algorithms

MDPI

Article

Consensus Big Data Clustering for Bayesian Mixture Models

Christos Karras [1,*], Aristeidis Karras [1], Konstantinos C. Giotopoulos [2], Markos Avlonitis [3] and Spyros Sioutas [1,*]

[1] Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece; akarras@ceid.upatras.gr
[2] Department of Management Science and Technology, University of Patras, 26334 Patras, Greece
[3] Department of Informatics, Ionian University, 49100 Kerkira, Greece
* Correspondence: c.karras@ceid.upatras.gr (C.K.); sioutas@ceid.upatras.gr (S.S.)

Abstract: In the context of big-data analysis, the clustering technique holds significant importance for the effective categorization and organization of extensive datasets. However, pinpointing the ideal number of clusters and handling high-dimensional data can be challenging. To tackle these issues, several strategies have been suggested, such as a consensus clustering ensemble that yields more significant outcomes compared to individual models. Another valuable technique for cluster analysis is Bayesian mixture modelling, which is known for its adaptability in determining cluster numbers. Traditional inference methods such as Markov chain Monte Carlo may be computationally demanding and limit the exploration of the posterior distribution. In this work, we introduce an innovative approach that combines consensus clustering and Bayesian mixture models to improve big-data management and simplify the process of identifying the optimal number of clusters in diverse real-world scenarios. By addressing the aforementioned hurdles and boosting accuracy and efficiency, our method considerably enhances cluster analysis. This fusion of techniques offers a powerful tool for managing and examining large and intricate datasets, with possible applications across various industries.

Keywords: stochastic data engineering; cluster analysis; Bayesian mixture modelling; consensus clustering; big-data management and analytics

check for updates

Citation: Karras, C.; Karras, A.; Giotopoulos, K.C.; Avlonitis, M.; Sioutas, S. Consensus Big Data Clustering for Bayesian Mixture Models. *Algorithms* **2023**, *16*, 245. https://doi.org/10.3390/a16050245

Academic Editor: Frank Werner

Received: 31 March 2023
Revised: 27 April 2023
Accepted: 6 May 2023
Published: 9 May 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Clustering is a key technique in unsupervised learning and is employed across various domains such as computer vision, natural language processing, and bioinformatics. Its primary objective is to assemble related items and disclose hidden patterns within data. Confronting complex datasets, however, can prove challenging, as conventional clustering approaches may not be effective. In response to this issue, Bayesian nonparametric methods have gained popularity in recent years as a potent means of organising large datasets. These approaches offer a versatile and potent solution for managing the data's unpredictability and complexity, making them a crucial tool in the field of clustering. Clustering is crucial in the fields of information science and big-data management for organizing and handling huge volumes of data. In recent years, exponential data proliferation has increased the demand for efficient and effective solutions to handle, manage, and analyse enormous data volumes. Clustering can accomplish this by grouping comparable data points together, hence lowering the dataset's size and making it simpler to examine. Apart from traditional techniques, there are much more promising ones. The product partition model (PPM) is one of the most widely used Bayesian nonparametric clustering algorithms. PPMs are a class of models that classify data into clusters and assign a set of parameters to each cluster. They use a prior over the parameters to draw conclusions about the clusters. Despite the efficacy of PPMs, a single clustering solution may not be enough for complicated datasets, resulting in the development of consensus clustering. Consensus clustering is a kind of

ensemble clustering that produces a final grouping by combining the results of numerous clustering methods [1,2].

The motivation behind this work lies in addressing the challenges associated with clustering complex datasets, which is crucial for efficient big-data management and analysis. The determination of the number of clusters and handling of high-dimensional data are significant challenges that arise while dealing with these complex datasets. To tackle these challenges, we propose an innovative method that combines Bayesian mixture models with consensus clustering.

Our method is designed to address the challenges of clustering extensive datasets and identifying the ideal number of clusters. By merging the strengths of PPMs, Markov chain Monte Carlo (MCMC), and consensus clustering, we aim to produce reliable and precise clustering outcomes. MCMC methods enable the estimation of PPM parameters, making them a powerful tool for sampling from intricate distributions. Moreover, split-and-merge techniques allow the MCMC algorithm to navigate the parameter space and generate samples from the posterior distribution of the parameters [3–5].

The incorporation of consensus clustering with Bayesian mixture models facilitates the examination of complex and high-dimensional datasets, thereby improving the effectiveness and efficiency of big-data management. Our suggested approach also holds the potential to uncover hidden data patterns, which can lead to improved decision-making processes and offer a competitive edge across various industries.

The proposed utilization of Bayesian nonparametric ensemble methods for clustering intricate datasets demonstrates considerable promise in the realms of information science and big-data management. Combining PPMs, MCMC, and consensus clustering results in a robust and accurate clustering solution. Further research could refine this method and explore its application in real-world situations.

The remainder of this article is structured as follows: Section 2 presents a concise overview of Bayesian nonparametric methods, particularly PPMs, and their applicability in clustering. Section 3 describes consensus clustering and its application to ensemble methods. In Section 3.3, we present our proposed method, which incorporates PPMs and consensus clustering. Section 4 conducts experiments to illustrate the efficacy of the proposed method for clustering complex datasets. Finally, Section 6 concludes the paper, discussing potential future research and the significance of our proposed method in the field of big-data analysis and management.

2. Related Work

Cluster analysis has been utilised extensively in numerous disciplines to identify patterns and structures within data. Caruso et al. [6] applied cluster analysis to an actual mixed-type dataset and reported their findings. Meanwhile, Absalom et al. [7] provided a comprehensive survey of clustering algorithms, discussing the state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Jiang et al. [8] conducted a survey of cluster analysis for gene expression data. Furthermore, Huang et al. [9] proposed a locally weighted ensemble clustering method that assigns weights to individual partitions based on local information. These studies demonstrate the diversity of clustering methods and their applications, emphasizing the importance of choosing the appropriate method for specific datasets.

Consensus clustering utilises W runs of a base model or learner (such as K-means clustering) and combines the W suggested partitions into a consensus matrix, where the (i, j)-th entries reflect the percentage of model runs in which the ith and jth individuals co-cluster. This ratio indicates the degree of confidence in the co-clustering of any two elements. Moreover, ensembles may reduce computational execution time. This occurs because individual learners may be weaker (and hence consume less of the available data or stop before complete convergence), and the learners in the vast majority of ensemble techniques are independent of one another, enabling the use of a parallel environment for each of the faster model runs [10].

Bayesian clustering is a popular machine-learning technique for grouping data points into clusters based on their probability distributions. Hidden Markov models (HMM) [11] have been used to model the underlying probabilistic structure of data in Bayesian clustering. Accelerating hyperparameters via Bayesian optimizations can also help in building automated machine learning (AutoML) schemes [12], while such optimizations can also be applied in Tiny Machine Learning (TinyML) environments wherein devices can be trained to fulfil ML tasks [13]. Ensemble Bayesian Clustering [14] is a variation of Bayesian clustering that combines multiple models to produce more robust results, while cluster analysis [15] extends traditional clustering methods by considering the uncertainty in the data, which leads to more accurate results.

Traditional clustering algorithms require a preset selection of the number of clusters K, which can be challenging as it plagues many investigations, with researchers often depending on certain rules to choose a final model. Various selections of K are compared, for instance, using an evaluation metric for K. Techniques for selecting K using the consensus matrix are offered in [16]; however, this implies that any uncertainty over K is not reflected in the final clustering, and each model run utilises the same, fixed number of clusters. An alternative clustering technique incorporates cluster analysis within a statistical framework [17], which implies that models may be formally compared and issues such as choosing K can be represented as a model-selection problem using relevant tools.

