Sādhanā (2020)45:73
https://doi.org/10.1007/s12046-020-1313-8

Deep Gaussian processes for music mood estimation and retrieval with locally aggregated acoustic Fisher vector

SANTOSH CHAPANERI* and DEEPAK JAYASWAL

Department of Electronics and Telecommunication Engineering, St. Francis Institute of Technology, University of Mumbai, Mumbai, India
e-mail: santoshchapaneri@sfit.ac.in; djjayaswal@sfit.ac.in

MS received 14 October 2019; revised 11 January 2020; accepted 13 January 2020

Abstract. Due to the subjective nature of music mood, it is challenging to computationally model the affective content of the music. In this work, we propose novel features known as locally aggregated acoustic Fisher vectors based on the Fisher kernel paradigm. To preserve the temporal context, onset-detected variable-length segments of the audio songs are obtained, for which a variational Bayesian approach is used to learn the universal background Gaussian mixture model (GMM) representation of the standard acoustic features. The local Fisher vectors obtained with the soft assignment of GMM are aggregated to obtain a better performance relative to the global Fisher vector. A deep Gaussian process (DGP) regression model inspired by the deep learning architectures is proposed to learn the mapping between the proposed Fisher vector features and the mood dimensions of valence and arousal. Since the exact inference on DGP is intractable, the pseudo-data approximation is used to reduce the training complexity and the Monte Carlo sampling technique is used to solve the intractability problem during training. A detailed derivation of a 3-layer DGP is presented that can be easily generalized to an L-layer DGP. The proposed work is evaluated on the PMEmo dataset containing valence and arousal annotations of Western popular music and achieves an improvement in R^2 of 25% for arousal and 52% for valence for music mood estimation and an improvement in the Gamma statistic of 68% for music mood retrieval relative to the baseline single-layer Gaussian process.

Keywords. Deep Gaussian process; Fisher vector; music mood; regression.

1. Introduction

Music mood estimation and retrieval research has attracted increasing attention in recent years since music mood is one of the most frequently used queries to search for related music in music libraries. For example, music applications such as Spotify, Wynk, Musicovery, YouTube Music, etc. specify the option of retrieving music relevant to a specific mood. However, due to the subjective nature of mood, the so-called *semantic gap* exists to mathematically model music mood as per the perception of varied human subjects. Mood (also referred to as emotion in literature, but often used interchangeably) can be modelled using the categorical approach (sad, happy, surprise, etc.) or the dimensional approach (valence and arousal (VA) dimensions). Using the dimensional approach is beneficial since the categorical approach cannot exhaustively cover the wide range of moods/emotions [1]. In the dimensional approach, the third dimension of dominance was also studied in [1], however, it is usually discarded due to its correlation with the VA dimensions.

To address the subjectivity issue, several datasets have been developed for dimensional music mood estimation via crowdsourced annotations. However, in most datasets, the inter-annotator agreement is not high and thus estimating the ground truth can be a challenging issue. The second issue is designing an appropriate feature representation to model the music mood. Conventional acoustic features are typically computed at the frame-level for each song, yielding a high feature dimensionality. Fisher vectors have been studied and applied successfully in the area of image categorization but not yet explored for the study of music mood estimation and retrieval. The third issue is the design of an appropriate regressor model to learn the mapping of features to VA mood dimensions. Models that can automatically determine optimal hyper-parameters without over-fitting and providing uncertainty estimates are preferred over models such as Support Vector Regression (SVR), where grid search is required to learn the hyper-parameters. Gaussian processes (GPs) offer these advantages by offering error bars on the prediction but they are

*For correspondence
Published online: 19 March 2020

not scalable and cannot learn non-stationary functions without the use of a non-stationary kernel, which can make the inference computationally intractable.

Contributions: The contributions of this work are threefold.

(a) We use the PMEmo dataset [2] having a high inter-annotator agreement; this resolves the subjectivity issue due to the availability of ground truth of mood annotations. Onset detection is performed as a pre-processing step for each audio song to determine variable-length segments for considering the temporal context. Standard acoustic features are computed for each such segment and a universal background model (UBM) is learned from these features using the Gaussian mixture model (GMM). For learning this background model, variational Bayesian inference with conjugate priors on Gaussian parameters is used to automatically determine the optimal number of clusters required to explain the data.
(b) We propose novel features termed as locally aggregated acoustic Fisher vector (LAFV) that describe the acoustic features by their deviation from the GMM. The LAFV features are analysed for interesting properties that effectively discriminate the audio songs.
(c) A deep Gaussian Process (DGP) regression model is proposed to learn the mapping of LAFV features to VA mood dimensions. Pseudo-data approximation and Monte Carlo sampling techniques are used to overcome the intractability of DGP model learning and the detailed working of a 3-layer DGP is presented.

The rest of this paper is organized as follows. Section 2 discusses the related work in literature and Sect. 3 explains the proposed methodology. Section 4 presents the experimental validation, followed by conclusions in Sect. 5.

2. Related work

Music mood estimation is an active research topic [3–7] since modelling the valence dimension is a difficult task. The perceived mood is studied rather than the felt mood as this alleviates the burden of several physiological factors that come into play for a layman listener [8]. Several benchmark datasets such as AMG [8], DEAM [9] and MoodSwings [10] have been created that help in modelling the subjectivity issue at song level (static mood) as well as segment level (dynamic mood) but they lack the golden truth as annotations are mostly obtained via crowd-sourcing, which raises the question of reliability of annotators and their agreement. A typical approach is to estimate the ground truth of each music clip by averaging multiple annotations given to it. However, this assumes that all annotators are equally reliable, which may not be a valid assumption in practice [5], since it often ignores the annotator errors (e.g. low-attention) and outliers (e.g. adversarial behaviour) that can have a significant impact on the consensus. To solve this issue, the PMEmo dataset [2] is used in this work due to its superior quality of annotations.

Standard acoustic features are widely used to represent the audio data; however, the dimensionality of such features is quite high as they are usually extracted at the frame-level. In [4], novel rhythmic and melodic features are proposed using musical concepts for music emotion classification. Kernel density estimation is used in [5] to represent the VA space as a probability density function (PDF) and an audio space dictionary is learned to map the acoustic features. Histogram density modelling (HDM) approach is used in [7] to represent the VA space as a heatmap and the block-level GMM posterior probability feature vectors are mapped to the latent histograms. Feature selection techniques are used in [11] to select the appropriate features using shrinkage methods for the emotion classification task. Novel features based on acoustic GMM using Bayesian inference are proposed in [12] by automatically determining the number of latent audio topics (mixtures) without risking over-fitting. In [13], the frame-level features are stacked by computing their statistics over multiple windows.

The concept of the Fisher kernel was first introduced in [14] and further studied in [15] for the application of image classification and retrieval. It has been also successfully applied in the areas of web audio classification [16], speaker verification [17], image aesthetic quality assessment [18], etc. Fisher kernels retain the advantages of the generative model in a discriminative framework [14]. The central idea of the Fisher kernel is to characterize a signal with a gradient vector of the log-likelihood PDF that models the signal's generation process. For the acoustic features extracted from the audio signals the distribution can be modelled using UBM-GMM, resulting in a generative model. This can be further extended in a discriminative setting by computing the gradient of UBM-GMM's log-likelihood with respect to the model parameters, resulting in a fixed-length representation vector known as the Fisher vector. The Fisher vector has been shown to outperform the bag-of-visual-words (BoV) approach [19] with l_2 and power normalization in [20].

While conventional regression techniques such as Multiple Linear Regression (MLR), SVR, etc. can be used to predict the VA values of songs, they do not have a probabilistic interpretation since the uncertainty measure of output predictions is missing. Though GPs can model a rich class of functions with a few hyper-parameters, their training complexity is prohibitive for large-scale applications. Recently, deep learning architectures based on Convolutional Neural Network [21] and Recurrent Neural Network [22] were proposed for music emotion classification task.

Inspired by the recent surge and success of deep learning architectures, a DGP model is proposed in [23] that acts as

a hierarchical composition of GPs to overcome the limitations of single-layer GP, due to which it can learn more complex non-stationary functions. DGP can be viewed as a multi-layer fully connected neural network with multiple infinitely-wide hidden layers. Unfortunately, the exact inference in DGP is no longer analytically feasible and variational approaches are proposed to solve this problem: mean-field variational inference in [23], approximate expectation propagation inference in [24], etc. The drawback of these variational methods is that they rely on integral approximations that depend on the specific kernel being used and thus they cannot be extended easily to arbitrary kernels such as the Matérn kernel. In [25], the Monte Carlo sampling technique is proposed, which is intuitive to understand and does not depend on the choice of a specific kernel for GP. DGP is used in [26] for the music emotion classification task of nine emotion classes using the mean-field variational inference approach of [23].

To the best of our knowledge, the Fisher vector is yet to be applied to the field of music mood estimation and retrieval. In this work, we propose novel features termed as LAFV to effectively discriminate the audio songs. For extracting the LAFV features, the generative model of Bayesian Acoustic Gaussian Mixture Model (BAGMM) is learned using the variational inference approach to avoid the drawbacks of the standard EM (Expectation-Maximization) algorithm and to also automatically learn the optimal number of Gaussian components to explain the audio data. Further, there is no existing related work on music mood dimensional estimation using DGPs. In this work, we propose an L-layer DGP regression model for effective music mood estimation and retrieval. The proposed work is shown to outperform the existing state-of-the-art techniques for feature extraction as well as regression modelling.

3. Proposed work

The proposed LAFV features are given as the input to a DGP regression model. Figure 1 shows the steps for obtaining the LAFV features for a given audio song. Using onset detection, variable-length segments are determined (Sect. 3.1) for which the acoustic features are extracted and a UBM using GMM (UBM-GMM) is fitted to the data (Sect. 3.2). Using the UBM-GMM model the LAFV features are computed (Sect. 3.3), which are used for learning the DGP regression model (Sect. 3.4).

