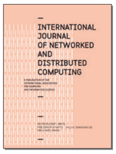
International Journal of Networked and Distributed Computing
Vol. **8**(**2**); March (2020), *pp.* 58–66
DOI: https://doi.org/10.2991/ijndc.k.200217.001; ISSN 2211-7938; eISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



Research Article

Multimedia Analysis and Fusion via Wasserstein Barycenter

Cong Jin[1,*], Junhao Wang[1], Jin Wei[1], Lifeng Tan[1], Shouxun Liu[1], Wei Zhao[1], Shan Liu[1], Xin Lv[2]

[1]*School of Information and Communication Engineering, Communication University of China, Beijing 100024, China*
[2]*School of Animation and Digital Arts, Communication University of China, Beijing 100024, China*

ARTICLE INFO

Article History

Received 13 August 2019
Accepted 18 October 2019

Keywords

Multimedia analysis
Wasserstein Barycenter
fusion

ABSTRACT

Optimal transport distance, otherwise known as Wasserstein distance, recently has attracted attention in music signal processing and machine learning as powerful discrepancy measures for probability distributions. In this paper, we propose an ensemble approach with Wasserstein distance to integrate various music transcription methods and combine different music classification models so as to achieve a more robust solution. The main idea is to model the ensemble as a problem of Wasserstein Barycenter, where our two experimental results show that our ensemble approach outperforms existing methods to a significant extent. Our proposal offers a new visual angle on the application of Wasserstein distance through music transcription and music classification in multimedia analysis and fusion tasks.

1. INTRODUCTION

Many multimedia analysis algorithms rely on probability distributions that characterize audio or image features as generally high dimensions. For example, music analysis methods, such as automatic music transcription (AMT) [1] and music classification [2], in these applications, having sufficient similarity (or equivalent difference) between distributions becomes crucial. The classical distance or difference of probability density includes Kullback Leibler divergence, Kolmogorov distance, Bhattacharyya distance (also known as Hellinger distance), etc. Recently, the framework of optimal transportation and Wasserstein distance [3] are also called earth mover's distance (EMD) [4], which has aroused great interest in computer vision [5], machine learning [6] and data fusion. Wasserstein distance calculates the best warped starter to map the measure μ to the second ν for a given input probability. Optimality corresponds to a loss function that measures the predicted value of the displacement in the warped starter. Generally, considering the accumulation of μ and ν, Wasserstein distance calculates the definition of the displacement of every particle from traces of its mass to the displacement of μ to ν.

In this paper, the applications of Wasserstein Barycenter [7] algorithm in music transcription and classification are discussed. The first application described in this paper is music transcription. In this study, we use non-negative matrix factorization (NMF) as the method of converting audio signal to musical instrument digital interface (MIDI) format and Wasserstein Barycenter as the algorithm of data fusion. The second is music classification, first we used jSymbolic software to extract the key features, and then input these features into our traditional machine learning model including eXtreme gradient boosting (XGB), back propagation neural network (BPNN), support vector machine (SVM). Finally, we propose Wasserstein Barycenter as the ensemble method of these models. Wasserstein distance, also known as EMD, is used to measure the distance between two distributions. Compared with Kullback Leibler (KL) and Jenson-Shannon (JS)–divergence, Wasserstein distance has the advantage that even if the support sets of two distributions do not overlap or overlap very little, it can still reflect the distance between the two distributions.

The remaining sections are organized as follows: In Section 2, we first review some of the theorems in the literature and mainly present the theorems of Wasserstein Barycenter, methodology of music transcription and model of music classification. Section 3 conducts experiment result of music transcription and music classification. Section 4 introduces the evaluation methods and the comparison of different models. In Section 5, we summarize our work and present future research directions in the field.

2. RELATED WORK

Materials show that the Wasserstein distance presents a useful methodology for quantifying geometric differences between the different distributions. In particular, they are mostly applied as variables in content-based image retrieval [8], modeling and visualization of image intensity value [9–12], estimated average probability metrics (i.e. Barycenter of gravity) [13,14], cancer detection [15,16], super resolution [17] and other applications. Recent advances in variation minimization [18,19], particle approximation [20], multi-scale schemes [21,22], and entropy regularization [5,6,23], transmission metrics can be effectively utilized to pattern recognition, machine learning and signal processing issues. Moreover, Wang et al. [10] describes a theory for computing the transport distance (expressed as

[*] *Corresponding author. Email: jincong0623@cuc.edu.cn*

linear optimal transport) between N image data sets requiring N minimized distance problems. Rabin et al. [14] and Bonneel et al. [23] proposes the truth that these problems are easy to solve for distribution of one-dimension, and introduces a change in the local distance defined as the Sliced Wasserstein distance. Finally, recent work [24–26] shows that the transmission frames can be treated as a reversible signal conversion framework allowing signal classes to be more linearly separated for various pattern recognition and machine learning tasks.

Due to the benefits of using the above transport distances and Wasserstein distance, and taking into account the flexibility and strength of Wasserstein Barycenter [7] algorithm, ensemble methods using Wasserstein Barycenter in dealing with data fusion have been described with applications in music transcription and music classification.

Various research groups of polyphonic pitch detection used different techniques for music transcriptions. Yeh et al. [27] presented a cross pitch estimation algorithm based on the score function of a pitch candidate set. Nam et al. [28] posed a transcription approach which uses deep belief networks to calculate a mid-level time-pitch representation. Duan et al. [29] and Emiya et al. [30] proposed a model of spectral peak, non-peak region and the residual noise via Maximum Likelihood (ML) methods. More recently, Peeling and Godsill [31] raised a F0 estimation function and an inhomogeneous Poisson in the frequency domain. In spectrogram factorization-based multi-pitch detection, resulting in harmonic and inharmonic NMF, Vincent et al. [32] merged harmonic constraints in the NMF model. Bertin et al. [33] presented a Bayesian model based on NMF, and each pitch in harmonic positions is treated as a model of Gaussian components. Fuentes et al. [34] modeled each note as a weighted amount of narrowband log spectrum, and switched to log frequency with the convoluted PLCA algorithm. Abdallah and Plumbley [35] combined machine learning and dictionary learning via non-negative sparse coding.