In recent years, various clustering techniques have been developed to address the challenges associated with traditional clustering methods. Locally weighted ensemble clustering [9] leverages the advantages of ensemble clustering while accounting for the local structure of the data, leading to more accurate and robust results. Consensus clustering, a type of ensemble clustering, combines multiple runs of a base model into a consensus matrix to increase confidence in co-clustering [16]. Enhanced ensemble clustering via fast propagation of cluster-wise similarities [18,19] improves the efficiency and effectiveness of clustering by propagating cluster-wise similarities more rapidly. Real-world applications of these clustering techniques can be found in various domains, such as gene expression analysis, cell classification in flow cytometry experiments, and protein localization estimation [20–22].

Recent advancements in ensemble clustering have addressed various challenges posed by high-dimensional data and complex structures. Yan and Liu [23] proposed a consensus clustering approach specifically designed for high-dimensional data, while Niu et al. [24] developed a multi-view ensemble clustering approach using a joint affinity matrix to improve the quality of clustering. Huang et al. [25] introduced an ensemble hierarchical clustering algorithm that considers merits at both cluster and partition levels. In addition, Zhou et al. [26] presented a clustering ensemble method based on structured hypergraph learning, and Zamora and Sublime [27] proposed an ensemble and multi-view clustering method based on Kolmogorov complexity. Huang et al. [28] tackled the challenge of high-dimensional data by developing a multidiversified ensemble clustering approach, focusing on various aspects such as subspaces, metrics, and more. Huang et al. [29] also proposed an ultra-scalable spectral clustering and ensemble clustering technique. Wang et al. [30] developed a Markov clustering ensemble method, and Huang et al. [31] presented a fast multi-view clustering approach via ensembles for scalability, superiority, and simplicity. These studies showcase the diverse range of ensemble clustering techniques developed to address complex data challenges and improve the performance of clustering algorithms.

Clustering ensemble techniques have been developed and applied across various domains, addressing the challenges and limitations of traditional clustering methods. Nie et al. [32] concentrated on the analysis of scRNA-seq data, discussing the methods, applications, and difficulties associated with ensemble clustering in this field. Boongoen and Iam-On [33] presented an exhaustive review of cluster ensembles, highlighting recent extensions and applications. Troyanovsky [34] examined the ensemble of specialised cadherin clusters in adherens junctions, demonstrating the versatility of ensemble clustering methods. Zhang and Zhu [35] introduced Ensemble Clustering based on Bayesian Net-

work (ECBN) inference for single-cell RNA-seq data analysis, offering a novel method for addressing the difficulties inherent to this data format. Hu et al. [36] proposed an ultra-scalable ensemble clustering method for cell-type recognition using scRNA-seq data of Alzheimer's disease. Bian et al. [37] developed an ensemble consensus clustering method, scEFSC, for accurate single-cell RNA-seq data analysis based on multiple feature selections. Wang and Pan [38] introduced a semi-supervised consensus clustering method for gene expression data analysis, while Yu et al. [39] explored knowledge-based cluster ensemble approaches for cancer discovery from biomolecular data. Finally, Yang et al. [40] proposed a consensus clustering approach using a constrained self-organizing map and an improved Cop-Kmeans ensemble for intelligent decision support systems, showcasing the broad applicability of ensemble clustering techniques in various fields.

Bayesian mixture models, with their adaptable densities, are highly attractive for data analysis across various types. The number of clusters K can be inferred directly from the data as a random variable, resulting in joint modelling of K and the clustering process [41–46]. Inference of the number of clusters can be achieved through methods such as the Dirichlet process [41], finite mixture models [42,43], or over-fitting mixture models [44]. These models have found success in a wide range of biological applications, including gene expression profiles [20], cell classification in flow cytometry experiments [21,47] and scRNAseq experiments [48], as well as protein localization estimation [22]. Bayesian mixture models can also be extended to jointly cluster multiple datasets [49,50].

MCMC techniques are the most-used method for executing Bayesian inference, and they are used to build a chain of clusterings. The convergence of the chain is evaluated to see if its behaviour conforms to the asymptotic theory predicted. However, despite the ergodicity of MCMC approaches, individual chains often fail to investigate the complete support of the posterior distribution and have lengthy runtimes. Some MCMC algorithms attempt to overcome these issues, often at the expense of higher computing cost every iteration (see [51,52]).

Preliminaries

Dirichlet processes (DPs) are a family of stochastic processes. A Dirichlet process defines a distribution over probability measures $G : \Theta \to \mathbb{R}^+$, where for any finite partition of Θ, say $\{\boldsymbol{\theta}_k\}_{k=1}^K$, the random vector

$$(G(\boldsymbol{\theta}_1), G(\boldsymbol{\theta}_2) \dots G(\boldsymbol{\theta}_K)) \tag{1}$$

is jointly generalized under a Dirichlet Distribution ($G(\theta)$ is a random variable since G itself is a random measure and is sampled from the Dirichlet process), written as

$$(G(\boldsymbol{\theta}_i), G(\boldsymbol{\theta}_2) \dots G(\boldsymbol{\theta}_K)) \sim \mathrm{Dir}(\alpha G_0(\boldsymbol{\theta}_1), \alpha G_0(\boldsymbol{\theta}_2) \dots \alpha G_0(\boldsymbol{\theta}_K)) \tag{2}$$

where α is called the concentration parameter and G_0 is the base distribution; αG_0 collectively is called the base measure.

Dirichlet processes are really useful in the task of nonparametric clustering via using a mixture of Dirichlet processes (commonly DP mixture models or Infinite mixture models). In fact, a DP mixture model can be seen as an extension of Gaussian mixture models over a nonparametric setting. The basic DP mixture model follows the following generative story:

$$\begin{gathered}\text{Likelihood: } \mathbf{y}_n \mid \boldsymbol{\theta}_n \sim F(\mathbf{y}_n \mid \boldsymbol{\theta}_n) \\ \text{Conditional Prior: } \boldsymbol{\theta}_n \mid G \sim G \\ \text{Hyperparameter: } G \sim \mathrm{DP}(G_0, \alpha)\end{gathered}$$

where $G \sim \mathrm{DP}$ denotes sampling from a Dirichlet process given a base measure.

When we are dealing with DP mixture models for clustering, it helps to integrate out G with respect to the prior on G [53]. Therefore, we can write the clustering problem in an alternate representation, although the underlying model remains the same.

$$\begin{aligned} \text{Likelihood:} \quad & \mathbf{y}_n \mid c_n, \Phi \sim F(\mathbf{y}_n \mid \phi_{c_n}) \\ \text{Latent Distribution: } & c_n \mid \mathbf{p} \sim \text{Discrete } (p_1, p_2 \cdots p_K) \\ \text{Priors: } & \phi_k \sim G_0 \ \mathbf{p} \sim \text{Dir}(p_1, p_2 \ldots p_K) \end{aligned}$$

where c_n is the cluster assignment for the n^{th} point and $\Phi = \{\phi_k\}_{k=1}^{K}$ are the likelihood parameters for each cluster. K denotes the number of clusters, and being a nonparametric model, we assume $K \to \infty$.

If the likelihood and the base distribution are conjugate, we can easily derive a posterior representation for the cluster assignments or the latent classes and use inference techniques such as mean-field VB and Markov chain Monte Carlo. The work [53] also describes various inference methods in the case of a non-conjugate base distribution.

Dirichlet processes are extremely useful for clustering purposes as they do not assume an inherent base distribution, and therefore it is possible to apply Dirichlet process priors over complex models.

