
Bayesian Hierarchical Clustering with Exponential Family: Small-Variance Asymptotics and Reducibility

Juho Lee and Seungjin Choi

Department of Computer Science and Engineering
Pohang University of Science and Technology
77 Cheongam-ro, Nam-gu, Pohang 790-784, Korea
{stonecold, seungjin}@postech.ac.kr

Abstract

Bayesian hierarchical clustering (BHC) is an agglomerative clustering method, where a probabilistic model is defined and its marginal likelihoods are evaluated to decide which clusters to merge. While BHC provides a few advantages over traditional distance-based agglomerative clustering algorithms, successive evaluation of marginal likelihoods and careful hyperparameter tuning are cumbersome and limit the scalability. In this paper we relax BHC into a non-probabilistic formulation, exploring small-variance asymptotics in conjugate-exponential models. We develop a novel clustering algorithm, referred to as *relaxed BHC* (RBHC), from the asymptotic limit of the BHC model that exhibits the scalability of distance-based agglomerative clustering algorithms as well as the flexibility of Bayesian nonparametric models. We also investigate the reducibility of the dissimilarity measure emerged from the asymptotic limit of the BHC model, allowing us to use scalable algorithms such as the nearest neighbor chain algorithm. Numerical experiments on both synthetic and real-world datasets demonstrate the validity and high performance of our method.

1 INTRODUCTION

Agglomerative hierarchical clustering, which is one of the most popular algorithms in cluster analysis, builds a binary tree representing the cluster structure of a dataset (Duda et al., 2001). Given a dataset and a dissimilarity mea-

sure between clusters, agglomerative hierarchical clustering starts from leaf nodes corresponding to individual data points and successively merges pairs of nodes with smallest dissimilarities to complete a binary tree.

Bayesian hierarchical clustering (BHC) (Heller and Ghahrahmani, 2005) is a probabilistic alternative of agglomerative hierarchical clustering. BHC defines a generative model on binary trees and compute the probability that nodes are merged under that generative model to evaluate the (dis)similarity between the nodes. Since this (dis)similarity is written as a probability, one can naturally decide a level, where to stop merging according to this probability. Hence, unlike traditional agglomerative clustering algorithms, BHC has a flexibility to infer a proper number of clusters for given data. The source of this flexibility is Dirichlet process mixtures (DPM) (Ferguson, 1973; Antoniak, 1974) used to define the generative model of binary trees. BHC was shown to provide a tight lower bound on the marginal likelihood of DPM (Heller and Ghahrahmani, 2005; Wallach et al., 2010) and to be an alternative posterior inference algorithm for DPM. However, when evaluating the dissimilarity between nodes, one has to repeatedly compute the marginal likelihood of clusters and careful tuning of hyperparameters are required.

In this paper, we study BHC when the underlying distributions are conjugate exponential families. Our contributions is twofold. First, we derive a non-probabilistic relaxation of BHC, referred to as RBHC, by performing *small variance asymptotics*, i.e., letting the variance of the underlying distribution in the model go to zero. To this end, we use the technique inspired by the recent work (Kulis and Jordan, 2012; Jiang et al., 2012), where the Gibbs sampling algorithm for DPM with conjugate exponential family was shown to approach a k -means-like hard clustering algorithm in the small variance limit. The dissimilarity measure in RBHC is of a simpler form, compared to the one in the original BHC. It does not require careful tuning of hyperparameters, and yet has the flexibility of the original BHC to infer a proper number of the clusters in data. It turns

Appearing in Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38. Copyright 2015 by the authors.

out to be equivalent to the dissimilarity proposed in (Telgarsky and Dasgupta, 2012), which was derived in different perspective, minimizing a cost function involving Bregman information (Banerjee et al., 2005). Second, we study the *reducibility* (Bruynooghe, 1978) of the dissimilarity measure in RBHC. If the dissimilarity is reducible, one can use *nearest-neighbor chain algorithm* (Bruynooghe, 1978) to build a binary tree with much smaller complexities, compared to the greedy algorithm. The nearest neighbor chain algorithm builds a binary tree of n data points in $O(n^2)$ time and $O(n)$ space, while the greedy algorithm does in $O(n^2 \log n)$ time and $O(n^2)$ space. We argue is that even though we cannot guarantee that the dissimilarity in RBHC is always reducible, it satisfies the reducibility in many cases, so it is fine to use the nearest-neighbor chain algorithm in practice to speed up building a binary tree using the RBHC. We also present the conditions where the dissimilarity in RBHC is more likely to be reducible.

2 BACKGROUND

We briefly review agglomerative hierarchical clustering, Bayesian hierarchical clustering, and Bregman clustering, on which we base the development of our clustering algorithm RBHC. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of n data points. Denote by $[n] = \{1, \dots, n\}$ a set of indices. A *partition* π_n of $[n]$ is a set of disjoint nonempty subsets of $[n]$ whose union is $[n]$. The set of all possible partitions of $[n]$ is denoted by Π_n . For instance, in the case of $[5] = \{1, 2, 3, 4, 5\}$, an exemplary random partition that consists of three clusters is $\pi_5 = \{\{1\}, \{2, 4\}, \{3, 5\}\}$; its members are indexed by $c \in \pi_5$. Data points in cluster c is denoted by $\mathbf{X}_c \stackrel{\text{def}}{=} \{\mathbf{x}_i | i \in c\}$ for $c \in \pi_n$. Dissimilarity between $c_0 \in \pi_n$ and $c_1 \in \pi_n$ is given by $d(c_0, c_1)$.

2.1 Agglomerative Hierarchical Clustering

Given \mathbf{X} and $d(\cdot, \cdot)$, a common approach to building a binary tree for agglomerative hierarchical clustering is the greedy algorithm, where pairs of nodes are merged as one moves up the hierarchy, starting in its leaf nodes (**Algorithm 1**). A naive implementation of the greedy algorithm requires $O(n^3)$ in time since each iteration needs $O(n^2)$ to find the pair of closest nodes and the algorithm runs over n iterations. It requires $O(n^2)$ in space to store pairwise dissimilarities. The time complexity can be reduced to $O(n^2 \log n)$ with priority queue, and can be reduced further for some special cases; for example, in single linkage clustering where

$$d(c_0, c_1) = \min_{i \in c_0, j \in c_1} d(\{i\}, \{j\}),$$

the time complexity is $O(n^2)$ since building a binary tree is equivalent to finding the minimum spanning tree in the dissimilarity graph. Also, in case of the centroid linkage

clustering where the distance between clusters are defined as the Euclidean distance between the centers of the clusters, one can reduce the time complexity to $O(kn^2)$ where $k \ll n$ (Day and Edelsbrunner, 1984).

