An Integrated Processing Method Based on Wasserstein Barycenter Algorithm for Automatic Music Transcription

Cong Jin[1], Zhongtong Li[1], Yuanyuan Sun[1], Haiyin Zhang[2], Xin Lv[3(✉)], Jianguang Li[4], and Shouxun Liu[4]

[1] School of Information and Communication Engineering, Communication University of China, Beijing 100024, China
{jincong0623,lizhongtong}@cuc.edu.cn
[2] School of Computer and Cyberspace Security, Communication University of China, Beijing 100024, China
hynn0633@outlook.com
[3] School of Animation and Digital Arts, Communication University of China, Beijing 100024, China
lvxincuc@163.com
[4] Communication University of China, Beijing 100024, China
{lijianguang,sxliu}@cuc.edu.cn

Abstract. Given a piece of acoustic musical signal, various automatic music transcription (AMT) processing methods have been proposed to generate the corresponding music notations without human intervention. However, the existing AMT methods based on signal processing or machine learning cannot perfectly restore the original music signal and have significant distortion. In this paper, we propose a novel processing method which integrates various AMT methods so as to achieve better performance on music transcription. This integrated method is based on the entropic regularized Wasserstein Barycenter algorithm to speed up the computation of the Wasserstein distance and minimize the distance between two discrete distributions. Moreover, we introduce the proportional transportation distance (PTD) to evaluate the performance of different methods. Experimental results show that the precision and accuracy of the proposed method increase by approximately 48% and 67% respectively compared with the existing methods.

Keywords: Automatic Music Transcription · Machine learning · Wasserstein Barycenter · Ensemble · NMF

Supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61631016, National Key Research and Development Plan of Ministry of Science and Technology No. 2018YFB1403903 and the Fundamental Research Funds for the Central Universities No. CUC2019E002, CUC19ZD003.

H. Gao et al. (Eds.): ChinaCom 2019, LNICST 313, pp. 230–240, 2020.
https://doi.org/10.1007/978-3-030-41117-6_19

1 Introduction

Famous audio researchers Moore [1], Pitztalski and Galler [2] proposed the term "Automatic Music Transcription" (AMT) firstly in 1977. These audio researchers believed that by programming computers, they can manage to analyze digital records of music, so that they could detect the pitch of melodies and chord patterns as well as the rhythm of percussion instruments. In music transcription system, a musical acoustic signal can be transformed to the format of music notation like a MIDI file [3]. As a basic problem in Music Information Retrieval (MIR), a complete AMT system would resolve the pitch, timing, and instrument of the sound events.

Various research groups of polyphonic pitch detection used different techniques for music transcriptions. Yeh [4] presented a cross pitch estimation algorithm based on the score function of a pitch candidate set. Nam et al. [5] posed a transcription approach which uses deep belief networks to calculate a mid-level time-pitch representation. Duan et al. [6] and Emiya et al. [7] proposed a model of spectral peak, non-peak region and the residual noise via Maximum Likelihood (ML) Methods. More recently, Peeling and Godsill [8] 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. [9] merged harmonic constraints in the NMF model. Bertin et al. [10] presented a Bayesian model based on NMF, and each pitch in harmonic positions is treated as a model of Gaussian components. Fuentes et al. [11] modeled each note as a weighted amount of narrowband log spectrum, and switched to log frequency with the convoluted PLCA algorithm. Abdallah and Plumbley [12] combined machine learning and dictionary learning via non-negative sparse coding.

In this paper, we propose a converged method based on Earth Mover's Distance and Wasserstein Barycenter, to compare our experimental result of music transcription with the ground truth. In Sect. 2, we introduce the algorithm of Earth Mover's Distance and Wasserstein Barycenter. In Sect. 3, we present an experiment including data preparation, music transcription with NMF, data trimming, merging and evaluation. At last, we conclude that our crowdsourcing method improves the robustness and accuracy of transcription result.

2 Algorithm

Our idea of music transcription ensemble is inspired by the recent study on Earth Mover's Distance and Wasserstein Barycenter in the area of machine learning. Here, we introduce their formal definitions first.