3. Methodology

3.1. Finite Mixture of Normals

Suppose we have a set of samples $X_1, X_2, \ldots, X_n$ that can be modelled as

$$p(X_i, |\mu_{1:k}, \tau_{1:k}, q_{1:k}) = \sum_{j}^{k} q_j \mathrm{N}(\mu_j, \tau_j^{-1}), \tag{3}$$

where N(.) denotes the Normal distribution and q_j represents the weight of the j-th component, with $q_j > 0$ and $\sum_{j=1}^{k} q_j = 1$. In addition, let us introduce the latent variable $Z_{1:n}$ to induce the mixture. Thus, we have that $X_i|Z_i = j \sim \mathrm{N}(\mu_j, \tau_j)$. Given the introduction of the latent variable, we can rewrite the likelihood function as

$$p(X_{1:n}|Z_{1:n}, \mu_{1:k}, \tau_{1:k}, Z_{1:n}) = \prod_{j=1}^{k} \prod_{i:I_{(Z_i=j)}}^{n} \mathrm{N}(\mu_j, \tau_j), \tag{4}$$

where $I_{(Z_i=j)} = 1$ if $Z_i = j$, and $I_{(Z_i=j)} = 0$ otherwise. In addition, the latent variables $Z_{1:n} \sim \text{Categorical}(1, q)$. That is,

$$p(Z_{1:n}|q_{1:k}) = \prod_{i=1} \prod_{j=1}^{k} q_j^{I_{(Z_i=j)}} = \prod_{j=1}^{k} q_j^{n_j}, \tag{5}$$

where $n_j = \sum_i^n I_{(Z_i=j)}$ represents the number of observations falling into component j. With that, we can define the joint distribution of $X_{1:n}$ and $Z_{1:n}$ as

$$p(X_{1:n}, Z_{1:n}|\mu_{1:k}, \tau_{1:k}, q_{1:k}) = p(X_{1:n}|Z_{1:n}, \mu_{1:k}, \tau_{1:k}) p(Z_{1:n}|q_{1:k}), \tag{6}$$

$$= \prod_{j=1}^{k} \left[\prod_{i:I_{(Z_i=j)}}^{n} \mathrm{N}(\mu_j, \tau_j) \right] q_j^{n_j}. \tag{7}$$

For notational convenience, let us denote $\theta_k = \{\mu_{1:k}, \tau_{1:k}, q_{1:k}\}$, and $\omega_{ij} = p(Z_i = j|\theta_k, X_{1:n}) / \sum_j p(Z_i = j|\theta_k, X_{1:n})$. Given the expressions above, we have that $Z_{1:n}$ conditioned on $X_{1:n}$ are independent with a probability of classification given by

$$p(Z_i = j|\theta_k, X_{1:n}) \propto p(X_i|Z_i, \mu_j, \tau_j, Z_i) p(Z_i), \tag{8}$$

$$\propto \mathrm{N}(\mu_j, \tau_j) q_j. \tag{9}$$

In the end, we have that

$$p(Z_i|\theta_k, X_{1:n}) \sim \text{Categorical}(1, \omega_{ij}). \tag{10}$$

To estimate the components of the finite mixture of Normals under the Bayesian paradigm, we consider the following priors:

$$\mu_j|\tau \sim \mathrm{N}(m_j, v_j/\tau_j), \tag{11}$$

$$\tau_j \sim \mathrm{G}(a_j, b_j), \tag{12}$$

$$q_{1:k} \sim \text{Dirichlet}(r_1, r_2, ..., r_k). \tag{13}$$

To construct the MCMC structure, we need the full conditionals for μ_j, τ_j, and q_j, which are given below.

$$p(\mu_j|-) \propto p(X_{1:n}|\theta_k, Z_{1:n})p(\mu_j), \mu_j|- \sim \mathrm{N}(M_j, V_j), \tag{14}$$

where

$$M_j = (n_j + 1/v_j)^{-1}\left(\sum_{i:Z_i=j} x_i + m_j/v_j\right) \tag{15}$$

and

$$V_j = \frac{v_j}{(n_j v_j + 1)\tau_j}, \tau|- \sim \mathrm{G}(A_j, B_j), \tag{16}$$

where

$$A_j = \frac{n_j}{2} + a_j, B_j = b_j + \frac{m_j^2}{2v_j} + \frac{\sum_{i:Z_i=j} X_i^2}{2} - \frac{1}{2}(n_j + 1/v_j)M_j^2. \tag{17}$$

From the above, we have:

$$\begin{aligned} p(q_{1:k}|-) &\propto p(Z_{1:n}|q_{1:k})p(q_{1:k}), \\ &\propto \prod_{j=1}^{k} q_j^{n_j} p(q_{1:k}), \\ &\propto \text{Multinomial}(n, q_{1:k}) \times \text{Dirichlet}(r_{1:k}). \end{aligned}$$

$$q_{1:k}|- \sim \text{Dirichlet}(r_{1:k} + n_{1:k}) \tag{18}$$

where $n = \sum_j n_j$.

3.2. Product Partition Models

Let $\mathbf{y} = (y_1, ..., y_n)$ be an n-dimensional vector of a variable we have interest in clustering. We define a partition ρ as a collection of clusters S_j, which are assumed to be non-empty and mutually exclusive. Following [54], the parametric PPM is presented as

$$p(\mathbf{y}, \boldsymbol{\theta}, \rho) = p(\mathbf{y}|\boldsymbol{\theta}, \rho)p(\boldsymbol{\theta})p(\rho), \tag{19}$$

$$= \frac{1}{T}\prod_{j=1}^{k_n}\left[\left(\prod_{i\in S_j} p(y_i|\boldsymbol{\theta}_j)\right)p(\boldsymbol{\theta}_j)c(S_j)\right], \tag{20}$$

where

$$c(S_j) = M \times (|S| - 1)! \tag{21}$$

for some $M > 0$ is the cohesion function. From the above, T can be approximated as

$$T = \sum_{\rho\in\mathcal{P}_n}\prod_{j=1}^{k_n(\rho)} c(S_j) \tag{22}$$

and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ such that $\theta_i = \{\theta_j : i \in S_j\}$.

3.3. Integration of PPM and Consensus Clustering

Following Section 5 in [54], let us consider that

$$\begin{aligned} y_i|\mu_j, \sigma_j^2 &\sim \mathrm{N}(\mu_j, \sigma_j^2), \\ \mu_j|\mu_0, \sigma_0^2 &\sim \mathrm{N}(\mu_0, \sigma_0^2), \\ \sigma_j^2 &\sim \mathrm{U}(0,1), \\ \sigma_0^2 &\sim \mathrm{U}(0,2), \\ \mu_0 &\sim \mathrm{N}(0,100). \end{aligned}$$

Further, let us denote $n_j = |S_j|$ and k as the number of distinct clusters. Below, we present the full conditionals of the quantities/parameters of interest.

$$\begin{aligned} p(\mu_j|-) \propto p(\mathbf{y}|\mu_j, \sigma_j^2)p(\mu_j), &\propto \prod_{i \in S_j} \left[\mathrm{N}(y_i|\mu_j, \sigma_j^2)\right] p(\mu_j), \\ &\propto \exp\left(-\frac{1}{2\sigma_j^2}\sum_{i \in S_j}(y_i - \mu_j)^2\right) \exp\left(-\frac{1}{2\sigma_0^2}(\mu_j - \mu_0)^2\right), \\ &\propto \exp\left\{-\frac{1}{2}\left(\mu^2\left[\frac{n_j}{\sigma_j^2} + \frac{1}{\sigma_0^2}\right] - 2\mu\left[\sum_{i \in S_j}\frac{y_i}{\sigma_j^2} + \frac{\mu_0}{\sigma_0^2}\right]\right)\right\}, \end{aligned}$$

which is

$$\mu_j|- \sim \mathrm{N}\left(\frac{\sigma_j^{-2}\sum_{i \in S_j} y_i + \mu_0/\sigma_0^2}{n_j/\sigma_j^2 + 1/\sigma_0^2}, \frac{1}{n_j/\sigma_j^2 + 1/\sigma_0^2}\right). \tag{23}$$

$$\begin{aligned} p(\sigma_j^2|-) \propto p(\mathbf{y}|\mu_j, \sigma_j^2)p(\sigma_j^2), &\propto \prod_{i \in S_j} \left[\mathrm{N}(y_i|\mu_j, \sigma_j^2)\right] p(\sigma_j^2), \\ &\propto (\sigma_j^2)^{-n_j/2} \exp\left(-\frac{1}{2\sigma_j^2}\sum_{i \in S_j}(y_i - \mu_j)^2\right) \times 1, \end{aligned}$$