3.1 *PMEmo dataset and onset detection*

The PMEmo dataset [2] contains chorus sections of 794 music clips annotated by 457 subjects on the two-dimensional VA axis in the range [0, 1]. The chorus sections were manually selected by participants majoring in music studies and the songs were carefully chosen from *Billboard Hot 100*, *iTunes Top 100* and *UK Top 40* charts covering a wide range of popular music. Out of 457 annotators, 413 had a non-music major background and a mix of English and Chinese speaking annotators were recruited to reduce the impact of cultural background. Each song was annotated by at least 10 annotators. The filtering of annotated data is done to avoid the crowd-sourcing bias and the inter-annotator agreement measured using Cronbach's α is $\alpha_{Valence} = 0.998$ and $\alpha_{Arousal} = 0.998$, indicating that the annotations are of high quality compared with other datasets. Figure 2 shows the scatter plot of the average values (per song) of VA annotations of 794 songs, along with the marginal VA distributions. It can be inferred that most songs are located in the first quadrant, implying that popular music results in high valence and high arousal perception in listeners.

For onset detection the audio signal is split into variable-length segments to preserve the audio signal rhythm continuity instead of fixed-length segments, where the rhythm continuity may be lost. This can be done by determining the onset detection function (ODF) of the input audio signal to consider musically related events. We refer to [27], which uses a constrained linear reconstruction unsupervised method to detect the onsets using a wider temporal context. Given an input audio signal, the short-time Fourier transform (STFT) is applied to compute the magnitude spectrum with a frame length of 2048 samples and a frame rate of 200 frames/s. The magnitude spectrum is then processed by 141 triangular filter banks in the range of 30 Hz–17 kHz with an interval of 24 bands per octave. Using the logarithm mapping $\mathbf{u} \leftarrow \log(1 + \mathbf{u})$, the resulting feature vector of the nth frame is obtained as $\mathbf{u}_n \in \mathbb{R}^{141}$. Instead of comparing the spectral difference only between two successive frames, the method of [27] considers up to τ previous

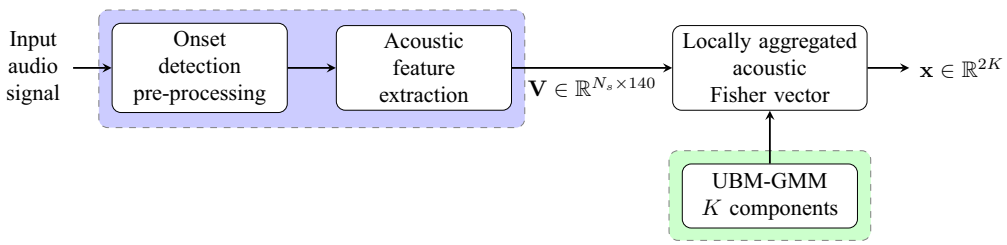


Figure 1. Block diagram for computing the LAFV features.

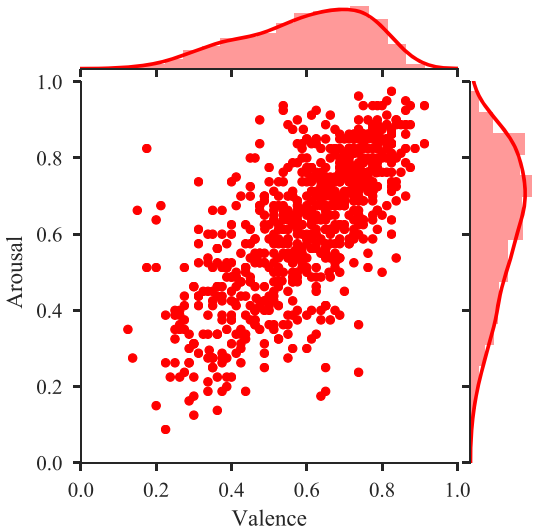


Figure 2. Scatter plot of VA annotations of the PMEmo dataset.

frames for computing the ODF via the linear reconstruction problem stated in Eq. (1). Here, $\boldsymbol{\alpha}_n \in \mathbb{R}^{\tau}$ represents the weighting coefficients of the τ previous frames $\overline{\mathbf{U}}_n \in \mathbb{R}^{141\times\tau}$ for reconstructing the current frame and $a(\cdot)$ represents the penalty regularization term parameterized by λ:

$$\{\mathbf{r}_n^*, \boldsymbol{\alpha}_n^*\} = \underset{\mathbf{r}_n,\boldsymbol{\alpha}_n}{\operatorname{argmin}} ||\mathbf{r}_n||_2^2 + \lambda a(\boldsymbol{\alpha}_n), \quad \mathbf{r}_n = \overline{\mathbf{u}}_n - \overline{\mathbf{U}}_n\boldsymbol{\alpha}_n. \tag{1}$$

The input feature vectors are l_2 normalized, resulting in $\overline{\mathbf{u}}_n = \mathbf{u}_n/||\mathbf{u}_n||_2$, and we have $\overline{\mathbf{U}}_n = [\overline{\mathbf{u}}_{n-\mu}, \overline{\mathbf{u}}_{n-\mu-1}, \ldots, \overline{\mathbf{u}}_{n-\mu-(\tau-1)}]$, where $\mu > 0$ indicates the temporal offset. The reconstruction length is denoted by the parameter $\tau = 5$ and the reconstruction error given by $||\mathbf{r}_n||_2 = ||\overline{\mathbf{u}}_n - \overline{\mathbf{U}}_n\boldsymbol{\alpha}_n||_2$ is an indication of the audio onset events. The non-negative least squares (NNLS) approach is proposed in [27] to solve Eq. (1) with $\lambda \to \infty$ and $a(\boldsymbol{\alpha}_n) = \sum_{i=1}^{\tau}(\alpha_{ni})_-$, where α_{ni} refers to the ith elements of $\boldsymbol{\alpha}_n$ and the function $(x)_-$ returns 1 if $x<0$ and 0 otherwise, i.e. $a(\boldsymbol{\alpha}_n) = 0$ if and only if all the elements in $\boldsymbol{\alpha}_n$ are non-negative.

Further, rectification is applied as given by Eq. (2) to calculate the ODF where $\odot$ is the element-wise product and $(x)_+ = \max(x, 0)$ is the element-wise rectification operation. For calculating the reconstruction error, the frequency bands with increased energy from $n-\mu$ to n in the original un-normalized feature vectors are considered and the rectified reconstruction error is multiplied by the l_2 norm of the original feature vector $\mathbf{u}_n$. Using the peak-picking criteria mentioned in [27], the onsets are detected at various time instants in the audio song.

$$\text{ODF}_{LR}(n) = ||\mathbf{r}_n \odot (\mathbf{u}_n - \mathbf{u}_{n-\mu})_+||_2||\mathbf{u}_n||_2. \tag{2}$$

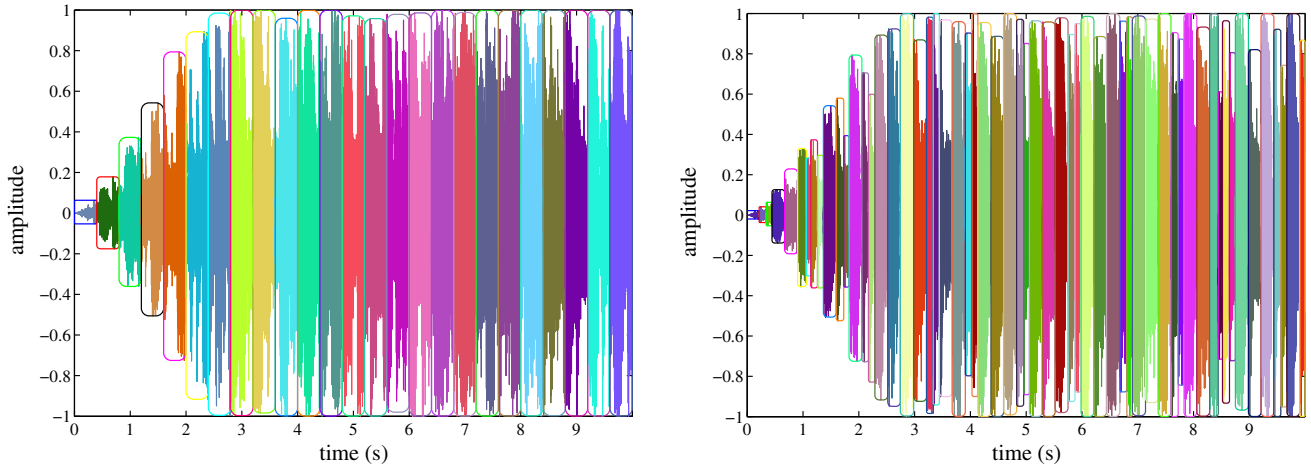
With the onset information, the frames belonging to certain onset duration are grouped, e.g. 1.2–1.4 s constitute one segment. An illustration for a specific song of the PMEmo dataset is shown in figure 3, where figure 3(a) shows the fixed-length segments and figure 3(b) shows the variable-length segments obtained via onset-detected events. To capture the temporal characteristics across frames with the fixed-length segments approach, each segment comprises 16 consecutive frames with an overlap of 12 frames [13]. However, the temporal continuity of the audio gets lost due to abrupt transitions between such fixed-length segments. On the contrary, the temporal continuity is maintained with the variable-length segments approach since a new segment reflects the beginning of a new onset event.

3.2 *Feature representation*

The PMEmo dataset consists of 6373-dimensional extracted feature set per song in accordance with INTERSPEECH ComParE (Computational Paralinguistics Campaign) [28]; these features are frame-wise acoustic low-level descriptors (LLD) including MFCC, energy, logarithmic harmonic-to-noise ratio, spectral flux, etc. and are extracted using the open-source tool *openSMILE* [29]. The training data $\mathcal{D}$ comprise $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{Y} \in \mathbb{R}^{N\times 2}$, where $N = 794$ songs and $D = 6373$ features.