Algorithm 1 Greedy algorithm for agglomerative hierarchical clustering

Input: $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n, d(\cdot, \cdot)$.

Output: Binary tree.

- 1: Assign data points to leaves.
 - 2: Compute $d(\{i\}, \{j\})$ for all $i, j \in [n]$.
 - 3: **while** the number of nodes > 1 **do**
 - 4: Find a pair (c_0, c_1) with minimum $d(c_0, c_1)$.
 - 5: Merge $c \leftarrow c_0 \cup c_1$ and compute $d(c, c')$ for all $c' \neq c$.
 - 6: **end while**
-

Algorithm 2 Nearest neighbor chain algorithm

Input: $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, reducible $d(\cdot, \cdot)$.

Output: Binary tree.

- 1: **while** the number of nodes > 1 **do**
 - 2: Pick any c_0 .
 - 3: Build a chain $c_1 = \text{nn}(c_0), c_2 = \text{nn}(c_1), \dots$, where $\text{nn}(c) \stackrel{\text{def}}{=} \arg \min_{c'} d(c, c')$. Extend the chain until $c_i = \text{nn}(c_{i-1})$ and $c_{i-1} = \text{nn}(c_i)$.
 - 4: Merge $c \leftarrow c_i \cup c_{i-1}$.
 - 5: If $i \geq 2$, go to line 3 and extend the chain from c_{i-2} . Otherwise, go to line 2.
 - 6: **end while**
-

2.2 Reducibility

Two nodes c_0 and c_1 are *reciprocal nearest neighbors* (RNNs) if the dissimilarity $d(c_0, c_1)$ is minimal among all dissimilarities from c_0 to elements in π_n and also minimal among all dissimilarities from c_1 . Dissimilarity $d(\cdot, \cdot)$ is *reducible* (Bruynooghe, 1978), if for any $c_0, c_1, c_2 \in \pi_n$,

$$\begin{aligned} d(c_0, c_1) &\leq \min\{d(c_0, c_2), d(c_1, c_2)\} \\ \Rightarrow \min\{d(c_0, c_2), d(c_1, c_2)\} &\leq d(c_0 \cup c_1, c_2). \end{aligned} \quad (1)$$

The reducibility ensures that if c_0 and c_1 are reciprocal nearest neighbors (RNNs), then this pair of nodes are the closest pair that the greedy algorithm will eventually find by searching on an entire space. Thus, the reducibility saves the effort of finding a pair of nodes with minimal distance. Assume that (c_0, c_1) are RNNs. Merging (c_0, c_1) become problematic only if, for other RNNs (c_2, c_3) , merging (c_2, c_3) changes the nearest neighbor of c_0 (or c_1) to $c_2 \cup c_3$. However this does not happen since

$$\begin{aligned} d(c_2, c_3) &\leq \min\{d(c_2, c_0), d(c_3, c_0)\} \\ \Rightarrow \min\{d(c_2, c_0), d(c_3, c_0)\} &\leq d(c_2 \cup c_3, c_0) \\ \Rightarrow d(c_0, c_1) &\leq d(c_2 \cup c_3, c_0). \end{aligned}$$

The nearest neighbor chain algorithm (Bruynooghe, 1978) enjoys this property and find pairs of nodes to merge by

following paths in the nearest neighbor graph of the nodes until the paths terminate in pairs of mutual nearest neighbors (**Algorithm 2**). The time and space complexity of the nearest neighbor chain algorithm are $O(n^2)$ and $O(n)$, respectively. The reducible dissimilarity includes those of single linkage and Ward’s method (Ward, 1963).

2.3 Agglomerative Bregman Clustering

Agglomerative clustering with Bregman divergence (Bregman, 1967) as a dissimilarity measure was recently developed in (Telgarsky and Dasgupta, 2012), where the clustering was formulated as the minimization of the sum of cost based on the Bregman divergence between the elements in a cluster and center of the cluster. This cost is closely related with the Bregman Information used for Bregman hard clustering (Banerjee et al., 2005)). In (Telgarsky and Dasgupta, 2012), the dissimilarity between two clusters (c_0, c_1) is defined as the change of cost function when they are merged. As will be shown in this paper, this dissimilarity turns out to be identical to the one we derive from the asymptotic limit of BHC. Agglomerative clustering with Bregman divergence showed better accuracies than traditional agglomerative hierarchical clustering algorithms on various real datasets (Telgarsky and Dasgupta, 2012).

2.4 Bayesian Hierarchical Clustering

Denote by \mathcal{T}_c a tree whose leaves are \mathbf{X}_c for $c \in \pi_n$. A binary tree constructed by BHC (Heller and Ghahrahmani, 2005) explains the generation of \mathbf{X}_c with two hypotheses compared in considering each merge: (1) the first hypothesis \mathcal{H}_c where all elements in \mathbf{X}_c were generated from a single cluster c ; (2) the alternative hypothesis where \mathbf{X}_c has two sub-clusters \mathbf{X}_{c_0} and \mathbf{X}_{c_1} , each of which is associated with subtrees \mathcal{T}_{c_0} and \mathcal{T}_{c_1} , respectively. Thus, the probability of \mathbf{X}_c in tree \mathcal{T}_c is written as:

$$p(\mathbf{X}_c|\mathcal{T}_c) = p(\mathcal{H}_c)p(\mathbf{X}_c|\mathcal{H}_c) + \{1 - p(\mathcal{H}_c)\}p(\mathbf{X}_{c_0}|\mathcal{T}_{c_0})p(\mathbf{X}_{c_1}|\mathcal{T}_{c_1}), \quad (2)$$

where the prior $p(\mathcal{H}_c)$ is recursively defined as

$$\gamma_{\{i\}} \stackrel{\text{def}}{=} \alpha, \quad \gamma_c \stackrel{\text{def}}{=} \alpha\Gamma(|c|) + \gamma_{c_0}\gamma_{c_1}, \quad (3)$$

$$p(\mathcal{H}_c) \stackrel{\text{def}}{=} \alpha\Gamma(|c|)/\gamma_c, \quad (4)$$

and the likelihood of \mathbf{X}_c under \mathcal{H}_c is given by

$$p(\mathbf{X}_c|\mathcal{H}_c) \stackrel{\text{def}}{=} \int \left\{ \prod_{i \in c} p(\mathbf{x}_i|\boldsymbol{\theta}) \right\} p(d\boldsymbol{\theta}). \quad (5)$$