which is

$$\sigma_j^2|- \sim \mathrm{IG}\left(\frac{n_j}{2}, \frac{\sum_{i \in S_j}(y_i - \mu_j)^2}{2}\right). \tag{24}$$

$$\begin{aligned} p(\mu_0|-) \propto p(\mu_j|\mu_0, \sigma_0^2)p(\mu_0), &\propto \prod_j \left[\mathrm{N}(\mu_j|\mu_0, \sigma_0^2)\right] p(\mu_0), \\ &\propto \exp\left(-\frac{1}{2\sigma_0^2}\sum_j(\mu_j - \mu_0)^2\right), \end{aligned}$$

which is

$$\mu_0|- \sim \mathrm{N}\left(\frac{\sum_j \mu_j}{k}, \frac{\sigma_0^2}{k}\right). \tag{25}$$

$$p(\sigma_0^2|-) \propto p(\mu_j|\mu_0,\sigma_0^2)p(\sigma_0^2), \propto \prod_j \Big[\mathrm{N}(\mu_j|\mu_0,\sigma_0^2)\Big] p(\sigma_0^2),$$
$$\propto (\sigma_0^2)^{-k/2} \exp\left(-\frac{1}{2\sigma_0^2}\sum_j(\mu_j-\mu_0)^2\right),$$

$$\sigma_0^2|- \sim \mathrm{IG}\left(\frac{k}{2}, \frac{\sum_j(\mu_j-\mu_0)^2}{2}\right). \tag{26}$$

For more details about the marginalization of μ and σ_2 for the full conditional of ρ, see Appendix A. To simulate the posterior distribution of the PPM, we use Algorithm 8 introduced by [53]. This algorithm was proposed in the context of Dirichlet process mixture models, but it can be used for PPMs as well.

Definition 1 (Singleton). *The definition of a singleton is a cluster consisting of only one observation. In contrast, any cluster comprising more than one observation is not considered a singleton.*

Let the cluster labels be denoted as $c_i = j : i \in S_j$ with values ranging from $1, \ldots, k$. For each i, where $i = 1, \cdots, n$, let $h = k + m$, where k represents the number of distinct cluster labels c_j excluding observation i.

- If observation i belongs to a singleton cluster such that $c_i \neq c_j$ for all $j \neq i$, the cluster label c_i is assigned the value of $k+1$. Independent values are then drawn from the prior distribution of μ_j and σ_j^2 for all $k+1 < c \leq h$.
- If observation i does not belong to a singleton cluster such that $c_i = c_j$ for some $j \neq i$, independent values are drawn from the prior distribution of μ_j and σ_j^2 for all $k < c \leq h$.

  For both cases, draw a new value for c_i from $\{1, \cdots, h\}$ using the following probabilities:

$$p(c_i = c|c_{-i}, y_i, \{\mu_c\}, \{\sigma_c^2\}) = \begin{cases} b_i \frac{n_{-i,c}}{n-1+\alpha} p(y_i|\mu_c, \sigma_c^2) & \text{for } 1 \leq c \leq k, \\ \\ b_i \frac{\alpha/m}{n-1+\alpha} p(y_i|\mu_c, \sigma_c^2) & \text{for } k \leq c \leq h, \end{cases} \tag{27}$$

where $n_{-i,c}$ is the number of observations (excluding i) that have $c_j = c$, and α is the Dirichlet process concentration parameter. Change the state to contain only those μ_j and σ_j^2 that are now associated with one or more observations. Here, b_i is an appropriate normalising constant given by (28).

$$b_i^{-1} = \sum_{c=1}^{k} \frac{n_{-i,c}}{n-1+\alpha} p(y_i|\mu_c, \sigma_c^2) + \sum_{c=k}^{h} \frac{\alpha/m}{n-1+\alpha} p(y_i|\mu_c, \sigma_c^2). \tag{28}$$

For all $c \in \{c_1, \cdots, c_n\}$: draw new values from $\mu_j|-$, $\sigma_j^2|-$, $\mu_0|-$, and $\sigma_0^2|-$.

The singleton (Definition 1) is an essential concept in the proposed Bayesian nonparametric clustering approach for several reasons:

- Algorithm efficiency: By identifying singleton clusters, the algorithm can handle them differently in the MCMC sampling process. This distinction allows the algorithm to efficiently explore the space of possible cluster assignments, improving the overall performance of the algorithm and potentially leading to more accurate cluster assignments.
- Flexibility in clustering: The Bayesian nonparametric clustering approach is designed to accommodate an unknown and potentially infinite number of clusters. The notion

of singleton clusters enables the method to contemplate the possibility of new cluster construction, resulting in a more flexible, data-adaptive clustering solution.

- Addressing overfitting: In the MCMC sampling process, the presence of singleton clusters helps prevent overfitting. By allowing new clusters to be created, the algorithm can effectively control the complexity of the clustering model, avoiding the risk of assigning too many data points to the same cluster when they may, in fact, belong to separate clusters.
- Model interpretability: Singleton clusters can provide insights into the underlying structure of the data. Identifying singleton clusters can help reveal potential outliers or unique observations, allowing for a more granular understanding of the data's patterns and relationships.

Generally, the concept of singletons plays a crucial role in the proposed Bayesian nonparametric clustering approach, improving the algorithm's efficiency, flexibility, and interpretability, as well as addressing potential overfitting issues.

Based on all the preceding information, we propose a Bayesian nonparametric clustering method within an MCMC framework. This is illustrated as a flowchart in Figure 1, while the full inner structure is given in Algorithm 1. The algorithm initiates by inputting the y observations and α, m hyperparameters as well as the MCMC iterations of the program. The outputs of the algorithm are the K clusters as well as their means and variances.

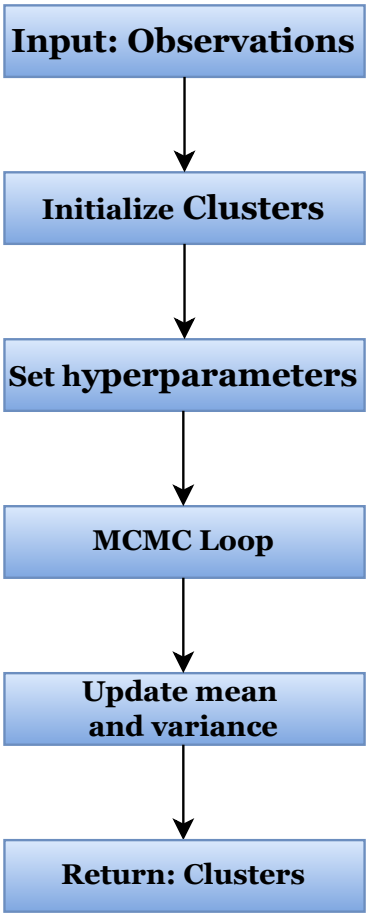


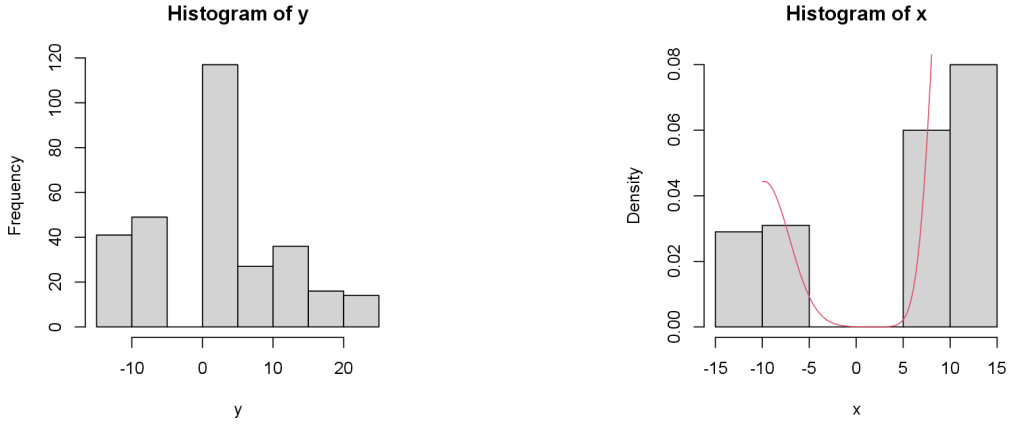
Figure 1. Flowchart of Bayesian nonparametric clustering in MCMC framework.

