found in [9], and we also present a brief review on these techniques in [10]. For example, Expected Error Reduction or Expected Variance Reduction AL techniques guarantee an improvement on the results, although they have a much higher computational cost.

Acknowledgments This work has been partially supported by the projects Classical Planet: TSI-070100- 2009-407 (MITYC) and DRIMS: TIN2009-14247-C02-01 (MICINN). We want to thank Nicolas Wack and Hendrik Purwins for their very valuable feedback.

References

1. Juslin, P.N., Sloboda, J.A.: Music and emotion: Theory and research. Oxford University Press Oxford, England (2001).
2. Ekman, P.: An argument for basic emotions. Cognition and Emotion. 6, 3, 169-200 (1992).
3. Laurier, C.: Automatic Classification of Musical Mood by Content Based Analysis. Universitat Pompeu Fabra, Barcelona (2011).
4. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP 08. 1070 (2008).
5. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research. 45-66 (2001).
6. Chang, E.Y. et al.: Support Vector Machine Concept-Dependent Active Learning for Image Retrieval. IEEE Transactions on Multimedia. 1-35 (2005).
7. Mandel, M.I. et al.: Support vector machine active learning for music retrieval. Multimedia Systems. 12, 1, 3-13 (2006).
8. Wang, T.-J. et al.: Music retrieval based on a multi-samples selection strategy for support vector machine active learning. Proceedings of the 2009 ACM symposium on Applied Computing - SAC 09. 1750 (2009).
9. Settles, B.: Active learning literature survey. SciencesNew York. 15, 2, (2010).
10. Sarasúa, A.: Active Learning for User-Tailored Refined Music Mood Detection. Universitat Pompeu Fabra, Barcelona (2011).
11. Guaus, E.: Audio content processing for automatic music genre classification: descriptors, databases, and classifiers. Universitat Pompeu Fabra, Barcelona (2009).
12. Gouyon, F.: A Computation approach to rhythm description. Universitat Pompeu Fabra, Barcelona (2005).
13. Gómez, E.: Tonal description of music audio signals. Universitat Pompeu Fabra, Barcelona (2006).
14. Essentia & Gaia: audio analysis and music matching C++ libraries developed by the MTG (Resp.: Nicolas Wack), `http://mtg.upf.edu/technologies/essentia`