
Music Preference Learning with Partial Information

Yvonne Moh, Peter Orbanz and Joachim M. Buhmann
Institute of Computational Science
ETH Zurich
{tmoh,porbanz,jbuhmann}@inf.ethz.ch

Abstract

We consider the problem of online learning in a changing environment under sparse user feedback. Specifically, we address the classification of music types according to a user’s preferences for a hearing aid application. The classifier, operating under limited computational resources, must be capable of adjusting to types of data not represented in the training set, and to changing user demands. The user provides feedback only occasionally, prompting the classifier to change its state. We propose an online learning algorithm capable of incorporating information from unlabeled data by a semi-supervised strategy, and demonstrate that the use of unlabeled examples significantly improves classification performance if the ratio of labeled points is small.

1 Introduction

Sound classification in changing acoustic environments in the context of a hearing instrument application poses a challenging problem that requires sophisticated online learning strategies. The long-term goal is to design “smart” hearing devices, which incorporate an adaptive environment sensitive controller. The controller changes the program settings according to the current acoustic field and user preferences. The application of machine learning algorithms to classify sounds for hearing aids [1] has been explored, but such classifiers are primarily non-adaptive. Emerging hardware technology allows the incorporation of user feedback. We consider the sub-task of music classification according to user preference.

Our focus here is on online adaptation of classifiers. We propose a classification algorithm based on additive expert ensembles [6] to tackle this problem. To cope with the sparse feedback problem, we show how the online learning algorithm can be combined with a label propagation algorithm for semi-supervised learning [3]. Experimental results show that the proposed classifier meets the requirements: It can adjust to both new music and changes in preference. Moreover, incorporating unlabeled data by label propagation significantly improves prediction performance when labels are sparse.

2 Background

Online Learning. Most supervised learning algorithms operate under a batch assumption: A complete, static set of training data is available prior to prediction. In online learning [2] data points are available one at a time, with each observation serving first as test, and then as training point. For a new data value, a prediction is made. After prediction, a label is obtained, and the observation is included in the training set. These methods only assume that the complete data sequence is generated by the same instance of the generative process – if the process is restarted, the classifier has to be trained anew. The data is not required to be i.i.d. Online learning algorithms are expected to perform worse than a well-chosen batch learner, but they are capable of dealing with both incrementally available data and data distributions that change over time.

Semi-supervised learning. In semi-supervised learning [3], the system is presented with both labeled data X_L , and unlabeled data X_U . Unlabeled data is commonly exploited in either of two ways: Directly, e.g. by nonparametric density estimates used for risk estimation, or indirectly, by transferring labels from labeled to unlabeled data. Both approaches are based on the notion that points sufficiently “close” to each other are likely to belong to the same class, which implies regularity assumptions on the class distributions: One is that the individual class densities are sufficiently smooth. The other is that classes are well-separated, that is, the density in overlap regions is small (and hence has small risk contribution).

3 Online Learning with random partial information

The learning problem described in the introduction is formalized as follows: We start with a baseline classifier. New data values x_t (sound features) are provided sequentially, some of which are labeled by the user as $y_t \in \{-1, +1\}$. The feedback label y_t is assumed to be available between observations x_t and x_{t+1} . If no feedback is provided, then $y_t = 0$. Changes in the input data distribution may occur, representing two cases:

- New concept: Data with a distribution not previously used in training is introduced.
- Concept change: Labels are contradictory to previous ones.

The online aspect of the learning problem is addressed by means of an additive expert ensemble [6]. The overall classifier is a weighted ensemble of up to K_{\max} experts (component classifiers), denoted $\eta_{t,k}$ for time step t and component k . The experts are combined linearly with non-negative weights. Given a new, labeled observation (x_{t+1}, y_{t+1}) , the algorithm adjusts the classifier weights according to current error rates of the experts. Good experts receive increased weights, whilst poor experts are penalized or discarded.