Algorithm 1 Bayesian nonparametric clustering in MCMC framework

Input: Observations **y**, Hyperparameters α, m, MCMC iterations *MCMCiter*
Output: Clusters and their means and variances
Initialize clusters and assign observations:
    Set up **y**
    Assign all observations into a cluster
Set hyperparameters:
    Set values for α and m
MCMC Loop:
for *mcmc* in 1 to *MCMCiter* **do**
  for i in 1 to n **do**
    if c_i is NOT a singleton **then**
      Draw a new value from c_{-i}, y_i from Equation (27)
    else
      Draw a new value from $c_i \mid c_{-i}, y_i$ from Equation (27)
    end if
    if any cluster is removed **then**
      Adjust the labels to maintain a sequence from 1 to k
    end if
  end for
  Update mean and variance for each cluster j
  Update mean and variance for base distribution μ_0 and σ_0^2
end for
Return: Clusters and their means and variances

4. Experimental Results

In this section, we present the experimental results based on the methods from the preceding sections. Figure 2a shows the frequency histogram of y observations of the data, while Figure 2b shows the histogram of x and the density of the data.



(**a**) Frequency histogram of data.

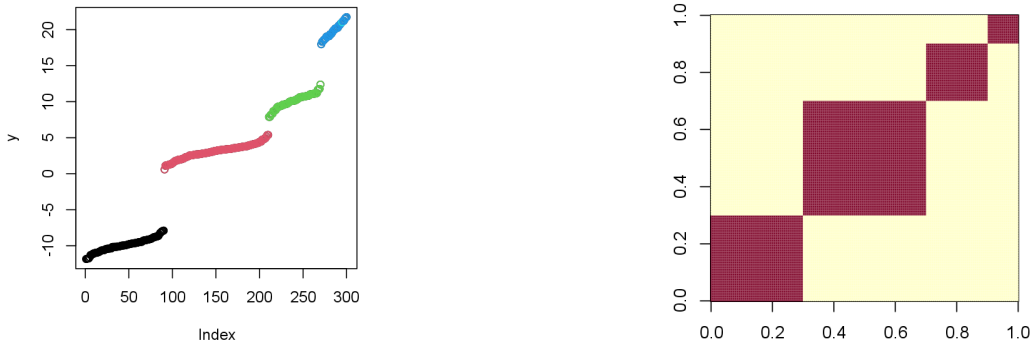
(**b**) Density histogram of data.

Figure 2. Statistical sampling analysis of the PPM model.

Figure 3a shows the MCMC sampling structure and the repetitive areas on y points over 300 iterations, while Figure 3b shows the posterior similarity matrix.

For the experimental results, the following posterior parameters were utilized $\mu_1, \mu_2, \mu_3, \mu_4$, which represent the means of the posterior distributions for each of the four clusters; $\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2$ represent the variances of the posterior distributions for each of the four clusters; $\eta_1, \eta_2, \eta_3, \eta_4$ represent the proportions (or mixing weights) of the posterior distributions for each of the four clusters. The simulation parameters are the means μ and variances σ^2 of the clusters, as well as the mixing proportions η of the Poisson process mixture. The values

represent the estimated proportion of each of the four clusters in the PPM. The standard deviations give the uncertainty around these estimated proportions, with the lower and upper bounds indicating the credible interval. It is worth noting that the results presented in the table are just one possible output of the simulation study, and other simulations with different parameter settings may produce different results. Table 1 shows the results of a simulation study of PPM and consensus clustering.



(**a**) MCMC sampling scheme. (**b**) Posterior similarity matrix.

Figure 3. Statistical sampling analysis of the PPM and consensus clustering.

Table 1. Results of PPM and consensus clustering.

Par. Posterior	Mean	SD	Lower Bound	Upper Bound
μ_1	3.30	0.119	3.11	3.58
μ_2	5.23	0.081	5.06	5.39
μ_3	7.29	0.244	6.64	7.63
μ_4	8.78	0.771	7.36	10.21
σ_1^2	0.159	0.096	0.063	0.416
σ_2^2	0.312	0.081	0.181	0.498
σ_3^2	0.488	0.327	0.122	1.406
σ_4^2	2.82	1.128	1.022	5.229
η_1	0.115	0.023	0.076	0.164
η_2	0.469	0.059	0.336	0.569
η_3	0.222	0.068	0.102	0.377
η_4	0.195	0.074	0.084	0.368

The clustering results of the proposed method in the MCMC framework are shown in Table 2. We conduct tests by adjusting the hyperparameters for different scenarios for which various numbers of clusters are produced, and their means and variances are calculated.

Table 2. Experimental results using the proposed Bayesian nonparametric clustering in MCMC framework.

Test Scenario	Hyperparameters	No. of Clusters	Mean	Variance
1	$\alpha = 1.0, m = 0.5$	4	1.25	0.12
2	$\alpha = 1.2, m = 0.6$	5	1.10	0.09
3	$\alpha = 1.5, m = 0.7$	3	1.45	0.15
4	$\alpha = 0.9, m = 0.4$	6	1.05	0.08
5	$\alpha = 1.3, m = 0.8$	2	1.60	0.17
6	$\alpha = 1.1, m = 0.3$	7	0.95	0.07
7	$\alpha = 1.4, m = 0.9$	4	1.30	0.11
8	$\alpha = 1.6, m = 0.2$	5	1.15	0.10

5. Further Extensions for Big Data Systems

In this section, we propose further extensions for big-data systems and how these methods can be applied to the information science sector. Algorithm 2 is a method for clustering big datasets into groups based on the similarities between observations, with a variety of applications in fields such as information science, big-data systems, and businesses. The algorithm uses Hamiltonian Monte Carlo (HMC) sampling to estimate the posterior distribution of the clusters and their means and variances. One of the main challenges in dealing with big data is the processing time required to analyse and cluster large datasets. By partitioning the MCMC iterations into equal portions and allocating each part to a worker, parallelization of the algorithm helps to surmount this difficulty. This parallelizes the clustering procedure, thereby substantially reducing the processing time.

Algorithm 2 Parallel Bayesian nonparametric clustering using HMC

Input: Observations $\mathbf{y}$, Hyperparameters α, m, MCMC iterations $MCMCiter$, Number of parallel workers $n_{workers}$
Output: Clusters and their means and variances
Initialize clusters and assign observations:
    Set up $\mathbf{y}$
    Assign all observations into a cluster
Set hyperparameters:
    Set values for α and m
Partition MCMC iterations:
    Partition the MCMC iterations into $n_{workers}$ equal parts
MCMC Loop:
for *worker* in 1 to $n_{workers}$ **do**
    Use HMC to sample the posterior distribution of K, μ, and σ^2
        Loop over the assigned MCMC iterations
    Update K, μ, and σ^2 based on the samples
end for
Combine results:
    Combine the results from $n_{workers}$ to obtain the final K, μ, and σ^2
Return: Clusters j and their μ and σ^2

Algorithm 2 boasts a wide range of applications spanning numerous fields such as computer science, big-data systems, and business. In the realm of information science, this algorithm can be employed to consolidate extensive datasets based on similarities, thereby uncovering the data's underlying structure. In the context of business, the algorithm can be applied to cluster customers according to their purchasing behaviours and preferences, yielding valuable insights that enable targeted marketing and sales strategies.

In the domain of big-data systems, the algorithm is capable of clustering massive datasets into groups sharing similar traits, which reduces data storage and processing demands. Moreover, clustering analogous data facilitates parallel processing, consequently boosting efficiency and accelerating processing times. Beyond business and information science, the algorithm can also be utilized in human resource management, where it can group employees based on their skill sets and experiences. This clustering empowers organizations to streamline talent acquisition, leading to increased productivity and employee satisfaction.

