Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Analysis and optimization of pitch change patterns in double bass performance based on weighted clustering algorithm

Chao Liu[1, ✉]

[1] *Zhengzhou Sias University, Zhengzhou, Henan, 450000, China*

ABSTRACT

The double bass, as the instrument with the lowest timbre and the largest volume in the string section of a symphony orchestra, is the “mainstay” of the orchestra's acoustic effect, and grasping the bass performance mode in double bass performance is a problem that all double bass players need to explore in depth. A cluster-weighted multi-view kernel k-means clustering model (CWK2M) is proposed to study the local quality differences of the bass performance score views at the cluster level. The proposed weighted multiview clustering algorithm is then compared with several multiview clustering algorithms on several real multiview data for experiments and analysis of pitch change patterns. The experimental results show that, on the whole, the proposed algorithm in this paper obtains a relatively good clustering effect on each multiview data, especially on the Sens IT dataset of bass performance scores, the performance of each metrics is significantly improved, and the precision, recall, F1 value and NMI metrics are 0.632, 0.653, 0.687, and 0.713, respectively.In addition, the algorithm of this paper is utilized for the three bass playing patterns such as TaS1, Py11 and Mla1 are further analyzed, which further validates the universality and performance effect of the improved weighted clustering algorithm proposed in this paper for the analysis of pitch change patterns in bass playing.

Keywords: spectral clustering, multi-view clustering, cluster weighting, kernel k-means clustering, double bass

1. Introduction

In the orchestra, the double bass is an extremely important instrument, seldom solo, mainly play the role of accompaniment, but in the process of playing also has a position and influence that can not be ignored [1-2]. If the performance effect is not good, the connotation of the musical work can not be fully reflected, can not get the emotional resonance of the audience, will have an impact on the overall effect of the performance, so grasp the bass playing skills can make the audience better understand

✉ Corresponding author.
E-mail address: 15238313445@163.com (C. Liu).
Received 10 January 2024; Revised 25 May 2024; Accepted 20 October 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-206

the musical work [3-6]. The double bass is a bowed string instrument, which is one of the earlier instruments in Western four-stringed music, and is generally regarded as the deepest low voice that supports the acoustic weight of the whole orchestra in modern symphonic orchestra performance [7-9]. The double bass not only influences the whole orchestra to some extent, but also its strong and powerful bass can make the orchestra's acoustic equilibrium can be maintained, and it is an indispensable part of the establishment of the modern orchestra's acoustic luster [10-13].

Bass performance development so far, in addition to the orchestra in the performance of good harmonic effect, but also in the pursuit of art on the road of continuous innovation and exploration, in order to explore the bass more professional, more standardized, more humane way of playing [14-17]. In this process, players should implement their own performance characteristics and double bass performance characteristics, implement the whole orchestra's overall performance and acoustics, pay attention to the connotation of the music works to grasp, standardize the performance action, control the breath to adjust the rhythm [18-21]. Then through the integration and innovation of various performance methods, enhance their professional skills, integrate the performance skills into the actual performance, accurately express the emotions of the musical works, cause the audience's inner resonance, so that people are more aware of and familiar with the double bass, and jointly feel the unique charm of music [22-25].

In this study, for the problem of analyzing the tenor change pattern in double bass performance, an improved weighted multi-view clustering algorithm based on the performance of the score is proposed. This paper firstly introduces spectral clustering and subspace clustering, and deeply analyzes multi-view space clustering. On this basis, this paper constructs a model of kernel k-means clustering algorithm based on multi-view weighting by using cluster weighting. Subsequently, through the comparison experiments between this paper's algorithm and other clustering algorithms, the optimized and improved weighted clustering algorithm in this paper is verified to be effective in the application of the bass performance atlas dataset, and three bass performance modes are fully analyzed to prove its application value in the analysis of the change of performance modes.

2. Clustering algorithms for multiple views of double bass scores

The double bass is the largest instrument in the string family, with the lowest tone and the lowest range of any stringed instrument. It was created to meet the demand for bass in orchestral playing, so its extremely low range has led to the double bass becoming the only transposed instrument in the string family (i.e., the actual pitch is an octave lower than the notation). With the development of the double bass and the improvement of the violin making process, more and more double bass models have appeared which are more suitable for solo playing. In double bass performance, the players are required to have a unified understanding and grasp of the composer's creative intent, to reach a consensus on the melody, rhythm, and harmonic treatment of the work, to have a common artistic understanding and musical experience, the players to the score shall prevail in the strict performance norms, and is divided into different modes of performance. Therefore, in this paper, in order to analyze the pitch change mode in double bass performance, we start to construct a weighted multi-view clustering algorithm model based on the double bass performance score.

Multi-view clustering method can make full use of the intrinsic consistency information and diversity information among multiple views to improve the utilization of information and enhance the clustering performance, and thus has received wide attention. This chapter will introduce the classical spectral clustering algorithm and subspace clustering algorithm.

2.1. Spectral clustering

Spectral clustering algorithm is an algorithm evolved from graph theory. Compared with the more widely known K-mean clustering algorithm, spectral clustering is more adaptable to the distribution

of different data, and its computational amount is small and simple to implement, which makes it the most representative clustering algorithm at present. The algorithm transforms the traditional clustering problem into the optimal division problem of the undirected assignment graph. Its advantage is that it can perform clustering in an arbitrarily shaped data space composed of the target data point set, and obtain the global optimal solution of the clustering problem. The core idea of the spectral clustering algorithm is to treat each data point in each sample as a vertex in the space, and then connect different vertices by edges to assign weights to each edge. The weights of the edges are generally measured in terms of size in terms of distance, with edges between two vertices that are relatively far away from each other having a smaller weight, and conversely edges between two vertices that are relatively close to each other having a larger weight. This can get a graph, by cutting the graph, so that the subgraph obtained after cutting the graph has the same subgraph of the edge weight should be as large as possible, the weight of the edge between different subgraphs should be as small as possible, in order to achieve the purpose of clustering.

For a graph G, it is typically described using the set of points P and the set of edges E, i.e., $G(P,E)$. Where P is all the points $(p_1, p_2, \ldots, p_n)$ inside the sample data set. For any two different points in P, the connection of the edges is not necessary. Define weight w_{ij} to be the weight between point p_i and point p_j. Since spectral clustering is based on undirected graphs, $w_{ij} = w_{ji}$. If two points p_i and p_j are connected by an edge, then the weight between the two points is $w_{ij} > 0$, and if there is no edge connection, then the weight is $w_{ij} = 0$. The degree d_i for any point p_i in $G(P,E)$ is the sum of the weights of each edge connected to it, i.e. $d_i = \sum_{j=1}^{n} w_{ij}$. In this way, a $n \times n$ degree matrix D is obtained:

$$D = \begin{pmatrix} d_1 & \ldots & \ldots \\ \ldots & d_2 & \ldots \\ \vdots & \vdots & \ddots \\ \ldots & \ldots & d_n \end{pmatrix} \tag{1}$$

Obviously the degree matrix obtained is a diagonal matrix. Meanwhile, according to the weight values w_{ij} between different data points, a $n \times n$-sized adjacency matrix w can be obtained for this figure, and w_{ii} represents the weight values of the i th row and j th column positions.