Now, the posterior probability of \mathcal{H}_c , which is the probability of merging (c_0, c_1) , is computed by Bayes rule:

$$p(\mathcal{H}_c|\mathbf{X}_c, \mathcal{T}_c) = \frac{p(\mathcal{H}_c)p(\mathbf{X}_c|\mathcal{H}_c)}{p(\mathbf{X}_c|\mathcal{T}_c)}. \quad (6)$$

In (Lee and Choi, 2014), an alternative formulation for the generative probability was proposed, which writes the generative process via the unnormalized probabilities (potential functions):

$$\tilde{p}_{\mathcal{H}}(\mathbf{X}_c) \stackrel{\text{def}}{=} \alpha\Gamma(|c|)p(\mathbf{X}_c|\mathcal{H}_c), \quad (7)$$

$$\tilde{p}_{\mathcal{T}}(\mathbf{X}_c) \stackrel{\text{def}}{=} \gamma_c p(\mathbf{X}_c|\mathcal{T}_c). \quad (8)$$

With these definitions, (2) is written as

$$\tilde{p}_{\mathcal{T}}(\mathbf{X}_c) = \tilde{p}_{\mathcal{H}}(\mathbf{X}_c) + \tilde{p}_{\mathcal{T}}(\mathbf{X}_{c_0})\tilde{p}_{\mathcal{T}}(\mathbf{X}_{c_1}), \quad (9)$$

and the posterior probability of \mathcal{H}_c is written as

$$p(\mathcal{H}_c|\mathbf{X}_c, \mathcal{T}_c) = \left\{ 1 + \frac{\tilde{p}_{\mathcal{T}}(\mathbf{X}_{c_0})\tilde{p}_{\mathcal{T}}(\mathbf{X}_{c_1})}{\tilde{p}_{\mathcal{H}}(\mathbf{X}_c)} \right\}^{-1}. \quad (10)$$

One can see that the ratio inside behaves as the dissimilarity between c_0 and c_1 :

$$d(c_0, c_1) \stackrel{\text{def}}{=} \frac{\tilde{p}_{\mathcal{T}}(\mathbf{X}_{c_0})\tilde{p}_{\mathcal{T}}(\mathbf{X}_{c_1})}{\tilde{p}_{\mathcal{H}}(\mathbf{X}_c)}. \quad (11)$$

Now, the building binary tree proceeds as **Algorithm 1** with Eq. (11). Beside this, BHC has a scheme to determine the number of clusters. At each iteration, if the minimum $d(c_0, c_1) > 1$, then $p(\mathcal{H}_c|\mathbf{X}_c, \mathcal{T}_c) < 0.5$ ($c = c_0 \cup c_1$). This means that the maximum probability of merging nodes is smaller than 0.5, and so we can stop the merging and cut the tree to determine the number of clusters. Note that when tree is cut, the result become a forest where each tree embeds a cluster in a partition.

BHC is closely related to the marginal likelihood of DPM; actually, the prior $p(\mathcal{H}_c)$ comes from the predictive distribution of DP prior. Moreover, it was shown that computing $\tilde{p}_{\mathcal{T}}(\mathbf{X}_c)$ to build a tree naturally induces a lower bound on the marginal likelihood of DPM, $p_{\text{DPM}}(\mathbf{X}_c)$ (Heller and Ghahrahmani, 2005):

$$\frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \tilde{p}_{\mathcal{T}}(\mathbf{X}_c) \leq p_{\text{DPM}}(\mathbf{X}_c). \quad (12)$$

Hence, in the perspective of the posterior inference algorithm for DPM, building tree in BHC is equivalent to computing the approximate marginal likelihood. Also, cutting the tree at the level where $d(c_0, c_1) > 1$ corresponds finding the MAP clustering of \mathbf{X} .

In (Heller and Ghahrahmani, 2005), the time complexity was claimed to be $O(n^2)$. However, this does not count the complexity required to find a pair with the smallest dissimilarity via sorting. For instance, with a sorting algorithm using priority queues, BHC requires $O(n^2 \log n)$ in time.

The dissimilarity is very sensitive to the hyperparameters of $p(\boldsymbol{\theta})$ needed to compute $p(\mathbf{X}_c|\mathcal{H}_c)$; The authors proposed an EM-like iterative algorithm to tune the hyperparameters, but the repeated execution of the algorithm is infeasible for large-scale data.

2.5 Bregman Divergences and Exponential Families

Definition 1. ((Bregman, 1967)) Let ϕ be a strictly convex differentiable function defined on a convex set. The Bregman divergence is defined as

$$B_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\phi(\mathbf{y}) \rangle. \quad (13)$$

The Bregman divergence includes many divergences as its special case; for example, when $\phi = \langle \mathbf{x}, \mathbf{x} \rangle$, corresponding Bregman divergence is Euclidean distance. Also, when ϕ is the negative entropy, corresponding Bregman divergence is KL divergence.

The exponential family distribution for $\mathbf{x} \in \mathbb{R}^d$ with parameter $\boldsymbol{\theta} \in \Theta$ has the form:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\{\langle t(\mathbf{x}), \boldsymbol{\theta} \rangle - \psi(\boldsymbol{\theta}) - h(\mathbf{x})\}, \quad (14)$$

where $t(\mathbf{x})$ is sufficient statistics, $\psi(\boldsymbol{\theta})$ is a log-partition function and $\exp\{-h(\mathbf{x})\}$ is a base distribution. We assume that $p(\mathbf{x}|\boldsymbol{\theta})$ is regular (Θ is open) and $t(\mathbf{x})$ is minimal ($\exists \mathbf{a} \in \mathbb{R}^d$ s.t. $\forall \mathbf{x}, \langle \mathbf{a}, t(\mathbf{x}) \rangle = \text{const}$). Let $\phi(\boldsymbol{\mu})$ be the convex conjugate of ψ :

$$\phi(\boldsymbol{\mu}) = \sup_{\boldsymbol{\theta} \in \Theta} \{\langle \boldsymbol{\mu}, \boldsymbol{\theta} \rangle - \psi(\boldsymbol{\theta})\}. \quad (15)$$

Then, the Bregman divergence and the exponential family has the following relationship:

Theorem 1. ((Banerjee et al., 2005)) $p(\mathbf{x}|\boldsymbol{\theta})$ is uniquely expressed as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\{-B_\phi(t(\mathbf{x}), \boldsymbol{\mu})\} \exp\{\phi(t(\mathbf{x})) - h(\mathbf{x})\}, \quad (16)$$

where $\boldsymbol{\mu} = \mathbb{E}[t(\mathbf{x})] = \nabla\psi(\boldsymbol{\theta})$.