To reduce the computational burden during regression training due to the high value of D, the Bayesian Acoustic Gaussian Mixture Model (BAGMM) features proposed in [12] are computed using the variational Bayesian inference framework. For each frame, standard acoustic features across four categories are computed using the *MIRToolBox* [30]: dynamics (root mean square energy), spectral (centroid, spread, skewness, kurtosis, entropy, flatness, 85% roll-off, 95% roll-off, brightness, roughness, irregularity), timbral (zero-crossing rate, flux, 13-dimensional MFCCs, delta MFCCs, delta-delta MFCCs) and tonal (key clarity, musical mode, harmonic change likelihood, 12-bin chroma vector, chroma peak, chroma centroid), resulting in a 70-dimensional feature vector per frame. Each feature dimension is normalized to zero mean and unit standard deviation. Segment-level features are computed to capture the temporal characteristics across frames belonging to a particular onset-detected segment. The segment-level feature vector $\mathbf{v}_t \in \mathbb{R}^{140}$ consists of the mean and standard deviation of the frame-based feature vectors. This results in the audio feature matrix $\mathbf{V} \in \mathbb{R}^{N_s\times 140}$ where N_s is the number of segments of the given audio song.

For an effective prototypical representation, we learn the UBM with parameters $\boldsymbol{\Theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}_{k=1}^{K}$ denoting the weight, mean and (diagonal) covariance of the kth latent audio topic (or mixture) with a mixture of Gaussians given by Eq. (3). Using the EM algorithm [31] the UBM model is trained with randomly selected 25% segment-level feature vectors $\mathbf{v}_t$ across the entire dataset (spanning the whole range of VA space), resulting in 246, 000 vectors. The



(**a**) Fixed-length segments

(**b**) Onset-detected variable-length segments.

Figure 3. Illustration of onset detection.

responsibility that the component k takes for explaining the observation $\mathbf{v}_t$ is given by Eq. (4):

$$p(\mathbf{v}_t|\boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{v}_t|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) \tag{3}$$

$$\gamma_{tk} = \frac{\pi_k \mathcal{N}(\mathbf{v}_t|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{v}_t|\boldsymbol{\mu}_l, \boldsymbol{\sigma}_l)} \tag{4}$$

To determine the optimal number of mixtures, we resort to variational Bayesian inference framework [31] where all parameters of GMM are assigned conjugate priors: the weight is assigned Dirichlet prior given by Eq. (5) with hyper-parameter α_0 and $C(\alpha_0)$ as the normalizing constant; the mean and covariance ($\boldsymbol{\Lambda}^{-1}$) are assigned Gaussian-Wishart prior given by Eq. (6) with hyper-parameters $\boldsymbol{m}_0, \beta_0, \boldsymbol{W}_0$ and v_0. Due to the intractability of true posterior distribution, variational Bayesian inference is used for its approximation by minimizing the Kullback–Leibler (KL) divergence between the true and approximate posteriors. The algorithm is initialized with $\alpha_0 = 0.001$ (so that the posterior will be influenced primarily by the data), $\mathbf{m}_0$ as the k-mean centroid of training data (to speed up the convergence), $\beta_0 = 1$, $v_0 = D + 1$ and $\mathbf{W}_0 = 10\mathbf{I}$ to avoid the mixtures getting trapped in local maximum, i.e. no single Gaussian can stay in the neighborhood of a possible bad saddle point. Here, $\mathbf{I}$ is the identity matrix and D is the dimensionality of segment-level features. We refer the reader to [31] (Chapter 10) for a detailed formulation of variational posteriors and the variational EM algorithm.

$$p(\boldsymbol{\pi}) = C(\alpha_0) \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1}, \tag{5}$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\beta_0 \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_0, v_0). \tag{6}$$

The Bayesian GMM framework determines the optimal number of latent audio topics ($K = 57$ in this work) without using cross-validation, overcomes the problem of singularity [31] (that occurs in GMM) due to the positive-definite hyper-parameter $\boldsymbol{W}_0$ and prevents over-fitting of data. Instead of using the acoustic posterior probability values as the final feature vector, we propose to use Fisher vectors as a step ahead of the generative modelling of GMM. This idea is used successfully in the area of image classification, but not yet studied in the context of music mood.

3.3 *LAFV*

The gradient of the log-likelihood with respect to $\boldsymbol{\Theta}$ given by Eq. (7), also known as the empirical Fisher score, describes the sensitivity of the model to changes in the parameters. The expected value of this Fisher score is zero as shown in Appendix equation (32).

$$g_{\boldsymbol{\Theta}}^{\mathbf{v}} = \frac{1}{N_s} \sum_{t=1}^{N_s} \nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}_t|\boldsymbol{\Theta}). \tag{7}$$

The uncertainty estimate of the Fisher score can be obtained using its covariance, resulting in the Fisher Information Matrix (FIM) given by Eq. (8). The FIM is equivalent to the negative expected Hessian of the model's log-likelihood as shown in Appendix Eq. (35) and thus it can serve as a measure of the curvature of the log-likelihood function. The Fisher kernel on the gradients for

two audio feature matrices $\mathbf{V}$ and $\mathbf{W}$ is then given by Eq. (9):

$$\begin{aligned}\mathbf{F}_{\boldsymbol{\Theta}} &= \mathbb{E}_{\mathbf{v}\sim\text{GMM}_{\boldsymbol{\Theta}}}\left[(g_{\boldsymbol{\Theta}}^{\mathbf{V}}-0)(g_{\boldsymbol{\Theta}}^{\mathbf{V}}-0)^{\mathsf{T}}\right] \\ &= \mathbb{E}_{\mathbf{v}\sim\text{GMM}_{\boldsymbol{\Theta}}}\left[\nabla_{\boldsymbol{\Theta}}\log p(\mathbf{v}|\boldsymbol{\Theta})\cdot\nabla_{\boldsymbol{\Theta}}\log p(\mathbf{v}|\boldsymbol{\Theta})^{\mathsf{T}}\right],\end{aligned} \tag{8}$$

$$FK(\mathbf{V},\mathbf{W}) = g_{\boldsymbol{\Theta}}^{\mathbf{V}\mathsf{T}}\mathbf{F}_{\boldsymbol{\Theta}}^{-1}g_{\boldsymbol{\Theta}}^{\mathbf{W}}. \tag{9}$$

Since the FIM $\mathbf{F}_{\boldsymbol{\Theta}}$ is semi-positive definite, its inverse has the Cholesky decomposition given by $\mathbf{F}_{\boldsymbol{\Theta}}^{-1} = \mathbf{L}_{\boldsymbol{\Theta}}^{\mathsf{T}}\mathbf{L}_{\boldsymbol{\Theta}}$. The normalized gradient obtained through whitening operation is thus given by $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathbf{V}} = \mathbf{L}_{\boldsymbol{\Theta}}g_{\boldsymbol{\Theta}}^{\mathbf{V}}$. A closed-form expression of FIM for GMM is derived in [32], using which the normalized Fisher vector with respect to the mean $\mathbf{g}_{\boldsymbol{\mu}}^{\mathbf{V}} \in \mathbb{R}^{KD}$ and standard deviation $\mathbf{g}_{\boldsymbol{\sigma}}^{\mathbf{V}} \in \mathbb{R}^{KD}$ are given by Eq. (10), where $d = 1,\ldots,D(=140)$ and $k = 1,\ldots,K(=57)$. The final Fisher vector is the concatenation of these two quantities given by Eq. (11), resulting in $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathbf{V}} \in \mathbb{R}^{2KD}$. Note that the gradients with respect to the mixing weights π_k of the GMM are generally ignored since they contribute little discriminative power to the Fisher vector [33].

$$\begin{aligned}g_{\boldsymbol{\mu}}^{\mathbf{V}}[(k-1)D+d] &= \frac{1}{N_s\sqrt{\pi_k}}\sum_{t=1}^{N_s}\gamma_{tk}\left(\frac{v_t[d]-\mu_k[d]}{\sigma_k[d]}\right) \\ g_{\boldsymbol{\sigma}}^{\mathbf{V}}[(k-1)D+d] &= \frac{1}{N_s\sqrt{2\pi_k}}\sum_{t=1}^{N_s}\gamma_{tk}\left[\left(\frac{v_t[d]-\mu_k[d]}{\sigma_k[d]}\right)^2-1\right],\end{aligned} \tag{10}$$

$$\mathbf{g}_{\boldsymbol{\Theta}}^{\mathbf{V}} = \left[\cdots\mathbf{g}_{\boldsymbol{\mu}}^{\mathbf{V}}\cdots\mathbf{g}_{\boldsymbol{\sigma}}^{\mathbf{V}}\cdots\right]^{\mathsf{T}}. \tag{11}$$

The LAFV features are extracted as presented in Algorithm 1, where the input is the audio feature matrix $\mathbf{V} \in \mathbb{R}^{N_s\times 140}$ with N_s segments along with the UBM-GMM parameters $\boldsymbol{\Theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}_{k=1}^{K}$. In lines 1–3, for every segment feature vector $\mathbf{v}_t$, the responsibilities $\boldsymbol{\gamma}_t \in \mathbb{R}^K$ are obtained using Eq. (4) and the cluster membership c_t is obtained to identify the dominant Gaussian cluster. Thus, each Gaussian cluster contains a set of relevant segment feature vectors. The Fisher vector $\mathbf{s}$ is initialized to $\mathbf{0}$ in line 4 and is computed in lines 5–8, where the set $\mathcal{S}_k$ contains all feature vectors belonging to the kth cluster and the local Fisher vector are determined using Eq. (11) by concatenating the gradients with respect to the mean and standard deviation parameters, resulting in $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_k} \in \mathbb{R}^{2KD}$. All such local Fisher vectors are aggregated to obtain the overall Fisher vector $\mathbf{s}$ in line 8. It has been shown in [33] that power normalization and l_2 normalization improve the performance of Fisher vectors; these normalizations are done in lines 9–11. The Fisher kernel, being a gradient, measures the partial derivatives with respect to the changes in each dimension of each Gaussian's parameter. Thus, a compact feature can be computed as the magnitude of the gradient with respect to each Gaussian's mean and standard deviation, which is calculated in lines 12–15. The final LAFV is obtained as $\mathbf{x} \in \mathbb{R}^{2K}$.

