symmetry

MDPI

Article

Sampling Based Histogram PCA and Its Mapreduce Parallel Implementation on Multicore

Cheng Wang [1], Huiwen Wang [1,2], Siyang Wang [3,*], Edwin Diday [4] and Richard Emilion [5]

[1] School of Economics and Management, Beihang University, Beijing 100191, China; wangcheng@buaa.edu.cn (C.W.); wanghw@vip.sina.com (H.W.)
[2] Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations, Beijing 100191, China
[3] School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 100081, China
[4] CEREMADE, Paris-Dauphine University, 75775 Paris, France; diday@ceremade.dauphine.fr
[5] MAPMO, University of Orleans, 45067 Orleans, France; richard.emilion@univ-orleans.fr
* Correspondence: siyangw@163.com

Received: 7 April 2018; Accepted: 11 May 2018; Published: 15 May 2018

Abstract: In existing principle component analysis (PCA) methods for histogram-valued symbolic data, projection results are approximated based on Moore's algebra and fail to reflect the data's true structure, mainly because there is no precise, unified calculation method for the linear combination of histogram data. In this paper, we propose a new PCA method for histogram data that distinguishes itself from various well-established methods in that it can project observations onto the space spanned by principal components more accurately and rapidly by sampling through a MapReduce framework. The new histogram PCA method is implemented under the same assumption of "orthogonal dimensions for every observation" with the existing literatures. To project observations, the method first samples from the original histogram variables to acquire single-valued data, on which linear combination operations can be performed. Then, the projection of observations can be given by linear combination of loading vectors and single-valued samples, which is close to accurate projection results. Finally, the projection is summarized to histogram data. These procedures involve complex algorithms and large-scale data, which makes the new method time-consuming. To speed it up, we undertake a parallel implementation of the new method in a multicore MapReduce framework. A simulation study and an empirical study confirm that the new method is effective and time-saving.

Keywords: histogram-valued symbolic data; Principal component analysis; sampling; mapreduce; parallel

1. Introduction

Generalized Principal Component Analysis (PCA) is an important research tool in the Symbolic Data Analysis (SDA) [1]. PCA is a statistical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (PCs) [2–4]. PCA is commonly used for dimension reduction [5,6] by specifying a few PCs that account for as much of the variability in the original dataset as possible. It is well known that PCA has been primarily developed for single-valued variables. However, the explosion of big data from a wide range of application domains presents new challenges for traditional PCA, as it is difficult to gain insight from mass observation even in a low-dimensional space. Symbolic data analysis [7–10] has developed a new path for solving this problem.

Symbolic data was first introduced by [7] and aims to summarize large-scale data with conceptual observations described by symbolic descriptors, such as interval data, histogram, and continuous distribution data. Application fields of symbolic data include economic management, finance, and sociodemographic surveys. Thus, some researchers devoted themselves to studying new PCA methods for symbolic data. Here, we are interested in PCA for histogram data, where each variable value for each unit is a set of weights associated with the bins of variable values. The weights can be considered to be the frequency or probability of their associated bin for the corresponding unit. The sum of the weights is equal to 1.

Several studies contribute to the extension of PCA to histogram data. Reference [11] attempted to analyze a symbolic dataset for dimensionality reduction when the features are of histogram type. This approach assumes that the modal variables have the same number of ordered bins and uses the PCA of tables associated with each bin. References [12,13] assumed that the same number of bins was ordered for each variable. References [1,14] proposed approaches based on distributions of histogram transformation, which is not possible in the case of nonordinal bins. Nevertheless, the Ichino method solves this case by ranking bins by their frequency among the whole population of individuals. Reference [15] presented a strategy for extending standard PCA to histogram data. This method uses metabins that mix bins from different bar charts and enhance interpretability. Then it proposes Copular PCA to handle the probabilities and underlying dependencies. The method presented by [1] builds metabins and metabin trajectories for each individual from the nominal histogram variables. Reference [15] also proved that the method proposed by [1] is an alternate solution to Copular PCA. Reference [16] expressed an accurate distribution of linear combination of quantitative symbolic variables by convolutions of their probability distribution functions. Reference [17] use Multiple Factor Analysis (MFA) approach for the analysis of data described by distributional variables. Each distributional variable induces a set new numeric variable related to the quantiles of each distribution. Reference [18] deals with the current situation of histogram use through histogram PCA.

Despite the number of experts who study PCA for histogram data, there are still many challenging issues that have not been fully addressed in previous publications. In PCA methods for histogram data, projection results are approximated based on Moore's algebra [19] and fail to reflect the data's true structure, mainly because methods of linear combinations of histogram data are imprecise and not unified. The PSPCA method proposed by [16] can obtain precise projections, but the process of calculating convolutions is very complicated and time consuming, requiring a more efficient method of PCA for histogram data.

In this paper, we investigate a sampling-based PCA method for histogram data to solve the dimension reduction problem of histogram data. This method can project observations onto the space spanned by PCs more accurately and rapidly using a MapReduce framework. Assuming orthogonal dimensions for every observation [20], as in existing literature, to project observations onto the space spanned by PCs, we first sample from the original histogram variables to acquire single-valued data on which linear combination operations can be performed. Then, the projection of observations can be described by a linear combination of loading vectors and single-valued samples. Lastly, the projection is summarized to histogram data. As random sampling is used in this process, the projection results of different samplings may be different, resulting in the projection results of the proposed method being only close to accurate projections. However, we have proved that for a sufficiently large sample size, the projection results tend towards stability and are close to the accurate projections.

Because this method uses complex algorithms and large-scale data, it is quite time consuming. To speed up the method, we undertake a parallel implementation of the new method through a multicore MapReduce framework. MapReduce is a popular parallel programming model for cloud computing platforms and has been effective in processing large datasets using multicore or multiprocessor systems [21–25]. We conducted a simulation experiment to confirm that the new method is effective and can be substantially accelerated.

It should be emphasized that we only propose the MapReduce framework for sampling-based histogram PCA for big data. When there is sufficient data, we can use a big data platform like Hadoop and adopt parallel computing methods, such as GPU computing and MPI (Message Passing Interface). In this paper, we simply consider a multicore MapReduce parallel implementation.

The remainder of this paper is organized as follows. In Section 2, we propose a sampling-based Histogram PCA method. Section 3 describes the implementation of the new method in a multicore MapReduce framework. Next, in Section 4, a simulation study of the new method and its parallel implementation are presented. We also compare the execution times of serial and parallel processing. In Section 5, we report the results of our empirical study. Finally, Section 6 outlines our conclusions.

2. Sampling-Based Histogram PCA

In this section, we first define the inner product operator for histogram-valued data, then derive histogram PCA, and finally project the observations onto the space spanned by principal components using sampling.