The conjugate prior for $\boldsymbol{\theta}$ has the form:

$$p(\boldsymbol{\theta}|\nu, \boldsymbol{\tau}) = \exp\{\langle \boldsymbol{\tau}, \boldsymbol{\theta} \rangle - \nu\psi(\boldsymbol{\theta}) - \xi(\nu, \boldsymbol{\tau})\}. \quad (17)$$

$p(\boldsymbol{\theta}|\nu, \boldsymbol{\tau})$ can also be expressed with the Bregman divergence:

$$p(\boldsymbol{\theta}|\nu, \boldsymbol{\tau}) = \exp\{-\nu B_\phi(\boldsymbol{\tau}/\nu, \boldsymbol{\mu})\} \times \exp\{\nu\phi(\boldsymbol{\tau}/\nu) - \xi(\nu, \boldsymbol{\tau})\}. \quad (18)$$

2.6 Scaled Exponential Families

Let $\tilde{\boldsymbol{\theta}} = \beta\boldsymbol{\theta}$, and $\tilde{\psi}(\tilde{\boldsymbol{\theta}}) = \beta\psi(\tilde{\boldsymbol{\theta}}/\beta) = \beta\psi(\boldsymbol{\theta})$. The scaled exponential family with scale β is defined as follows (Jiang et al., 2012):

$$\begin{aligned} p(\mathbf{x}|\tilde{\boldsymbol{\theta}}) &= \exp\{\langle t(\mathbf{x}), \tilde{\boldsymbol{\theta}} \rangle - \tilde{\psi}(\tilde{\boldsymbol{\theta}}) - h_\beta(\mathbf{x})\} \\ &= \exp\{\beta\langle t(\mathbf{x}), \boldsymbol{\theta} \rangle - \beta\psi(\boldsymbol{\theta}) - h_\beta(\mathbf{x})\}. \end{aligned} \quad (19)$$

For this scaled distribution, the mean $\boldsymbol{\mu}$ remains the same, and the covariance Σ becomes Σ/β (Jiang et al., 2012).

Hence, the distribution is more concentrated around its mean. The scaled distribution in the Bregman divergence form is

$$p(\mathbf{x}|\tilde{\boldsymbol{\theta}}) = \exp\{-\beta B_\phi(t(\mathbf{x}), \boldsymbol{\mu})\} \exp\{\beta\phi(t(\mathbf{x})) - h_\beta(\mathbf{x})\}. \quad (20)$$

According to $p(\mathbf{x}|\tilde{\boldsymbol{\theta}})$, $p(\tilde{\boldsymbol{\theta}}|\tilde{\boldsymbol{\tau}}, \tilde{\nu})$ is defined with $\tilde{\boldsymbol{\tau}} = \boldsymbol{\tau}/\beta$, $\tilde{\nu} = \nu/\beta$. Actually, this yields the same prior as non-scaled distribution.

$$\begin{aligned} p(\tilde{\boldsymbol{\theta}}|\tilde{\nu}, \tilde{\boldsymbol{\tau}}) &= \exp\{\langle \tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\theta}} \rangle - \tilde{\nu}\tilde{\psi}(\tilde{\boldsymbol{\theta}}) - \xi_\beta(\tilde{\nu}, \tilde{\boldsymbol{\tau}})\} \\ &= \exp\{\langle \boldsymbol{\tau}, \boldsymbol{\theta} \rangle - \nu\psi(\boldsymbol{\theta}) - \xi(\nu, \boldsymbol{\tau})\}. \end{aligned} \quad (21)$$

3 MAIN RESULTS

We present the main contribution of this paper. From now on, we assume that the likelihood and prior in Eq. (5) are scaled exponential families defined in Section 2.6.

3.1 Small-Variance Asymptotics for BHC

The dissimilarity in BHC can be rewritten as follows:

$$\begin{aligned} d(c_0, c_1) &= \frac{\tilde{p}_T(\mathbf{X}_{c_0})\tilde{p}_T(\mathbf{X}_{c_1})}{\tilde{p}_H(\mathbf{X}_c)} \\ &= \frac{\alpha\Gamma(|c_0|)\Gamma(|c_1|)p(\mathbf{X}_{c_0}|\mathcal{H}_{c_0})p(\mathbf{X}_{c_1}|\mathcal{H}_{c_1})}{\Gamma(|c|)p(\mathbf{X}_c|\mathcal{H}_c)} \\ &\quad \times \{1 + d(c_{00}, c_{01})\}\{1 + d(c_{10}, c_{11})\}, \end{aligned} \quad (22)$$

where $c_0 = c_{00} \cup c_{01}$ and $c_1 = c_{10} \cup c_{11}$. We first analyze the term $p(\mathbf{X}_c|\mathcal{H}_c)$, as in (Jiang et al., 2012).

$$\begin{aligned} p(\mathbf{X}_c|\mathcal{H}_c) &= \int \left\{ \prod_{i \in c} p(\mathbf{x}_i|\tilde{\boldsymbol{\theta}}) \right\} p(\tilde{\boldsymbol{\theta}}|\nu, \boldsymbol{\tau}) d\tilde{\boldsymbol{\theta}} \\ &= \beta^d \int \exp \left\{ \langle \boldsymbol{\theta}, \boldsymbol{\tau} + \beta \sum_{i \in c} t(\mathbf{x}_i) \rangle - (\nu + \beta|c|)\psi(\boldsymbol{\theta}) \right. \\ &\quad \left. - \sum_{i \in c} h_\beta(\mathbf{x}_i) - \xi(\nu, \boldsymbol{\tau}) \right\} d\boldsymbol{\theta} \\ &= \beta^d \exp \left\{ (\nu + \beta|c|)\phi(\boldsymbol{\mu}_c) - \sum_{i \in c} h_\beta(\mathbf{x}_i) - \xi(\nu, \boldsymbol{\tau}) \right\} \\ &\quad \times \int \exp\{-\nu B_\phi(\boldsymbol{\mu}_c, \boldsymbol{\mu})\} d\boldsymbol{\theta}, \end{aligned} \quad (23)$$

where

$$\boldsymbol{\mu}_c \stackrel{\text{def}}{=} \frac{\boldsymbol{\tau} + \beta \sum_{i \in c} t(\mathbf{x}_i)}{\nu + \beta|c|}. \quad (24)$$

Note that $\boldsymbol{\mu} = \nabla\psi(\boldsymbol{\theta})$ is a function of $\boldsymbol{\theta}$. The term inside the integral of Eq. (23) has a local minimum at $\boldsymbol{\mu} = \boldsymbol{\mu}_c$, and thus can be approximated by Laplace's method:

$$\begin{aligned} &= \beta^d \exp \left\{ (\nu + \beta|c|)\phi(\boldsymbol{\mu}_c) - \sum_{i \in c} h_\beta(\mathbf{x}_i) - \xi(\nu, \boldsymbol{\tau}) \right\} \\ &\quad \left(\frac{2\pi}{\nu + \beta|c|} \right)^{\frac{d}{2}} \left| \frac{\partial^2 B_\phi(\boldsymbol{\mu}_c, \boldsymbol{\mu})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right|_{\boldsymbol{\mu}=\boldsymbol{\mu}_c}^{-\frac{1}{2}} \left\{ 1 + O(\beta^{-1}) \right\}. \end{aligned} \quad (25)$$