Previously, only a broad definition of the weight value w_{ij} was given, and the specific weight value between the given samples could not be obtained, so it is not possible to construct the adjacency matrix directly. However, it is possible to obtain the adjacency matrix W from the similarity matrix s measured by the distance of the sample points according to the core idea of spectral clustering where the edge weights of the points close together have a high value and the edge weights of the points far away have a low value.

There are generally three common methods to construct the adjacency matrix W, which are ò− proximity method, K-neighborhood method and fully connected method. Only the ò− proximity method and the fully connected method are described here.

The ò− proximity method sets a distance threshold ò and uses the Euclidean distance $s_{ij} = \left\| x_i - x_j \right\|_2^2$ to represent the whole similarity matrix, and then sets the adjacency matrix w according to the magnitude relationship between the distance and the distance threshold. The value of w is taken as:

$$w_{ij} = \begin{cases} 0, & s_{ij} > \text{ò} \\ \text{ò}, & s_{ij} \le \text{ò} \end{cases} \tag{2}$$

However, the weight values obtained in this method have only two fixed values and are not very accurate, so ò−-neighbor method is rarely used.

The weight values of the neighbor matrices obtained in the fully connected method are all greater than 0, hence it is called the fully connected method. This method determines the edge weights by using different kernel functions, commonly used are Gaussian kernel function, Sigmoid kernel function and polynomial kernel function. The most commonly used is the Gaussian kernel function because the similarity matrix and the adjacency matrix obtained through the Gaussian kernel function are the same. w for:

$$w_{ij} = s_{ij} = \exp\left(-\frac{\left\|x_i - x_j\right\|_2^2}{2\sigma^2}\right) \tag{3}$$

After obtaining the degree matrix D and the adjacency matrix w, the Laplacian matrix $L = D - W$ can be obtained.

L is a diagonal matrix which has some good properties: L is a symmetric matrix with all eigenvalues real and also a semi-positive definite matrix with all eigenvalues greater than or equal to 0 and can be obtained for any n dimensional vectors f. $f^T Lf = \frac{1}{2}\sum_{i,j}^{n} w_{ij}(f_i - f_j)^2$. Spectral clustering is mainly based on the Laplacian matrix L computed from the degree matrix D and the adjacency matrix w to get the final clustering result.

2.2. Subspace clustering

Subspace clustering methods find effective clusters defined only by subsets of dimensions. Higher dimensional spaces have a large number of uncorrelated attributes, so the distribution of data information is sparser in higher dimensional spaces compared to lower dimensional spaces. The data in the dataset is generally located in the underlying low-dimensional subspace and very little data exists in the high-dimensional space. In this case, the low-dimensional subspace can be used to represent the data points. Clustering is mainly based on the subspace structure obtained from learning rather than clustering based on the whole space. The current subspace clustering is commonly categorized into two types: single view subspace clustering and multi-view subspace clustering.

2.2.1. Single view subspace clustering. Single-view subspace clustering algorithms are mainly based on a self-representation model, i.e., sample data from the subspace itself can be linearly represented by itself. Formally, the self-representation model can be expressed as:

$$X = XZ + E \tag{4}$$

Here $X = [x_1, \ldots, x_n] \in \square^{m \times n}$ is the sample data represented in matrix form, where n is the number of data points and each column represents an m-dimensional data sample. $Z = [z_1, z_2, ..., z_n] \in \square^{n \times n}$ represents the learned self-representation matrix. Each subspace representation z_i corresponds to a sample x_i. $E \in \square^{m \times n}$ is the reconstructed error matrix. Under the basic assumption that the observed data points are generally taken from low-dimensional subspaces, the popular single-view subspace clustering method can generally be formulated as:

$$\min_{Z,E} \theta(X, XZ) + \alpha\Psi(Z) \quad \text{s.t.} X = XZ + E \tag{5}$$

Here $\theta(\cdot,\cdot)$ is the loss function, $\psi_{(\cdot)}$ is the regularization term that imposes desired properties on the self-representation matrix z and $\alpha > 0$ is the balance parameter. The main difference between some subspace clustering methods is how the Ψ regularization term is chosen, and different choices of the regularization term can affect the clustering results differently. For example, SSC minimizes the

ℓ_1 paradigm on Z to learn the local structure of the data, while LRR imposes a low-rank constraint on Z to capture the global structure.LSR models Z and E using the Frobenius paradigm.

Finally, using $s = (|Z| + |Z^T|)/2$, the learned self-representation matrix is used to represent the final similarity matrix.

Subspace clustering methods have been used with great success in face and scene image clustering, motion segmentation, target detection, social network community clustering, and biclustering. Several literatures describe more information about subspace clustering in detail. Since these methods assume that the data lies in multiple linear subspaces, they may not be able to handle nonlinear data. On the other hand, for multi-view data, especially heterogeneous features, LRR, SSC, LSR and their variants may lead to significant performance degradation due to their focus on single-view features. Therefore, they cannot handle multi-view clustering tasks.

2.2.2. Multi-view subspace clustering. Multi-view subspace clustering utilizes multi-view features to improve clustering performance, which is often superior to single-view clustering. Multi-view subspace clustering aims to utilize the multi-view feature of data points to reveal the intrinsic low-dimensional structure of the data points.

Let $X = [X^{(1)}, X^{(2)}, ..., X^{(V)}] \in \square$ denote a multi-view data matrix consisting of V different views, where $X^{(v)} \in \square^{m_v \times n}$ denotes the v th feature matrix, V is the number of views, and m_v is the dimension of the data in the v th data matrix. The objective function can be expressed as follows:

$$\min_{Z^{(v)}, E^{(v)}} \sum_{v=1}^{V} \theta(X^{(v)}, X^{(v)}Z^{(v)}) + \alpha\Psi(Z^{(v)}) \tag{6}$$

$$\text{s.t.} X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)} \tag{7}$$

where $Z^{(v)} \in \square^{n \times n}$ is the v nd corresponding representation matrix and $E^{(v)}$ represents the v th reconstruction error matrix.

$Z^{(v)}, v = 1, 2, \ldots, V$ is obtained by solving Eq. After calculating and then obtaining the similarity matrix s, the final clustering result can be obtained by spectral clustering of the similarity matrix:

$$S = \frac{1}{V} \sum_{v=1}^{V} \frac{\left|z^{(v)}\right| + \left|z^{(v)T}\right|}{2} \tag{8}$$

Different multi-view subspace clustering methods differ in the choice of loss functions and regularization terms. The loss function is usually determined by the type of error, e.g., ℓ_2 loss for white noise, ℓ_1 loss for random corruption, and $\ell_{2,1}$ loss for sample-specific outliers. In contrast to the generality of the loss term, the regularization term characterizes the essential difference between the various methods.