Since the emergence of the Internet, music classification has been a widely studied field. Researchers around the world have put a lot of energy into the field of music classification. Although researchers have proposed different algorithms from different perspectives, most of them rely on excellent and well-designed designs and the construction of appropriate classifiers for music data. Traditional music classification methods are based on supervised machine learning [36]. They used k-Nearest Neighbor (KNN) and Gaussian Mixture model. The above methods as well as Mel-frequency Cepstral coefficients were used for noisy classification. Lee et al. [37] introduced a multiclass SVM approach that translated multiple classification problems into a single optimization problem rather than breaking it down into multiple binary classification problems. There are many favorable properties of Wasserstein distance, which are recorded in theories [3] and practice [38]. With the recent success of Deep Neural Networks (DNN), a number of studies apply these techniques to speech and other forms of audio data. Hussain and Haque [39] developed SwishNet— a fast CNN for audio data classification and segmentation. AzarNet, a DNN was created by Azar et al. [40] to recognize classical music. Liu et al. [41] fully exploited of low-level information in the audio from spectrograms to develop a new CNN algorithm. Nasrullah and Zhao [42] reviewed the classification method of artists under this framework and conducted an empirical study on the impact of introducing time structure into feature representation. Under the comprehensive conditions, they applied the convolutional recursive neural network to the music artist recognition data set and established the classification architecture.

Based on the recent works on EMD [4] and Wasserstein Barycenter [7], we propose a converged method and have concrete theoretical and practical advantages in music transcription and classification. We derive mathematical results that enable Wasserstein Barycenter ensemble to be applied in music transcription and music classification. Finally, we prove through experiments that Wasserstein Barycenter ensemble outperform the commonly used models such as SVM, XGB and BPNN.

3. METHODOLOGY

Our idea of music signal processing ensemble is inspired by the recent study on EMD and Wasserstein Barycenter in the area of machine learning. Here, we introduce their formal definitions first. And then we propose music classification models with compared classifiers.

3.1. Music Transcription with Earth Mover's Distance

Let $X = \{x_1, x_2, ..., x_{n1}\}$ and $Y = \{y_1, y_2, ..., y_{n1}\}$ be two sets of weighted points in R^d with non-negative weights α_i and β_j for each $\alpha_i \in X$ and $\beta_j \in Y$ respectively, and W_X and W_Y be their corresponding total weights. The EMD [4,43] between X and Y is

$$\mathrm{EMD}(X,Y) = \frac{1}{\min\{W_X, W_Y\}} \min_F \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f_{ij} \,||x_i - y_j||^2, \tag{1}$$

where $F = \{f_{ij}\}$ is a feasible flow from X to Y, with each $f_{ij} \geq 0$, $\sum_{i=1}^{n_1} f_{ij} \leq \beta_j$, $\sum_{j=1}^{n_2} f_{ij} \leq \alpha_i$, and $\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} f_{ij} = \min\{W_X, W_Y\}$.

Roughly speaking, EMD is an example of the least cost and maximum flow problem in the Euclidean space R^d. Therefore, the problem of computing EMD can be solved by linear programming [44]. In addition, several faster algorithms have been proposed by using the techniques developed in computational geometry [43,45,46]. Following EMD, we have the definition of Wasserstein Barycenter.

3.2. Music Classification with Compared Classifiers

3.2.1. Multiclass SVM

Support vector machine is a useful technique for data classification. Even though it is considered that neural networks are easier to use than this; however, sometimes unsatisfactory results are obtained. A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one target values and several attributes. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes.

Classification in SVM is an example of Supervised Learning. Known labels help indicate whether the system is performing in a right way or not. This information points to a desired response, validating the accuracy of the system, or be used to help the system learn to act correctly. A step in SVM classification involves identification as which are intimately connected to the known classes. This is called feature selection or feature extraction. Feature selection and SVM classification together have a use even when prediction of unknown samples is not necessary. They can be used to identify key sets which are involved in whatever processes distinguish the classes.

Computing the SVM classifier amounts to minimizing an expression of the form

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1-y_i(w\cdot x_i - b))\right] + \lambda\,\|w\|^2. \tag{2}$$

Minimizing (2) can be rewritten as a constrained optimization problem with a differentiable objective function in the following way.

For each $i \in \{1, ..., n\}$, we introduce a variable $\zeta_i = \max(0, 1 - y_i(w \cdot x_i - b))$. Note that ζ_i is the smallest non-negative number satisfying $y_i(w \cdot x_i - b) \geq 1 - \zeta_i$.

Thus we can rewrite the optimization problem as to minimize $\frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda\,\|w\|^2$, subject to $y_i(w \cdot x_i - b) \geq 1 - \zeta_i$ and $\zeta_1 \geq 0$, for all i. This is called the primal problem.

By solving for the Lagrangian dual of the above problem, one obtains the simplified problem, which is to maximize $f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (x_i \cdot x_j) y_j c_j$ subject to $\sum_{i=1}^{n} c_i y_i = 0$, and $0 \leq c_i \leq \frac{1}{2n\lambda}$ for all i. This is called the dual problem. Since the dual maximization problem is a quadratic function of the c_i subject to linear constraints, it is efficiently solvable by quadratic programming algorithms. Here, the variables c_i are defined such that

$$\vec{w} = \sum_{i=1}^{n} c_i y_i \vec{x}_i \tag{3}$$

Moreover, $c_i = 0$ exactly when $\vec{x}_i$ lies on the correct side of the margin, and $0 < c_i < (2n\lambda)^{-1}$ when c_i lies on the margin's boundary. It follows that $\vec{w}$ can be written as a linear combination of the support vectors.