It follows from this result that we can approximate the marginal likelihood ratio term in Eq. (22), when the scale β approaches ∞ :

$$\lim_{\beta \rightarrow \infty} \frac{\alpha \Gamma(|c_0|) \Gamma(|c_1|) p(\mathbf{X}_{c_0} | \mathcal{H}_{c_0}) p(\mathbf{X}_{c_1} | \mathcal{H}_{c_1})}{\Gamma(|c|) p(\mathbf{X}_c | \mathcal{H}_c)} \\ \propto \lim_{\beta \rightarrow \infty} \alpha \beta^{\frac{d}{2}} \exp\left\{\beta(|c_0| \phi(\boldsymbol{\mu}_{c_0}) + |c_1| \phi(\boldsymbol{\mu}_{c_1}) - |c| \phi(\boldsymbol{\mu}_c))\right\},$$

Let $\alpha = \beta^{-\frac{d}{2}} \exp(-\beta\lambda)$, then we have

$$= \lim_{\beta \rightarrow \infty} \exp\{\beta(|c_0| \phi(\boldsymbol{\mu}_{c_0}) + |c_1| \phi(\boldsymbol{\mu}_{c_1}) - |c| \phi(\boldsymbol{\mu}_c) - \lambda)\}.$$

When $\beta \rightarrow \infty$, the term inside the exponent converges to

$$|c_0| \phi(\bar{t}_{c_0}) + |c_1| \phi(\bar{t}_{c_1}) - |c| \phi(\bar{t}_c) - \lambda, \quad (26)$$

where

$$\bar{t}_c \stackrel{\text{def}}{=} \frac{1}{|c|} \sum_{i \in c} t(\mathbf{x}_i), \quad (27)$$

and this is the average of sufficient statistics for cluster c . From this, we define a new dissimilarity $d_*(c_0, c_1)$ as

$$d_*(c_0, c_1) \stackrel{\text{def}}{=} |c_0| \phi(\bar{t}_{c_0}) + |c_1| \phi(\bar{t}_{c_1}) - |c| \phi(\bar{t}_c) \\ = |c_0| \phi(\bar{t}_{c_0}) + |c_1| \phi(\bar{t}_{c_1}) \\ - (|c_0| + |c_1|) \phi\left(\frac{|c_0| \bar{t}_{c_0} + |c_1| \bar{t}_{c_1}}{|c_0| + |c_1|}\right). \quad (28)$$

Note that $d_*(\cdot, \cdot)$ is always positive since ϕ is convex. If $d_*(c_0, c_1) \geq \lambda$, the limit Eq. (26) diverges to ∞ , and converges to zero otherwise. When $|c| = 1$, Eq. (26) is the same as the limit of the dissimilarity $d(c_0, c_1)$, and thus the dissimilarity diverges when $d_*(c_0, c_1) \geq \lambda$ and converges otherwise. When $|c| > 1$, assume that the dissimilarities of children $d(c_{00}, c_{01})$ and $d(c_{10}, c_{11})$ converges to zero. From Eq. (22), we can easily see that $d(c_0, c_1)$ converges only if $d_*(c_0, c_1) < \lambda$. In summary,

$$\lim_{\beta \rightarrow \infty} d(c_0, c_1) = \begin{cases} 0 & \text{if } d_*(c_0, c_1) < \lambda, \\ \infty & \text{otherwise.} \end{cases} \quad (29)$$

In similar way, we can also prove the following:

$$\lim_{\beta \rightarrow \infty} \frac{d(c_0, c_1)}{d(c_2, c_3)} = \begin{cases} 0 & \text{if } d_*(c_0, c_1) < d_*(c_2, c_3), \\ \infty & \text{otherwise.} \end{cases}, \quad (30)$$

which means that comparing two dissimilarities in original BHC is equivalent to comparing the new dissimilarities $d_*(\cdot, \cdot)$, and we can choose the next pair to merge by comparing $d_*(\cdot, \cdot)$ instead of $d(\cdot, \cdot)$.

With Eqs. (29) and (30), we conclude that when $\beta \rightarrow \infty$, BHC reduces to **Algorithm 1** with dissimilarity measure $d_*(\cdot, \cdot)$ and threshold λ , where the algorithm terminates when the minimum $d_*(\cdot, \cdot)$ exceeds λ .

On the other hand, a simple calculation yields

$$d_*(c_0, c_1) = |c_0| B_\phi(\bar{t}_{c_0}, \bar{t}_c) + |c_1| B_\phi(\bar{t}_{c_1}, \bar{t}_c),$$

which is exactly same as the dissimilarity proposed in (Telgarsky and Dasgupta, 2012). Due to the close relationship between exponential family and the Bregman divergence, the dissimilarities derived from two different perspective has the same form.

As an example, assume that $p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \sigma^2 \mathbf{I})$ and $p(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}|0, \rho^2 \mathbf{I})$. We have $\phi(\mathbf{x}) = \|\mathbf{x}\|^2 / (2\sigma^2)$ and

$$d_*(c_0, c_1) = \frac{|c_0| |c_1| \|\mathbf{x}_{c_0} - \mathbf{x}_{c_1}\|^2}{2\sigma^2(|c_0| + |c_1|)}, \quad (31)$$

which is same as the Ward's merge cost (Ward, 1963), except for the constant $1/(2\sigma^2)$. Other examples can be found in (Banerjee et al., 2005).

Note that $d_*(\cdot, \cdot)$ does not need hyperparameter tunings, since the effect of prior $p(\boldsymbol{\theta})$ is ignored as $\beta \rightarrow 0$. This provides a great advantage over BHC which is sensitive to the hyperparameter settings.