Algorithm 1 LAFV

Input: Audio feature matrix $\mathbf{V} \in \mathbb{R}^{N_s\times 140}$, UBM-GMM $\boldsymbol{\Theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}_{k=1}^{K}$
Output: LAFV $\mathbf{x} \in \mathbb{R}^{2K}$ of the input audio

1: **for** each segment-level feature vector $\{\mathbf{v}_t\}_{t=1}^{N_s}$ **do**
2: find the responsibilities $\{\gamma_{tk}\}_{k=1}^{K}$ using Eq. (4)
3: find the cluster membership $c_t \leftarrow \underset{k}{\text{argmax}}\ \gamma_{tk}$
4: **end for**
5: $\mathbf{s} \leftarrow \mathbf{0}$
6: **for** $k = 1,\ldots,K$ **do**
7: $\mathcal{S}_k \leftarrow \{\mathbf{v}_t | c_t == k\}$ ▷ feature vectors of k^{th} cluster
8: compute $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_k} \in \mathbb{R}^{2KD}$ using Eq. (11) ▷ local FV
9: $\mathbf{s} \leftarrow \mathbf{s} + \mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_k}$ ▷ aggregated acoustic Fisher vector
10: **end for**
11: **for** $i = 1,\ldots,2KD$ **do** ▷ power normalization
12: $s[i] \leftarrow \text{sign}(s[i])\sqrt{|s[i]|}$
13: **end for**
14: $\mathbf{s} \leftarrow \mathbf{s}/\|\mathbf{s}\|_2$ ▷ l_2 normalization
15: **for** $k = 1,\ldots,2K$ **do** ▷ compact LAFV
16: **for** $d = 1,\ldots,D$ **do**
17: $m[d] \leftarrow s[(k-1)D+d]$
18: **end for**
19: $x[k] \leftarrow \|\mathbf{m}\|_2$ ▷ l_2 norm of D-dimensional vector
20: **end for**

An illustration is shown in figure 4 to demonstrate the importance of aggregated Fisher vectors (LAFV) relative to the global Fisher vector. The top left panel shows 2000 data points sampled from three Gaussian clusters: C_1 with mean $[0.1, 0.3]^{\mathsf{T}}$ and covariance matrix $[0.03, -0.01; -0.01, 0.02]$, C_2 with mean $[0.4, 0.8]^{\mathsf{T}}$ and covariance matrix

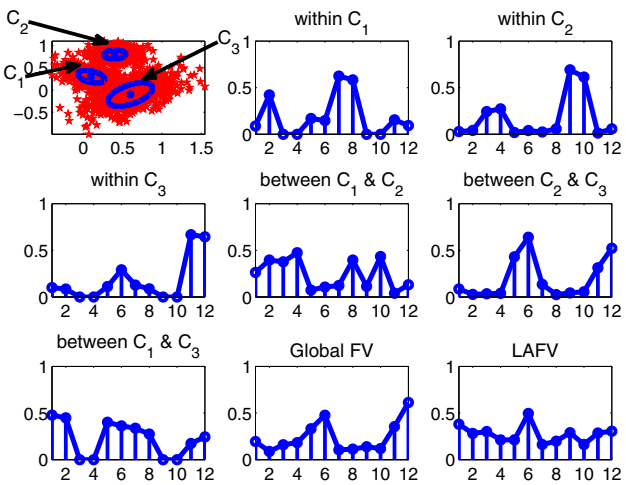


Figure 4. Fisher vectors (global FV v/s locally aggregated FV).

[0.02, 0.001; 0.001, 0.01], C_3 with mean $[0.6, -0.1]^\intercal$ and covariance matrix [0.08, 0.04; 0.04, 0.07] with the prior weights $\boldsymbol{\pi} = [0.35, 0.35, 0.3]^\intercal$. Since $D = 2$ and $K = 3$, the local Fisher vector consists of $2KD = 12$ values.

The global Fisher vector is obtained directly as $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathbf{V}}$ from Eq. (11) without considering the soft assignment of GMM. The local Fisher vectors $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_k}$ are shown for randomly selected points within clusters C_1, C_2, C_3, points between clusters C_1 and C_2, points between clusters C_2 and C_3 and points between clusters C_1 and C_3. For the LAFV representation $\mathbf{s} \in \mathbb{R}^{2KD}$, the clusters are represented by three local Fisher vectors $\mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_1}, \mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_2}, \mathbf{g}_{\boldsymbol{\Theta}}^{\mathcal{S}_3}$ and summed and normalized to obtain the aggregated Fisher vector $\mathbf{s}$. We observe that for the points within specific clusters, only those elements corresponding to the specific Gaussian are high, e.g. for cluster C_2, only the elements `3` and `4` (gradient relative to the mean) and `9` and `10` (gradient relative to the standard deviation) are high. For points between two clusters, the elements corresponding to both clusters are high, e.g. for clusters C_2 and C_3, only the elements `1`, `2`, `5`, `6` (gradient relative to the mean) and `7`, `8`, `11`, `12` (gradient relative to the standard deviation) are high. In global Fisher vector, the energy of the standard deviation gradients is higher than the energy of the mean gradients, whereas in LAFV, the energy of the mean gradients is higher than the energy of the standard deviation gradients. The energy distribution in LAFV is more loyal to the individual energy distributions in local Fisher vectors and hence to the respective feature clusters. In contrast, the global Fisher vector is biased towards the contribution of only some gradients. Thus, the LAFV feature is a better representation of the variations in the features compared with the global Fisher vector. The LAFV feature vector $\mathbf{s}$ is then compactly represented using the gradient magnitude, resulting in $\mathbf{x} \in \mathbb{R}^{2K}$.

Analysis of LAFV: Assume that the segment-level features $\mathbf{v}_t$ are *i.i.d.* and follow a distribution z that is different from the universal background distribution $\text{GMM}_{\boldsymbol{\Theta}}$. Then. according to the law of large numbers from probability theory, the gradient of log-likelihood, i.e. the empirical Fisher score can be represented as Eq. (12):

$$\begin{aligned} g_{\boldsymbol{\Theta}}^{\mathbf{V}} &= \frac{1}{N_s} \sum_{t=1}^{N_s} \nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}_t|\boldsymbol{\Theta}) \\ &\approx \nabla_{\boldsymbol{\Theta}} \mathbb{E}_{\mathbf{v}\sim z} \log p(\mathbf{v}_t|\boldsymbol{\Theta}) \\ &= \nabla_{\boldsymbol{\Theta}} \int_{\mathbf{v}} z(\mathbf{v}) \log p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v}. \end{aligned} \tag{12}$$

Suppose that we can decompose the probability distribution z into two parts as given by Eq. (13): an universal audio-independent part that follows the background model $\text{GMM}_{\boldsymbol{\Theta}}$ and an audio-specific distribution q. Here, $0 \leq \zeta \leq 1$ is the proportion of audio-specific information contained in the audio signal:

$$z(\mathbf{v}) = \zeta\, q(\mathbf{v}) + (1 - \zeta)\, p(\mathbf{v}|\boldsymbol{\Theta}). \tag{13}$$

Accordingly, the empirical Fisher score can be rewritten as Eq. (14):

$$\begin{aligned} g_{\boldsymbol{\Theta}}^{\mathbf{V}} \approx{}& \zeta\, \nabla_{\boldsymbol{\Theta}} \int_{\mathbf{v}} q(\mathbf{v}) \log p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} \\ &+ (1 - \zeta)\, \nabla_{\boldsymbol{\Theta}} \int_{\mathbf{v}} p(\mathbf{v}|\boldsymbol{\Theta}) \log p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v}. \end{aligned} \tag{14}$$

The second integral of Eq. (14) is zero, as shown in Appendix equation (33), since the parameters $\boldsymbol{\Theta}$ of the UBM-GMM are estimated with the Maximum Likelihood (ML) estimation approach. Rewriting Eq. (14) as Eq. (15) shows that the audio-independent information is approximately discarded from the Fisher vector and that an audio signal is described on an average by what makes it different from other audio signals. This makes LAFV a highly discriminative feature for regression modelling.

$$g_{\boldsymbol{\Theta}}^{\mathbf{V}} \approx \zeta\, \nabla_{\boldsymbol{\Theta}} \int_{\mathbf{v}} q(\mathbf{v}) \log p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} = \zeta\, \nabla_{\boldsymbol{\Theta}} \mathbb{E}_{\mathbf{v}\sim q}[\log p(\mathbf{v}|\boldsymbol{\Theta})]. \tag{15}$$

An example of the global Fisher vector and LAFV is shown in figure 5 for a specific song of the PMEmo dataset, where we observe that the gradient magnitudes of LAFV are better preserved across the dimensions relative to the global Fisher vector's gradient magnitude. The extracted LAFV features $\mathbf{X} \in \mathbb{R}^{N\times 2K}$ serve as the input to the DGP regression modelling.

3.4 *DGP regression*

Gaussian process regression (GPR) is a non-parametric probabilistic model that can provide uncertainty estimates of the predicted output without the need for manual hyper-

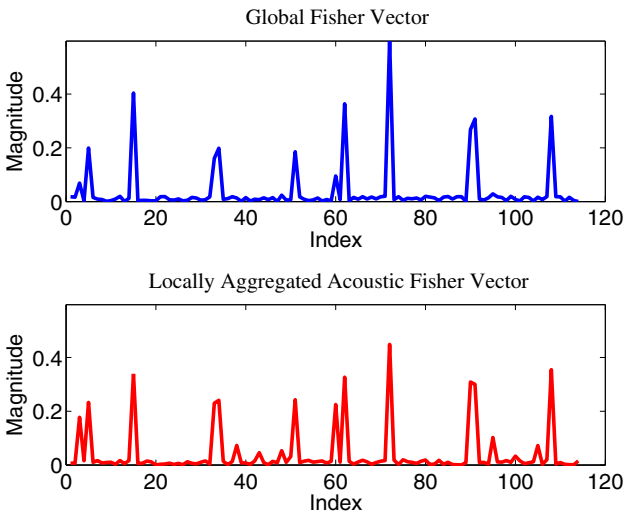


Figure 5. Acoustic Fisher vectors for *SongID*: 210.