2.1. Basic Operators

Let **X** be a data matrix that can be expressed as a vector of observations or of variables.

$$\mathbf{X} = [\mathbf{e}_1^T \ldots \mathbf{e}_n^T]^T = [\mathbf{X}_1 \ldots \mathbf{X}_p], \tag{1}$$

where $\mathbf{e}_i = (x_{i1}, \ldots, x_{ip}), i = 1, \ldots, n$ denotes the ith observation and $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})^T, j = 1, \ldots, p$ denotes the jth variable. Additionally, the random variable x_{ij} represents the jth variable of the ith observation, with realization x_{ij} in the classical case and ξ_{ij} in the symbolic case.

Next, we give the empirical distribution functions and descriptive statistics for histogram data. For further detail, refer to [9,26].

Let $\mathbf{e}_i$, $i = 1, \ldots, n$, be a random sample. Let the jth variable have a histogram-valued realization ξ_{ij},

$$\xi_{ij} = \{[a_{ij}^1, b_{ij}^1), p_{ij}^1; [a_{ij}^2, b_{ij}^2), p_{ij}^2; \ldots; [a_{ij}^{s_{ij}}, b_{ij}^{s_{ij}}], p_{ij}^{s_{ij}}\} \tag{2}$$

where $[a_{ij}^l, b_{ij}^l)$ is the lth subinterval of ξ_{ij} and p_{ij}^l is its associated relative frequency. Let s_{ij} denote the number of subintervals in ξ_{ij}. Then, $a_{ij}^l \leq b_{ij}^l$ for all $l = 1, \ldots, s_{ij}$, and $\sum_{l=1}^{s_{ij}} p_{ij}^l = 1$.

Reference [27] extended the empirical distribution function derived by [26] for interval data to histogram data. Based on the assumption that all values within each subinterval $[a_{ij}^1, b_{ij}^1)$ are uniformly distributed, they defined the empirical distribution of a point W_s within subinterval $[a_{ij}^1, b_{ij}^1)$ as

$$P(W_s \leq w) = \begin{cases} 0, & w < a_{ij}^l, \\ (w - a_{ij}^l)/(b_{ij}^l - a_{ij}^l), & a_{ij}^l \leq w < b_{ij}^l, \\ 1, & b_{ij}^l \leq w. \end{cases} \tag{3}$$

Furthermore, assume that each object is equally likely to be observed with probability $1/n$; then, the empirical distribution function of W is

$$F_W(w) = \frac{1}{n}\sum_{i=1}^{n}[\sum_{l:w\in\xi_{ij}^l} p_{ij}^l(w - a_{ij}^l)/(b_{ij}^l - a_{ij}^l) + \sum_{l:w\geq b_{ij}^l} p_{ij}^l], \tag{4}$$

where $\xi_{ij}^l = [a_{ij}^1, b_{ij}^1)$. The empirical density function of W is defined as

$$f_W(w) = \frac{1}{n}\sum_{i=1}^{n}\sum_{l:w\in\xi_{ij}^l} p_{ij}^l/(b_{ij}^l - a_{ij}^l). \tag{5}$$

The symbolic sample mean derived from the density function shown in Equation (5) is

$$\overline{W} = \frac{1}{2n}\sum_{i=1}^{n}\sum_{l=1}^{s_{ij}} p_{ij}^l(a_{ij}^l + b_{ij}^l). \tag{6}$$

Accordingly, the centralization of x_{ij} is given by

$$\widetilde{x_{ij}} = x_{ij} - \overline{W}. \tag{7}$$

After centralization, the symbolic sample variance and covariance are, respectively,

$$S^2 = \frac{1}{3n}\sum_{i=1}^{n}\sum_{l=1}^{s_{ij}} p_{ij}^l[(a_{ij}^l)^2 + a_{ij}^l b_{ij}^l + (b_{ij}^l)^2], \tag{8}$$

and

$$S_{ij} = \frac{1}{4n}\sum_{i=1}^{n}[\sum_{l=1}^{s_{ik}} p_{ik}^l(a_{ik}^l + b_{ik}^l)][\sum_{l=1}^{s_{jk}} p_{jk}^l(a_{jk}^l + b_{jk}^l)], i \neq j. \tag{9}$$

2.2. The PCA Algorithm of Histogram Data

Based on the operators above, we begin to derive histogram-valued PCs. For simplicity of notation, we assume that all histogram-valued data units have been centralized.

The kth histogram-valued PC $\mathbf{Y}_k(k = 1 \ldots p)$ is a linear combination of $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p$,

$$\mathbf{Y}_k = \mathbf{X}\mathbf{u}_k = \sum_{j=1}^{p} u_{jk}\mathbf{X}_j, \tag{10}$$

where $\mathbf{u}_k = (u_{1k} \ldots u_{pk})^T \in \mathbb{R}^p$ is subject to $\mathbf{u}_k^T\mathbf{u}_k = 1$ and $\mathbf{u}_k^T\mathbf{u}_l = 0, \forall l \neq k$. Then, the symbolic sample variance of $\mathbf{Y}_k(k = 1 \ldots p)$ can be derived from

$$S_{\mathbf{Y}_k}^2 = (u_{1k}, u_{2k}, \ldots, u_{pk})\begin{pmatrix} S_1^2 & S_{12} & \cdots & S_{1p} \\ S_{21} & S_2^2 & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ S_{p1} & S_{p2} & \cdots & S_p^2 \end{pmatrix}\begin{pmatrix} u_{1k} \\ u_{2k} \\ \vdots \\ u_{pk} \end{pmatrix} = \mathbf{u}_k^T\mathbf{\Sigma}\mathbf{u}_k, \tag{11}$$

where $\mathbf{\Sigma}$ represents the sample covariance matrix as $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p$, in which S_i^2 and $S_{ij}(i, j = 1 \ldots p)$ are present in Equations (8) and (9).

The following process is the same as that for classical PCA. (for details, see the original sources).

Set $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ as the eigenvectors of $\mathbf{\Sigma}$, corresponding to the eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_m$. The derivation of PC coefficients is transformed into the eigendecomposition of the covariance matrix. The contribution rate(CR) of the mth PC to the total variance can be measured by

$$CR_m = \frac{S_{\mathbf{Y}_m}^2}{\sum\limits_{j=1}^{p} S_{\mathbf{Y}_j}^2} = \frac{\lambda_m}{\sum\limits_{j=1}^{p} \lambda_j}. \tag{12}$$

2.3. The Projection Process

In existing PCA methods for histogram data, the projection results are approximated based on Moore's algebra [19] and fail to reflect true data structures, mainly because there are no precise, unified methods for linearly combining histogram data.

In this paper, we project observations onto the space spanned by PCs with a sampling method. Using the concept of symbolic data, histogram-valued data can be generated from a mass of numerical data. Retroactively, we can also sample from histogram-valued data to obtain numerical data on which linear combination operations can be performed. As PC coefficients and numerical variables corresponding to the original histogram-valued variables are obtained, we can calculate the numerical projections by linearly combining them. Then, histogram-valued projections of the original observations can be summarized from numerical projections.