Definition 1 (Earth Mover's Distance (EMD) [13,14]). *Let* $X = \{x_1, x_2, \cdots, x_{n_1}\}$ *and* $Y = \{y_1, y_2, \cdots, y_{n_2}\}$ *be two sets of weighted points in* $\mathbb{R}^d$ *with non-negative weights* α_i *and* β_j *for each* $x_i \in X$ *and* $y_j \in Y$ *respectively, and* W_X *and* W_Y *be their corresponding total weights. The Earth Mover's Distance between* X *and* Y *is* $\mathcal{EMD}(X, Y)$

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

where $F = \{f_{ij}\}$ *is a feasible flow from* X *to* Y*, i.e., 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 Euclidean space $\mathbb{R}^d$. Therefore, the problem of computing EMD can be solved by linear programming [15]. In addition, several faster algorithms have been proposed by using the techniques developed in computational geometry [16–18]. Following EMD, we have the definition of Wasserstein Barycenter.

Definition 2 (Wasserstein Barycenter (WB) [19]**).** *Given a set of point sets* $X_1, X_2, \cdots, X_k \subset \mathbb{R}^d$*, where each* X_j *has the same total weight, the problem of Wasserstein Barycenter is to build a new point set* Q*, such that the total EMDs* $\sum_{j=1}^{k} \mathcal{EMD}(X_j, Q)$ *is minimized.*

Intuitively, the WB Q can be treated as the representation of all the given patterns $X_1, X_2, \cdots, X_k$. As mentioned in [14], WB has extensive applications in practical areas. For example, it can be applied to compute the average of a large set of images, so as to obtain a robust pattern or compress the image dataset. Prior works include [19–25]. Recently, researchers also use WB to handle Bayesian inference problem [26].

In theory, Ding and Liu have systematically studied the problem of WB (they call it as geometric prototype in the paper) [14]. Given an instance of geometric prototype problem, they show that a small core-set, which is independent of any geometric prototype algorithm, can be effectively computed. That is, one can achieve a similar result via running any available black box algorithm on the core-set. The benefit of computing the core-set is that the data size can be significantly reduced and thus the existing algorithms can run much faster. The reader can find more details about core-set in [27,28].

In this paper, we adopt the method from [22] for computing WB. For the sake of completeness, we briefly describe their algorithm below.

In Ye's paper [22], they have developed an improved Bregman ADMM (B-ADMM) method to optimize the centroid of big clusters. With the calculation of centroid distribution, clustering has a serious expandability problem, and they introduced the Wasserstein barycenter algorithm, which can calculate the sum of least squared distances with cluster members.

Suppose a set of discrete distributions $\{Q^{(1)}, ..., Q^{(N)}\}$, N stands for the size of a Wasserstein barycenter's computation. They intend to get a centroid Q: $\{(\omega_1, x_1), ..., (\omega_m, x_m)\}$, such that

$$\min_Q \frac{1}{N} \sum\nolimits_{n=1}^{N} W^2(Q, Q^{(n)}) \tag{2}$$

where includes the weights of the centroids $\{\omega_i \in \mathbb{R}+\}$, the supporting points $\{x_i \in \mathbb{R}^d\}$, and the optimum coupling between Q and $Q^{(n)}$ for each n, expressed as $\{\pi_{i,j}^{(n)}\}$.

Clustering in B-ADMM method optimizes $\{\omega_i\}$ and $\{\pi_{i,j}^{(n)}\}$ in turn, n = $1, 2, \ldots, \mathrm{N}$, versus $\{x_i\}$. Δ_n is defined a Probabilistic simplex of n dimensions. To solve the optimal transport issue, they have introduced two sets of variables $\pi_{(n,1)} = (\pi_{i,j}^{(n,1)}), i \in L', j \in L_n$, and $\pi_{(n,2)} = (\pi_{i,j}^{(n,2)}), i \in L', j \in L_n$, for $n = 1, 2, \cdots, N$ the constraints as follows. Let

$$\Delta_{n,1} := \{\pi_{i,j}^{(n,1)} \geqslant 0 : \sum\nolimits_{i=1}^{m} \pi_{i,j}^{(n,1)} = \omega_j^{(n)}, j \in L_n\} \tag{3}$$

$$\Delta_{n,2}(\omega) := \{\pi_{i,j}^{(n,2)} \geqslant 0 : \sum\nolimits_{j=1}^{m} \pi_{i,j}^{(n,2)} = \omega_i, i \in L'\} \tag{4}$$

then $\pi^{(n,1)} \in \Delta_{n,1}$ and $\pi^{(n,2)} \in \Delta_{n,2}(\omega)$.

3 Experiment

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 employ 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.

3.1 Data Preparation in Different Scenes

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 noises would be added to dry source during recording due to the sound card device or background. What's more, some instrumental sounds were 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.