parameter tuning [34]. A GP can be considered to be an ∞ dimension extension of a multi-variate normal (MVN); given N training observations in a dataset, the output can be considered to be a vector sampled from an N-dimensional MVN. For training data $\{\mathbf{x}_n, y_n\}_{n=1}^N$, the output of GPR is modelled as $y_n = f(\mathbf{x}_n) + \epsilon$, where $\epsilon \sim \mathcal{N}(\epsilon \,|\, 0, \sigma_n^2)$, $f \sim GP(\mathbf{m}(\mathbf{x}), \mathbf{K_{XX}})$, σ_n^2 is the noise variance, the mean $\mathbf{m}(\mathbf{x})$ is typically assumed to be zero and the elements of the kernel matrix $\mathbf{K_{XX}}$ are specified by $k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\intercal \Lambda (\mathbf{x}_i - \mathbf{x}_j)\right)$, where Λ is the diagonal matrix having elements l_n^{-2} representing the length-scales indicating smoothness of the function. For new inputs $\mathbf{X}_*$, the outputs can be predicted according to Eq. (16), where $\boldsymbol{\phi}$ refers to the set of hyper-parameters $\{\sigma_n^2, \sigma_f^2, l_n\}$. To determine the appropriate $\boldsymbol{\phi}$, the model log-likelihood given by Eq. (17) is optimized with respect to each hyper-parameter. Scalabilty is an issue with GPR, since the training complexity is $\mathrm{O}(N^3)$. However, the test complexity is only $\mathrm{O}(N^2)$ if the inverse of the kernel matrix $\mathbf{K_{XX}}$ is pre-computed as it depends only on the training data:

$$P(\mathbf{y}_* \,|\, \mathbf{X}_*, \mathbf{X}, \mathbf{y}, \boldsymbol{\phi}) \sim \mathcal{N}(\mathbf{y}_* \,|\, \mathbf{k}_{\mathbf{X}_*\mathbf{X}} \mathbf{K}_{\mathbf{XX}}^{-1} \mathbf{y}, \; k_{\mathbf{X}_*\mathbf{X}_*} - \mathbf{k}_{\mathbf{X}_*\mathbf{X}} \mathbf{K}_{\mathbf{XX}}^{-1} \mathbf{k}_{\mathbf{X}\mathbf{X}_*}), \tag{16}$$

$$\log P(\mathbf{y} \,|\, \mathbf{X}, \boldsymbol{\phi}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K_{XX}}| - \frac{1}{2}\mathbf{y}^\intercal \mathbf{K}_{\mathbf{XX}}^{-1} \mathbf{y}. \tag{17}$$

As a form of deep learning method, a deep architecture called DGP is proposed in [23] while still retaining the advantages of GP. Essentially, DGP is a process composition of multiple GPs, i.e. it is a distribution of functions constructed by composing GPs as shown in Eq. (18), where the GP of dth unit in lth hidden layer is given by Eq. (19) for every $f_d^{(l)} \in \boldsymbol{f}^{(l)}$. An example of a 3-layer DGP is shown in figure 6, which consists of 4-dimensional inputs, 1-dimensional output and two 5-dimensional hidden layers and all layers are fully connected.

$$\boldsymbol{f}^{(1:L)}(\mathbf{x}) = \boldsymbol{f}^{(L)}(\boldsymbol{f}^{(L-1)}(\ldots(\boldsymbol{f}^{(2)}(\boldsymbol{f}^{(1)}(\mathbf{x})))\ldots)) \tag{18}$$

$$f_d^{(l)} \sim GP\left(0, k_d^{(l)}(\mathbf{x}, \mathbf{x}')\right). \tag{19}$$

FITC (Fully Independent Training Conditional) approximation is proposed in [23] to reduce the training complexity using pseudo-data instead of the actual training data. Using FITC GP, instead of learning the model on N training input–output pairs, the model is learned on the pseudo-dataset of size $M\,(<<N)$ consisting of pseudo-inputs $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_m\}_{m=1}^M$ and the corresponding pseudo-outputs $\bar{\mathbf{y}} = \{\bar{y}_m\}_{m=1}^M$, which correspond to the function values at the pseudo-inputs. Conditioned on the pseudo-data, the observed output value y_n is generated from the

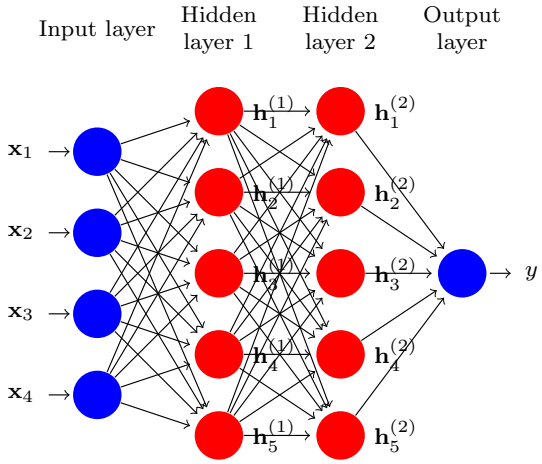


Figure 6. Illustration of a 3-layer DGP

corresponding input $\mathbf{x}_n$ by conditioning an MVN on the pseudo-data according to Eq. (20), where $k_{\mathbf{x}_n} = k(\mathbf{x}_n, \mathbf{x}_n)$, $\mathbf{k}_{\mathbf{x}_n} = [k(\mathbf{x}_n, \bar{\mathbf{x}}_1), \ldots, k(\mathbf{x}_n, \bar{\mathbf{x}}_M)]$ and $\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}$ is the kernel matrix evaluated at the pseudo-inputs:

$$P(y_n \,|\, \mathbf{x}_n, \bar{\mathbf{X}}, \bar{\mathbf{y}}, \boldsymbol{\phi}) \sim \mathcal{N}\left(y_n \,|\, \mathbf{k}_{\mathbf{x}_n}^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{y}}, k_{\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n}^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \mathbf{k}_{\mathbf{x}_n}\right). \tag{20}$$

A fundamental assumption [25] is that given the pseudo-data, the training data are generated independently according to Eq. (21). As a result, the output covariance matrix becomes diagonal and it removes the complexity of $\mathrm{O}(N^3)$ matrix inversion. To avoid over-fitting, a normal prior is put on the pseudo-outputs as $P(\bar{\mathbf{y}} \,|\, \bar{\mathbf{X}}) \sim \mathcal{N}(\bar{\mathbf{y}} \,|\, \mathbf{0}, \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}})$; this helps the pseudo-data to closely resemble the training data and also serves as regularization. The computational complexity is now dominated by inverting the kernel matrix of size M, which is diagonal, and hence the training complexity reduces to $\mathrm{O}(NM^2)$ where $M<<N$. The hyper-parameters to be learned now include the kernel parameters as well as the pseudo-data: $\boldsymbol{\Phi} = \left\{\left\{\bar{\mathbf{X}}_d^{(l)}, \bar{\mathbf{y}}_d^{(l)}, \boldsymbol{\phi}_d^{(l)}\right\}_{d=1}^{D^{(l)}}\right\}_{l=1}^{L}$ for an L-layer DGP with $D^{(l)}$ dimensionality of hidden layers.

$$P(\mathbf{y} \,|\, \mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{y}}, \boldsymbol{\phi}) = \prod_{n=1}^{N} P(y_n \,|\, \mathbf{x}_n, \bar{\mathbf{X}}, \bar{\mathbf{y}}, \boldsymbol{\phi}). \tag{21}$$

Since the exact inference in DGP is infeasible due to the computational intractability of marginal likelihood [23], we refer to the method of [25] that uses Monte Carlo samples to approximate the marginal likelihood instead of using the variational inference proposed in [23]. The sampling algorithm shown in Algorithm 2 can be easily extended to most kernels as opposed to a restricted kernel family as in [23]. For an L-layer DGP with one input layer $\mathbf{X} \in \mathbb{R}^{N \times D^{(0)}}$, one output layer $\mathbf{y} \in \mathbb{R}^N$ and $L-1$ hidden layers, the

pseudo-data are denoted as $\bar{\mathbf{X}}_d^{(l)} \in \mathbb{R}^{M \times D^{(l-1)}}, \bar{\mathbf{y}}_d^{(l)} \in \mathbb{R}^M$ and the samples of hidden layer are denoted as $\tilde{\mathbf{H}}^{(l-1)}$. For the 1st layer, we have $\tilde{\mathbf{H}}^{(0)} = \mathbf{X}$ and for the dth unit in lth hidden layer the samples are drawn from $\tilde{\mathbf{H}}_{:,d}^{(l)}$, where the mean $\tilde{\boldsymbol{\mu}}_d^{(l)}$ and covariance $\tilde{\boldsymbol{\Sigma}}_d^{(l)}$ are given by Eq. (22). This procedure of sampling hidden values $\tilde{\mathbf{H}}_{:,d}^{(l)}$ from the previous layer $\tilde{\mathbf{H}}_{:,d}^{(l-1)}$ is repeated till the last layer:

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_d^{(l)} &= \mathbf{k}_{\tilde{\mathbf{H}}_{:,d}^{(l-1)} \bar{\mathbf{X}}_d^{(l)}} \mathbf{K}^{-1}_{\bar{\mathbf{X}}_d^{(l)} \bar{\mathbf{X}}_d^{(l)}} \bar{\mathbf{y}}_d^{(l)}, \\ \tilde{\boldsymbol{\Sigma}}_d^{(l)} &= diag\left(k_{\tilde{\mathbf{H}}_{:,d}^{(l-1)} \tilde{\mathbf{H}}_{:,d}^{(l-1)}} - \mathbf{k}_{\tilde{\mathbf{H}}_{:,d}^{(l-1)} \bar{\mathbf{X}}_d^{(l)}} \mathbf{K}^{-1}_{\bar{\mathbf{X}}_d^{(l)} \bar{\mathbf{X}}_d^{(l)}} \mathbf{k}_{\bar{\mathbf{X}}_d^{(l)} \tilde{\mathbf{H}}_{:,d}^{(l-1)}} \right).\end{aligned} \tag{22}$$