Let x_{ij} represent the ith observation unit of the jth variable $\mathbf{X}_j$ in histogram-valued matrix $\mathbf{X}$. ξ_{ij} in Equation (2) is a realization of x_{ij}. Based on the assumption that all values within each subinterval $[a_{ij}^1, b_{ij}^1)$ are uniformly distributed, to obtain numerical data corresponding to ξ_{ij}, we select m_i samples from those subintervals of ξ_{ij} through random sampling. To maintain the coherence of the numerical sample matrix, the sample sizes of different variables from the same observation are assumed to be equal. After the random sampling process, the numerical sample matrix corresponding to the original histogram-valued matrix, which we call matrix $\mathbf{P}$, can be obtained. As the dimension of $\mathbf{X}$ is $n \times p$, the dimension of $\mathbf{P}$ is $\sum\limits_{i=1}^{n} m_i \times p$.

According to Equation (10), the kth numerical PC $\mathbf{NY}_k (k = 1 \ldots p)$ is a linear combination of $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_p$,

$$\mathbf{NY}_k = \mathbf{P}\mathbf{u}_k = \sum_{j=1}^{p} u_{jk}\mathbf{P}_j. \tag{13}$$

The calculation in Equation (13) can be performed because $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_p$ are numerical variables. The projection approach in Equation (13) is similar to the classic PCA approach. Finally, we obtain histogram-valued PCs through generating numerical PCs. In this way, histogram-valued observations are successfully projected onto the space spanned by PCs.

The above analysis can be summarized by the following algorithm:

- *Step* 1: Centralize the histogram-valued matrix $\mathbf{X}$ in Equation (1) and keep the notations consistent for simplicity.
- *Step* 2: Calculate the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{X}$ using Equations (8) and (9).
- *Step* 3: Eigendecompose $\mathbf{\Sigma}$ for orthonormalized eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m (m \leq p)$ in accordance with the eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_m$, which are PC coefficients and PC variances, respectively.
- *Step* 4: Obtain numerical matrix $\mathbf{P}$ through random sampling from $\mathbf{X}$.
- *Step* 5: Compute the numerical PCs $\mathbf{NY}_1, \mathbf{NY}_2, \ldots, \mathbf{NY}_m$ using Equation (13).
- *Step* 6: Generate histogram-valued PCs $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m$ by summarizing numerical PCs $\mathbf{NY}_1, \mathbf{NY}_2, \ldots, \mathbf{NY}_m$.

During this procedure, it is crucial to guarantee that any column of the matrix $\mathbf{P}$ obtained by sampling has the same distribution as the corresponding column in $\mathbf{X}$, or that the distance between these two distributions is small enough to be negligible. Considering the empirical cumulative distribution approximates the true cumulative distribution favorably, we can determine that the difference between these two distributions converges as the sample size goes to infinity. Here, we give some key conclusions from the perspective of asymptotic theory.

Kolmogorov-Smirnov distance is a useful measure of global closeness between distributions $F_m(x)$ and $F(x)$

$$D_m = \sup_{-\infty < x < \infty} |F_m(x) - F(x)|. \tag{14}$$

The asymptotic properties of Kolmogorov-Smirnov distance D_m have been studied and many related conclusions have been built. Extreme fluctuations in the convergence of D_m to 0 can be characterized by [28]

$$\overline{\lim}_{m \to \infty} \frac{m^{1/2} D_m}{(2 \log \log m)^{1/2}} = \sup_x \sqrt{F(x)(1 - F(x))}. \tag{15}$$

Here, we deal with histogram data, in which case $F(x)$ is not continuous; thus, we cannot give $\sup_x \sqrt{F(x)(1 - F(x))}$. According to Equation (15), D_m converges with a rate slower than $m^{-1/2}$ by a factor $(\log \log m)^{1/2}$.

From the pointwise closeness of $F_m(x)$ to $F(x)$, we can easily observe that

$$F_m(x) \quad is \quad AN\left(F(x), \frac{F(x)[1-F(x)]}{n}\right), for \quad fixed \quad x, \tag{16}$$

where AN is asymptotic normality.

Therefore, global and local differences between the histogram obtained from sampling data and the original can be controlled when the sample size is large, illustrating the feasibility of the proposed sampling method. It should be noted that the original histogram data, which is not continuous, is treated as the true distribution, but the conclusions above can cover discontinuous cases, although the formula expression may be complex. Related theoretical issues will be considered in the future.

3. MapReduce for Multicore, Sampling-Based Histogram PCA

The sampling-based histogram PCA approach presented in Section 2 involves complex algorithms and large-scale data, which makes the method time-consuming and degrades its computation efficiency. To speed up the method, we undertake its parallel implementation using a multicore MapReduce framework in this section.

MapReduce is a programming model for processing large datasets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a map procedure that filters and sorts data and a reduce procedure that performs a summary operation [21]. Reference [22] show that algorithms that can be written in a "summation form" can be easily parallelized on multicore computers. Coincidentally, Equations (8), (9) and (13) are written in this form. In addition, the random sampling procedure can also be parallelized. Consequently, we adapt the MapReduce paradigm to implement our histogram PCA algorithm.

The multicore MapReduce frame for histogram PCA is presented in Figure 1, which shows a high-level view of our architecture and how it processes the data. In *Step* 1, the MapReduce engine calculates the mean of each histogram variable using Equation (6) using pseudocode shown in Algorithm 1.

Algorithm 1 MapReduce for calculating the mean of each histogram variable

```
function MAP1($key, value$)
    // $key$: row ID of the observation in $\mathbf{X}$
    // $value$: observations in $\mathbf{X}$
    for each element $\xi_{ij}$ in $value$ do
        $sum_{ij} = \frac{1}{2}\sum_{l=1}^{s_{ij}} p_{ij}^{l}(a_{ij}^{l} + b_{ij}^{l})$
    end for
    return ($coordinate, sum_{ij}$)// $coordinate$: coordinates of elements in $\mathbf{X}$
end function
function REDUCE1($coordinate, sum_{ij}$)
    for $j = 1 \to p$ do
        $\overline{W}_j = \frac{1}{n}\sum_{i=1}^{n} sum_{ij}$ // Calculating the mean of each histogram variable
    end for
    return ($column\_id, \overline{W}_j$)
end function
```

Then, the variables are centralized using Equation (7). Every variable has its own engine instance, and every MapReduce task is delegated to its engine. Similarly, calculating the covariance matrix of histogram data using Equations (8) and (9) in *Step* 2, random sampling from $\mathbf{X}$ in *Step* 4, and computing the numerical principal components $\mathbf{NY}_1, \mathbf{NY}_2, \ldots, \mathbf{NY}_m$ using Equation (13) in *Step* 5 can also be delegated to MapReduce engines. The map and reduce functions are presented in Algorithms 2 and 3.