3.2 Experimental Settings and Transcription

In this paper, we first proposed a method based on non-negative matrix factorization. Non-negative matrix factorization (NMF) algorithm is utilized as a tool for music transcription [29]. The NMF model in its simplest form decomposes an input spectrogram $A \in \mathbb{R}_+^{X \times Y}$ with X frequency bins and Y frames as:

$$\boldsymbol{A \approx FT} \tag{5}$$

where R $\ll$ X,Y; $F \in \mathbb{R}_{+}^{X \times Y}$ contains the spectral cardinality of each R tone component; and $T \in \mathbb{R}_{+}^{X \times Y}$ is the matrix of pitch activity across time.

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 16384-point DFT was calculated on every frame via double zero padding. Smoothing the spectrum through a median filter covered 100 ms. The algorithms is updated and iterated 50 times. Each row of the transcription results showed: onset time, offset time, notations of Midi are as followed in Fig. 1.

```
[[ 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.  ]
```

Fig. 1. The transcription result

3.3 Data Trimming with Random Forest

When we conducted the transcription experiment, we found that the results had some differences in dimensions of matrixes and maybe some data were lost or added when transcription was performing. In order to obtain a better merging result, we applied Random Forest Regression to complete data trimming through inserting the predicted value or removing the large deviation. Random forest is an integrated algorithm of decision tree. Random forests contain multiple decision trees to reduce the risk of over-fitting.

Random forests train is a series of decision trees, so the training process is analogical. Due to the addition of random processes to the algorithm, there is a small difference among each decision tree. By combining the prediction results of each tree, the variance of the prediction is reduced and the performance on the test set is improved. Random representation:

1. At each iteration, the original data are subsampled to obtain different training data.
2. For each tree node, considering different random feature subsets as split.
3. The training process of decision making is the same as that of decision tree.

We first made an initial guess at the missing value, such as filling it with mean/median, then sorting it from small to large according to the missing rate of variable. Using Random Forest Regression to fill in the missing value of variable first, and then iterating it until the latest and final filling result no longer change (with little change). As is shown in Fig. 2, according to Random Forest Regression which we can obtain the predict value and then by comparing with true value, the result is great.

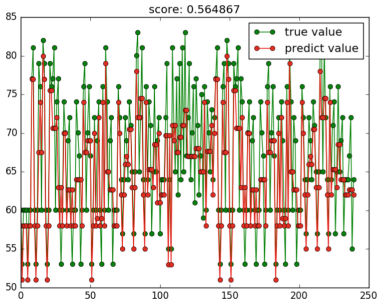

Fig. 2. Data prediction with Random Forest Regression (red line stands for predict value and green line stands for true value) (Color figure online)

3.4 Ensemble and Comparison

In this section, we used transcribed data sets adding four kinds of noises to see the properties of the ensemble method through the Wasserstein Barycenter algorithm which we described above. In ensemble experiments, we examined the conditions under which ensemble method could estimate clusters of transcription data. In comparison experiments, we compared the ensemble method with single transcription method in four scenes through Proportional Transportation Distance (PTD) to see the advantages of the ensemble method.

(i) Ensemble. Firstly, 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 employed the Wasserstein Barycenter algorithm as our ensemble method to obtain results. For example, as is shown in Fig. 3, we put forward the transcription data with reverb noises before ensemble. It can be seen that there is a large gap between unmerged data and raw data.

While, we generated the 10 transcription data adding with different reverb noises and then merged them through Wasserstein means algorithm. The comparison between merged data and raw data is shown in Fig. 4.

(ii) Comparison. 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

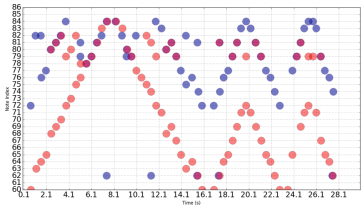

Fig. 3. The raw data and unmerged data with reverb noises. (The blue dots represent the raw data) (Color figure online)

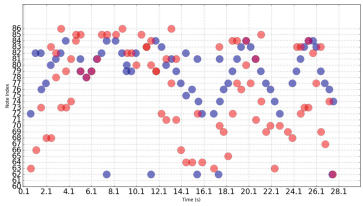

Fig. 4. The raw data and merged data with reverb noises. (The blue dots represent raw data) (Color figure online)

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 [30]. 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. Thus, the distance between two notes with the coordinates (x_i, y_i) and (x_j, y_j) is

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Then it is used to calculate the EMD of the two audio matrices. At last, we switch the EMD to PTD as the comparison of merged and unmerged data. The result is shown in Fig. 5.