Ultimately, the algorithm serves as a potent instrument for analysing vast datasets, generating insightful information, and improving decision-making processes across various sectors. Future research could explore the algorithm's additional applications and assess its performance in diverse contexts.

Algorithm 1 can be adapted in Apache Spark by dividing the data into smaller chunks and distributing them among different nodes in a cluster. The algorithm can be executed in parallel on each node and the results can be combined to get the final clustering results.

Each iteration of the MCMC loop can be implemented as a map-reduce operation, wherein the map operation performs the MCMC updates on a portion of the data and the reduce operation aggregates the results from all the map operations to get the updated clustering results. The map operation includes the operations mentioned in the initial algorithm: updating the μ and σ^2 and adjusting the cluster labels if needed. The reduce operation combines the results from all the map operations and produces the updated clustering results, allowing for efficient parallel processing of large datasets and meeting big-data processing requirements in a scalable manner using Apache Spark.

The results of Algorithm 2 are shown in Table 3. This method is similar to Algorithm 1; however, here we utilize Hamiltonian Monte Carlo instead of MCMC.

Table 3. Experimental results using parallel Bayesian nonparametric clustering with HMC.

Test Scenario	Hyperparameters	No. of Workers	No. of Clusters	Mean	Variance
1	$\alpha = 1.0, m = 0.5$	2	4	1.25	0.12
2	$\alpha = 1.2, m = 0.6$	4	5	1.10	0.09
3	$\alpha = 1.5, m = 0.7$	3	3	1.45	0.15
4	$\alpha = 0.9, m = 0.4$	5	6	1.05	0.08
5	$\alpha = 1.3, m = 0.8$	2	2	1.60	0.17
6	$\alpha = 1.1, m = 0.3$	6	7	0.95	0.07
7	$\alpha = 1.4, m = 0.9$	3	4	1.30	0.11
8	$\alpha = 1.6, m = 0.2$	4	5	1.15	0.10

Integrating Algorithm 2 with Apache Spark, as demonstrated in Algorithm 3, allows for efficient and scalable processing of massive datasets in a distributed computing environment. Leveraging Spark's capabilities, the algorithm can handle the increasing demands of big-data systems by dividing the data into P partitions and executing parallel MCMC iterations across multiple nodes within a cluster. This approach enables faster processing times and accommodates growing data sizes while maintaining the accuracy and effectiveness of the clustering method. Additionally, the implementation in Spark paves the way for further development and optimization of Bayesian nonparametric clustering techniques in distributed computing environments, enabling better insights and more effective decision-making processes in various application domains. Finally, the results of Algorithm 3 are shown in Table 4.

Table 4. Experimental results using the Bayesian nonparametric clustering in MCMC on Apache Spark.

Test Scenario	Hyperparameters	No. of Partitions	No. of Clusters	Mean	Variance
1	$\alpha = 1.0, m = 0.5$	2	4	1.25	0.12
2	$\alpha = 1.2, m = 0.6$	4	5	1.10	0.09
3	$\alpha = 1.5, m = 0.7$	3	3	1.45	0.15
4	$\alpha = 0.9, m = 0.4$	5	6	1.05	0.08
5	$\alpha = 1.3, m = 0.8$	2	2	1.60	0.17
6	$\alpha = 1.1, m = 0.3$	6	7	0.95	0.07
7	$\alpha = 1.4, m = 0.9$	3	4	1.30	0.11
8	$\alpha = 1.6, m = 0.2$	4	5	1.15	0.10

Ultimately, we present the performance of all of the proposed algorithms in Table 5. We utilize famous real-world datasets such as CIFAR-10, MNIST, and Iris. The hyperparameters for each experiment are α and m, and we evaluate the clustering accuracy and the time required for the method to complete. The time column is measured in seconds. As we can observe from the table, the fastest clustering was on the Iris dataset, but note that this dataset is smaller in terms of size compared with the other two. As for the method with the highest accuracy, it appears that BNP-MCMC (Algorithm 1) has the highest accuracy on the Iris dataset while having satisfactory accuracy on the other methods. Moreover, the

parallel method appears to further improve the accuracy by some decimal points. Lastly, the Spark version of the proposed method produces similar results but outperforms the other methods with regards to time.

Algorithm 3 Bayesian nonparametric clustering in MCMC on Apache Spark

Input: Observations $\mathbf{y}$, Hyperparameters α, m, MCMC iterations
Output: Clusters j and their μ and σ^2
  Divide the observations $\mathbf{y}$ into P partitions
Initialize clusters and assign observations:
  Assign all observations in each partition into a cluster
  Set hyperparameter values for α and m
for *mcmc* in 1 to *MCMCiter* **do**
  Parallel Processing:
  for p in 1 to P **do**
    for i in 1 to n in partition p **do**
      if c_i is NOT a singleton **then**
        Draw a new value from c_{-i}, y_i from Equation (27)
      else
        Draw a new value from $c_i \mid c_{-i}, y_i$ from Equation (27)
      end if
      if any cluster is removed **then**
        Adjust the labels to maintain a sequence from 1 to k
      end if
    end for
    Update mean and variance for each cluster j in partition p
  end for
  Merge the updated μ and σ^2 from partitions to obtain the global values
  Update mean and variance for base distribution μ_0 and σ_0^2
end for
Return: Clusters j and their μ and σ^2

Table 5. Experimental results for real-world datasets using different algorithms and hyperparameters.

Dataset	Algorithm	α	m	Clustering Accuracy	Time (s)
CIFAR-10	BNP-MCMC	1.0	1.0	0.75	120
		1.5	1.0	0.78	130
		1.0	1.5	0.77	125
	Parallel BNP-HMC	1.0	1.0	0.76	100
		1.5	1.0	0.80	110
		1.0	1.5	0.79	105
	BNP-Spark	1.0	1.0	0.74	90
		1.5	1.0	0.78	95
		1.0	1.5	0.76	92
MNIST	BNP-MCMC	1.0	1.0	0.85	150
		1.5	1.0	0.88	160
		1.0	1.5	0.87	155
	Parallel BNP-HMC	1.0	1.0	0.86	130
		1.5	1.0	0.89	140
		1.0	1.5	0.88	135
	BNP-Spark	1.0	1.0	0.84	120
		1.5	1.0	0.88	125
		1.0	1.5	0.86	122

Table 5. *Cont.*

Dataset	Algorithm	α	m	Clustering Accuracy	Time (s)
Iris	BNP-MCMC	1.0	1.0	0.95	10
		1.5	1.0	0.96	11
		1.0	1.5	0.96	10.5
	Parallel BNP-HMC	1.0	1.0	0.95	9
		1.5	1.0	0.96	9.5
		1.0	1.5	0.96	9.2
	BNP-Spark	1.0	1.0	0.94	8
		1.5	1.0	0.96	8.5
		1.0	1.5	0.95	8.3

6. Conclusions and Future Work

In the context of this work, we have proposed a novel approach for clustering complex datasets using Bayesian nonparametric ensemble methods that have the potential to revolutionize the field of information science and big-data management. Our approach generates a robust and accurate final clustering solution, addressing the challenges related to determining the number of clusters and managing high-dimensional datasets. By combining the strengths of PPMs, MCMC, and consensus clustering, our proposed method provides a more comprehensive and informative clustering solution, enabling efficient and effective management of massive datasets.

Further study could concentrate on improving the proposed method in a variety of ways. One area of research could be the creation of more efficient algorithms for computing the consensus matrix, thereby reducing the execution time of the approach. In addition, the incorporation of additional data sources into the model could result in more exhaustive clustering solutions. In addition, the use of more adaptive methods for determining optimal hyperparameters and the development of sophisticated techniques for dealing with data noise and outliers could improve the efficacy of the approach. Improving the comprehensibility and clarity of the proposed method for non-expert users could foster its widespread adoption across a more extensive array of applications and industries.