The offset b can be recovered by finding an $\vec{x}_i$ on the margin's boundary and solving

$$y_i\left(\vec{w}\cdot\vec{x}_i - b\right) = 1 \Leftrightarrow b = \vec{w}\cdot\vec{x}_i - y_i. \tag{4}$$

(Note that $y_i^{-1} = y_i$ since $y_i = \pm 1$.)

Sub-gradient decent algorithms for the SVM work directly with the expression

$$f(\vec{w}, b) = \left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1-y_i(w\cdot x_i - b))\right] + \lambda\,\|w\|^2. \tag{5}$$

Note that f is a convex function of $\vec{w}$ and b. As such, traditional gradient descent (or SGD) methods can be adapted, where instead of taking a step in the direction of the function's gradient, a step is taken in the direction of a vector selected from the function's sub-gradient. This approach has the advantage that, for certain implementations, the number of iterations does not scale with n, the number of data points.

Coordinate descent algorithms for the SVM work from the dual problem, which is to maximize $f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (x_i \cdot x_j) y_j c_j$ subject to $\sum_{i=1}^{n} c_i y_i = 0$, and $0 \leq c_i \leq \frac{1}{2n\lambda}$ for all i. For each $i \in \{1, ..., n\}$, iteratively, the coefficient c_i is adjusted in the direction of $\frac{\partial f}{\partial c_i}$. Then, the resulting vector of coefficients $(c_1', \ldots, c_n')$ is projected onto the nearest vector of coefficients that satisfies the given constraints. The process is repeated until a near-optimal vector of coefficients is obtained. The resulting algorithm is extremely fast in practice, although few performance guarantees have been proven.

One of the major strengths of SVM is that the training is relatively easy. No local optimal, unlike in neural networks. It scales relatively well to high dimensional data and the trade-off between classifier complexity and error can be controlled explicitly. The weakness includes the need for a good kernel function.

3.2.2. Multilevel Wasserstein means

For any given subset $\Theta \subset R^d$, let $P(\Theta)$ denote the space of Borel probability measures on Θ. The Wasserstein space of order $r \in [1, \infty)$ of probability measures on Θ is defined as $P_r(\Theta) = \left\{G \in P(\Theta) : \int \|x\|^r \, dG(x) < \infty\right\}$, where $\|\cdot\|$ denotes Euclidean metric in R^d. Additionally, for any $k \geq 1$ the probability simplex is denoted as

$$\Delta_k = \left\{u \in R^k : u_i \geq 0, \sum_{i=1}^{k} u_i = 1\right\} \tag{6}$$

Finally, let $O_k(\Theta)$(resp., $\mathcal{E}_k(\Theta)$) be the set of probability measures with at most (resp., exactly) k support points in Θ.

- Wasserstein distances

For any elements G and G' in $P_r(\Theta)$ where $r \geq 1$, the Wasserstein distance of order r between G and G' is defined as:

$$W_r(G, G') = \left(\inf \int_{\theta^2} \|x - y\|^r \, d\pi(x, y)\right)^{\frac{1}{r}}, \pi \in \prod(G, G') \tag{7}$$

where $\prod(G, G')$ is the set of all probability measures on $\Theta \times \Theta$ that have marginals G and G'. In other words, $W_r^r(G, G')$ is the optimal cost of moving mass from G to G', where the cost of moving unit mass is proportional to r, the power of Euclidean distance in Θ. When G and G' are two discrete measures with finite number of atoms, fast computation of $W_r(G, G')$ can be achieved. The details of this are deferred to the Supplement.

By a recursion of concepts, we can speak of measures of measures, and define a suitable distance metric on this abstract space: the space of Borel measures on $P_r(\Theta)$, to be denoted by $P_r(P_r(\Theta))$.

This is also a Polish space (that is, complete and separable metric space) as $P_r(\Theta)$ is a Polish space. It will be endowed with a Wasserstein metric of order r that is induced by a metric W_r on $P_r(\Theta)$ as follows: for any $D, D' \in P_r(P_r(\Theta))$,

$$W_r(D,D') = \left(\inf \int_{P_r(\Theta)^2} W_r^r(G,G')d\pi(G,G')\right)^{\frac{1}{r}} \tag{8}$$

where the infimum in the above ranges over all $\pi \in \prod(D, D')$ such that $\prod(D, D')$ is the set of all probability measures on $P_r(\Theta) \times P_r(\Theta)$ that has marginals D and D'. In words, $W_r(D, D')$ corresponds to the optimal cost of moving mass from D to D' where the cost of moving unit mass in its space of support $P_r(\Theta)$ is proportional to the r-power of the W_r distance in $P_r(\Theta)$. Note a slight notational abuse $-W_r$ is used for both $P_r(\Theta)$ and $P_r(P_r(\Theta))$, but it should be clear which one is being used from context.

- Wasserstein barycenter

Next, we present a brief overview of Wasserstein barycenter problem. Given probability measures (P_1, P_2, ..., $P_N \in P_2(\Theta)$), for $N \geq 1$, their Wasserstein barycenter $P_{N,\lambda}$ is such that

$$\overline{P_{N,\lambda}} = \operatorname{argmin} \sum_{i=1}^{N} \lambda_i W_2^2(P, P_i) \tag{9}$$

where $\lambda \in \Delta N$ denote weights associated with P_1, P_2, ..., P_N. When P_1, P_2, ..., P_N are discrete measures with finite number of atoms and the weights λ are uniform, it was shown by Gramfort et al. [50] that the problem of finding Wasserstein barycenter $\overline{P_{N,\lambda}}$ over the space $P_2(\Theta)$ is reduced to search only over a much simpler space $O_l(\Theta)$, where $l = \sum_{i=1}^{N} s_i - N + 1$ and s_i is the number of components of P_i or all $1 \leq \mathbf{i} \leq N$.