Let $X = [X^{(1)}, X^{(2)}, ..., X^{(V)}] \in \square$ denote a multi-view data matrix consisting of V different views, where $X^{(v)} \in \square^{m_v \times n}$ denotes the v th feature matrix, V is the number of views, and m_v is the dimension of the data in the v th data matrix. The objective function can be expressed as follows:

$$\begin{aligned} &\min_{Z^{(v)}, E^{(v)}} \sum_{v=1}^{V} \theta(X^{(v)}, X^{(v)}Z^{(v)}) + \alpha\Psi(Z^{(v)}) \\ &\text{s.t.} X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)} \end{aligned} \tag{9}$$

where $Z^{(v)} \in \square^{n \times n}$ is the v nd corresponding representation matrix and $E^{(v)}$ represents the v th reconstruction error matrix.

$Z^{(v)}, v = 1, 2, \ldots, V$ is obtained by solving Eq. After calculating and then obtaining the similarity matrix s, the final clustering result can be obtained by spectral clustering of the similarity matrix:

$$S = \frac{1}{V}\sum_{v=1}^{V}\frac{\left|z^{(v)}\right| + \left|z^{(v)T}\right|}{2} \tag{10}$$

3. Optimization method based on weighted multi-view clustering algorithm

3.1. Multi-view clustering model based on cluster weighting

Oriented to bass performance scores, the scores are viewed as multiview data, and in real multiview data, the intracluster similarity of different clusters in a view may be very different, and a view with a low clustering loss does not guarantee that all the clusters in the view are better separable than the clusters in the other views. The intra-cluster similarity of the corresponding clusters may be low in some atlas views, while it may be high in other views. If each view is treated as a whole to roughly assign weights, local quality differences at the cluster level for each view cannot be recognized. To address this problem, this chapter proposes a multi-view kernel k-means clustering model (CWK2M) based on cluster weighting, which does not assign weights to the whole view, but rather assigns weights to each cluster within the view to determine the importance of each cluster in each view to the final clustering result.

3.1.1. Nuclear k-means clustering models. The classical k-mcans clustering model is able to achieve effective segmentation of convex clusters, while it is not effective in segmenting data with non-convex clusters. The kernel k-mcans model, as a kernel space-based version of the classical k-mcans model, can achieve effective delineation of non-convex clusters by utilizing the kernel function to map the original data to a non-linear high-dimensional feature space (or kernel space). Suppose the input dataset $\{x_1, x_2, ..., x_N\} \in \square^{d}$ is partitioned into M disjoint clusters $\{\pi_k\}_{k=1}^{M}$. Each data point in the original space is transformed to the regenerated kernel Hilbert space H through a nonlinear mapping $\phi : \square^{d} \mapsto \mathrm{H}$, and then by applying the k-mcans model on the higher dimensional mapping $\{\phi(x_1), \phi(x_2), ..., \phi(x_N)\}$ under this space, i.e., minimizing the sum of squares of the distances of each mapped point to its nearest cluster center, clustering under the nonlinear partitioning of the samples can be achieved, which results in an efficient partitioning of the nonconvex clusters. The objective function of kernel k-mcans is:

$$\begin{aligned} &\min_{U,\mu_k} \sum_{k=1}^{M}\sum_{i=1}^{N} U_{ik} \,\|\, \phi(x_i) - \mu_k \,\|^2 \\ &s.t. U \in \{0,1\}^{N\times M}, \sum_{k=1}^{M} U_{ik} = 1, \end{aligned} \tag{11}$$

where μ_k is the center of the k nd cluster, denoted as:

$$\mu_k = \frac{\sum_{i=1}^{N} U_{ik}\phi(x_i)}{\sum_{i=1}^{N} U_{ik}} \tag{12}$$

$U \in \square^{N\times M}$ is the cluster assignment matrix where each mapping point $\phi(x_i)$ is assigned to the cluster where the cluster center nearest to it is located, then the elements of row i and column $k^{'}$ of U are defined as:

$$U_{ik'} = \begin{cases} 1, & k' = arg\min_{k} \,\|\, \phi(x_i) - \mu_k \,\|^2 \\ 0, & otherwise \end{cases} \tag{13}$$

Usually the nonlinear mapping $\phi(x_i)$ cannot be computed explicitly, but the inner product $\phi(x_i)^T\phi(x_i)$ of any two mappings can be computed by the kernel function $\mathrm{K}(x_i, x_i) = \phi(x_i)^T\phi(x_i)$.

Thus, the entire data set can be represented in the high dimensional space as a kernel matrix $K \in \square^{N\times N}$, where the elements $K_{ij} = \mathrm{K}(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ in row i and column j represent the inner product between the i th mapping and the j th mapping. In the first iteration, M mapping points are selected as the initial cluster centers, and assuming that the k th mapping point among these M mapping points corresponds to a subscript of m in $\{\phi(x_1), \phi(x_2), ..., \phi(x_N)\}$, then $\mu_k = \phi(x_m)$. The square of the distance from each mapping to the cluster center in the first iteration can be expressed as:

$$\| \phi(x_i) - \mu_k \|^2 = \| \phi(x_i) - \phi(x_m) \|^2 = K_{im} - 2K_{im} + K_{mm} \tag{14}$$

In each of the following iterations, the cluster centers cannot be computed explicitly by Eq. However, the square of the distance from each mapping to the cluster center can be computed by the following equation when updating the cluster assignment matrix U:

$$\| \phi(x_i) - \mu_k \|^2 = K_{ii} - \frac{2\sum_{j=1}^{N} U_{jk} K_{ij}}{\sum_{j=1}^{N} U_{jk}} + \frac{\sum_{l=1}^{N}\sum_{m=1}^{N} U_{lk} U_{mk} K_{lm}}{\sum_{l=1}^{N}\sum_{m=1}^{N} U_{lk} U_{mk}} \tag{15}$$

Continuously and iteratively updating the cluster assignment matrix U until the objective function converges gives the final cluster partitioning result.

3.1.2. Multi-view kernel k-means model based on multi-view weighting. The weighted multi-view kernel k-means clustering model performs joint kernel k-means clustering on the corresponding sample points of each view and optimizes a consistent cluster partitioning result by adaptively weighting the kernel k-means loss for each view. Where $\mu_k^{(v)} = \frac{\sum_{i=1}^{N} U_{ik} \phi^{(v)}(x_i^{(v)})}{\sum_{i=1}^{N} U_{ik}}$ is the center of the k rd cluster in the v nd view, U is a consistent cluster assignment matrix across views, and U and ω_v are optimized in turn during the iteration process. The index p is a hyperparameter to control the distribution of weights ω_v over the views. When $p \to 1$, only one optimal view is selected, and when $p \to \infty$, the weights ω_v on each view tend to be equal. The model is able to reflect the importance of each view as a whole for clustering division by learning the view weights, but not the importance of each cluster in each view for clustering division. For this reason, in the next section, a cluster weighting mechanism is proposed for the kernel k-mcans-based multi-view clustering model.