We selected three classical music pieces by Bach, Mozart and Beethoven in four scenes by adding different noises including filter noise, distortion noise, reverb noise and dynamic noise. We calculate the PTD between unmerged data and ground truth (or raw data), and then between merged data and ground truth. The PTD comparison of merged and unmerged data in four scenes are shown in Tables 1, 2, 3 and 4.

From these tables we can see that the transcription has a more robust result after merging. By comparing the PTD between merged and unmerged data, we can find that the merged data are already very close to the ground truth, and the PTD of merged data decreased more than 3 times compared with unmerged data.

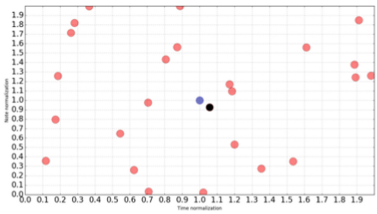

Fig. 5. The PTD Comparison. (The middle blue dot represents the raw data, the black dots represent the merged data, and other points represent the unmerged data.) (Color figure online)

Table 1. The PTD comparison in scene1 with filter noises. (Filter 1–5 are unmerged data, mix are merged data)

PTD	Filter1	Filter2	Filter3	Filter4	Filter5	Mix
Bach	4.9777	4.1040	4.9818	4.7306	3.9892	0.3624
Beethoven	3.6474	3.4409	2.9725	2.9948	2.9216	0.7622
Mozart	1.8165	2.2874	1.8099	1.7664	1.7765	0.5321

Table 2. The PTD comparison in scene2 with reverb noises. (Reverb 1–5 are unmerged data, mix are merged data)

PTD	Reverb1	Reverb2	Reverb3	Reverb4	Reverb5	Mix
Bach	1.8228	1.9929	2.4563	2.4874	3.4708	0.4978
Beethoven	3.0975	3.2011	4.3783	3.1690	2.9124	0.4137
Mozart	2.4403	2.0241	1.8956	1.8765	1.8224	0.4167

Table 3. The PTD comparison in scene3 with dynamic noises. (Dynamic 1–5 are unmerged data, mix are merged data)

PTD	Dynamic1	Dynamic2	Dynamic3	Dynamic4	Dynamic5	Mix
Bach	2.3392	3.0378	2.8648	2.6319	2.4494	0.3909
Beethoven	2.4482	2.8423	2.4436	2.4857	2.9649	0.7432
Mozart	1.9077	1.9662	1.9327	1.8483	1.8813	0.6417

Table 4. The PTD comparison in scene4 with distortion noises. (Distortion 1–5 are unmerged data, mix are merged data)

PTD	Distortion1	Distortion2	Distortion3	Distortion4	Distortion5	Mix
Bach	3.2819	3.8431	3.1133	2.8937	2.2559	0.6667
Beethoven	4.2782	4.3418	3.5581	5.0861	2.7737	0.4621
Mozart	2.6864	2.6495	2.5223	2.1681	1.9183	0.8932

3.5 Evaluation and Performance

We employed the evaluation by calculating precision ($P = \frac{N_{tp}}{N_{tp}+N_{fp}}$), recall ($R = \frac{N_{tp}}{N_{tp}+N_{fn}}$), F-measure ($F = \frac{2PR}{P+R}$) and accuracy ($A = \frac{N_{tp}}{N_{tp}+N_{fp}+N_{fn}}$), where N_{tp}, N_{fp} and N_{fn} are the values of true positives, false positives and false negatives, respectively. If the pitch is correct and its starting time is within 50ms of the ground truth, we computed the notes as true positives [31].

The results are shown in Table 5. First of all, we averaged precision, recall, F-measure and accuracy of unmerged data from three 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 2 times in F-measure and accuracy and 1.5 times in precision and recall.

Table 5. Performance comparison on the real date 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
Mix	0.6421	0.9231	0.7371	0.6642

4 Conclusion

In this paper we showed that Wasserstein Barycenter is effective in multiple scenes ensemble in machine learning. 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. We also proposed an objective evaluation to measuring the differences between music notation transcriptions in different scenes and the ground truth scores. Finally, we drew a conclusion that our crowdsourcing method is very useful in improving the robustness and accuracy of transcription results.