Algorithm 2 MapReduce for centralizing histogram data and calculating the covariance matrix of histogram data

```
function MAP2($key, value$)
    // $key$: row ID of the observation in $\mathbf{X}$
    // $value$: observations in $\mathbf{X}$
    for each element $\xi_{ij}$ in $value$ do
        $\xi_{ij} = \xi_{ij} - \overline{W}_j$ //Centralizing histogram data
        $sum_{ij} = \frac{1}{2}\sum_{l=1}^{s_{ij}} p_{ij}^l(a_{ij}^l + b_{ij}^l)$
    end for
    return ($coordinate, sum_{ij}$)// $coordinate$: coordinates of elements in $\mathbf{X}$
end function
function REDUCE2($coordinate, sum_{ij}$)
    for $i = 1 \rightarrow p$ do
        for $j = 1 \rightarrow p$ do
            if $i == j$ then
                $S_{ij} = \frac{1}{3n}\sum_{i=1}^{n}\sum_{l=1}^{s_{ij}} p_{ij}^l[(a_{ij}^l)^2 + a_{ij}^l b_{ij}^l + (b_{ij}^l)^2]$
            else
                $S_{ij} = \frac{1}{4n}\sum_{i=1}^{n}[\sum_{l=1}^{s_{ik}} p_{ik}^l(a_{ik}^l + b_{ik}^l)][\sum_{l=1}^{s_{jk}} p_{jk}^l(a_{jk}^l + b_{jk}^l)]$
                // Calculating the covariance matrix of histogram data
            end if
        end for
    end for
    return ($cov\_id, S_{ij}$)// $cov\_id$: coordinates of elements in $\mathbf{S}$
end function
```

Algorithm 3 MapReduce for sampling-based histogram PCA

```
function MAP3($key, value$)
    // $key$: row ID of the observation in $\mathbf{X}$
    // $value$: observations in $\mathbf{X}$
    for each element $\xi_{ij}$ in $value$ do
        $\mathbf{P}_{ij}$ ←Randomly sampling numerical data from $\xi_{ij}$, the sample size is $K$.
    end for
    return ($coordinate, \mathbf{P}_{ij}$)
end function
function REDUCE3($coordinate, \mathbf{P}_{ij}$)
    for each element $\mathbf{P}_{ij}$ in $\mathbf{P}$ do
        $\mathbf{NY}_{ik} = \sum_{j=1}^{p} u_{jk}\mathbf{P}_{ij}$
        //Calculating the $i$th observation of the $k$th numerical PC
        $\mathbf{Y}_{ik}$ ← Summarizing numerical PCs to obtain histogram PCs
    end for
    return ($pc\_id, \mathbf{Y}_k$)// $\mathbf{Y}_k$: the $k$th histogram PC
end function
```

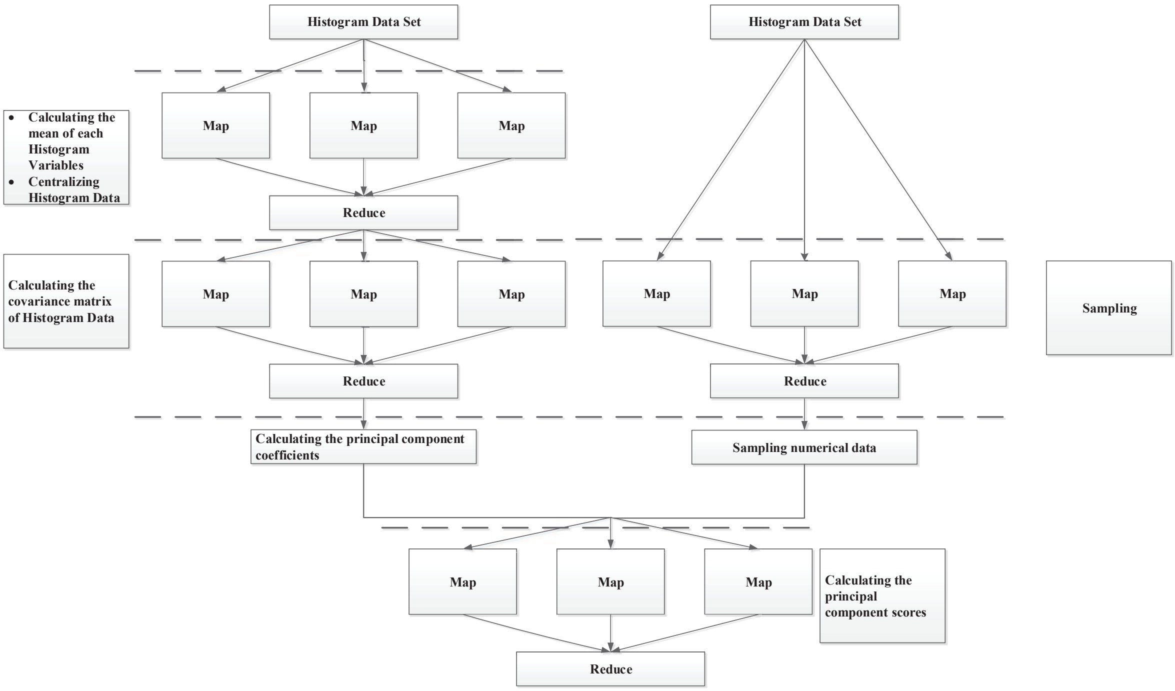

Histogram Data Set
Histogram Data Set
Calculating the mean of each Histogram Variables
Centralizing Histogram Data
Map
Map
Map
Reduce
Calculating the covariance matrix of Histogram Data
Map
Map
Map
Map
Map
Map
Sampling
Reduce
Reduce
Calculating the principal component coefficients
Sampling numerical data
Map
Map
Map
Calculating the principal component scores
Reduce

Figure 1. The multicore MapReduce frame for histogram PCA.

4. Simulation Study

To analyze the performance of the new histogram PCA method and demonstrate whether MapReduce parallelization can speed it up, we carried out a simulation study involving three examples.

$$\{U(1,3), N(1,1), \chi^2(1), lnN(0,0.5), -lnN(0,0.5)\}.$$

Of the examples, Example 1 investigates the histogram PCA accuracy in Section 2.2, Example 2 evaluates the precision of the sampling-based projection process of histogram PCA in Section 2.3, and Example 3 compares the method's execution time in parallel and serial systems.

The environment in which we conducted experiments is a PC with 8 Core Intel i7-6700K 4.00 GHz CPU and 8 GB RAM running Windows 7.

4.1. Example 1

With Example 1, we aim to investigate the accuracy of the histogram PCA algorithm. The simulated symbolic data tables are built as in [29]. Based on the concept of symbolic variables, to obtain n observations associated with a histogram-valued variable $\mathbf{X}_k$, we simulate K real values x_{ik}^{NUM} corresponding to each unit. These values are then organized in histograms that represent the empirical distribution for each unit. The distribution of the microdata that allow histogram generation, corresponding to each observation of the variables $\mathbf{X}_k(k = 1 \ldots p)$, is randomly selected from a mixture of distributions. Without loss of generality, in all observations, the widths of the subintervals in each histogram are the same.