Exploration of the proposed implementation in various real-world contexts presents a promising avenue for future research. For example, the technique could be deployed to discern and classify diverse tumour types by examining their presentation in medical imaging data. Within the realm of natural language processing, the method could be harnessed to categorize text into specific topics. Additionally, in the financial industry, the suggested approach could be employed to cluster financial data, facilitating the recognition of patterns for stock price forecasting and fraud detection.

In summary, the proposed strategy of employing Bayesian nonparametric ensemble methods for clustering intricate datasets holds substantial promise for the proficient and effective handling of enormous datasets and has the potential to transform the landscape of information science and big-data management. Future research in this area could lead to significant advancements in the field, enabling the solution of increasingly complex problems in various disciplines.

Author Contributions: C.K., A.K., K.C.G., M.A. and S.S. conceived of the idea, designed and performed the experiments, analysed the results, drafted the initial manuscript and revised the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Appendix A

In this section, we try to marginalise μ and σ_2 for the full conditional of ρ.

$$
\begin{aligned}
p(\rho|\mathbf{y}) &\propto \prod_{j=1}^{k_n}\left\{c(S_j)\int\int\prod_{i\in S_j}p(y_i|\mu_j,\sigma_j^2)p(\mu_j)p(\sigma_j^2)d\mu_j d\sigma_j^2\right\},\\
&\propto \prod_{j=1}^{k_n}\left\{(|S_j|-1)!\int\int\prod_{i\in S_j}N(y_i|\mu_j,\sigma_j^2)N(\mu_j|\mu_0,\sigma_0^2)U(\sigma_j^2|0,1)d\mu_j d\sigma_j^2\right\},\\
&\propto \prod_{j=1}^{k_n}\left\{(|S_j|-1)!\right\}\times\\
&\quad\times\prod_{j=1}^{k_n}\int\int(\sigma^j)^{-n_j/2}\exp\left\{-\frac{1}{2\sigma_j^2}\sum_{i\in S_j}(y_i-\mu_j)^2\right\}\exp\left\{-\frac{1}{2\sigma_0^2}\sum_{i\in S_j}(\mu_j-\mu_0)^2\right\}d\mu_j d\sigma_j^2,\\
&\propto \prod_{j=1}^{k_n}\left\{(|S_j|-1)!\right\}\times\\
&\quad\times\prod_{j=1}^{k_n}\int\int(\sigma^j)^{-n_j/2}\exp\left\{-\frac{1}{2}\left(\mu_j^2\left[\frac{|S_j|}{\sigma_j^2}+\frac{1}{\sigma_0^2}\right]-2\mu_j\left[\frac{\sum_{i\in S_j}y_i}{\sigma_j^2}+\frac{\mu_0}{\sigma_0^2}\right]\right)\right\}d\mu_j d\sigma_j^2\\
&\propto \prod_{j=1}^{k_n}\left\{(|S_j|-1)!\right\}\times\\
&\quad\times\prod_{j=1}^{k_n}\int(\sigma_j^2)^{-n_j/2}\left(n_j/\sigma_j^2+1/\sigma_0^2\right)^{-1/2}\exp\left\{-\frac{1}{2\sigma_j^2}\sum_{i\in S_j}y_i^2+\frac{\left(\sigma_j^{-2}\sum_{i\in S_j}y_i+\mu_0/\sigma_0^2\right)^2}{n_j/\sigma_j^2+1/\sigma_0^2}\right\}d\sigma_j^2
\end{aligned}
$$

The expression for $p(\rho|\mathbf{y})$ is a product of integrals, where each integral is over the variables μ_j and σ_j^2. The integral of a Gaussian distribution is a Gaussian distribution with a modified mean and variance. Hence, we can simplify the expression as follows:

$$
p(\rho|\mathbf{y})\propto\prod_{j=1}^{k_n}(|S_j|-1)!\times\prod_{j=1}^{k_n}\left(\frac{n_j}{\sigma^2 j}+\frac{1}{\sigma^2 0}\right)^{-1/2}\int(\sigma_j^2)^{-\frac{n_j}{2}}\exp\left(-\frac{1}{2\sigma^2 j}\sum_{i\in S_j}y_i^2+\frac{\left(\frac{\sigma^{-2}j\sum_{i\in S_j}y_i+\mu_0}{\sigma^2 0}\right)^2}{\frac{nj}{\sigma^2 j}+\frac{1}{\sigma^2 0}}\right)d\sigma_j^2. \tag{A1}
$$

References

1. Coleman, S.; Kirk, P.D.; Wallace, C. Consensus clustering for Bayesian mixture models. *BMC Bioinform.* **2022**, *23*, 1–21. [CrossRef] [PubMed]
2. Lock, E.F.; Dunson, D.B. Bayesian consensus clustering. *Bioinformatics* **2013**, *29*, 2610–2616. [CrossRef] [PubMed]
3. Jain, S.; Neal, R.M. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *J. Comput. Graph. Stat.* **2004**, *13*, 158–182. [CrossRef]
4. Jain, S.; Neal, R.M. Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Anal.* **2007**, *2*, 445–472. [CrossRef]
5. Bouchard-Côté, A.; Doucet, A.; Roth, A. Particle Gibbs split-merge sampling for Bayesian inference in mixture models. *J. Mach. Learn. Res.* **2017**, *18*, 868–906.
6. Caruso, G.; Gattone, S.A.; Balzanella, A.; Di Battista, T. Cluster Analysis: An Application to a Real Mixed-Type Data Set. In *Models and Theories in Social Systems*; Springer International Publishing: Cham, Switzerland, 2019; pp. 525–533. [CrossRef]
7. Ezugwu, A.E.; Ikotun, A.M.; Oyelade, O.O.; Abualigah, L.; Agushaka, J.O.; Eke, C.I.; Akinyelu, A.A. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104743. [CrossRef]
8. Jiang, D.; Tang, C.; Zhang, A. Cluster analysis for gene expression data: A survey. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1370–1386. [CrossRef]
9. Huang, D.; Wang, C.D.; Lai, J.H. Locally weighted ensemble clustering. *IEEE Trans. Cybern.* **2017**, *48*, 1460–1473. [CrossRef]