Smoothing: In some particular choice of ϕ , the singleton clusters may have degenerate values (Telgarsky and Dasgupta, 2012). For example, when $p(\mathbf{x}|\boldsymbol{\theta}) = \text{Mult}(\mathbf{x}|m, q)$, the function $\phi(\mathbf{x}) = \sum_{j=1}^d x_j \log(x_j/m)$ has degenerate values when $x_j = 0$. To handle this, we use the smoothing strategy proposed in (Telgarsky and Dasgupta, 2012); instead of the original function $\phi(\mathbf{x})$, we use the smoothed functions $\phi_0(\mathbf{x})$ and $\phi_1(\mathbf{x})$ defined as follows:

$$\phi_0(\mathbf{x}) \stackrel{\text{def}}{=} \phi((1 - \alpha)\mathbf{x} + \alpha\boldsymbol{\gamma}), \quad (32)$$

$$\phi_1(\mathbf{x}) \stackrel{\text{def}}{=} \phi(\mathbf{x} + \alpha\boldsymbol{\gamma}), \quad (33)$$

where $\alpha \in (0, 1)$ be arbitrary constant and $\boldsymbol{\gamma}$ must in the relative interior of the domain of ϕ . In general, we use $\phi_0(\mathbf{x})$ as a smoothed function, but we can also use $\phi_1(\mathbf{x})$ when the domain of ϕ is a convex cone.

Heuristics for choosing λ : As in (Kulis and Jordan, 2012), we choose the threshold value λ . Fortunately, we found that the clustering accuracy was not extremely sensitive to the choice of λ ; merely selecting the scale of λ could result in reasonable accuracy. There can be many simple heuristics, and the one we found effective is to use the k -means clustering. With the very rough guess on the desired number of clusters \tilde{k} , we first run the k -means clustering (with Euclidean distance) with $k = a\tilde{k}$ (we fixed $a = 4$ for all experiments). Then, λ was set to the average value of dissimilarities $d_*(\cdot, \cdot)$ between the all pair of k centers.

3.2 Reducibility of $d_*(\cdot, \cdot)$

The relaxed BHC with small-variance asymptotics still has the same complexities to BHC. If we can show that $d_*(\cdot, \cdot)$ is reducible, we can reduce the complexities by adapting the nearest neighbor chain algorithm. Unfortunately, $d_*(\cdot, \cdot)$ is not reducible in general (one can easily find

counter-examples for some distributions). However, we argue that $d_*(\cdot, \cdot)$ is reducible in many cases, and thus applying the nearest neighbor chain algorithm as if $d_*(\cdot, \cdot)$ is reducible does not degrade the clustering accuracy. In this section, we show the reason by analyzing $d_*(\cdot, \cdot)$.

At first, we investigate a term inside the dissimilarity:

$$f(\bar{t}_c) \stackrel{\text{def}}{=} |c| \phi(\bar{t}_c). \quad (34)$$

The second-order Taylor expansion of this function around the mean $\boldsymbol{\mu}$ yields:

$$f(\bar{t}_c) = |c| \phi(\boldsymbol{\mu}) + |c| \phi^{(1)}(\boldsymbol{\mu})^\top (\bar{t}_c - \boldsymbol{\mu}) + \Delta_\phi(\bar{t}_c, \boldsymbol{\mu}) + \epsilon_\phi(\bar{t}_c, \boldsymbol{\mu}), \quad (35)$$

where $\phi^{(k)}$ is the k th order derivative of ϕ , and

$$\Delta_\phi(\bar{t}_c) \stackrel{\text{def}}{=} \frac{|c|}{2} (\bar{t}_c - \boldsymbol{\mu})^\top \phi^{(2)}(\boldsymbol{\mu}) (\bar{t}_c - \boldsymbol{\mu}), \quad (36)$$

$$\epsilon_\phi(\bar{t}_c) \stackrel{\text{def}}{=} |c| \sum_{|\boldsymbol{\alpha}|=3} \frac{\partial^\alpha \phi(\boldsymbol{\nu})}{\boldsymbol{\alpha}!} (\bar{t}_c - \boldsymbol{\mu})^\alpha. \quad (37)$$

Here, $\boldsymbol{\alpha}$ is the multi-index notation, and $\boldsymbol{\nu} = \boldsymbol{\mu} + k(\bar{t}_c - \boldsymbol{\mu})$ for some $k \in (0, 1)$. The term $\Delta_\phi(\bar{t}_c)$ plays an important role in analyzing the reducibility of $d_*(\cdot, \cdot)$. To bound the error term $\epsilon_\phi(\bar{t}_c)$, we assume that $|\phi^{(3)}| \leq M^1$. As earlier, assume that \bar{t}_c is a average of $|c|$ observations generated from the same scaled-exponential family distribution:

$$\mathbf{x}_1, \dots, \mathbf{x}_n \stackrel{\text{iid}}{\sim} p(\cdot | \beta, \boldsymbol{\theta}), \quad \bar{t}_c = \frac{1}{|c|} \sum_{i \in c} t(\mathbf{x}_i). \quad (38)$$

By the property of the log-partition function of the exponential family distribution, we get the following results:

$$\mathbb{E}[\epsilon_\phi(\bar{t}_c)] = \frac{1}{\beta^{2|c|}} \sum_{|\boldsymbol{\alpha}|=3} \frac{\partial^\alpha \phi(\boldsymbol{\nu}) \partial^\alpha \psi(\boldsymbol{\theta})}{\boldsymbol{\alpha}!}. \quad (39)$$

One can see that the expected error converges to zero as $\beta^{2|c|} \rightarrow \infty$. Also, it can be shown that the expectation of the ratio of two terms converges to zero as $\beta \rightarrow \infty$:

$$\lim_{\beta \rightarrow \infty} \mathbb{E} \left[\frac{\epsilon_\phi(\bar{t}_c)}{\Delta_\phi(\bar{t}_c)} \right] = 0, \quad (40)$$

which means that $\Delta_\phi(\bar{t}_c)$ asymptotically dominates $\epsilon_\phi(\bar{t}_c)$ (detailed derivations are given in the supplementary material). Hence, we can safely approximate $f(\bar{t}_c)$ up to second order term.

Now, let c_0 and c_1 be clusters belong to the same super-cluster (i.e. \mathbf{X}_{c_0} and \mathbf{X}_{c_1} were generated from the same mean vector $\boldsymbol{\mu}$). We don't need to investigate the case where the pair belong to a different cluster, since then they

¹This assumption holds for the most of distributions we will discuss (if properly smoothed), but not holds in general.

Algorithm 3 Nearest neighbor chain algorithm for BHC with small-variance asymptotics

Input: $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, $d_*(\cdot, \cdot)$, λ .