Thus, we generate n histogram observations and the corresponding $n \times K$ single-value data points. The single-value data points can be viewed as the original data, which we aim to approximate using histogram-valued data. Consequently, experimental PCA results obtained from single-value data can work as an evaluation benchmark for the histogram PCA algorithm.

The detailed process of Example 1 is as follows:

(1) Generate a synthetic histogram dataset $\mathbf{X}$ with p variables and n observations and the corresponding single-value dataset, denoted as $\mathbf{X}^{NUM}$.

(2) Adopt the new histogram PCA method on $\mathbf{X}$ and perform classical PCA on $\mathbf{X}^{NUM}$. The obtained PC coefficients are denoted as $\mathbf{u}_j$ and $\mathbf{u}_j^{NUM}$, and the corresponding PC variances are represented by λ_j and λ_j^{NUM}, $j = 1, 2, \ldots, p$.

(3) Compute the following two indicators [30]:

(a) Absolute cosine value (ACV) of PC coefficients:

$$ACV(\mathbf{u}_j) = \left| \frac{(\mathbf{u}_j)^T \mathbf{u}_j^{NUM}}{\|\mathbf{u}_j\| \|\mathbf{u}_j^{NUM}\|} \right|. \tag{17}$$

Since cosine measures the angle between two vectors, ACV describes the similarity between PC coefficients $\mathbf{u}_j$ and the benchmark $\mathbf{u}_j^{NUM}$. A higher ACV indicates a better performance of the new histogram PCA method. (b) Relative error (RE) of PC variances:

$$RE(\lambda_j) = \left| \frac{\lambda_j - \lambda_j^{NUM}}{\lambda_j^{NUM}} \right|. \tag{18}$$

Taking PC variances obtained from $\mathbf{X}^{NUM}$ as a benchmark, the lower the RE is, the better the new histogram PCA method has performed.

The parameters for the experiments are set as follows: $n = 30, p = 3, K = 50, 100, 500, 1000, 5000$.

We can compare the performance of the new histogram PCA method with the single-value benchmark based on the resulting PC coefficients and variances.

The comparative results of the PC coefficients are shown in Table 1 using the ACV measure, and the comparative results of PC variances are shown in Table 2 using the RE measure. In general, the new histogram method yields fairly good results. The ACV values are all very close to 1, which indicates that the PC coefficients of the new histogram PCA method are similar to the benchmark PC coefficients. The values of RE all fluctuate near zero, which verifies that the PC variances of the new histogram PCA method have only small differences with benchmark PC variances.

Table 1. ACV values of the new histogram PCA.

K	$ACV(\mathbf{u}_1)$	$ACV(\mathbf{u}_2)$	$ACV(\mathbf{u}_3)$
50	1.0000	0.9999	0.9999
100	0.9988	0.9976	0.9988
500	0.9956	0.9927	0.9956
1000	0.9990	0.9994	0.9986
5000	0.9977	0.9995	0.9980

Furthermore, Tables 1 and 2 show that the values of ACV and RE change little, as the number of single-value data points K increases. Additionally, there is no obvious regularity between the indicators and K. Consequently, it can be concluded that the performance of the new histogram PCA method is accurate and not influenced by K.

Table 2. RE values for the new histogram PCA method.

K	$RE(\lambda_1)$	$RE(\lambda_2)$	$RE(\lambda_3)$
50	0.0056	0.0386	0.0215
100	0.0019	0.0108	0.0212
500	0.0139	0.0120	0.0179
1000	0.0331	0.0370	0.0288
5000	0.0464	0.0862	0.0359

4.2. Example 2

In this example, we consider the precision of the sampling-based projection process. The simulated symbolic data tables were built in the same way as in Example 1. To obtain n observations associated with histogram-valued variable $\mathbf{X}_k$, we simulated 5000 real values x_{ik}^{NUM} corresponding to each unit. These values were then organized in histograms. The distribution of the microdata was also randomly selected from a mixture of distributions:

$$\{U(1,3), N(1,1), \chi^2(1), lnN(0,0.5), -lnN(0,0.5)\}.$$

As in Example 1, for all observations, the widths of the subintervals in each histogram are the same.

Thus, we generate n histogram observations and the corresponding $n \times 5000$ single-value data points. The histogram dataset was adopted for comparing our sampling-based histogram PCA with the PSPCA method proposed by [16]. Single-value data points can also be viewed as the original data we aim to approximate using histogram-valued data. Consequently, experimental PCA results obtained from single-value data can work as the evaluation benchmark for the results of the histogram PCA algorithm.

Per Section 2.3, during the projection process, we select m_i single-value samples from the ith observation. In this example, we assume that the sizes of single-value samples in each observation are all the same and are denoted as M. The detailed process of Example 2 follows:

(1) Generate a synthetic histogram dataset $\mathbf{X}$ with p variables and n observations and the corresponding single-value data set, denoted as $\mathbf{X}^{NUM}$.

(2) Adopt sampling-based histogram PCA and PSPCA methods to $\mathbf{X}$. The obtained histogram PCs are denoted as $\mathbf{Y}_j(M)$ and $\mathbf{Y}_j^{PSPCA}$ respectively, where M is the sample size of each histogram unit. Classical PCA is performed on $\mathbf{X}^{NUM}$. The obtained histogram PCs are $\mathbf{Y}_j^{NUM}$.

(3) Compute the Wasserstein distance $d(\mathbf{Y}_j, \mathbf{Y}_j^{NUM})$ between $\mathbf{Y}_j$ and $\mathbf{Y}_j^{NUM}$ [31].

$$d(\mathbf{Y}_j, \mathbf{Y}_j^{NUM}) = \frac{1}{n}\sum_{i=1}^{n} d(Y_{ij}, Y_{ij}^{NUM}). \tag{19}$$

(4) Compute the Wasserstein distance $d(\mathbf{Y}_j, \mathbf{Y}_j^{PSPCA})$ between $\mathbf{Y}_j$ and $\mathbf{Y}_j^{PSPCA}$.

$$d(\mathbf{Y}_j, \mathbf{Y}_j^{PSPCA}) = \frac{1}{n}\sum_{i=1}^{n} d(Y_{ij}, Y_{ij}^{PSPCA}). \tag{20}$$

The parameters for the experiments are set as $n = 30, p = 3$, and $M = 50, 100, 500, 1000, 5000$.

Then, we can compare the performance of our sampling-based histogram PCA method with the single-value benchmark and the PSPCA method with the resulting PCs, as shown in Figure 2.

The solid line shows the Wasserstein distance between $\mathbf{Y}_j$ and $\mathbf{Y}_j^{PSPCA}$ and the dotted line denotes the Wasserstein distance between $\mathbf{Y}_j$ and $\mathbf{Y}_j^{NUM}$. The results shown in this figure indicate that the projections of sampling-based histogram PCA, PSPCA, and the single-value benchmarks are generally close. More specifically, the distances decrease as the random sampling size increases during sampling-based projection; when the sample size is more than 1000, the difference is close to zero. These results demonstrate the accuracy of sampling-based histogram PCA projection.