References

1. Moorer, J.A.: On the transcription of musical sound by computer. Comput. Music J. **1**(4), 32–38 (1977)
2. Piszczalski, M., Galler, B.A.: Automatic music transcription. Comput. Music J. **1**(4), 22–31 (1977)

3. Duan, Z., Benetos, E.: Automatic music transcription. In: Proceedings of the International Society for Music Information Retrieval Conference, Malaga, Spain (2015)
4. Chunghsin, Y.: Multiple fundamental frequency estimation of polyphonic recordings (2008)
5. Nam, J., Ngiam, J., Lee, H., Slaney, M.: A classification-based polyphonic piano transcription approach using learned feature representations (2011)
6. Duan, Z., Pardo, B., Zhang, C.: Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. IEEE Trans. Audio Speech Lang. Process. **18**(8), 2121–2133 (2010)
7. Emiya, V., Badeau, R., David, B.: Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. IEEE Trans. Audio Speech Lang. Process. **18**(6), 1643–1654 (2010)
8. Peeling, P.H., Godsill, S.J.: Multiple pitch estimation using non-homogeneous poisson processes. IEEE J. Sel. Top. Signal Process. **5**(6), 1133–1143 (2011)
9. Vincent, E., Bertin, N., Badeau, R.: Adaptive harmonic spectral decomposition for multiple pitch estimation. IEEE Trans. Audio Speech Lang. Process. **18**(3), 528–537 (2010)
10. Bertin, N., Badeau, R., Vincent, E.: Enforcing harmonicity and smoothness in Bayesian nonnegative matrix factorization applied to polyphonic music transcription. IEEE Trans. Audio Speech Lang. Process. **18**(3), 538–549 (2010)
11. Fuentes, B., Badeau, R., Richard, G.: Adaptive harmonic time-frequency decomposition of audio using shift-invariant PLCA. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 401–404 (2011)
12. Abdallah, S.M., Plumbley, M.D.: Polyphonic transcription by non-negative sparse coding of power spectra. In: Proceedings of the International Society for Music Information Retrieval Conference (2004)
13. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis. **40**(2), 99–121 (2000)
14. Ding, H., Liu, M.: On geometric prototype and applications. In: 26th Annual European Symposium on Algorithms, pp. 1–15 (2018)
15. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Saddle River (1993)
16. Agarwal, P.K., Fox, K., Panigrahi, D., Varadarajan, K.R., Xiao, A.: Faster algorithms for the geometric transportation problem. In: 33rd International Symposium on Computational Geometry, pp. 1–16 (2017)
17. Cabello, S., Giannopoulos, P., Knauer, C., Rote, G.: Matching point sets with respect to the Earth Mover's Distance. Comput. Geom. **39**(2), 118–133 (2008)
18. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
19. Cuturi, M., Doucet, A.: Fast computation of Wasserstein Barycenters. In: International Conference on Machine Learning, pp. 685–693 (2014)
20. Baum, M., Willett, P., Hanebeck, U.D.: On Wasserstein Barycenters and MMOSPA estimation. IEEE Signal Process. Lett. **22**(10), 1511–1515 (2015)
21. Gramfort, A., Peyré, G., Cuturi, M.: Fast optimal transport averaging of neuroimaging data. In: Ourselin, S., Alexander, D.C., Westin, C.-F., Cardoso, M.J. (eds.) IPMI 2015. LNCS, vol. 9123, pp. 261–272. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19992-4_20
22. Ye, J., Wu, P., Wang, J.Z., Li, J.: Fast discrete distribution clustering using Wasserstein Barycenter with sparse support. IEEE Trans. Signal Process. **65**(9), 2317–2332 (2017)

23. Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., Peyré, G.: Iterative Bregman projections for regularized transportation problems. SIAM J. Sci. Comput. **37**(2), 1111–1138 (2015)
24. Ding, H., Berezney, R., Xu, J.: k-prototype learning for 3d rigid structures. In: Advances in Neural Information Processing Systems, pp. 2589–2597 (2013)
25. Ding, H., Xu, J.: Finding median point-set using earth mover's distance. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
26. Staib, M., Claici, S., Solomon, J., Jegelka, S.: Parallel streaming Wasserstein Barycenters. In: Advances in Neural Information Processing Systems, pp. 2647–2658 (2017)
27. Phillips, J.M.: Coresets and sketches. Comput. Res. Repos. (2016)
28. Agarwal, P.K., Har-Peled, S., Varadarajan, K.R.: Geometric approximation via coresets. Comb. Comput. Geom. **52**, 1–30 (2005)
29. Smaragdis, P., Brown, J.C.: Non-negative matrix factorization for polyphonic music transcription. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 177–180 (2003)
30. Typke, R., Veltkamp, R.C., Wiering, F.: Searching notated polyphonic music using transportation distances. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 128–135 (2004)
31. Gao, L., Su, L., Yang, Y.H., Tan, L.: Polyphonic piano note transcription with non-negative matrix factorization of differential spectrogram. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 291–295 (2017)