Specifically, assume that a multi-view dataset consists of N samples under V view, denoted as $\{x_1^{(v)}, x_2^{(v)}, ..., x_N^{(v)}\}_{v=1}^{V} \in \square^{d^{(v)}}$, where $x_i^{(v)}$ is the i th sample under the v th view and $d^{(v)}$ is the feature dimension of the v th view. The model utilizes a multi-view joint kernel k-mcans approach to partition the mapping $\{\phi^{(v)}(x_1^{(v)}), \phi^{(v)}(x_2^{(v)}), ..., \phi^{(v)}(x_N^{(v)})\}_{v=1}^{V}$ of the multi-view samples in high-dimensional space into M disjoint clusters, and multiplies the kernel k-mcans loss under consistent partitioning for each view by a view weight ω_v and adaptively learns the weights of each view during the clustering process. The objective function of the model is:

$$\begin{gathered} \min_{\omega_v, U, \mu_k^{(v)}} \sum_{v=1}^{V} \omega_v^p \sum_{k=1}^{M}\sum_{i=1}^{N} U_{ik} \| \phi^{(v)}(x_i^{(v)}) - \mu_k^{(v)} \|^2 \\ s.t. U \in \{0,1\}^{N\times M}, \sum_{k=1}^{M} U_{ik} = 1 \\ \omega_v > 0, \sum_{v=1}^{V} \omega_v = 1, p > 1 \end{gathered} \tag{16}$$

3.2. Objective function

In order to realize the cluster-weighting based multi-view clustering model, based on the objective function of the view-weighting based multi-view kernel k-mcans model, the weight multiplied on the clustering loss (sum of squared distances) of each view in Eqn. is changed to multiplying the clustering loss of each cluster by a cluster weight and ensuring that the sum of the corresponding clusters' weights among different views is one. Therefore, the multi-view kernel k-mcans objective function under the cluster weighting mechanism can be obtained by replacing the weight ω_v of the v st view in Eq. with the weight ω_{vk} of the k rd cluster in the v th view:

$$\begin{gathered}\min_{\omega_{vk},U,\mu_k}\sum_{v=1}^{V}\sum_{k=1}^{M}\omega_{vk}^{p}\sum_{i=1}^{N}U_{ik}\,\|\,\phi^{(v)}(x_i^{(v)})-\mu_k^{(v)}\,\|^2 \\ s.t.U\in\{0,1\}^{N\times M},\sum_{k=1}^{M}U_{ik}=1 \\ \omega_{vk}>0,\sum_{v=1}^{V}\omega_{vk}=1,\forall k\end{gathered} \tag{17}$$

In order for the losses of the corresponding clusters under the high-dimensional mapping space to be comparable across views, the high-dimensional space of each view needs to be normalized to the same measure, and thus the kernel matrix element $K_{ij}^{(v)}$ under each view is divided by the mean of the squares of the distances between all the two-by-two mappings, viz:

$$K_ij^{(v)}/(\sum_{i=1}^{N}\sum_{j=1}^{N}(K_{ii}^{(v)}-2K_{ij}^{(v)}+K_{jj}^{(v)})/N^2) \tag{18}$$

3.3. Optimization process

In optimizing the objective function in Eq. Cluster assignment matrix U and cluster weights ω_{vk} are updated in turn using rotational optimization and when updating one variable, the other variable is fixed. In order to obtain relatively accurate cluster partitioning results and thus more meaningful cluster weights at the initial stage, M samples are selected as initial cluster centers using the global kernel k-mcans initialization algorithm. Initialize the cluster weights ω_{vk} to 1/V.

Update the cluster assignment matrix U: Fixing the cluster weights ω_{vk}, the cluster assignment matrix U can be updated by assigning each sample to the cluster that minimizes the allocation loss under the high-dimensional mapping. The allocation loss of each sample to each cluster is obtained by performing a weighted sum of the squared distances of $\phi^{(v)}(x_i^{(v)})$ and $\mu_k^{(v)}$ in the different views. Thus, the i th row and j th column elements of the cluster assignment matrix U are:

$$U_{ij}=\begin{cases}1, & j=\arg\min_k\sum_{v=1}^{V}\omega_{vk}^{p}\,\|\,\phi^{(v)}(x_i^{(v)})-\mu_k^{(v)}\,\|^2 \\ 0, & otherwise.\end{cases} \tag{19}$$

In the first iteration, the initial cluster center $\mu_k^{(v)}$ is the real sample and $\|\,\phi^{(v)}(x_i^{(v)})-\mu_k^{(v)}\,\|^2$ for the v nd view can be calculated according to Eq. In the next iteration, the cluster center is:

$$\mu_k^{(v)}=\frac{\sum_{i=1}^{N}U_{ik}\phi^{(v)}(x_i^{(v)})}{\sum_{i=1}^{N}U_{ik}},\|\,\phi^{(v)}(x_i^{(v)})-\mu_k^{(v)}\,\|^2 \tag{20}$$

Updating Cluster Weights ω_{vk}: Fixed Cluster Assignment Matrix U, Cluster Weights ω_{vk} can be updated by Lagrange multiplier method. First, the sum of squared distances (cluster loss) of the k th cluster in view v is denoted as:

$$D_{vk} = \sum_{i=1}^{N} U_{ik} \| \phi^{(v)}(x_i^{(v)}) - \mu_k^{(v)} \|^2 \tag{21}$$

Then construct the Lagrangian function of the objective function about ω_{vk} and λ_k in Eq:

$$L(\omega_{vk}, \lambda_k) = \sum_{v=1}^{V} \sum_{k=1}^{M} \omega_{vk}^p D_{vk} + \sum_{k=1}^{M} \lambda_k (\sum_{v=1}^{V} \omega_{vk} - 1) \tag{22}$$

Derivation of the above equation with respect to ω_{vk} yields:

$$\frac{\partial L(\omega_{vk}, \lambda_k)}{\partial \omega_{vk}} = p\omega_{vk}^{p-1} D_{vk} + \lambda_k = 0 \tag{23}$$

Then make the derivative equal to 0 to get:

$$p\omega_{vk}^{p-1} D_{vk} + \lambda_k = 0 \Rightarrow \omega_{vk} = (\frac{-\lambda_k}{pD_{vk}})^{\frac{1}{p-1}} \tag{24}$$