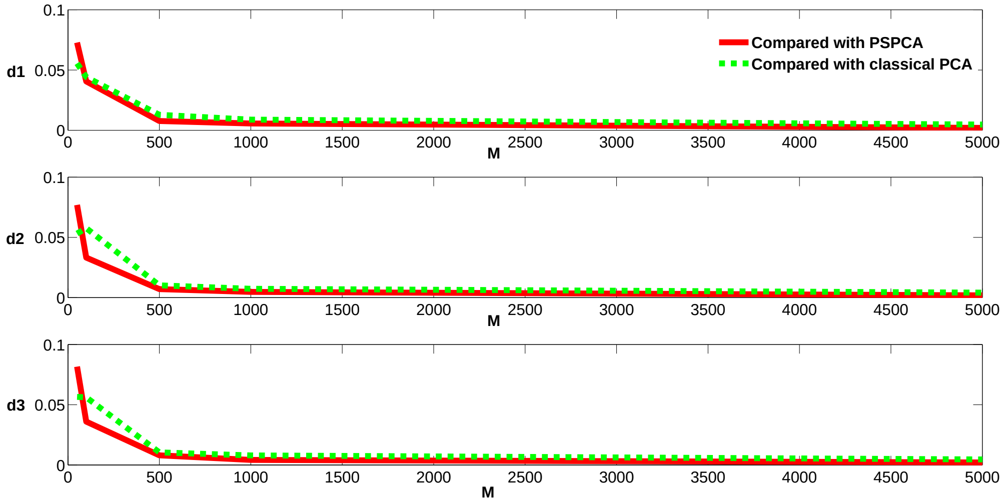


Figure 2. Wasserstein distance between $\mathbf{Y}_j$, $\mathbf{Y}_j^{PSPCA}$ and $\mathbf{Y}_j^{NUM}$.

4.3. Example 3

In the third example, we verify the time-saving effects of the multicore MapReduce implementation of the new histogram PCA method. First, we generated observations of the histogram-valued variables $\mathbf{X}_k, (k = 1 \dots p)$. Then, we implemented the method in two different versions: one running MapReduce and the other a serial implementation without the new framework. The execution times of both approaches are compared. Here, we compare results using 1, 2, 4 and 8 cores.

First, we create observations for each histogram-valued variable $\mathbf{x}_k$. The simulated symbolic data tables are built as in Examples 1 and 2.

Based on the concept of symbolic variables, to obtain n observations associated with a histogram-valued variable $\mathbf{X}_k$, we simulate 5000 real values corresponding to each unit. These values are then organized in histograms that represent the empirical distribution for each unit. In all observations, the widths of the histograms subintervals are the same.

To perform the simulation experiment, symbolic tables that illustrate different situations were created. In this study, a full factorial design is employed, with the following factors:

- Distribution of the microdata that allow histogram generation corresponding to each observation of the variables $\mathbf{X}_k(k = 1 \dots p)$:

  1. Uniform distribution:

  $$x_{kj} \sim U(\delta_1(j), \delta_2(j)), \delta_1(j) \sim U(-2, 0), \delta_1(j) \sim U(0, 2), j = 1, \dots, n; \tag{21}$$

  2. Normal distribution:

  $$x_{kj} \sim N(\mu(j), \sigma^2(j)), \mu(j) \sim U(0, 1), \sigma^2(j) \sim U(0, 2), j = 1, \dots, n; \tag{22}$$

  3. Log-Normal distribution:

  $$x_{kj} \sim lnN(\mu(j), \sigma^2(j)), \tag{23}$$

  $$\mu(j) \sim U(-0.5, 0.5), \sigma^2(j) \sim U(0.5, 1), j = 1, \dots, n;$$

  4. Mixture of distributions: Randomly selected from

  $$\{U(1, 3), N(1, 1), \chi^2(1), lnN(0, 0.5), -lnN(0, 0.5)\}. \tag{24}$$

- Number of histogram-valued variables: p = 3; 5.
- Observation size: n = 100; 500; 1000; 3000.

Based on the simulated histogram-valued variables, we conducted the new histogram PCA method in both parallel and serial. The random sampling size from histogram-valued variables in *Step* 4 is also 5000, the same as the number of real values we began with to simulate histogram-valued variables. Index CR_1 and CR_2 are computed using Equation (12) to analyze the behavior of the new histogram PCA method.Moreover, the execution time of both approach are compared to evaluate the speedup effect.

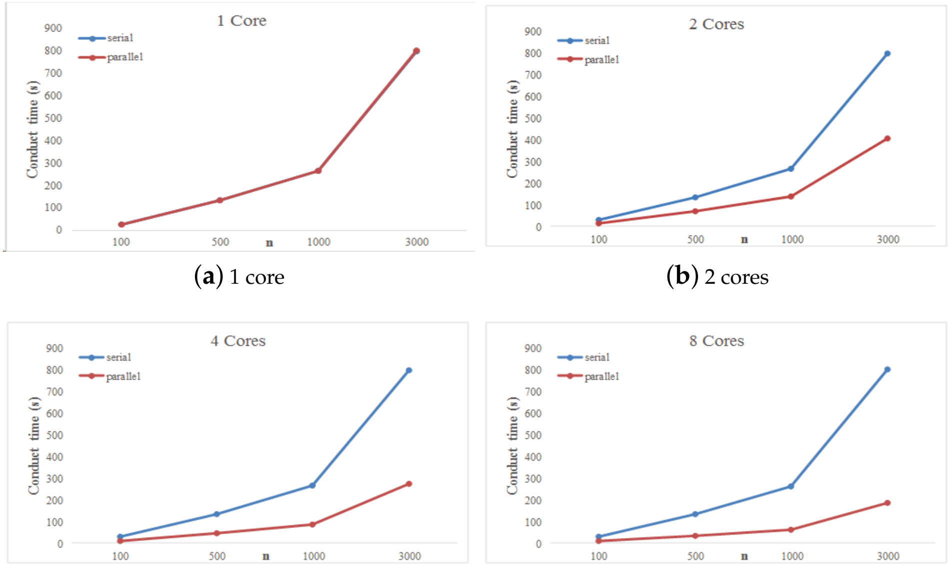
As the conclusion of different situations is almost the same, for the sake of brevity, in this section, we only provide the results for p = 3 and the uniform distribution.

Table 3 shows the contribution rate of the 1th and 2th PCs. As can be seen from the table, CR_1 and CR_2 explain most of the original information contained in the histogram-valued matrix, indicating that the new histogram PCA method performs well.