Efficient algorithms for finding local solutions of the Wasserstein barycenter problem over $O_k(\Theta)$ for some $k \geq 1$ have been studied recently in Cuturi and Doucet [48].

3.2.3. eXtreme gradient boosting

eXtreme gradient boosting, proposed by Dr. Chen in 2016, is a large-scale machine learning method for tree boosting and the optimization of gradient boosting decision tree (GBDT). As a lot of researches have mentioned, GBDT is an ensemble learning algorithm, which aims to achieve accurate classifications by combining a number of iterative computation of weak classifiers (such as decision trees). However, unlike GBDT, XGB can take advantage of multi-threaded parallel computing by using central processing unit (CPU) automatically to shorten the process of iteration. Besides, additional regularization terms help decrease the complexity of the model.

In Supervised Learning, there are objective function as well as predictive function. In XGB, objective function in Equation (10) consists of training loss $L(\phi)$ which measures whether model is fit on training data and regularization $\Omega(\phi)$ which measures complexity of model. If there is no regularization or regularization parameter is zero, the model returns to the traditional gradient tree boosting.

$$\mathrm{Obj}(\phi) = L(\phi) + \Omega(\phi) \tag{10}$$

When it comes to $L(\phi)$:

$$L(\phi) = \sum_{i=1}^{n} (y_i, \dot{y}_i) \tag{11}$$

where y_i denotes the true value and $\dot{y}_i$ denotes the predicted value. If a model after an iteration is:

$$\dot{y}_l = \sum_{m=1}^{M} f_m(x_i), f_m \in A \tag{12}$$

Then the corresponding objective function is:

$$\mathrm{Obj}(\phi) = \sum_{i=1}^{n} l(y_i, \dot{y}_i) + \sum_{m=1}^{M} \Omega(f_m) \tag{13}$$

And the model after t times iteration:

$$\begin{aligned} \mathrm{Obj}^{(t)} &= \sum_{i}^{n} I(y_i, \dot{y}_l) + \sum_{i=1}^{1} \Omega(f_i) \\ &= \sum_{i=1}^{n} l\left(y_i, \dot{y}_l^{(t-1)} + f_t(x_i)\right) + \sum_{i=1}^{t} \Omega(f_i) \end{aligned} \tag{14}$$

where $f_t(x_i)$ is a predictable function newly added in the t times iteration.

The formula of second-order Taylor expansion is:

$$f(x + \Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 \tag{15}$$

using the second-order Taylor expansion of $l\left(y_i, \dot{y}_i^{(t-1)} + f_t(x_i)\right)$:

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} \left[(y_i, \dot{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \sum_{i=1}^{t} \Omega(f_i) \tag{16}$$

In this formula, $g_i = \frac{\partial l(y_i, \dot{y}_i^{(i-1)})}{\partial \dot{y}_l^{(t-1)}}$ is the first-order derivative of $l(y_i, \dot{y}_l^{(t-1)})$. And $h_i = \frac{\partial^2 l(y_i, \dot{y}_l^{(t-1)})}{(\partial \dot{y}_l^{(t-1)})^2}$ is the second-order derivative of $l(y_i, \dot{y}_l^{(t-1)})$.

When it comes to $\Omega(f)$:

$$f_t(x_i) = \omega_{q(x_i)} \tag{17}$$

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2 \tag{18}$$

In Equation (17), $q_{(x_i)}$ structure function, which describes the structure of a decision tree ω is the weight of the leaves on the tree. Equation (18) describes the complexity of a tree. γ is a coefficient of leaf nodes, taking pre-processing to prune leaves while optimizing the objective function. λ is another coefficient to prevent the model from over-fitting.

Define $P_i = \{i | q(x_i) = j\}$ as the sample set for each leaf j. Then the objective function can be simplified as:

$$\mathrm{Obj}^{(t)} = \gamma T + \sum_{j=1}^{T} \left[\left(\sum_{i \in P_j} g_i \right) \omega_j + \frac{1}{2} \left(\sum_{i \in P_j} h_i + \lambda \right) \omega_j^2 \right] \tag{19}$$

When structures of trees q are known, this equation has solutions:

$$\omega_j^* = -\frac{\sum_{i\in P_j} g_i}{\sum_{i\in P_j} h_i + \lambda} \tag{20}$$

$$\text{Obj}^* = \gamma T + \frac{1}{2}\sum_{j=1}^{T}\left(\frac{\sum_{i\in P_j} g_i}{\sum_{i\in P_j} h_i + \lambda}\right)^2 \tag{21}$$

Blessed with traits mentioned above, XGB has the following advantages compared to traditional methods:

(i) Avoiding over-fitting. According to Biasvariance trade-off, the regularization term simplifies the model. Simpler models tends to have smaller variance, thus avoiding overfitting as well as improving accuracy of the solution.

(ii) Supporting for parallelism. Before training, XGB sorts the data in advance, and saves it as a block structure. When splitting nodes, we can calculate the greatest gain of each feature with multi-threading by using this block structure.

(iii) Flexibility. XGB supports user-defined objective function and evaluation function as long as the objective function is second-order derivable.

(iv) Built-in cross validation. XGB allows cross validation in each round of iterations. Therefore, the optimal number of iterations can be easily obtained.

(v) Process of missing feature values. For a sample with missing feature values, XGB can automatically learn its splitting direction.

3.2.4. Back propagation neural network

Back propagation neural network is a multi-layer feedforward neural network trained by error back propagation learning algorithm. It was firstly coined by Rumelhart and McClelland in 1986. Blessed with strong ability of nonlinear mapping, generalization and fault tolerance, it has become one of the most widely used neural network models. The core of BPNN mainly includes two parts: the forward propagation of signals as well as the reverse propagation of errors. In the former, input signals as input cells activate the cells of hidden layer and transfer information to them with weights. The hidden layer also acts on the output layer in this way, thus finally getting the output results. If those results are not fit on the expected output results, it turns to the latter process. The output layer error will be back-propagated layer by layer. The weights of the network are adjusted at the same time to make the output of the forward propagation process closer to the ideal output.