For the last layer, the mean $\tilde{\boldsymbol{\mu}}^{(L)}$ and covariance $\tilde{\boldsymbol{\Sigma}}^{(L)}$ at the output layer L are used to compute the MVN density, from which the marginal likelihood is obtained using J samples $\left\{ \tilde{\boldsymbol{\mu}}_j^{(L)}, \tilde{\boldsymbol{\Sigma}}_j^{(L)} \right\}_{j=1}^{J}$ according to the Monte Carlo sampling technique. The log-likelihood LL of DGP is given by Eq. (23), where LSE refers to the logsumexp function, the likelihood is

$$P\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_j^{(L)}, \tilde{\boldsymbol{\Sigma}}_j^{(L)} \right) \sim \mathcal{N}\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_j^{(L)}, \tilde{\boldsymbol{\Sigma}}_j^{(L)} \right)$$

for $j = 1, \ldots, J$ and

$$P\left(\bar{\mathbf{y}}_d^{(l)} \mid \bar{\mathbf{X}}_d^{(l)} \right) \sim \mathcal{N}\left(\bar{\mathbf{y}}_d^{(l)} \mid \mathbf{0}, \mathbf{K}_{\bar{\mathbf{X}}_d^{(l)} \bar{\mathbf{X}}_d^{(l)}} \right)$$

is the normal prior on the pseudo-data:

$$\begin{aligned}LL = LSE\left[\log P\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_1^{(L)}, \tilde{\boldsymbol{\Sigma}}_1^{(L)} \right), \ldots, \log P\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_J^{(L)}, \tilde{\boldsymbol{\Sigma}}_J^{(L)} \right) \right] \\ - \log(N) + \sum_{l=1}^{L} \sum_{d=1}^{D^{(l)}} \log P\left(\bar{\mathbf{y}}_d^{(l)} \mid \bar{\mathbf{X}}_d^{(l)} \right).\end{aligned} \tag{23}$$

Algorithm 2 DGP model with FITC pseudo-data and Monte Carlo sampling.

Input: Data $\left\{ \mathbf{X} \in \mathbb{R}^{N \times 2K}, \mathbf{y} \in \mathbb{R}^N \right\}$, hyper-parameters $\boldsymbol{\Phi}$
Output: DGP model log-likelihood

```
H̃^(0) = X
for j = 1, ..., J do
    for each layer l = 1, ..., L − 1 do
        for each hidden unit d = 1, ..., D^(l) do
            Compute μ̃_d^(l) and Σ̃_d^(l) using Eq. (22)
            Sample H̃_{:,d}^(l) ~ N(H̃_{:,d}^(l) | μ̃_d^(l), Σ̃_d^(l))
        end for
    end for
    Compute μ̃_j^(L) and Σ̃_j^(L) at layer L using Eq. (22)
end for
Compute the model log-likelihood using Eq. (23)
```

The optimization is done by differentiating the model log-likelihood LL with respect to all hyper-parameters in $\boldsymbol{\Phi}$; for this, an automatic differentiation library in Python called *Autograd* [35] is used that can handle gradient-based optimization. The training complexity for L-layer DGP with H hidden units per layer and M pseudo-inputs and outputs is $O(NM^2LH)$ where $M << N$. Having learned the optimal $\boldsymbol{\Phi}$ parameters, to predict estimates of unseen test inputs $\mathbf{X}_*$, sampling is done as before till the last layer from which the final mean $\tilde{\boldsymbol{\mu}}^{(L)}$ and covariance $\tilde{\boldsymbol{\Sigma}}^{(L)}$ are obtained; J samples are drawn from this MVN and the average of these samples results in the output estimate $\mathbf{y}_*$.

The detailed working of training a 3-layer DGP using FITC approximation and Monte Carlo sampling technique is as follows: the training data are $\left\{ \mathbf{X} \in \mathbb{R}^{N \times 2K}, \mathbf{y} \in \mathbb{R}^N \right\}$ (here, $\mathbf{y}$ refers to either valence or arousal mood dimension), the hyper-parameters $\boldsymbol{\Phi} = \left\{ \bar{\mathbf{X}}^{(1)}, \bar{\mathbf{y}}^{(1)}, \boldsymbol{\phi}^{(1)}, \bar{\mathbf{X}}^{(2)}, \bar{\mathbf{y}}^{(2)}, \boldsymbol{\phi}^{(2)}, \bar{\mathbf{X}}^{(3)}, \bar{\mathbf{y}}^{(3)}, \boldsymbol{\phi}^{(3)} \right\}$ are initialized to random values, except for the pseudo-data of the first layer $\bar{\mathbf{X}}^{(1)}$, which is initialized as the M $(= 100)$ cluster centres of $\mathbf{X}$ obtained using the K-Mean clustering algorithm (this helps in distributing the pseudo-inputs over the entire range of $\mathbf{X}$) and $D^{(1)} = D^{(2)}$ $(= 200)$. Additionally, the length-scale l_n is initialized as the median distance between all pairs of input features $\mathbf{x}_n$ for $n = 1, \ldots, N$. For the first layer, hidden values $\tilde{\mathbf{H}}_{:,d}^{(1)}$ are sampled given the training data $\mathbf{X}$ as per Eq. (24). For the second layer, hidden values $\tilde{\mathbf{H}}_{:,d}^{(2)}$ are sampled given the samples of previous hidden layer $\tilde{\mathbf{H}}_{:,d}^{(1)}$ as per Eq. (25). For the output (third) layer, J $(= 100)$ samples of $\tilde{\mathbf{H}}_{:,d}^{(2)}$ are obtained resulting in $\left\{ \tilde{\mathbf{H}}_j \right\}_{j=1}^{J}$, where $\tilde{\mathbf{H}}_j = \left(\tilde{\mathbf{H}}_{:,d}^{(2)} \right)_j$. For each such sample, the conditional probability of observed output is given by Eq. (26), where the mean $\tilde{\boldsymbol{\mu}}_j^{(3)}$ and covariance $\tilde{\boldsymbol{\Sigma}}_j^{(3)}$ are given by Eq. (27):

$$P\left(\tilde{\mathbf{H}}_{:,d}^{(1)} \mid \mathbf{X}, \bar{\mathbf{X}}^{(1)}, \bar{\mathbf{y}}^{(1)}, \boldsymbol{\phi}^{(1)} \right) \sim \mathcal{N}\left(\tilde{\mathbf{H}}_{:,d}^{(1)} \mid \tilde{\boldsymbol{\mu}}_d^{(1)}, \tilde{\boldsymbol{\Sigma}}_d^{(1)} \right), \tag{24}$$

$$P\left(\tilde{\mathbf{H}}_{:,d}^{(2)} \mid \tilde{\mathbf{H}}_{:,d}^{(1)}, \bar{\mathbf{X}}^{(2)}, \bar{\mathbf{y}}^{(2)}, \boldsymbol{\phi}^{(2)} \right) \sim \mathcal{N}\left(\tilde{\mathbf{H}}_{:,d}^{(2)} \mid \tilde{\boldsymbol{\mu}}_d^{(2)}, \tilde{\boldsymbol{\Sigma}}_d^{(2)} \right), \tag{25}$$

$$P\left(\mathbf{y} \mid \tilde{\mathbf{H}}_j, \bar{\mathbf{X}}^{(3)}, \bar{\mathbf{y}}^{(3)}, \boldsymbol{\phi}^{(3)} \right) \sim \mathcal{N}\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_j^{(3)}, \tilde{\boldsymbol{\Sigma}}_j^{(3)} \right), \tag{26}$$

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_j^{(3)} &= \mathbf{k}_{\tilde{\mathbf{H}}_j \bar{\mathbf{X}}_d^{(3)}} \mathbf{K}^{-1}_{\bar{\mathbf{X}}_d^{(3)} \bar{\mathbf{X}}_d^{(3)}} \bar{\mathbf{y}}_d^{(3)}, \\ \tilde{\boldsymbol{\Sigma}}_j^{(3)} &= diag\left(k_{\tilde{\mathbf{H}}_j \tilde{\mathbf{H}}_j} - \mathbf{k}_{\tilde{\mathbf{H}}_j \bar{\mathbf{X}}_d^{(3)}} \mathbf{K}^{-1}_{\bar{\mathbf{X}}_d^{(3)} \bar{\mathbf{X}}_d^{(3)}} \mathbf{k}_{\bar{\mathbf{X}}_d^{(3)} \tilde{\mathbf{H}}_j} \right).\end{aligned} \tag{27}$$

The marginal likelihood can then be approximated from these samples using Eq. (28). The approximation reaches the true value as the number of samples J increases. The log-likelihood LL given by Eq. (29) is optimized using

Table 1. Performance evaluation of estimating music mood with 10-fold cross-validation.

Method	R^2_{Arousal}	$RMSE_{\text{Arousal}}$	R^2_{Valence}	$RMSE_{\text{Valence}}$
SVR (LLD)	0.603 ± 0.026	0.135 ± 0.039	0.411 ± 0.043	0.158 ± 0.046
GPR (LLD) [13]	0.627 ± 0.083	0.126 ± 0.061	0.403 ± 0.051	0.146 ± 0.017
HDM ($G = 7$) [7]	0.641 ± 0.019	0.112 ± 0.043	0.435 ± 0.032	0.133 ± 0.055
Aggregate GPR [6]	0.635 ± 0.071	0.109 ± 0.019	0.429 ± 0.074	0.127 ± 0.023
2-layer DGP (LLD)	0.673 ± 0.047	0.104 ± 0.072	0.458 ± 0.086	0.131 ± 0.035
2-layer DGP (BAGMM)	0.691 ± 0.065	0.095 ± 0.015	0.481 ± 0.052	0.119 ± 0.027
2-layer DGP (LAFV)	$\mathbf{0.738 \pm 0.038}$	$\mathbf{0.082 \pm 0.011}$	$\mathbf{0.507 \pm 0.028}$	$\mathbf{0.102 \pm 0.079}$
3-layer DGP (LLD)	0.724 ± 0.017	0.091 ± 0.022	0.476 ± 0.018	0.116 ± 0.042
3-layer DGP (BAGMM)	0.733 ± 0.021	0.081 ± 0.031	0.519 ± 0.032	0.104 ± 0.021
3-layer DGP (LAFV)	$\mathbf{0.786 \pm 0.048}$	$\mathbf{0.064 \pm 0.053}$	$\mathbf{0.613 \pm 0.081}$	$\mathbf{0.093 \pm 0.056}$