Table 3. Contribution rate of first and second PCs.

	n	CR_1	CR_2
p = 3, Uniform	100	51.19%	24.44%
	500	52.18%	23.92%
	1000	53.10%	23.46%
	3000	53.21%	23.40%

Figure 3 demonstrates the speedup of the method for 1, 2, 4 and 8 processing cores under the MapReduce frame. In Figure 3, we can see that for a given number of cores, execution time increases with sample size. In addition, when sample size is constant, the execution time of the parallel and serial approaches is nearly equivalent when using 1 core. As the number of cores increases, the execution time of the serial approach is essentially unchanged, whereas the execution time of the parallel approach is gradually reduced; the ratio of the execution time between the serial and parallel approaches is nearly the same as the core number. We conclude that the speed increase using a parallel approach with a multicore MapReduce frame is significant.



(**a**) 1 core (**b**) 2 cores

(**c**) 4 cores (**d**) 8 cores

Figure 3. Execution time (s) of serial and parallel approaches with 1, 2, 4 and 8 cores.

5. Empirical Study

Nowadays, the rapid development of the Internet has created opportunities for the development of movie sites. The major functions of movie sites are to calculate movie ratings based on user rating data and to then recommend high-quality movies. Since the set of user rating data is massive, symbolic data can be utilized.

In this section, we use real data to evaluate the performance of sampling-based histogram PCA. The data consists of 198,315 ratings from a Chinese movie site from October 2009 to May 2014. The ratings come from three different types of users: visitors, normal registered users, and expert users. In the first step, we summarize movie ratings from the different user types to obtain scoring histograms for each type of user for a movie. Then, the dataset becomes a histogram symbolic table of 500 observations and 3 variables. Finally, sampling-based histogram PCA can be conducted on the generated histogram symbolic table. We computed the CR index of each PC and show the loading plot of the first and the second PCs in Table 4 and Figure 4.

Table 4. Contribution rate of PCs.

PC	CR
F_1	73.87%
F_2	23.21%
F_3	2.92%

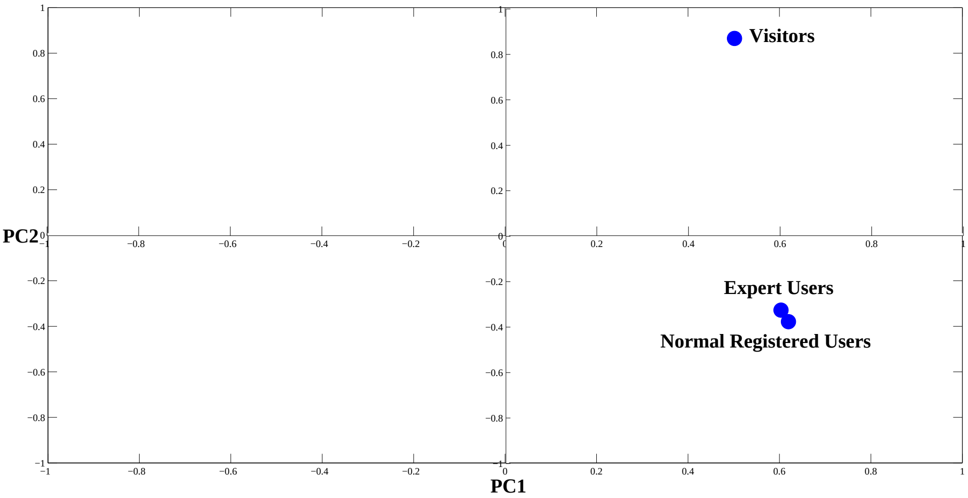


Figure 4. Loading plot on the first and the second PCs.

The results show that the first PC summarizes 73.87% of the total information carried by the original variables. The first PC is positively associated with visitors', normal registered users', and expert users' rating histograms. Thus, we could simply use the first PC as an integrated embodiment of the three types of user ratings, simplifying the evaluation and comparison of different movies.

Next, we verify the performance of the sampling-based projection process using the projection results on the first PC. As the explanatory power of the first PC is very good, if the scores of the first PC can identify the different characteristics of the movies, the projection results are reasonable.

In this empirical study, clustering analysis is implemented on these movies based on the Wasserstein distances [31] between pairs of projections on the first PC and movies are divided into five clusters. During the projection process, we randomly sample 1000 single-value samples from each histogram observation. The projection results of the clusters are showed in Figure 5, where the curves in the first row are the projections of the movies in each cluster. For convenience, we use curve instead of histogram to show the a movie's scoring probabilities in each score section. The histograms in the second row are the centers of the five clusters in the first row.

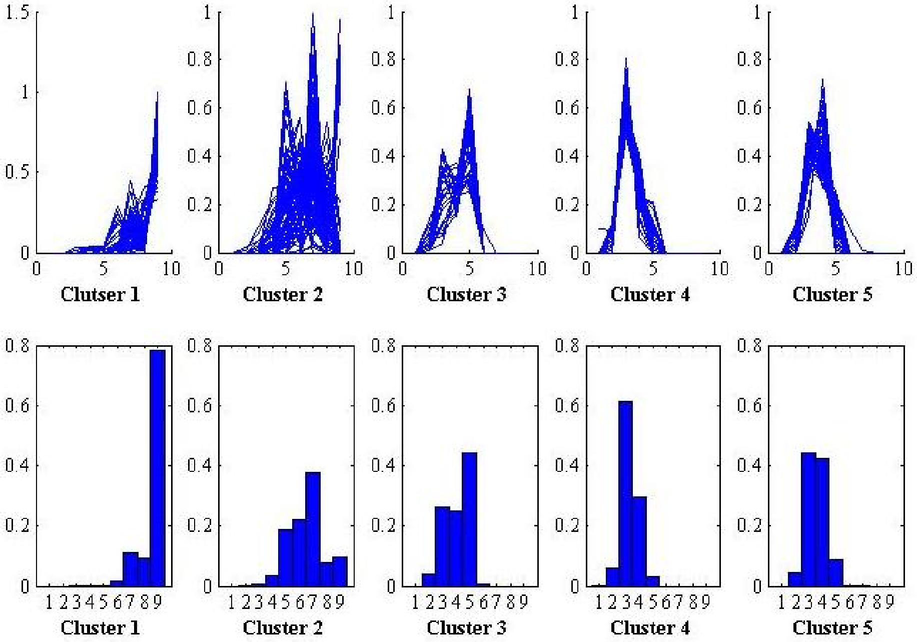


Figure 5. Projection Results.

The results show that the characteristics of different categories of movie are different. The movies in Cluster 1 receive high ratings; movies in Cluster 2 are distributed relatively equally across score sections; movies in Cluster 3 are distributed relatively equally among low-score sections; and the movies in Clusters 4 and 5 are low grade, mainly distributed in 3 and 3-4 point ratings, respectively.

As can be seen, the final ratings histograms clearly reflect different types of movies, which indicate that the method proposed in this paper performs effectively.