- Forward propagation

First, we need to introduce activation functions. It helps to solve complex non-linear problems. The widely used activation function is the *sigmoid* function:

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{22}$$

$$\text{sigmoid}'(x) = \text{sigmoid}(x)[1-\text{sigmoid}(x)] \tag{23}$$

In input layer, the input and output of the ith cell are the same. And the number of input cells is $n1$.

$$O_i = I_i \tag{24}$$

In hidden layer, the input and output of the ith cell are as follows. And the number of input cells is $n2$.

$$I_j = \sum_{i=1}^{n1}\omega_{ji}O_i + \delta_j \tag{25}$$

$$O_j = \text{sigmoid}(I_j) \tag{26}$$

ω_{ji} is the weight connecting the ith input cell and the ith hidden cell. δ_j represents the thresholds of the jth hidden cell.

In output layer, the input and output of the kth cell are as follows:

$$I_j = \sum_{i=1}^{n1}\omega_{kj}O_j + \delta_k \tag{27}$$

$$O_k = \text{sigmoid}(I_k) \tag{28}$$

ω_{kj} is the weight connecting the jth hidden cell and the output cell. δ_k represents the thresholds of the output cell.

- Back propagation

We define the expected output as $\widehat{O}$ and the number of the output cells is $n3$. After training, the total error is:

$$E = \frac{1}{2}\sum_{k=1}^{n3}(\widehat{O}_k - O_k)^2 \tag{29}$$

According to the chain rule, we can adjust the weights.

$$\frac{\partial E}{\partial \omega_{\mu}} = \frac{\partial E}{\partial O_k}\cdot\frac{\partial O_k}{\partial I_k}\cdot\frac{\partial I_k}{\partial \omega_{kj}} = (\widehat{O}_k - O_k)\cdot O_k\cdot(1-O_k)\cdot O_j \tag{30}$$

Similarly, other weights can also be adjusted in this way.

4. EXPERIMENT

4.1. Experiment of Music Transcription

4.1.1. Database and data preprocessing

In this section, we start to describe training data and experimental settings, and then conduct the state-of-the-art method to merge different transcription results. In this experiment, we use anaconda3 and python3.5 to perform the transcription, and sklearn toolbox to deal with data; while adopted pycharm to merge the data of different transcription results.

In data preparation period, the instrumental sound records in studio were described as dry source; however, most of scenes were not ideal. For a large amount of ground noise would be added to dry source during recording due to the sound card device or background. What's more, some instrumental sound was recorded in different scenes and added different noises. We chose three classical music pieces by Bach, Mozart and Beethoven and preprocessed them with filter noise, distortion noise, reverb noise and dynamic noise.

4.1.2. Experimental settings

In this paper, we first propose a method based on NMF. Then we employed a fresh and simple Time-frequency representation, using the effectiveness of spectral features when highlighting the start time of notes. In addition, we adopted the NMF model to input the proposed features. In our system, we used different audio signals recorded in different scenes with a sample rate of 48 kHz. We split the frame with a hamming window of 8192 samples and a jump size of 1764 samples. The 16,384-point DFT was calculated on every frame via double zero padding. Smoothing the spectrum through a median filter covered 100 ms. The algorithm is updated and iterated 50 times. Each row of the transcription results showed: onset time, offset time, notations of MIDI are as follows in Figure 1.

4.2. Experiment of Music Classification

4.2.1. Database and data preprocessing

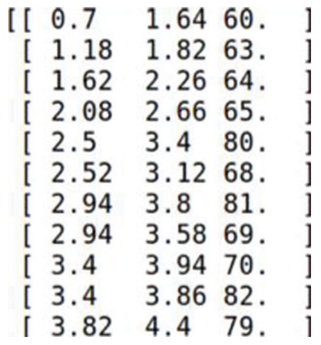
As for data sets, we selected classical music from five composers. They are Haydn, Mozart, Beethoven, Bach and Schubert. We get 200 pieces of music from each composer, a total of 1000 pieces. We use 90% of each composer's data as a training set and the remaining 10% is used as testing set. Music pieces are all in MIDI format.

4.2.2. Experimental settings

In this experiment, we use python 3, an interpretative scripting language. We mainly use sklearn toolbox to deal with data, which is simple but efficient tools for data mining and data analysis. It is not only accessible to beginners but also reusable in various contexts. Matplotlib toolbox is used to draw the receiver operating characteristic curve (ROC curve).

4.2.3. Features extraction

In this paper, we use jSymbolic as an open-source platform for extracting features from symbolic music. These features can serve as inputs to machine learning algorithms, or they can be analyzed statistically to derive musicological insights. jSymbolic implements 246 unique features, comprising 1497 different values, making it by far the most extensive symbolic feature extractor to date. These features are designed to be applicable to a diverse range of music, and may be extracted from both symbolic music files as a whole and from windowed subsets of them.



```
[[ 0.7   1.64 60.  ]
 [ 1.18  1.82 63.  ]
 [ 1.62  2.26 64.  ]
 [ 2.08  2.66 65.  ]
 [ 2.5   3.4  80.  ]
 [ 2.52  3.12 68.  ]
 [ 2.94  3.8  81.  ]
 [ 2.94  3.58 69.  ]
 [ 3.4   3.94 70.  ]
 [ 3.4   3.86 82.  ]
 [ 3.82  4.4  79.  ]
```

Figure 1 | The transcription result.