Bringing the expression of ω_{vk} in the above equation to constraint $\sum \omega_{vk} = 1$ yields:

$$(-\lambda_k)^{\frac{1}{p-1}} = \frac{1}{\sum_{v'=1}^{V} (\frac{1}{pD_{v'k}})^{\frac{1}{p-1}}}, p > 1 \tag{25}$$

Bringing this into Eq. gives a closed-form solution for cluster weight ω_{vk}, viz:

$$\omega_{vk} = \frac{1}{\sum_{v'=1}^{V} \left(\frac{D_{vk}}{D_{v'k}} \right)^{\frac{1}{p-1}}}, p > 1 \tag{26}$$

From the above equation, it can be seen that the smaller the cluster loss D_{vk} is, the larger the cluster weight ω_{vk} is. The smaller the cluster loss of a cluster in a particular view can reflect the higher intra-cluster similarity of this cluster in that view. Therefore, a cluster with higher intra-cluster similarity in a view will be given a higher weight in the clustering process. Parameter p is used to control the weight distribution of the corresponding clusters between views. For any two corresponding clusters belonging to view r and view s, it can be derived $\frac{\omega_{rk}}{\omega_{sk}} = (\frac{D_{sk}}{D_{rk}})^{\frac{1}{p-1}}, \frac{\omega_{rk}^p}{\omega_{sk}^p} = (\frac{D_{sk}}{D_{rk}})^{\frac{p}{p-1}}$ according to Eq. As p increases, the exponent $\frac{1}{p-1} \to 0$, the exponent $\frac{p}{p-1} \to 1$, and hence $\frac{\omega_{rk}}{\omega_{sk}}$ tends to 1, implying that the distribution of ω_{vk} tends to be equal across corresponding clusters, and $\frac{\omega_{rk}^p}{\omega_{sk}^p}$ tends to $\frac{D_{sk}}{D_{rk}}$, implying that the distribution of ω_{vk}^p across clusters tends to be inversely proportional to the loss between clusters. Conversely as p decreases, ω_{vk}^p it amplifies the relative difference in the losses D_{vk} of the corresponding clusters.

Since the division of clusters in the first iteration based on the initial cluster centers is usually coarse, the optimization process starts updating the cluster weights ω_{vk} from the second iteration (i.e., from the time the cluster assignment matrix is updated once). In the first iteration, only the cluster assignment matrix is updated, and in each subsequent iteration, the cluster assignment matrix U and the cluster weights ω_{vk} are updated in turn.

4. Experimental analysis based on weighted multi-view clustering algorithm

In order to validate the effectiveness and usability of the proposed weighted multiview clustering algorithm, two commonly used open source multiview datasets, Wikipedia, ALOI, and a homemade multiview dataset of bass playing (Sens IT), are selected for the comparison experiments in this paper. In this paper, we will mainly use Accuracy (ACC), Recall (P), F1 value and NMI to evaluate the clustering results of the model. In order to validate the performance of the algorithms, the algorithms in this paper will be compared with the single-view k-means algorithms ACOUSTIC and SEISMIC as well as the view concatenated multiview k-means (SMKM), the traditional multiview kernel k-means (MVKKM), the co-training-based multiview spectral clustering algorithms (Co-trainSC), and the multiview non-negative matrix factorization (Multi-NMF) Comparison.

4.1. Algorithm Performance Analysis

The comparison results of different algorithms on Wikipedia dataset are shown in Fig. 1.The Accuracy (ACC), Recall (P), F1 value and NMI metrics of this paper's algorithm on Wikipedia dataset are 0.587, 0.559, 0.521, and 0.315, respectively.The overall performance of this algorithm is significantly higher than that of other models on Wikipedia dataset. Compared to the traditional multi-view kernel k-means (MVKKM) the F1 value improves by 0.046 and the NMI improves by 0.044.

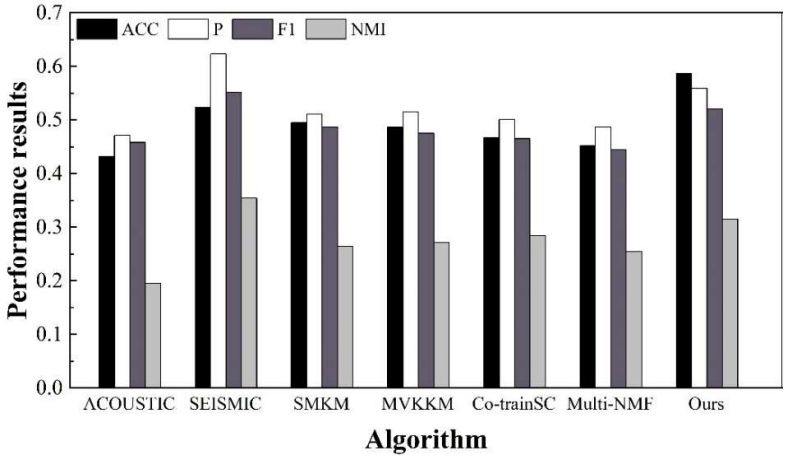


Fig. 1. Comparison results of different algorithms(Wikipedia)

The comparison results on the ALOI dataset are shown in Fig. 2, the performance of this paper's algorithm, Co-trainSC and Multi-NMF algorithm's on the ALOI dataset is better, with each index above 0.45, but this paper's method has better NMI value, which improves by 0.01~0.08 compared with other models.

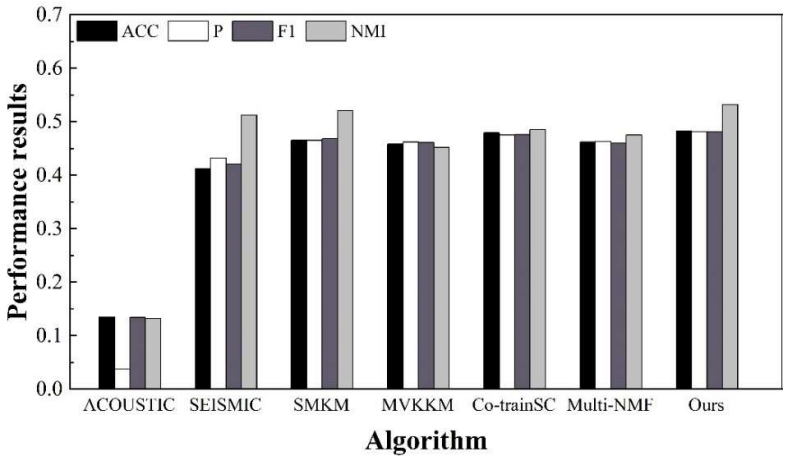


Fig. 2. Comparison results of different algorithms(ALOI)

The comparison results of the multi-view dataset of bass playing (Sens IT) are shown in Fig. 3. The performance of this paper's algorithm on the Sens IT dataset for each metric is significantly improved, with accuracy (ACC), recall (P), F1 value and NMI metrics of 0.632, 0.653, 0.687 and 0.713, respectively. Overall, the algorithm proposed in this paper gives better clustering results on each multi-view data.