6. Conclusions

As there is no precise, unified method of linear combination of histogram-valued symbolic data, the paper presents a more accurate histogram PCA method, which can project observations onto the space spanned by PCs by random sampling numerical data from the histogram-valued variables. Furthermore, considering that the sampling algorithm is time-consuming and that the sampling can be done separately, we adopt a MapReduce paradigm to implement the new histogram PCA algorithm. A simulation study and an empirical study were performed to analyze the behavior of the new histogram PCA method and to demonstrate the effect of MapReduce parallel implementation. The new method performs well and can be sped up significantly through a multicore MapReduce framework.

In practice, the sampling method for the linear combination of histogram data proposed in the paper can be popularized and applied widely on other multivariable statistical models for histogram data, such as linear regression model, linear discriminant analysis model, and so on. Using sampling method will provide a new perspective for these methods, but further research is needed in the future.

Author Contributions: Data curation, C.W.; Formal analysis, H.W.; Funding acquisition, H.W.; Methodology, H.W., C.W.; Supervision, H.W., E.D., R.E.; Writing and original draft, C.W.; Writing, review and editing, S.W.

Acknowledgments: The authors are grateful to the Editor and anonymous reviewers for their insightful comments which have helped to improve the quality of this paper. This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 7142010725, 71371019 and 11501586).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ichino, M. The quantile method for symbolic principal component analysis. *Stat. Anal. Data Min.* **2011**, *4*, 184–198. [CrossRef]
2. Pearson, K. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
3. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educat. Psychol.* **1933**, *24*, 417. [CrossRef]
4. Jolliffe, I. *Principal Component Analysis*; Spring: New York, NY, USA, 1986.
5. Wang, J.; Barreto, A.; Wang, L.; Chen, Y.; Rishe, N.; Andrian, J.; Adjouadi, M. Multilinear principal component analysis for face recognition with fewer features. *Neurocomputing* **2010**, *73*, 1550–1555. [CrossRef]
6. Fung, W.K.; Gu, H.; Xiang, L.; Yau, K.K. Assessing local influence in principal component analysis with application to haematology study data. *Stat. Med.* **2007**, *26*, 2730–2744. [CrossRef] [PubMed]
7. Diday, E. The symbolic approach in clustering and relating methods of data analysis: The basic choices. In Proceedings of the Conference of the International Federation of Classification Societies, Aachen, Germany, 29 June–1 July 1987; pp. 673–684.
8. Diday, E.; Bock, H.H. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*; Springer: Berlin, Germany, 2000.
9. Diday, E.; Billard, L. *Symbolic Data Analysis: Conceptual Statistics and Data Mining*; Wiley: New York, NY, USA, 2006.
10. Diday, E.; Noirhomme-Fraiture, M. *Symbolic Data Analysis and the SODAS Software*; Wiley Online Library: New York, NY, USA, 2008.
11. Nagabhushan, P.; Kumar, R.P. Histogram PCA. In *Advances in Neural Networks–ISNN 2007*; Springer: Berlin, Germany, 2007; pp. 1012–1021.
12. Rodrıguez, O.; Diday, E.; Winsberg, S. Generalization of the principal components analysis to histogram data. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, Lyon, France, 13–16 September 2000; pp. 12–16.
13. Makosso Kallyth, S. Analyse en Composantes Principales de Variables Symboliques de Type Histogramme. Ph.D. Thesis, Université Paris-Dauphine, Paris, France, 2010.
14. Ichino, M. Symbolic PCA for histogram-valued data. In Proceedings of the IASC, Yokohama, Japan, 5–8 December 2008; pp. 5–8.
15. Diday, E. Principal component analysis for bar charts and metabins tables. *Stat. Anal. Data Min.* **2013**, *6*, 403–430. [CrossRef]
16. Chen, M.; Wang, H.; Qin, Z. Principal component analysis for probabilistic symbolic data: A more generic and accurate algorithm. *Adv. Data Anal. Classif.* **2015**, *9*, 59–79. [CrossRef]
17. Verde, R.; Irpino, A. Multiple factor analysis of distributional data. *arXiv* **2018**, arXiv:1804.07192. [CrossRef]
18. Žák, J.; Vach, M. A histogram based radar signal identification process. In Proceedings of the 2017 18th International Radar Symposium (IRS), Prague, Czech Republic, 28–30 June 2017; pp. 1–9.
19. Moore, R.E. *Interval Analysis*; Prentice-Hall Englewood Cliffs: Upper Saddle River, NJ, USA, 1966; Volume 2.
20. Cazes, P.; Chouakria, A.; Diday, E.; Schektrman, Y. Entension de l'analyse en composantes principales à des données de type intervalle. *Revue de Statistique Appliquée* **1997**, *45*, 5–24.
21. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Oper. Syst. Des. Implement.* **2004**, 137–149. [CrossRef]
22. Chu, C.T.; Kim, S.K.; Lin, Y.A.; Yu, Y.; Bradski, G.; Ng, A.Y.; Olukotun, K. Map-reduce for machine learning on multicore. *NIPS* **2006**, *6*, 281–288.
23. Ranger, C.; Raghuraman, R.; Penmetsa, A.; Bradski, G.; Kozyrakis, C. Evaluating mapreduce for multi-core and multiprocessor systems. In Proceedings of the IEEE 13th International Symposium on High Performance Computer Architecture, Phoenix, AZ, USA, 10–14 February 2007; pp. 13–24.
24. Lee, D.; Kim, J.S.; Maeng, S. Large-scale incremental processing with MapReduce. *Future Gener. Comput. Syst.* **2013**, *36*, 66–79. [CrossRef]

25. Kiran, M.; Kumar, A.; Mukherjee, S.; Ravi Prakash, G. Verification and Validation of MapReduce Program Model for Parallel Support Vector Machine Algorithm on Hadoop Cluster. *Int. J. Comput. Sci. Issues (IJCSI)* **2013**, *10*. [CrossRef]
26. Bertrand, P.; Goupil, F. Descriptive statistics for symbolic data. In *Analysis of Symbolic Data*; Springer: Berlin, Germany, 2000; pp. 106–124.
27. Billard, L.; Diday, E. From the statistics of data to the statistics of knowledge: Symbolic data analysis. *J. Am. Stat. Assoc.* **2003**, *98*, 470–487. [CrossRef]
28. Kiefer, J. On large deviations of the empiric df of vector chance variables and a law of the iterated logarithm. *Pac. J. Math.* **1961**, *11*, 649–660. [CrossRef]
29. Dias, S.; Brito, P. Distribution and Symmetric Distribution Regression Model for Histogram-Valued Variables. *arXiv* **2013**, arXiv:1303.6199. [CrossRef]
30. Wang, H.; Guan, R.; Wu, J. CIPCA: Complete-Information-based Principal Component Analysis for interval-valued data. *Neurocomputing* **2012**, *86*, 158–169. [CrossRef]
31. Irpino, A.; Verde, R. A new Wasserstein based distance for the hierarchical clustering of histogram symbolic data. In *Data Science and Classification*; Springer: Berlin, Germany, 2006; pp. 185–192.