Features extracted with jSymbolic can be roughly divided into eight categories, which are: range, repeated notes, vertical perfect fourths, rhythmic variability, parallel motion, vertical tritones, chord duration, number of pitches. And the following is a brief introduction of some of them.

- Pitch Statistics: How common are various pitches and pitch classes relative to one another? How are they distributed and how much do they vary?
- Chords and Vertical Intervals: What vertical intervals are present? What types of chords do they represent? What kinds of harmonic movement are present?
- Rhythm: Information associated with note attacks, durations and rests, measured in ways that are both dependent and independent of tempo. Information on rhythmic variability, including rubato, and meter.

5. RESULTS AND COMPARISON

5.1. Comparison of Music Transcription

5.1.1. Ensemble and comparison

First, we examined data sets in four scenes (adding filter noise, distortion noise, reverb noise and dynamic noise) under which we could get reasonable clusters. Then, we used the Wasserstein Barycenter algorithm as our ensemble method to obtain results. For example, we put forward the transcription data with reverb noises before ensemble. While, we generated the 10 transcription data adding with different reverb noises and then merged them through Wasserstein means algorithm.

We show that ensemble method is more robust than single transcription method in four scenes through Proportional Transportation Distance (PTD). The experimental results are evaluated objectively by using PTD described above. The PTD is computed by first dividing each point's weight by the total weight of its point set, and then the EMD of resulting point sets is calculated [52]. According to the EMD and PTD method, we present notation as sets of weighted points. The weight represents note duration. Each note stands for a point distributed in the x and y coordinates, representing the start time and pitch, respectively. We use the Euclidean distance as the ground distance.

5.1.2. Evaluation

We conducted the evaluation by calculating precision ($P = N_{tp}/(N_{tp} + N_{fp})$), recall ($R = N_{tp}/(N_{tp} + N_{fn})$), F-measure ($F = 2PR/(P + R)$) and accuracy ($A = N_{tp}/(N_{tp} + N_{fp} + N_{fn})$), where N_{tp}, N_{fp} and N_{fn} are the numbers of true positives, false positives and false negatives respectively. If the pitch is correct and its starting time is within 50 ms of the ground truth, we computed the notes as true positives [53].

The results are shown in Table 1. First of all, we averaged precision, recall, F-measure and accuracy of unmerged data from three

Table 1 | Performance comparison on the real data set

	Precision	Recall	*F*-measure	Accuracy
Filter	0.4321	0.6667	0.3766	0.3232
Reverb	0.4405	0.6829	0.3841	0.3927
Dynamic	0.4272	0.6977	0.3385	0.3431
Distortion	0.4137	0.6914	0.3278	0.3703
Ensemble	0.6241	0.9231	0.7371	0.6642

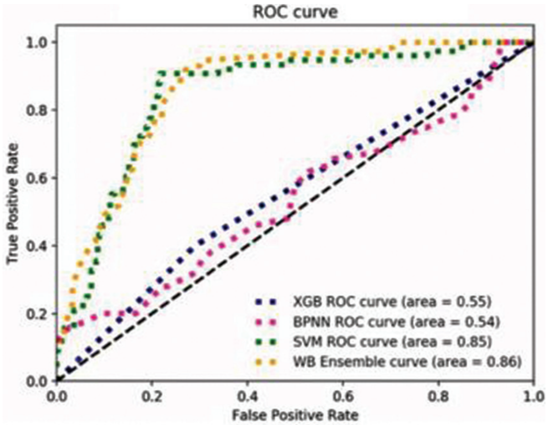


Figure 2 | ROC curves for the best performing models and their ensemble.

composers in four scenes. Then, we compared the values of them in four scenes with those of merged data. It can be seen that the ensemble method is better than single transcription method in four scenes and the rates of precision, recall, *F*-measure and accuracy are obviously higher than those of unmerged data. It has increased nearly two times in *F*-measure and accuracy and 1.5 times in precision and recall.

5.2. Comparison of Music Classification

5.2.1. Ensemble and comparison

In this section, we combine different classifiers including SVM, XGB and BPNN to the ensemble classifier called Wasserstein Barycenter ensemble, which is based on Wasserstein Barycenter described above. Such an ensemble scheme which combines the prediction powers of different classifiers makes the overall system more robust. In our case, each classifier outputs a prediction probability for each of the classification labels. Hence averaging the predicted probabilities from the different classifiers would be a straightforward way to do ensemble learning.

The methodologies described in Section 3.2 that introduce different models of classification and the Wasserstein Barycenter algorithm, which makes sense to combine the models via ensemble. The performance of SVM, XGB, BPNN and WB ensemble is shown in Figure 2. ROC curve of BPNN has more twists and turns, compared with that of XGB, and ROC curve of WB ensemble is more smooth than that of SVM. The ROC curve for the WB ensemble model is above that of SVM, XGB and BPNN as illustrated in Figure 2. As shown in Table 2, this WB ensemble is beneficial and is observed to outperform the all individual classifiers.

Table 2 | Performance comparison on classifiers and their ensemble

Classifiers	Accuracy	Precision	Recall	*F*-measure	AUC
SVM	0.46	0.43	0.45	0.4	0.85
XGB	0.63	0.31	0.4	0.35	0.55
BPNN	0.43	0.18	0.26	0.22	0.54
WB ensemble	0.65	0.44	0.47	0.44	0.86

5.2.2. Evaluation

In order to evaluate the performance of the models described in Section 3.2, we compute precision, recall, *F*-measure, accuracy and area under curve (AUC) as the evaluation metrics of these classifiers. Experiment results can be seen from Table 2. The best performance in terms of all metrics is observed for ensemble model based on Wasserstein Barycenter. As we can see, WB ensemble achieves a classification effect better than other classifiers in each evaluation metric. There is no notable difference in AUC between XGB and BPNN, with 0.55 and 0.54 representatively. In contrast, SVM and WB ensemble reach to 0.85 and 0.86, higher than XGB and BPNN. In terms of accuracy, XGB has achieved 63%, nearly the same accuracy with WB ensemble, which has reached to 65%. However, in other evaluation metrics, SVM obviously performs better than XGB and BPNN.