Output: A clustering C .

```

1: Set  $R = [n]$  and  $C = \emptyset$ .
2: while  $R \neq \emptyset$  do
3:   Pick any  $c_0 \in R$ .
4:   Build a chain  $c_1 = \text{nn}(c_0), c_2 = \text{nn}(c_1), \dots$ , where
       $\text{nn}(c) \stackrel{\text{def}}{=} \arg \min_{c'} d_{\text{star}}(c, c')$ . Extend the chain until
       $c_i = \text{nn}(c_{i-1})$  and  $c_{i-1} = \text{nn}(c_i)$ .
5:   Remove  $c_i$  and  $c_{i-1}$  from  $R$ .
6:   if  $d_*(c_{i-1}, c_i) < \lambda$  then
7:     Add  $c = c_i \cup c_{i-1}$  to  $R$ .
8:     If  $i \geq 2$ , go to line 3 and extend the chain from  $c_{i-2}$ .
       Otherwise, go to line 2.
9:   else
10:    Add  $c_i$  and  $c_{i-1}$  to  $C$ .
11:   end if
12: end while
    
```

will not be merged anyway ($d_*(\cdot, \cdot) \geq \lambda$) in our algorithm. By the independence, $\mathbb{E}[\bar{t}_{c_0}] = \mathbb{E}[\bar{t}_{c_1}] = \mathbb{E}[\bar{t}_{c_0 \cup c_1}] = \boldsymbol{\mu}$. Applying the approximation (35), we have

$$d_*(c_0, c_1) \approx \Delta_\phi(\bar{t}_{c_0}) + \Delta_\phi(\bar{t}_{c_1}) - \Delta_\phi(\bar{t}_{c_0 \cup c_1}) = \frac{|c_0||c_1|}{2(|c_0|+|c_1|)} (\bar{t}_{c_0} - \bar{t}_{c_1})^\top \phi^{(2)}(\boldsymbol{\mu}) (\bar{t}_{c_0} - \bar{t}_{c_1}). \quad (41)$$

This approximation, which we will denote as $\tilde{d}_*(c_0, c_1)$, is a generalization of the Ward's cost (31) from Euclidean distance to Mahalanobis distance with matrix $\phi^{(2)}(\boldsymbol{\mu})$ (note that this approximation is exact for the spherical Gaussian case). More importantly, $\tilde{d}_*(c_0, c_1)$ is reducible.

Theorem 2.

$$\begin{aligned} \tilde{d}_*(c_0, c_1) &\leq \min\{\tilde{d}_*(c_0, c_2), \tilde{d}_*(c_1, c_2)\} \\ \Rightarrow \min\{\tilde{d}_*(c_0, c_2), \tilde{d}_*(c_1, c_2)\} &\leq \tilde{d}_*(c_0 \cup c_1, c_2). \end{aligned} \quad (42)$$

Proof. As for the Ward's cost, the following Lance-Williams update formula (Lance and Williams, 1967) holds for $\tilde{d}_*(\cdot, \cdot)$:

$$\begin{aligned} \tilde{d}_*(c_0 \cup c_1, c_2) &= \frac{(|c_0| + |c_2|) \tilde{d}_*(c_0, c_2)}{|c_0| + |c_1| + |c_2|} \\ &\quad + \frac{(|c_1| + |c_2|) \tilde{d}_*(c_1, c_2) - |c_2| \tilde{d}_*(c_0, c_1)}{|c_0| + |c_1| + |c_2|}. \end{aligned} \quad (43)$$

Hence, by the assumption, we get

$$\tilde{d}_*(c_0 \cup c_1, c_2) \geq \min\{\tilde{d}_*(c_0, c_2), \tilde{d}_*(c_1, c_2)\}. \quad \square$$

As a result, the dissimilarity $d_*(\cdot, \cdot)$ is reducible provided that the Taylor's approximation (35) is accurate. In such case, we can apply the nearest-neighbor chain algorithm with $d_*(\cdot, \cdot)$ as if it is reducible to build a binary tree in

Table 1: Average value of the exact dissimilarity d_* , approximate dissimilarity \tilde{d}_* , and relative error, and number of not-reducible case among 100,000 trials.

	$\mathbb{E}[d_*]$	$\mathbb{E}[\tilde{d}_*]$	average relative error (%)	not reducible /100k
Poisson	0.501	0.500	1.577	0
Multinomial	3.594	3.642	4.675	87
Gaussian	8.739	8.717	7.716	48

$O(n^2)$ time and $O(n)$ space. Unlike **Algorithm 2**, we have a threshold λ to determine the number of clusters, and we present a slightly revised algorithm (**Algorithm 3**). Note again that the revised algorithm generates forests instead of trees.

4 EXPERIMENTS

4.1 Experiments on Synthetic Data

Testing the reducibility of $d_*(\cdot, \cdot)$: We tested the reducibility of $d_*(\cdot, \cdot)$ empirically. We repeatedly generated the three clusters c_0, c_1 and c_2 from the exponential family distributions, and counted the number of cases where the dissimilarities between those clusters are not reducible. We also measured the average value of the relative error to support our arguments in Section 3.2. We tested three distributions; Poisson, multinomial and Gaussian. At each iteration, we first sampled the size of the clusters $|c_0|, |c_1|$ and $|c_2|$ from $\text{Unif}([20, 100])$. Then we sampled the three clusters from one of the three distributions, and computed $d_*(c_0, c_1), d_*(c_1, c_2)$ and $d_*(c_0, c_2)$. We then first checked whether these three values satisfy the reducibility condition (1) (for example, if $d_*(c_0, c_2)$ is the smallest, we checked if $d_*(c_0 \cup c_2, c_1) \geq \min\{d_*(c_0, c_1), d_*(c_2, c_1)\}$). Then, for $d_*(c_0, c_1)$, we computed the approximate value $\tilde{d}_*(c_0, c_1)$ and measured the relative error

$$2 \times \left| \frac{d_*(c_0, c_1) - \tilde{d}_*(c_0, c_1)}{d_*(c_0, c_1) + \tilde{d}_*(c_0, c_1)} \right|. \quad (44)$$

We repeated this process for 100,000 times for the three distributions and measured the average values. For Poisson distribution, we sampled the mean $\rho \sim \text{Gamma}(2, 0.05)$ and sampled the data from $\text{Poisson}(\rho)$. We smoothed the function $\phi(\mathbf{x}) = \mathbf{x} \log(\mathbf{x}) - \mathbf{x}$ as $\phi(\mathbf{x} + 0.01)$ to prevent degenerate function values. For multinomial distribution, we tested the case where the dimension d is 10 and the number of trials m is 5. We sampled the parameter $\mathbf{q} \sim \text{Dir}(5 \cdot \mathbf{1}_d)$ where $\mathbf{1}_d$ is d -dimensional one vector, and sampled the data from $\text{Mult}(\mathbf{q})$. We smoothed the function $\phi(\mathbf{x}) = \sum_{j=1}^d x_j \log(x_j/m)$ as $\phi(0.9\mathbf{x} + 0.1m\mathbf{1}_d/d)$. For Gaussian, we tested with $d = 10$, and sampled the mean and covariance $\boldsymbol{\mu}, \boldsymbol{\Sigma} \sim \mathcal{N}(0, (0.08 \cdot \boldsymbol{\Lambda})^{-1})\mathcal{W}(d+2, \boldsymbol{\Psi})$ ($\boldsymbol{\Psi} = \mathbf{A}\mathbf{A}^\top + d\mathbf{I}$ where \mathbf{A} was sampled from unit Gaussian). We

smoothed the function $\phi(\mathbf{x}, \mathbf{X}) = -\frac{1}{2} \log \det(\mathbf{X} - \mathbf{x}\mathbf{x}^\top)$ as $-\frac{1}{2} \log \det(\mathbf{X} - \mathbf{x}\mathbf{x}^\top + 0.01\mathbf{I})$.