As the application scenario requires a bounded memory footprint, previously observed data cannot be stored indefinitely. We therefore window the learning algorithm, that is, updates in each round are performed on a moving window of constant size. Knowledge obtained from observations in previous rounds is stored only implicitly in the state of the classifier, until new, contradictory information votes against it.

Online learning algorithms adapt the classifier after each sample, and feedback is provided to change the state of the classifier. In our setting, we adopt a passive update scheme: If the performance of the classifier is perfect on the current window,

no learning updates are made. Furthermore, the learning algorithm is triggered only if feedback is obtained for the current data point x_t . In this case, both labeled and unlabeled observations in the current window up to x_t are used to change the classifier.

Since the window size limits the amount of data available at once, direct density estimation is not applicable. Hence, we employ a graph-based approach for label transfer. The particular method used here is learning with local and global consistency (LLGC) [5]. The data set is represented as a graph, with nodes encoding the observations. Edges encode pairwise similarities (exponential of the negative euclidean distance), collected in the adjacency matrix W of the graph. LLGC spreads label information from labeled to unlabeled points by a discrete diffusion process along the graph edges. The diffusion operator in Euclidean space is discretized according to the graph’s notion of distance by the normalized graph Laplacian L . The Laplacian is computed from the graph’s affinity matrix W and diagonal degree matrix $D_{ii} := \sum_j W_{ij}$, and is defined as $L := D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.

The algorithm that we now introduce has two phases: A prediction step and a learning step. For each sample x_t , the algorithm executes a prediction step, upon which, it may or may not obtain a label from the user. If feedback is obtained, the algorithm executes a learning step. It takes four scalar input parameters: $\alpha \in [0, 1)$ controls how rapidly label information is propagated. For the learning step, $\beta \in [0, 1]$ and $\gamma \in \mathbb{R}_+$ control the decrease of expert weights and the weights of new experts, respectively. Furthermore, τ specifies the window size.

Prediction step for x_t :

1. Get expert predictions $\eta_{t,1}, \dots, \eta_{t,N_t} \in \{-1, +1\}$.
2. Output prediction: $\hat{y}_t = \text{sign}(\sum_{i=1}^{N_t} w_{t,i}\eta_{t,i})$

Learning step (only if $y_t \neq 0$): The algorithm first propagates labels to unlabeled points, and then updates the classifier ensemble. The graph Laplacian L_t is updated for current window index t , for which the corresponding labels are $Y_t = [y_{t-\tau+1}, \dots, y_t]'$.

1. Propagation:

- (a) Initialize estimate vector as $\hat{Y}_t^{(0)} = Y_t$
- (b) Iterate $\hat{Y}_t^{(j+1)} = \alpha L_t \hat{Y}_t^{(j)} + (1 - \alpha)\hat{Y}_t^{(0)}$
- (c) Assign each x_i the label given by $\text{sign}(\hat{y}_i^{\text{final}})$

2. Learning:

- (a) Update expert weights: $w_{t+1,i} = w_{t,i}\beta^{|y_t \neq \eta_{t,i}|}$
- (b) If $\hat{y}_t \neq y_t$, then add a new expert N_{t+1} (and eliminate expert with lowest weight), where the new expert is trained on the current window of data: $w_{t+1,N_{t+1}} = \gamma \sum_{i=1}^{N_t} w_{t,i}$
- (c) Update each expert on example x_t, y_t

4 Experiments

For evaluation, we built a music database of 2000 files. The bulk of the database are “classical music”, yielding 12 subclasses: opera (4 composers), orchestral music (5 composers) and chamber music (3 types). A small set of very dissimilar music