Bold is used to highlight the results obtained with the proposed method.

automatic differentiation to learn the hyper-parameters $\mathbf{\Phi}$. This working can be easily generalized for an L-layer DGP.

$$P(\mathbf{y} \mid \mathbf{X}, \mathbf{\Phi}) \approx \frac{1}{J} \sum_{j=1}^{J} P\left(\mathbf{y} \mid \tilde{\mathbf{H}}_j, \bar{\mathbf{X}}^{(3)}, \bar{\mathbf{y}}^{(3)}, \boldsymbol{\phi}^{(3)}\right), \tag{28}$$

$$\begin{aligned} LL = & LSE\left[\log P\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_1^{(3)}, \tilde{\mathbf{\Sigma}}_1^{(3)}\right), \ldots, \log P\left(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}_J^{(3)}, \tilde{\mathbf{\Sigma}}_J^{(3)}\right)\right] \\ & - \log(N) + \sum_{l=1}^{3} \sum_{d=1}^{200} \log P\left(\bar{\mathbf{y}}_d^{(l)} \mid \bar{\mathbf{X}}_d^{(l)}\right). \end{aligned} \tag{29}$$

4. Experimental results

4.1 *Music mood estimation*

The evaluation of music mood estimation is performed on the PMEmo dataset with two metrics: R^2 (the coefficient of determination) and root mean square error *RMSE* given by Eq. (30) to measure the regression performance between the predicted VA values and the static VA annotations provided in the dataset. Here, $\hat{y}^{(i)}$ is the predicted value of ith song, $y^{(i)}$ is the corresponding ground truth and $\bar{y}$ is the average value of the mood dimension:

$$\begin{aligned} R^2 &= 1 - \frac{\sum_i (\hat{y}^{(i)} - y^{(i)})^2}{\sum_i (\hat{y}^{(i)} - \bar{y})^2}, \\ RMSE &= \sqrt{\frac{1}{N} \sum_i (\hat{y}^{(i)} - y^{(i)})^2}. \end{aligned} \tag{30}$$

The dataset is randomly partitioned into 10 mutually exclusive subsets $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_{10}$, each of approximately equal size. In the ith experiment ($i = 1, \ldots, 10$), the testing subset $\mathcal{Q}_i$ is used for model evaluation and the remaining training subsets are used for fitting the regression model. The means and standard deviations of the evaluation metric values resulting from these 10 experiments are reported to measure the performance (10-fold cross-validation) [36].

We evaluate the performance of SVR, baseline single-layer GPR [13], HDM [7], Aggregate GPR [6], the proposed 2-layer DGP and 3-layer DGP approaches for estimating the music mood. For HDM, the number of latent histograms for modelling audio topics is set to $K_H = 256$ as suggested in [7]; a limitation of HDM is that it requires fine-tuning of K_H and the choice of grid-size G. For the Aggregate GPR method, adaptive aggregation of GP regressors is used with standard acoustic features as described in [6]. The LLD features of the PMEmo dataset are used for the existing techniques; the LLD as well as the Bayesian Acoustic GMM features of [12] and the proposed LAFV features are used for the DGP models.

The music mood estimation performance results are shown in table 1, where we observe that the coefficient of determination R^2 is higher for both VA mood dimensions using 3-layer DGP with the proposed LAFV features. Also, the *RMSE* is lower for both mood dimensions with the proposed LAFV features relative to the existing techniques. Overall, we observe an improvement in R^2 of 25.4% for arousal and 52.1% for valence estimation relative to the baseline single layer GPR [13].

4.2 *Music mood retrieval*

The architecture of the music mood retrieval system is shown in figure 7, where the input is a query song with its

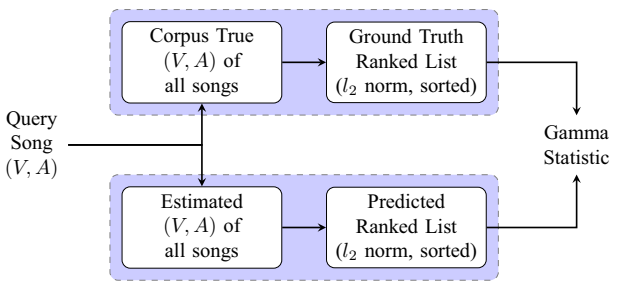


Figure 7. Architecture of music retrieval system.

Table 2. Retrieval evaluation results with 5-fold cross-validation.

	SVR	GPR [13]	HDM [7]	Aggregate GPR [6]	3-layer DGP (LLD)	3-layer DGP (BAGMM)	3-layer DGP (LAFV)
G	0.374 ± 0.069	0.44 ± 0.039	0.56 ± 0.019	0.52 ± 0.094	0.53 ± 0.045	0.68 ± 0.027	0.74 ± 0.062

known VA (V, A) values and the objective is to retrieve songs similar to the query song in the (V, A) space. The true (V, A) values of all songs are either available in the music corpus or learned from the crowdsourced annotator data using judgement analysis techniques [37]. From the music mood estimation system, the estimated (V, A) values are determined for every music clip in the corpus. In both cases, a list is created by determining the Euclidean distance (l_2 norm) between the query song (V, A) value and the corresponding true or estimated (V, A) values. The list is then sorted as per increasing Euclidean distance, resulting in a ranked list. The top retrieved songs (e.g. 10) can be presented as most similar to the query song, which can thus serve as a recommendation system to the end-user. For retrieval performance evaluation, the gamma statistic G is computed from the ground truth ranked list and the predicted ranked list using Eq. (31), where N_C is the number of concordant (correctly ranked) pairs and N_D is the number of discordant (incorrectly ranked) pairs. The value of gamma statistic G ranges from -1 to 1, with a larger value indicating better performance [5]:

$$G = \frac{N_C - N_D}{N_C + N_D} \tag{31}$$

We use 5-fold cross-validation for retrieval evaluation where each clip in the test fold is regarded as a query song to generate two ranked lists (true and predicted) over the remaining songs. The lists are derived by comparing the corpus (V, A) values to the estimated (V, A) values of the specific method using the Euclidean distance. The ranks are obtained by sorting the distances in increasing order. The average and standard deviation values of the gamma statistic G given by Eq. (31) are reported in table 2, where we observe that the proposed work outperforms the existing state-of-the-art approaches. The 3-layer DGP regression model with the proposed LAFV features results in a higher G value relative to the existing work, i.e. the ranks are preserved better using the DGP regression model with LAFV features. Overall, we observe an improvement in G of 68% relative to the baseline single layer GPR [13] for the retrieval performance.

5. Conclusion

We proposed novel features termed as LAFV that measure the deviation of the acoustic features from the average distribution of the features modelled by the GMM. Onset detection was applied to the audio as a pre-processing step to determine the variable-length segments for preserving the temporal context. A universal background GMM model is learned from a sample of the acoustic features extracted for the entire dataset. While the GMM acoustic posterior probabilities represent a discrete distribution as used in the BAGMM feature representation, we propose a continuous distribution with LAFV, resulting in highly discriminative features.

For the regression modelling, using FITC GP approximation and the Monte Carlo sampling procedure, a DGP algorithm is derived in general for an L-layer DGP; a specific example of 3-layer DGP is explained and it showed a significant performance improvement for both annotation and retrieval tasks relative to the existing techniques. As inferred from figure (2), since the VA dimensions are not uncorrelated, for further work, a structured DGP algorithm can be derived to jointly learn the VA dimensions.

Appendix

(i) The expected value of the Fisher score with respect to the UBM-GMM model is zero as shown in Eq. (32):

$$\begin{aligned}
\mathbb{E}_{\mathbf{v}\sim\mathrm{GMM}_{\boldsymbol{\Theta}}}[g_{\boldsymbol{\Theta}}^{\mathbf{V}}] &= \frac{1}{N_s}\sum_{t=1}^{N_s}\int_{\mathbf{v}_t} \nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}_t|\boldsymbol{\Theta}) \times p(\mathbf{v}_t|\boldsymbol{\Theta})\, d\mathbf{v}_t \\
&= \frac{1}{N_s}\sum_{t=1}^{N_s}\int_{\mathbf{v}_t} \frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}_t|\boldsymbol{\Theta})}{p(\mathbf{v}_t|\boldsymbol{\Theta})} p(\mathbf{v}_t|\boldsymbol{\Theta})\, d\mathbf{v}_t \\
&= \frac{1}{N_s}\sum_{t=1}^{N_s}\int_{\mathbf{v}_t} \nabla_{\boldsymbol{\Theta}} p(\mathbf{v}_t|\boldsymbol{\Theta})\, d\mathbf{v}_t = \frac{1}{N_s}\sum_{t=1}^{N_s}\nabla_{\boldsymbol{\Theta}}\int_{\mathbf{v}_t} p(\mathbf{v}_t|\boldsymbol{\Theta})\, d\mathbf{v}_t \\
&= \frac{1}{N_s}\sum_{t=1}^{N_s}\nabla_{\boldsymbol{\Theta}} 1 = 0.
\end{aligned} \tag{32}$$

(ii) Due to the use of ML estimation approach, the second integral of Eq. (14) is zero as shown in Eq. (33):

$$\begin{aligned}
\nabla_{\boldsymbol{\Theta}}\int_{\mathbf{v}} p(\mathbf{v}|\boldsymbol{\Theta}) \log p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} &= \int_{\mathbf{v}} \nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}|\boldsymbol{\Theta}) p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} \\
&= \int_{\mathbf{v}} \frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta})} p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} = \int_{\mathbf{v}} \nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} \\
&= \nabla_{\boldsymbol{\Theta}}\int_{\mathbf{v}} p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} = \nabla_{\boldsymbol{\Theta}} 1 = 0.
\end{aligned} \tag{33}$$

(iii) The FIM $\mathbf{F}_{\boldsymbol{\Theta}}$ is equivalent to the negative expected Hessian of the model's log-likelihood as shown in Eq. (35).