Among the models that use manually crafted features, the one with the least performance is the BPNN model. This is expected since BPNN mainly deal with the classification of big data while our datasets are very small. SVMs outperform random forests in terms of AUC. However, the XGB version of the gradient boosting algorithm performs the best among the feature engineered methods with the relatively high accuracy.

6. CONCLUSION

In this paper we showed that Wasserstein Barycenter is effective in multiple scenes ensemble in music transcription and multiple classifiers ensemble in music classification. Here, we proposed two applications of Wasserstein Barycenter ensemble. For music transcription, in different scenes and pieces of music, we presented their effectiveness in ensemble results, as well as in improving the robustness and accuracy of music transcriptions. In addition, for music classification, we compare the performance of single classifiers and ensemble model for music style classification with different composers. Then, we also conduct an objective evaluation to compare diverse scenes and diverse models through measuring the differences between music signal data and the ground truth scores. Finally, we drew a conclusion that our crowdsourcing method is very useful in improving the robustness and accuracy of music signal processing results.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

FUNDING

This research was supported by National Key R&D Program of China (Grant No. 2018YFB1403900), the National Natural Science

Foundation of China No. 61901421 and "the Fundamental Research Funds for the Central Universities" No. 2019E002, CUC200B017 and CUC19ZD003.

REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, S. Ewert, Automatic music transcription: an overview, IEEE Signal Process. Mag. 36 (2018), 20–30.

[2] H. Bahuleyan, Music genre classification using machine learning techniques, arXiv preprint arXiv:1804.01149, 2018.

[3] C. Villani, Optimal Transport: Old and New, vol. 338, Springer Science & Business Media, Berlin Heidelberg, 2008.

[4] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vis. 40 (2000), 99–121.

[5] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, et al., Convolutional Wasserstein distances: efficient optimal transportation on geometric domains, ACM Trans. Graph. 34 (2015), 1–11.

[6] M. Cuturi, Sinkhorn distances: lightspeed computation of optimal transport, Proceedings of the 26th International Conference on Neural Information Processing Systems, ACM, New York, NY, USA, 2013, pp. 2292–2300.

[7] J. Ye, P. Wu, J.Z. Wang, J. Li, Fast discrete distribution clustering using Wasserstein Barycenter with sparse support, IEEE Trans. Signal Process. 65 (2017), 2317–2332.

[8] J.D. Benamou, G. Carlier, M. Cuturi, L. Nenna, G. Peyré, Iterative Bregman projections for regularized transportation problems, SIAM J. Sci. Comput. 37 (2015), A1111–A1138.

[9] S. Basu, S. Kolouri, G.K. Rohde, Detecting and visualizing cell phenotype differences from microscopy images using transport-based morphometry, Proc. Natl. Acad. Sci. 111 (2014), 3448–3453.

[10] W. Wang, D. Slepčev, S. Basu, J.A. Ozolek, G.K. Rohde, A linear optimal transportation framework for quantifying and visualizing variations in sets of images, Int. J. Comput. Vis. 101 (2013), 254–269.

[11] V. Seguy, M. Cuturi, Principal geodesic analysis for probability measures under the optimal transport metric, Proceedings of the 28th International Conference on Neural Information Processing Systems, ACM, New York, NY, USA, 2015, pp. 3312–3320.

[12] J. Bigot, R. Gouet, T. Klein, A. López, Geodesic PCA in the Wasserstein space by convex PCA, Ann. Inst. H. Poincaré Probab. Statist. 53 (2017), 1–26.

[13] M. Agueh, G. Carlier, Barycenters in the Wasserstein space, SIAM J. Math. Anal. 43 (2011), 904–924.

[14] J. Rabin, G. Peyré, J. Delon, M. Bernot, Wasserstein barycenter and its application to texture mixing, International Conference on Scale Space and Variational Methods in Computer Vision, Springer, Berlin, Heidelberg, 2011, pp. 435–446.

[15] J.A. Ozolek, A.B. Tosun, W. Wang, C. Chen, S. Kolouri, S. Basu, et al., Accurate diagnosis of thyroid follicular lesions from nuclear morphology using supervised learning, Med. Image. Anal. 18 (2014), 772–780.

[16] A.B. Tosun, O. Yergiyev, S. Kolouri, J.F. Silverman, G.K. Rohde, Detection of malignant mesothelioma using nuclear structure of mesothelial cells in effusion cytology specimens, Cytometry A 87 (2015), 326–333.

[17] S. Kolouri, G.K. Rohde, Transport-based single frame super resolution of very low resolution face images, 2015 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Boston, MA, USA, 2015, pp. 4876–4884.

[18] S. Haker, L. Zhu, A. Tannenbaum, S. Angenent, Optimal mass transport for registration and warping, Int. J. Comput. Vis. 60 (2004), 225–240.

[19] R. Chartrand, B. Wohlberg, K.R. Vixie, E.M. Bollt, A gradient descent solution to the Monge-Kantorovich problem, Appl. Math. Sci. 3 (2009), 1071–1080.

[20] W. Wang, J.A. Ozolek, D. Slepčev, A.B. Lee, C. Chen, G.K. Rohde, An optimal transportation approach for nuclear structure-based pathology, IEEE Trans. Med. Imaging 30 (2010), 621–631.

[21] Q. Mérigot, A multiscale approach to optimal transport, Comput. Graph Forum. 30 (2011), 1583–1592.

[22] A.M. Oberman, Y. Ruan, An efficient linear programming method for optimal transportation, arXiv preprint arXiv:1509.03668, 2015.