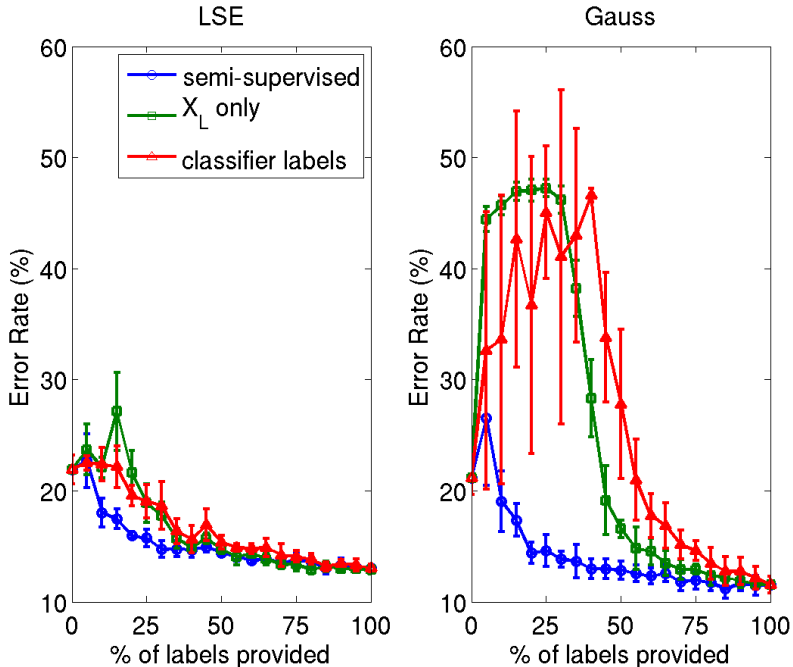


Figure 1: Cumulative Errors on learning concept changes versus ratio (percentage) of available labels, shown for LSE (left) and Gaussian (right) experts. Results are obtained by five-fold cross validation. Error rates at 0% correspond to the initial static classifier. The peak in error rates is discussed in Sec. 4.2.

(pop) is used as the 13th subclass. MFCC features were extracted [4] as well as means and variances of zero crossing, spectral center of gravity, spectral rolloff, and spectral flux over entire songs.

4.1 Classifier Setting

The additive expert is based on an ensemble of simple component classifiers. Two types of components were considered here: A least mean-squared error (LSE) classifier, and a full covariance Gaussian model (GM). The two principal differences between the two classifiers are the fact that the GM constitutes a generative model, whereas the LSE model is discriminative. The set of hyperplanes expressible in terms of LSE is contained in the GM, which can additionally model quadratic surfaces.

A baseline model is first learned on an initial set of data. During the evaluation phase, the remaining data is presented to the classifier sequentially. When no labels are provided, the classifier does not update, such that values reported for 0% of labels shows the performance of a static baseline classifier. When all labels are provided, we obtain the conventional, fully supervised online learning scenario. For both choices of experts, we compare the semi-supervised online algorithm to two other learning strategies. The three variants shown in each of the diagrams are:

1. X_U takes the label hypothesized by the LLGC (semi-supervised).
2. X_U is ignored and not used for learning (X_L only).

3. X_U takes the label hypothesized by the current classifier (classifier labels).

Results report cumulative error on the evaluation data, i.e. if \hat{y}_t denotes the label predicted by the classifier for x_t , the error is given as $\text{Err} = \frac{1}{T} \sum_{t=1}^T [\hat{y}_t \neq y_t]$.

4.2 Experimental Results

Results are presented separately for two mismatch scenarios: change of concepts, and introduction of new concepts. The experiments simulate behavior in adaptation phases.

Learning a changed concept. The baseline model is trained on two sets {opera, pop} (preferred) versus {orchestral, chamber} (not preferred). During the evaluation phase, two orchestral subclasses and pop are reassigned. Fig. 1 shows the results for both GM and LSE models. When the proportion of label data is low, using the unseen labels via LLGC yields significant improvements, consistently outperforming the other approaches. At 0% the performance of the initial baseline model is given. For very sparse labels, overfitting decreases prediction accuracy. The peaks in error rates are due to a trade-off effect between the information provided by the labels and the number of learning steps they trigger. The decrease in performance is most notable for Gaussian experts, which are more prone to overfitting. While the semi-supervised approach requires about 10% of labels to start improving upon the baseline method, between 20% (LSE) and 40% (Gaussian) are required if the unlabeled data is neglected. At large label ratios, the Gaussian model slightly outperforms the LSE. The semi-supervised version of the model requires only about 40% of labels to reach optimal performance.