Since the Hessian can be written as the Jacobian of the gradient, we have

$$\begin{aligned} H_{\log p(\mathbf{v}|\boldsymbol{\Theta})} &= J(\nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}|\boldsymbol{\Theta})) \\ &= J\left(\frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta})}\right) \\ &= \frac{H_{p(\mathbf{v}|\boldsymbol{\Theta})} p(\mathbf{v}|\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta}) \nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})^{\mathsf{T}}}{p(\mathbf{v}|\boldsymbol{\Theta}) p(\mathbf{v}|\boldsymbol{\Theta})} \\ &= \frac{H_{p(\mathbf{v}|\boldsymbol{\Theta})} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta}) p(\mathbf{v}|\boldsymbol{\Theta})} - \frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta}) \nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})^{\mathsf{T}}}{p(\mathbf{v}|\boldsymbol{\Theta}) p(\mathbf{v}|\boldsymbol{\Theta})} \\ &= \frac{H_{p(\mathbf{v}|\boldsymbol{\Theta})}}{p(\mathbf{v}|\boldsymbol{\Theta})} - \frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta})} \left(\frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta})}\right)^{\mathsf{T}} \end{aligned} \tag{34}$$

Taking the expectation of Eq. (34) with respect to the UBM-GMM model, we have

$$\begin{aligned} &\mathbb{E}_{\mathbf{v}\sim \mathrm{GMM}_{\boldsymbol{\Theta}}}\left[H_{\log p(\mathbf{v}|\boldsymbol{\Theta})}\right] \\ &= \mathbb{E}_{\mathbf{v}\sim \mathrm{GMM}_{\boldsymbol{\Theta}}}\left[\frac{H_{p(\mathbf{v}|\boldsymbol{\Theta})}}{p(\mathbf{v}|\boldsymbol{\Theta})}\right] - \\ &\mathbb{E}_{\mathbf{v}\sim \mathrm{GMM}_{\boldsymbol{\Theta}}}\left[\frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta})}\left(\frac{\nabla_{\boldsymbol{\Theta}} p(\mathbf{v}|\boldsymbol{\Theta})}{p(\mathbf{v}|\boldsymbol{\Theta})}\right)^{\mathsf{T}}\right] \\ &= \int_{\mathbf{v}} \frac{H_{p(\mathbf{v}|\boldsymbol{\Theta})}}{p(\mathbf{v}|\boldsymbol{\Theta})} p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v} \\ &- \mathbb{E}_{\mathbf{v}\sim \mathrm{GMM}_{\boldsymbol{\Theta}}}[\nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}|\boldsymbol{\Theta}) \cdot \nabla_{\boldsymbol{\Theta}} \log p(\mathbf{v}|\boldsymbol{\Theta})^{\mathsf{T}}] \\ &= H_{\int_{\mathbf{v}} p(\mathbf{v}|\boldsymbol{\Theta})\, d\mathbf{v}} - \mathbf{F}_{\boldsymbol{\Theta}} \\ &= H_1 - \mathbf{F}_{\boldsymbol{\Theta}} = -\mathbf{F}_{\boldsymbol{\Theta}}. \\ &\therefore \mathbf{F}_{\boldsymbol{\Theta}} = -\mathbb{E}_{\mathbf{v}\sim \mathrm{GMM}_{\boldsymbol{\Theta}}}\left[H_{\log p(\mathbf{v}|\boldsymbol{\Theta})}\right]. \end{aligned} \tag{35}$$

References

[1] Brinker B, Dinther R and Skowronek J 2012 Expressed music mood classification compared with valence and arousal ratings. *EURASIP Journal of Audio, Speech and Music Processing* 24: 1–14

[2] Zhang K, Zhang H, Li S, Yang C and Sun L 2018 The PMEmo dataset for music emotion recognition. In: *Proceedings of the* 8*thInternational Conference on Multimedia Retrieval*, ICMR 2018, pp. 135–142

[3] Wang J, Yang Y, Wang H and Jeng S 2015 Modeling the affective content of music with a Gaussian mixture model. *IEEE Transactions on Affective Computing* 6: 56–68

[4] Panda R, Malheiro R and Paiva R 2018 Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing* 1–14

[5] Chin Y, Wang J, Wang J and Yang Y 2016 Predicting the probability density function of music emotion using emotion space mapping. *IEEE Transactions on Affective Computing* 1–10

[6] Fukayama S and Goto M 2016 Music emotion recognition with adaptive aggregation of Gaussian process regressors. In: *Proceedings of the* 41*st IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP 2016, pp. 71–75

[7] Wang J, Wang H and Lanckriet G 2015 A histogram density modeling approach to music emotion recognition. In: *Proceedings of the* 40*thIEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP 2015, pp. 698–702

[8] Chen Y, Yang Y, Wang J and Chen H 2015 The AMG1608 dataset for music emotion recognition. In: *Proceedings of the* 40*thIEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP 2015, pp. 693–697

[9] Aljanaki A, Yang Y and Soleymani M 2017 Developing a benchmark for emotional analysis of music. *PLoS ONE* 12: e0173392

[10] Schmidt E and Kim Y 2011 Modeling musical emotion dynamics with conditional random fields. In: *Proceedings of the* 12*thConference International Society for Music Information Retrieval*, ISMIR 2011, pp. 777–782

[11] Zhang J, Huang X, Yang L, Xu Y and Sun S 2017 Feature selection and feature learning in arousal dimension of music emotion by using shrinkage methods. *Multimedia Systems* 23: 251–264

[12] Chapaneri S and Jayaswal D 2017 Structured prediction of music mood with twin Gaussian processes. In: Shankar B, Ghosh K, Mandal D, Ray S, Zhang D and Pal S (Eds.) *Pattern Recognition and Machine Intelligence, PReMI 2017, Lecture Notes in Computer Science*, vol. 10597, pp. 647–654

[13] Markov K and Matsui T 2014 Music genre and emotion recognition using Gaussian processes. *IEEE Access* 2: 688–697

[14] Jaakkola T and Haussler D 1999 Exploiting generative models in discriminative classifiers. In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 487–493

[15] Sánchez J, Perronnin F, Mensink T and Verbeek J 2015 Image classification with the Fisher vector: theory and practice. *International Journal of Computer Vision* 105: 222–245

[16] Moreno P and Rifkin R 2000 Using the Fisher kernel method for web audio classification. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2417–2420

[17] Mariethoz J, Grandvalet Y and Bengio S 2009 Kernel based text-independent speaker verification. In: *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, pp. 195–220

[18] Marchesotti L, Perronnin F, Larlus D and Csurka G 2011 Assessing the aesthetic quality of photographs using generic image descriptors. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1784–1791

[19.] Csurka G, Dance C, Fan L, Willamowski J and Bray C 2004 Visual categorization with bags of keypoints. In: *Proceedings of ECCV Statistical Learning in Computer Vision Workshop*, pp. 1–22

[20] Perronnin F, Sánchez J and Mensink T 2010 Improving the Fisher kernel for large-scale image classification. In: *Proceedings of ECCV Computer Vision Workshop*, pp. 143–156

[21] Liu X, Chen Q, Wu X, Liu Y and Liu Y 2017 CNN based music emotion classification. arXiv:1704.05665

[22] Malik M, Adavanne S, Drossos K, Virtanen T, Ticha D and Jarina R 2017 Stacked convolutional and recurrent neural networks for music emotion recognition. arXiv:1706.02292

[23] Damianou A and Lawrence N 2013 Deep Gaussian processes. In: *Proceedings of the* 16th*International Conference on Artificial Intelligence and Statistics*, AISTATS 2013, pp. 207–215

[24] Bui T, Lobato J, Lobato D, Li Y and Turner R 2016 Deep Gaussian processes for regression using approximate expectation propagation. In: *Proceedings of the* 33*rdInternational Conference on Machine Learning*, ICML 2016, pp. 1472–1481

[25] Vafa K 2016 Training deep Gaussian processes with sampling. In: *Proceedings of the* 3rd*NIPS Workshop on Advances in Approximate Bayesian Inference*, NIPS 2016, pp. 1–5

[26] Chen S, Lee Y, Hsieh W and Wang J 2015 Music emotion recognition using deep Gaussian process. In: *Proceedings of the* 7*thSignal and Information Processing Association Annual Summit and Conference*, APSIPA 2015, pp. 495–498

[27] Liang C, Su L and Yang Y 2015 Musical onset detection using constrained linear reconstruction. *IEEE Signal Processing Letters* 22: 2142–2146

[28] Schuller B, Steidl S, Batliner A, Vinciarelli A, Scherer K, Ringeval F, Chetouani M, Weninger F, Eyben F, Marchi E, Mortillaro M, Salamin H, Polychroniou A, Valente F and Kim S 2013 The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In: *Proceedings of the* 14th*Annual Conference of the International Speech Communication Association*, INTERSPEECH 2013, pp. 148–152

[29] Eyben F, Wöllmer M and Schuller B 2010 openSMILE: the Munich versatile and fast open-source audio feature extractor. In: *Proceedings of the* 18*thACM International Conference on Multimedia*, pp. 1459–1462

[30] Lartillot O and Toiviainen P 2007 A Matlab toolbox for musical feature extraction from audio. In: *Proceedings of the* 10*thInternational Conference on Digital Audio Effects*, DAFx 2007, pp. 237–244

[31] Bishop C 2006, *Pattern recognition and machine learning*. New York: Springer-Verlag

[32] Perronnin F and Dance C 2007 Fisher kernels on visual vocabularies for image categorization. *IEEE Computer Vision and Pattern Recognition*, pp. 1–8

[33] Jegou H, Perronnin F, Douze M, Sanchez J, Perez P and Schmid C 2012 Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34: 1704–1716

[34] Rasmussen C and Williams C 2006 *Gaussian processes for machine learning*. MIT Press

[35] Maclaurin D, Duvenaud D and Adams R 2015 Autograd: effortless gradients in pure Numpy. In: *Proceedings of the* 32*ndInternational Conference on Machine Learning AutoML Workshop*, ICML 2015, pp. 1–3

[36] Han J, Kamber M and Pei J 2011 *Data mining: concepts and techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[37] Chatterjee S, Mukhopadhyay A and Bhattacharyya M 2019 A review of judgment analysis algorithms for crowdsourced opinions. *IEEE Transactions on Knowledge and Data Engineering*