[23] N. Bonneel, J. Rabin, G. Peyré, H. Pfister, Sliced and radon Wasserstein barycenters of measures, J. Math. Imaging Vis. 51 (2015), 22–45.

[24] S.R. Park, S. Kolouri, S. Kundu, G.K. Rohde, The cumulative distribution transform and linear pattern classification, Appl. Comput. Harm. Anal. 45 (2018), 616–641.

[25] S. Kolouri, S.R. Park, G.K. Rohde, The radon cumulative distribution transform and its application to image classification, IEEE Trans. Image Process. 25 (2015), 920–934.

[26] S. Kolouri, A.B. Tosun, J.A. Ozolek, G.K. Rohde, A continuous linear optimal transport approach for pattern analysis in image datasets, Pattern Recognit. 51 (2016), 453–462.

[27] C. Yeh, A. Roebel, X. Rodet, Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals, IEEE Trans. on Speech and Audio Process. 18 (2009), 1116–1126.

[28] J. Nam, J. Ngiam, H. Lee, M. Slaney, A classification-based polyphonic piano transcription approach using learned feature representations, Proceedings of the 12th International Society for Music Information Retrieval Conference, University of Miami, Miami, Florida, USA, 2011, pp. 175–180.

[29] Z. Duan, B. Pardo, C. Zhang, Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions, IEEE Trans. Audio Speech Language Process. 18 (2010), 2121–2133.

[30] V. Emiya, R. Badeau, B. David, Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle, IEEE Trans. Audio Speech Language Process. 18 (2009), 1643–1654.

[31] P.H. Peeling, S.J. Godsill, Multiple pitch estimation using non-homogeneous poisson processes, IEEE J. Select. Top. Signal. Process. 5 (2011), 1133–1143.

[32] E. Vincent, N. Bertin, R. Badeau, Adaptive harmonic spectral decomposition for multiple pitch estimation, IEEE Trans. Audio Speech Language Process. 18 (2009), 528–537.

[33] N. Bertin, R. Badeau, E. Vincent, Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription, IEEE Trans. Audio Speech Language Process. 18 (2010), 538–549.

[34] B. Fuentes, R. Badeau, G. Richard, Adaptive harmonic time-frequency decomposition of audio using shift-invariant PLCA, 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, Prague, Czech Republic, 2011, pp. 401–404.

[35] S.A. Abdallah, M.D. Plumbley, Polyphonic music transcription by non-negative sparse coding of power spectra, Proceedings of the 5th International Conference on Music Information Retrieval, Barcelona, Spain, 2004, pp. 318–325.

[36] G. Tzanetakis, P. Cook, Musical genre classification of audio signals, IEEE Trans. Speech audio Process. 10 (2002), 293–302.

[37] Y. Lee, Y. Lin, G. Wahba, Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data, J. Am. Stat. Assoc. 99 (2004), 67–81.

[38] O. Pele, M. Werman, Fast and robust earth mover's distances, 2009 IEEE 12th International Conference on Computer Vision, IEEE, Kyoto, Japan, 2009, pp. 460–467.

[39] M. Hussain, M.A. Haque, SwishNet: a fast convolutional neural network for speech, music and noise classification and segmentation, arXiv preprint arXiv:1812.00149, 2018.

[40] S.R. Azar, A. Ahmadi, S. Malekzadeh, M. Samami, Instrument-independent Dastgah recognition of Iranian classical music using Azarnet, arXiv preprint arXiv:1812.07017, 2018.

[41] C. Liu, L. Feng, G. Liu, H. Wang, S. Liu, Bottom-up broadcast neural network for music genre classification, arXiv preprint arXiv:1901.08928, 2019.

[42] Z. Nasrullah, Y. Zhao, Music artist classification with convolutional recurrent neural networks, 2019 International Joint Conference on Neural Networks, IEEE, Budapest, Hungary, 2019, pp. 1–8.

[43] H. Ding, M. Liu, On geometric prototype and applications, arXiv preprint arXiv:1804.09655, 2018.

[44] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, K. Weihe, Network flows: theory, algorithms and applications, ZOR – Meth. Models Oper. Res. 41 (1995), 252–254.

[45] P.K. Agarwal, K. Fox, D. Panigrahi, K.R. Varadarajan, A. Xiao, Faster algorithms for the geometric transportation problem, arXiv preprint arXiv:1903.08263, 2019.

[46] S. Cabello, P. Giannopoulos, C. Knauer, G. Rote, Matching point sets with respect to the earth mover's distance, Comput. Geom. 39 (2008), 118–133.

[47] P. Indyk, A near linear time constant factor approximation for euclidean bichromatic matching (cost), Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, NY, USA, 2007, pp. 39–42.

[48] M. Cuturi, A. Doucet, Fast computation of Wasserstein barycenters, Proceedings of the 31st International Conference on Machine Learning, Proc. Mach. Learn. Res. 32 (2014), 685–693.

[49] M. Baum, P.K. Willett, U.D. Hanebeck, On Wasserstein barycenters and MMOSPA estimation, IEEE Signal Process. Lett. 22 (2015), 1511–1515.

[50] A. Gramfort, G. Peyré, M. Cuturi, Fast optimal transport averaging of neuroimaging data, International Conference on Information Processing in Medical Imaging, Springer, Cham, 2015, pp. 261–272.

[51] H. Ding, R. Berezney, J. Xu, k-Prototype learning for 3D rigid structures, Adv. Neural Inf. Process. Syst. 2013, 2589–2597.

[52] R. Typke, R.C. Veltkamp, F. Wiering, Searching notated polyphonic music using transportation distances, Proceedings of the 12th annual ACM international conference on Multimedia, ACM, New York, NY, USA, 2004, pp. 128–135.

[53] L. Gao, L. Su, Y.H. Yang, T. Lee, Polyphonic piano note transcription with non-negative matrix factorization of differential spectrogram, 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, New Orleans, LA, USA, 2017, pp. 291–295.