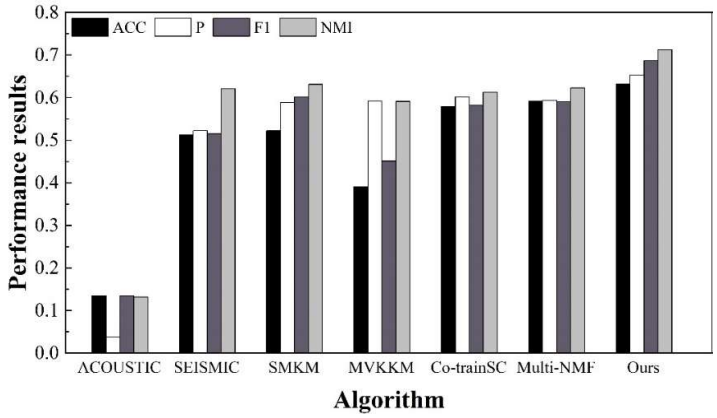


Fig. 3. Comparison results of different algorithms(Sens IT)

4.2. Time complexity analysis

For the traditional single view k-means clustering method, the computational complexity is O(IKNd), where I denotes the number of iterations, K denotes the number of clusters, N denotes the number of data points, and d denotes the feature dimension. Similar to the single-view k-means, the computational complexity of the weighted multi-view clustering method proposed in this paper is O(IKVNd), where V is the number of views, and the other variables are defined as in the single-view k-means algorithm. The results of time performance comparison are shown in Fig. 4. The time taken by this paper's algorithm on each dataset is 0.5s, 7.3s and 2.6s respectively, which is much less than that of the traditional Multi-View Mapping Clustering Algorithm (RMSC) and Multi-View Kernel k-means (MVKKM) on the same majority of the data and the difference in the time performance of the algorithms on the large-scale dataset, ALOI, will be even more obvious. In summary, the algorithm proposed in this chapter can dynamically assign view weights in the clustering process, and according to specific experimental data verified that the time complexity of this paper's algorithm is relatively small, and the algorithm is more effective, which confirms the algorithm's universality and at the same time also indicates its higher application value in the bass performance change mode.

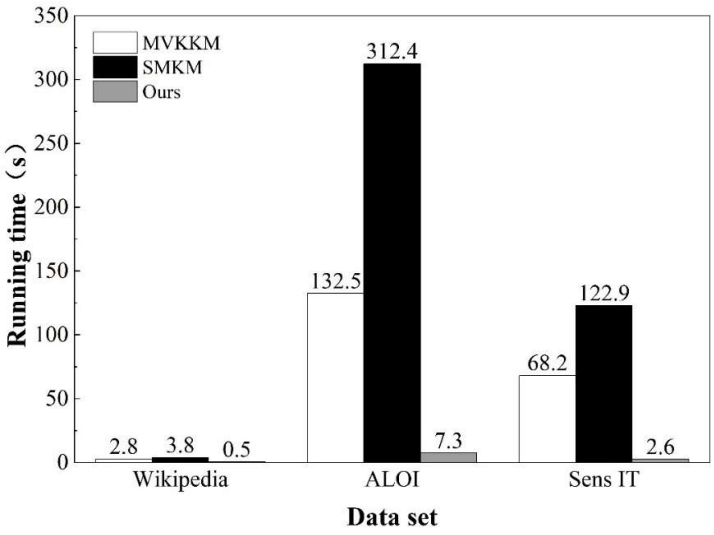


Fig. 4. Time can compare results

4.3. Robustness Tests for Double Bass Playing

In order to further validate the performance of this paper's algorithm in bass performance pattern analysis, the anti-noise ability of the score images in different modes of bass performance is tested. In this section, Gaussian noise, speckle noise and pretzel noise are added to the bass performance score images on the basis of Sens IT dataset, and then the clustering effect of this paper's algorithm is compared with other algorithms at the same time, and ACC, SA and mIoU are used as evaluation indexes.

The performance evaluation results of different algorithms for processing pretzel noise contaminated curvilinear images are shown in Table 1. After adding pretzel noise, the algorithm of this paper completes the training at 70 iterations, and the ACC, SA and mIoU of the algorithm are 0.902, 0.741 and 14.695, respectively. its performance metrics for processing pretzel noise polluted curved images are all the highest, which proves that its performance metrics for processing pretzel noise polluted curved images are all the highest.

Table 1. The evaluation results of different algorithms for the noise pollution of pretzels

Algorithm	ACC	SA	mIoU	Iteration number
ACOUSTIC	0.634	0.245	3.531	75
SEISMIC	0.854	0.601	10.512	16
SMKM	0.712	0.195	2.687	45
MVKKM	0.743	0.415	6.455	12
Co-trainSC	0.678	0.285	4.143	95
Multi-NMF	0.798	0.122	2.654	10
Ours	0.902	0.741	14.695	70

The performance evaluation results of different algorithms for processing Gaussian noise contaminated spectrum images are shown in Table 2, from which it can be seen that this paper's algorithm still achieves the best test results in processing Gaussian contaminated spectrum in terms of performance metrics, with an ACC as high as 0.891, and an SA and mIoU of 0.655 and 12.042, respectively, and its robustness is the best among all the compared algorithms.

Table 2. Different algorithms deal with the evaluation of gaussian pollution

Algorithm	ACC	SA	mIoU	Iteration number
ACOUSTIC	0.661	0.294	4.311	48
SEISMIC	0.753	0.068	0.855	95
SMKM	0.716	0.354	5.402	45
MVKKM	0.788	0.475	7.865	23
Co-trainSC	0.768	0.438	7.013	56
Multi-NMF	0.791	0.173	2.366	12
Ours	0.891	0.655	12.042	8

The performance evaluation results of different algorithms for processing speckle noise-polluted score images are shown in Table 3, and the performance indexes of this paper's algorithm for processing speckle noise-polluted score images are 0.879, 0.608, and 10.854, respectively, which are the best performance compared with other comparative algorithms. And from the performance comparison of the three kinds of noise, this paper's algorithm has achieved the best results, proving that this paper's algorithm can play a stable performance in the analysis of the change pattern in the double bass performance.