10. Ghaemi, R.; Sulaiman, M.N.; Ibrahim, H.; Mustapha, N. A survey: Clustering ensembles techniques. *Int. J. Comput. Inf. Eng.* **2009**, *3*, 365–374.
11. Can, C.E.; Ergun, G.; Soyer, R. Bayesian analysis of proportions via a hidden Markov model. *Methodol. Comput. Appl. Probab.* **2022**, *24*, 3121–3139. [CrossRef]
12. Karras, A.; Karras, C.; Schizas, N.; Avlonitis, M.; Sioutas, S. AutoML with Bayesian Optimizations for Big Data Management. *Information* **2023**, *14*, 223. [CrossRef]
13. Schizas, N.; Karras, A.; Karras, C.; Sioutas, S. TinyML for Ultra-Low Power AI and Large Scale IoT Deployments: A Systematic Review. *Future Internet* **2022**, *14*, 363. [CrossRef]
14. Zhu, Z.; Xu, M.; Ke, J.; Yang, H.; Chen, X.M. A Bayesian clustering ensemble Gaussian process model for network-wide traffic flow clustering and prediction. *Transp. Res. Part Emerg. Technol.* **2023**, *148*, 104032. [CrossRef]
15. Greve, J.; Grün, B.; Malsiner-Walli, G.; Frühwirth-Schnatter, S. Spying on the prior of the number of data clusters and the partition distribution in Bayesian cluster analysis. *Aust. N. Z. J. Stat.* **2022**, *64*, 205–229. [CrossRef]
16. Monti, S.; Tamayo, P.; Mesirov, J.; Golub, T. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.* **2003**, *52*, 91–118. [CrossRef]
17. Fraley, C.; Raftery, A.E. Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc.* **2002**, *97*, 611–631. [CrossRef]
18. Huang, D.; Wang, C.D.; Peng, H.; Lai, J.; Kwoh, C.K. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *51*, 508–520. [CrossRef]
19. Cai, X.; Huang, D. Link-Based Consensus Clustering with Random Walk Propagation. In Proceedings of the Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, 8–12 December 2021; Proceedings, Part V 28; Springer: Berlin/Heidelberg, Germany, 2021; pp. 693–700.
20. Medvedovic, M.; Sivaganesan, S. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics* **2002**, *18*, 1194–1206. [CrossRef]
21. Chan, C.; Feng, F.; Ottinger, J.; Foster, D.; West, M.; Kepler, T.B. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytom. Part A J. Int. Soc. Anal. Cytol.* **2008**, *73*, 693–701. [CrossRef]
22. Crook, O.M.; Mulvey, C.M.; Kirk, P.D.; Lilley, K.S.; Gatto, L. A Bayesian mixture modelling approach for spatial proteomics. *PLoS Comput. Biol.* **2018**, *14*, e1006516. [CrossRef]
23. Yan, J.; Liu, W. An ensemble clustering approach (consensus clustering) for high-dimensional data. *Secur. Commun. Netw.* **2022**, *2022*, 5629710. [CrossRef]
24. Niu, X.; Zhang, C.; Zhao, X.; Hu, L.; Zhang, J. A multi-view ensemble clustering approach using joint affinity matrix. *Expert Syst. Appl.* **2023**, *216*, 119484. [CrossRef]
25. Huang, Q.; Gao, R.; Akhavan, H. An ensemble hierarchical clustering algorithm based on merits at cluster and partition levels. *Pattern Recognit.* **2023**, *136*, 109255. [CrossRef]
26. Zhou, P.; Wang, X.; Du, L.; Li, X. Clustering ensemble via structured hypergraph learning. *Inf. Fusion* **2022**, *78*, 171–179. [CrossRef]
27. Zamora, J.; Sublime, J. An Ensemble and Multi-View Clustering Method Based on Kolmogorov Complexity. *Entropy* **2023**, *25*, 371. [CrossRef]
28. Huang, D.; Wang, C.D.; Lai, J.H.; Kwoh, C.K. Toward Multidiversified Ensemble Clustering of High-Dimensional Data: From Subspaces to Metrics and Beyond. *IEEE Trans. Cybern.* **2022**, *52*, 12231–12244. [CrossRef] [PubMed]
29. Huang, D.; Wang, C.D.; Wu, J.S.; Lai, J.H.; Kwoh, C.K. Ultra-Scalable Spectral Clustering and Ensemble Clustering. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1212–1226. [CrossRef]
30. Wang, L.; Luo, J.; Wang, H.; Li, T. Markov clustering ensemble. *Knowl.-Based Syst.* **2022**, *251*, 109196. [CrossRef]
31. Huang, D.; Wang, C.D.; Lai, J.H. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Trans. Knowl. Data Eng.* **2023**. [CrossRef]
32. Nie, X.; Qin, D.; Zhou, X.; Duo, H.; Hao, Y.; Li, B.; Liang, G. Clustering ensemble in scRNA-seq data analysis: Methods, applications and challenges. *Comput. Biol. Med.* **2023**, 106939. [CrossRef]
33. Boongoen, T.; Iam-On, N. Cluster ensembles: A survey of approaches with recent extensions and applications. *Comput. Sci. Rev.* **2018**, *28*, 1–25. [CrossRef]
34. Troyanovsky, S.M. Adherens junction: The ensemble of specialized cadherin clusters. *Trends Cell Biol.* **2022**, *33*, 374–387. [CrossRef] [PubMed]
35. Zhang, D.; Zhu, Y. ECBN: Ensemble Clustering based on Bayesian Network inference for Single-cell RNA-seq Data. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5884–5888.
36. Hu, L.; Zhou, J.; Qiu, Y.; Li, X. An Ultra-Scalable Ensemble Clustering Method for Cell Type Recognition Based on scRNA-seq Data of Alzheimer's Disease. In Proceedings of the 3rd Asia-Pacific Conference on Image Processing, Electronics and Computers, Dalian, China, 14–16 April 2022; pp. 275–280.
37. Bian, C.; Wang, X.; Su, Y.; Wang, Y.; Wong, K.C.; Li, X. scEFSC: Accurate single-cell RNA-seq data analysis via ensemble consensus clustering based on multiple feature selections. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 2181–2197. [CrossRef] [PubMed]
38. Wang, Y.; Pan, Y. Semi-supervised consensus clustering for gene expression data analysis. *BioData Min.* **2014**, *7*, 1–13. [CrossRef] [PubMed]

39. Yu, Z.; Wongb, H.S.; You, J.; Yang, Q.; Liao, H. Knowledge based cluster ensemble for cancer discovery from biomolecular data. *IEEE Trans. Nanobiosci.* **2011**, *10*, 76–85.
40. Yang, Y.; Tan, W.; Li, T.; Ruan, D. Consensus clustering based on constrained self-organizing map and improved Cop-Kmeans ensemble in intelligent decision support systems. *Knowl.-Based Syst.* **2012**, *32*, 101–115. [CrossRef]
41. Ferguson, T.S. A Bayesian analysis of some nonparametric problems. *Ann. Stat.* **1973**, *1*, 209–230. [CrossRef]
42. Miller, J.W.; Harrison, M.T. Mixture models with a prior on the number of components. *J. Am. Stat. Assoc.* **2018**, *113*, 340–356. [CrossRef]
43. Richardson, S.; Green, P.J. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1997**, *59*, 731–792. [CrossRef]
44. Rousseau, J.; Mengersen, K. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2011**, *73*, 689–710. [CrossRef]
45. Law, M.; Jain, A.; Figueiredo, M. Feature selection in mixture-based clustering. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2002; Volume 15.
46. Scrucca, L.; Fop, M.; Murphy, T.B.; Raftery, A.E. mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *R J.* **2016**, *8*, 289. [CrossRef] [PubMed]
47. Hejblum, B.P.; Alkhassim, C.; Gottardo, R.; Caron, F.; Thiébaut, R. Sequential Dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data. *Ann. Appl. Stat.* **2019**, *13*, 638–660. [CrossRef]
48. Prabhakaran, S.; Azizi, E.; Carr, A.; Pe'er, D. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; PMLR: Baltimore, MD, USA, 2016; pp. 1070–1079.
49. Gabasova, E.; Reid, J.; Wernisch, L. Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLoS Comput. Biol.* **2017**, *13*, e1005781. [CrossRef] [PubMed]
50. Kirk, P.; Griffin, J.E.; Savage, R.S.; Ghahramani, Z.; Wild, D.L. Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics* **2012**, *28*, 3290–3297. [CrossRef] [PubMed]
51. Karras, C.; Karras, A.; Avlonitis, M.; Giannoukou, I.; Sioutas, S. Maximum Likelihood Estimators on MCMC Sampling Algorithms for Decision Making. In Proceedings of the AIAI 2022 IFIP WG 12.5 International Workshops, Artificial Intelligence Applications and Innovations, Crete, Greece, 17–20 June 2022; Maglogiannis, I., Iliadis, L., Macintyre, J., Cortez, P., Eds.; Springer: Cham, Switzerland, 2022; pp. 345–356.
52. Karras, C.; Karras, A.; Avlonitis, M.; Sioutas, S. An Overview of MCMC Methods: From Theory to Applications. In Proceedings of the AIAI 2022 IFIP WG 12.5 International Workshops, Artificial Intelligence Applications and Innovations, Crete, Greece, 17–20 June 2022; Maglogiannis, I., Iliadis, L., Macintyre, J., Cortez, P., Eds.; Springer: Cham, Switzerland, 2022; pp. 319–332.
53. Neal, R.M. Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Stat.* **2000**, *9*, 249–265.
54. Quintana, F.A.; Loschi, R.H.; Page, G.L. Bayesian Product Partition Models. *Wiley StatsRef Stat. Ref. Online* **2018**, *1*, 1–15.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.