Learning a new concept. The second type of classifier adaptation is adjustment to previously unobserved music. Of particular interest is the classifiers behavior when the new concept substantially differs from those already incorporated in the baseline model. In this experiment, the baseline model is trained on opera, and orchestral/chamber music. During the evaluation phase, orchestral Mahler and piano are assigned to the opera class, whereas pop music and orchestral Shostakovitch to the other class. Fig. 2 shows the results for the LSE classifier. As in the concept change case, the amount of feedback required by online learning with LLGC is substantially smaller to achieve performance equivalent to that of the fully supervised method.

5 Discussions and Conclusions

We have presented an algorithm for music preference learning that combines an on-line approach to learning with a partial label scenario. The classifier tracks changes in class distributions and adapts to data that differs from previous observations. The integration of unlabeled data in the learning process improves performance despite low levels of user feedback. A window-based design copes with the limits of computational costs and memory requirements.

Our algorithm design does not make any assumptions about the base learner. In principle, any classification algorithm may be used, e.g., the proposed algorithm may be extended by kernelization of the LSE base learner, which generalizes decision boundaries beyond the linear case. We expect our method to be a step toward adaptivity in the control of “smart” hearing devices.

Acknowledgment. The authors gratefully acknowledge financial support by a

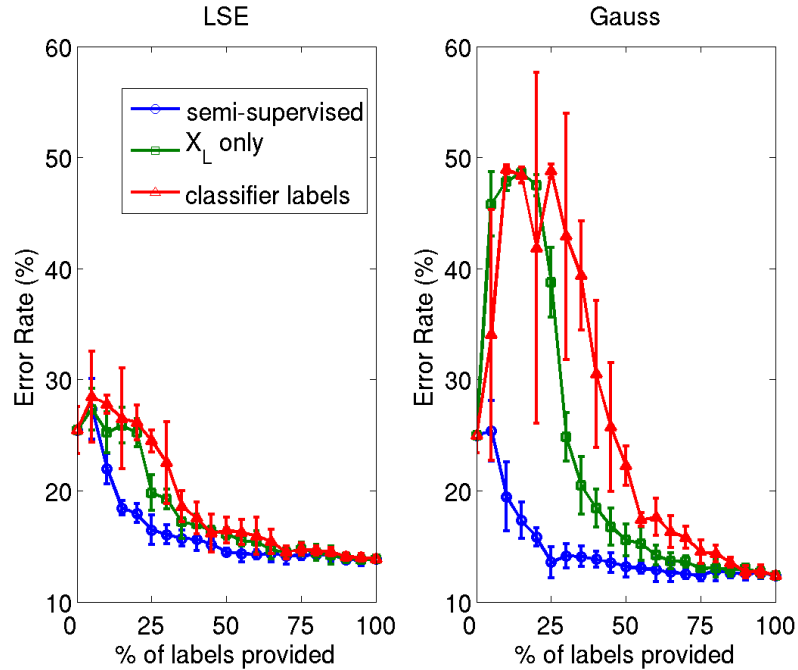


Figure 2: Cumulative Errors on learning new concepts

KTI grant. This work is under review at ICASSP'08.

References

- [1] M. Büchler, S. Allegro, S. Launer, and N. Dillier. Sound classification in hearing aids inspired by auditory scene analysis. *Journal of Applied Signal Processing*, 18:2991–3002, 2005.
- [2] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning and games*. Cambridge University Press, 2006.
- [3] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.
- [4] D. P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. online web resource.
- [5] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, volume 16, pages 321–328. MIT Press, 2004.
- [6] J. Z. Zolter and M. A. Maloof. Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22nd Intl Conference on Machine Learning*, 2005.