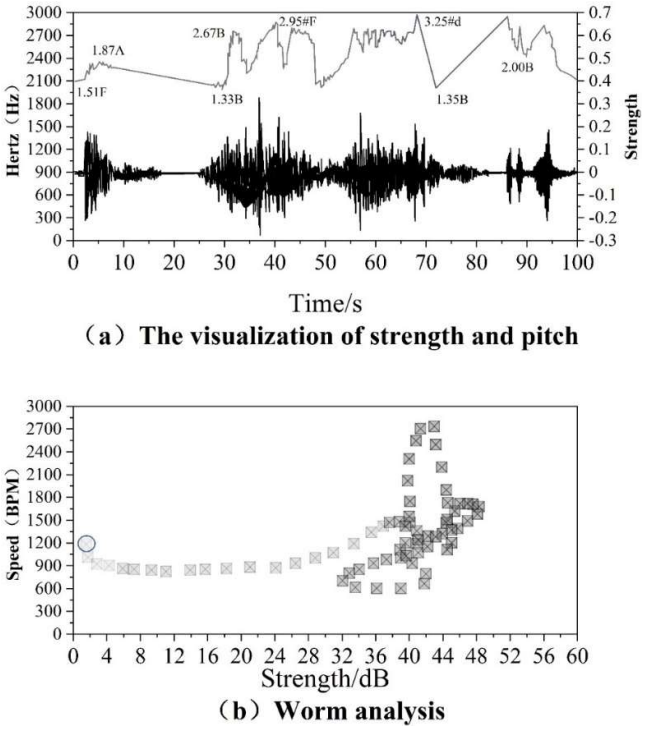
Table 3. Different algorithms handle the evaluation of spot noise pollution

Algorithm	ACC	SA	mIoU	Iteration number
ACOUSTIC	0.701	0.365	5.641	93
SEISMIC	0.712	0.384	5.956	45
SMKM	0.753	0.431	6.864	36
MVKKM	0.845	0.562	9.812	18
Co-trainSC	0.828	0.544	9.325	11
Multi-NMF	0.796	0.195	2.975	17
Ours	0.879	0.608	10.854	7

5. Analysis of pitch change patterns in double bass performance

According to the bass playing pattern, its pitch patterns are classified into three kinds, which are noted as TaS1, Py11 and Mla1 respectively.The three kinds of pitch change patterns are analyzed using the clustering algorithm in this paper, and the three patterns are visualized.

The analysis results of TaS1 are shown in Fig. 5, where Fig. (a) shows the visualization of intensity and pitch, and Fig. (b) shows the playing worm analysis. The horizontal coordinate of the performance worm analysis indicates the strength in decibels (dB), the vertical coordinate indicates the strength in beats per minute (BPM), and the blue circle is the beginning of the phrase as the lightest colored part. The graph shows that the TaS1 mode has frequent pitch changes, especially in the pitch details.

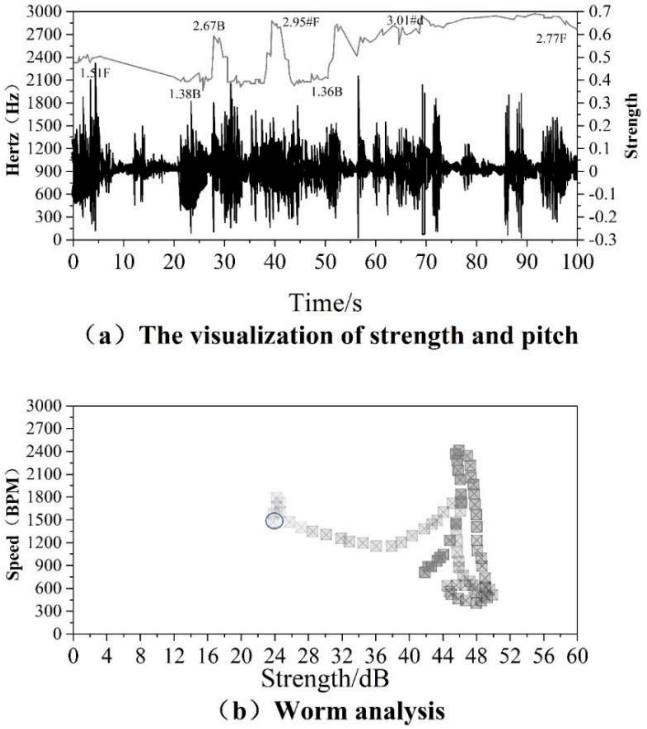


（a）The visualization of strength and pitch

（b）Worm analysis

Fig. 5. The results of TaS1

The results of the Py11 analysis are shown in Fig. 6, where Fig. (a) shows the visualization of intensity and pitch, and Fig. (b) shows the analysis of the playing worm. It can be seen that the Py11 model has the same frequent changes in playing pitch, but its pitch detailing changes are more subtle.



（a）The visualization of strength and pitch

（b）Worm analysis

Fig. 6. The results of Py11

The results of the analysis of Mla1 are shown in Figure 7, in which Figure (a) is the visualization of intensity and pitch, and Figure (b) is the performance worm analysis of the Mla1 mode of playing pitch change is more direct, and the pitch curve graph is more linearly increasing or decreasing.

Comprehensive comparison, from the aspect of pitch change can be seen TaS1, Py11 pitch change and Yu Xunfa playing pitch change is more similar. From the visualization graph of intensity, the playing intensity of TaS1 and Py11 is positively correlated with the melodic pitch change, the higher the pitch, the higher the intensity, the more obvious intensity change in the two modes of Py11, with more granularity of notes, and the playing intensity of Mla1 is weaker first and then stronger.

From the performance worms in the figure above, it can be seen that the performer is getting stronger and faster at the beginning of the phrase, with a greater range and variation of intensity in Mla1, and a more concentrated range of intensity in TaS1 and Py11, with Py11 having the smallest range of velocity variation. In terms of the smoothness of the playing worms, Py11 has the smoothest playing worm graph. TaS1's performance worms are more zigzag, indicating a richer hierarchical change in performance speed. mla1's worm graphs show more scattered dots, indicating exaggerated changes in performance strength and speed.

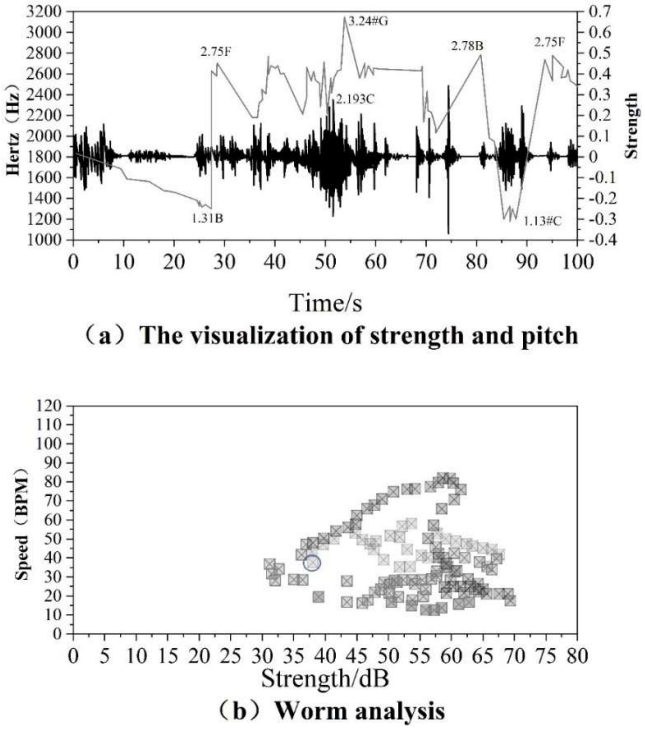


Fig. 7. The results of Mla1

6. Conclusion

This study is oriented towards the analysis of pitch change patterns in double bass performance using cluster weighting to propose a model of clustering algorithm based on multi-view weighting. The following conclusions are drawn through comparative experiments:

1) The weighted clustering algorithm proposed in this paper in terms of performance performance, in the Wikipedia dataset compared with the traditional multi-view kernel k-mean (MVKKM) F1 value improved by 0.046, NMI improved by 0.044. In the ALOI dataset the indexes are above 0.45 and the NMI value is improved by 0.01~0.08 compared with the other models. In addition, the bass violin playing multi-view view dataset (Sens IT), the performance of this paper's algorithm on each metric is significantly improved and significantly better than the other compared algorithms. The overall performance of the algorithm proposed in this paper is better than other comparison algorithms on

all data. In terms of time complexity, the time taken by this paper's algorithm on each dataset is not more than 8s, and the lowest time taken is only 0.5s, which is the least time taken on different datasets. And the comparison shows that this paper's algorithm is more advantageous in large-scale data sets.

2) The robustness of this paper's algorithm is verified through the noise processing of the image data of double bass performance score. The experimental results show that this paper's algorithm in the processing of Gaussian noise, speckle noise and pretzel noise and other three cases of the image, the ACC is higher than 0.87, the SA is not less than 0.6, and the mIoU is more than 10, compared with other comparative algorithms have a higher anti-noise ability, which further illustrates that this paper's optimization of improved weighted clustering algorithm of the practicality and effectiveness.

3) In the analysis of the pitch change of the bass playing mode, it can be seen that the three bass playing modes such as TaS1, Py11 and Mla1 have their own characteristics, TaS1 has frequent and delicate pitch change, the strength is positively correlated with the pitch, and it shows rich speed level, Py11 is more delicate in the pitch detail processing, the strength change is obvious, the sense of granularity is strong, and the playing speed change is naturally smooth, Mla1 pitch change is direct, and Mla1 pitch change is direct, the strength change is obvious, the sense of grain is strong, and the playing speed change is naturally smooth. The Mla1 has direct pitch changes, a wide range of intensity changes, exaggerated velocity changes, and an overall more decentralized playing style. These three modes of pitch change give the piece a variety of expression and emotional depth through their own unique playing techniques.

References

[1] Mick, J. P. (2024). An Analysis of Double Bass Vibrato. String Research Journal, 19484992241238989.

[2] Lavergne, P. J. (2021). The origin and the evolution of the double bass. Louisiana State University and Agricultural & Mechanical College.

[3] Nachtergaele, S. (2018). From divisions to divisi: improvisation, orchestration and the practice of double bass reduction. Early Music, 46(3), 483-500.

[4] Horner, C. (2019). Double Bass Parts: Challenges and Opportunities. American String Teacher, 69(4), 33-39.

[5] Chanphanitpornkit, T., & Yi, T. S. (2021). All About that Bass, No Treble: Shapeshifting from Violin to Double Bass. American String Teacher, 71(3), 33-38.

[6] Pagliaro, M. J. (2023). The Double Bass, How It Works: A Practical Guide to Double Bass Ownership. Rowman & Littlefield.

[7] Szczechowicz, J., & Kania, M. (2019). Pain in the spine and upper limbs among double bass players. Health Promotion & Physical Activity, 9(4), 27-39.

[8] Braun, J. (2020). An Overview of Pedagogical Materials for the Double Bass. American String Teacher, 70(3), 27-32.

[9] Benteler, B. (2016). Teaching select requisite fundamental double bass technique. Northeastern Illinois University.

[10] Chaichana, T. (2023). CLASSICAL DOUBLE BASS PEDAGOGY FOR INTERMEDIATE JAZZ BASS STUDENTS: A CASE STUDY. Mahidol Music Journal, 6(1), 99-113.

[11] Polianska, K. (2022). Double Bass in the Twentieth-Century Chamber Instrumental Music: Development of Double Bass Solo Performance. Artistic Culture. Topical Issues, 18(1).

[12] Booker, A. (2018). Double Bass Techniques of the Early Jazz Era. Jazz Perspectives, 11(3), 265-282.

[13] Wellington, K. K. D. (2018). The Double Bass and Theatricality: Technique as Choreography and Approaching a Theatrical Work. University of California, San Diego.

[14] Phipps, B. (2021). Reimagining the Double Bass:: Lloyd Swanton and The Necks. Journal of Music Research Online, 12.

[15] Borém, F. (2020). Developing a Crossover Idiomatic Writing for the Double Bass: Composing/Arranging & Playing, and... Da capo!. Revista Vórtex, 8(2), 7-7.

[16] Rawson, R. G. (2018). " For the Sake of Fullness of Music in the Choir"—Performance Practice and the Double Bass at the Kroměříž Court. Historical Performance, 1(1), 119-147.

[17] Geib, M. (2024). French versus German? The Over/Under on Double Bass Bow Holds. American String Teacher, 74(4), 60-63.

[18] Drabløs, P. E. (2012). From Jamerson to Spenner. A survey of the melodic eletric bass through performance practice. Norges musikkhøgskole.

[19] Ichim, M., & Dragulin, S. (2021). The sound color palette of the double bass: history and modernity. Bulletin of the Transilvania University of Braşov. Series VIII: Performing Arts, 65-74.

[20] Wang, P. (2021). The Double Bass in the Chinese Symphony Orchestra and the National Orchestra. Texas Christian University.

[21] Stuart, H. B. (2018). Shanti Nachtergaele From divisions to divisi: improvisation, orchestration and the practice of double bass reduction. Early Music.

[22] Berg, G. (2020). Allan Blank: Song Cycles with Double Bass and Piano. Journal of Singing, 76(3), 368-370.

[23] Hryhorovych, D. S. (2021). “CARMEN FANTASY” FOR DOUBLE BASS AND PIANO BY F. PROTO: INTERPRETATION AS A NEW LOOK AT OPERA. European Journal of Arts, (4), 19-24.

[24] Borém, F., & Lage, G. M. (2019). Intonation in the performance of the double bass: the role of vision and tact in undershoot and overshoot patterns. Online Journal of Bass Research.

[25] Xiaolong, F., & Yodwised, C. (2022). Promoting And Use of Applied Double Bass Workbook in The Conservatory of Music of Comprehensive University in Xiamen, China. Asia Pacific Journal of Religions and Cultures, 6(1), 112-123.