The result is summarized in Table 1. the generated dissimilarities were reducible in most case, as expected. The relative error was small, which supports our arguments of the reason why $d_*(\cdot, \cdot)$ is reducible with high probability. We also measured the change of average relative error by controlling two factors; the maximum cluster size n_{\max} and variance scale factor β . We plotted the average relative error of Gaussian distribution by changing those two factors, and the relative error decreased as predicted (Figure 1).

Clustering synthetic data: We evaluated the clustering accuracies of original BHC (BHC), BHC in small-variance limit with greedy algorithm (RBHC-greedy), BHC in small-variance limit with nearest neighbor chain method (RBHC-nnca), single linkage, complete linkage, and Ward’s method. We generated the datasets from Poisson, multinomial and Gaussian distribution. We tested two types of data; 1,000 elements with 6 clusters and 2,000 elements with 12 clusters. For Poisson distribution, each mixture component was generated from $\text{Poisson}(\rho)$ with $\rho \sim \text{Gamma}(2, 0.05)$ for both datasets. For multinomial distribution, we set $d = 20$ and $m = 10$ for 1,000 elements dataset, and set $d = 40$ and $m = 10$ for 2,000 elements datasets. For both dataset, we sampled the parameter $\mathbf{q} \sim \text{Dir}(0.5 \cdot \mathbf{1}_d)$. For Gaussian case, we set $d = 3$ for 1,000 elements dataset and set $d = 6$ for 2,000 elements dataset. We sampled the parameters from $\mathcal{N}(0, (0.08 \cdot \boldsymbol{\Lambda})^{-1})\mathcal{W}(\boldsymbol{\Psi}, 6)$ for 1,000 elements, and from $\mathcal{N}(0, (0.2 \cdot \boldsymbol{\Lambda})^{-1})\mathcal{W}(\boldsymbol{\Psi}, 9)$ for 2,000 elements. For each distribution and type (1,000 or 2,000), we generated 10 datasets for each type and measured the average clustering accuracies.

We evaluated the clustering accuracy using the adjusted Rand index (Hubert and Arabie, 1985). For traditional agglomerative clustering algorithms, we assumed that we know the true number of clusters k and cut the tree at corresponding level. For RBHC-greedy and RBHC-nnca, we selected the threshold λ with the heuristics described in Section 3.1. For original BHC, we have to carefully tune the hyperparameters, and the accuracy was very sensitive to this setting. In the case of Poisson distribution where $\rho \sim \text{Gamma}(a, b)$, we have to tune two hyperparameters $\{a, b\}$. For multinomial case where $\mathbf{q} \sim \text{Dir}(\boldsymbol{\alpha})$, we set $\boldsymbol{\alpha} = \alpha\mathbf{1}_d$ and tuned α . For Gaussian case where $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathcal{N}(\boldsymbol{\mu}, (r\boldsymbol{\Lambda})^{-1})\mathcal{W}(\nu, \boldsymbol{\Psi}^{-1})$, we have four hyperparameters $\{\boldsymbol{m}, r, \nu, \boldsymbol{\Psi}\}$. We set \boldsymbol{m} to be the empirical mean of \mathbf{X} and fixed $r = 0.1$ and $\nu = d + 6$. We set $\boldsymbol{\Psi} = k\mathbf{S}$ where \mathbf{S} is the empirical covariance of \mathbf{X} and controlled k according to the dimension and the size of the data. The result is summarized in Table 2.

The accuracies of RBHC-greedy and RBHC-nnca were best for most of the cases, and the accuracies of the two

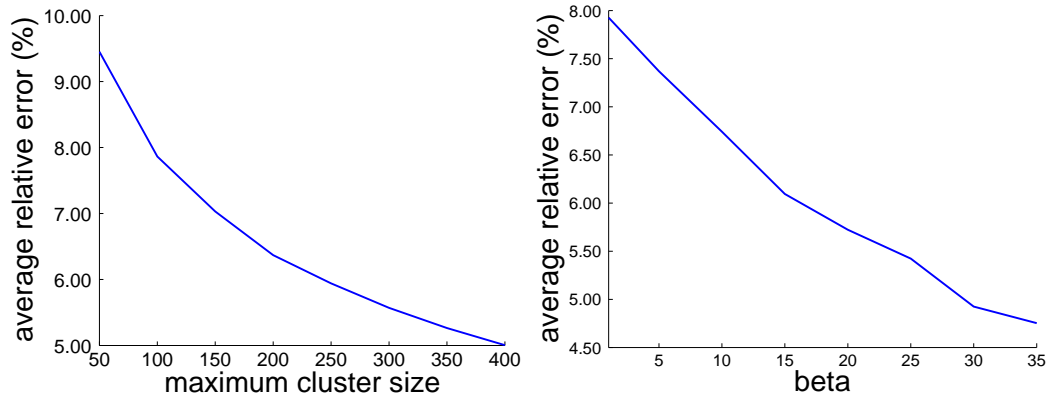

 Figure 1: Average relative error vs maximum cluster size and scale factor β .

Table 2: Average adjusted Rand index values for randomly generated datasets. Best ones are marked as bold face.

	Poisson (k)	Poisson ($2k$)	multinomial (k)	multinomial ($2k$)	Gaussian (k)	Gaussian ($2k$)
single linkage	0.091 (0.088)	0.015 (0.030)	0.000 (0.000)	0.000 (0.000)	0.270 (0.280)	0.057 (0.055)
complete linkage	0.381 (0.091)	0.263 (0.059)	0.266 (0.144)	0.090 (0.025)	0.565 (0.166)	0.475 (0.141)
Ward's method	0.465 (0.119)	0.273 (0.049)	0.770 (0.067)	0.564 (0.055)	0.779 (0.145)	0.763 (0.122)
RBHC-greedy	0.469 (0.112)	0.290 (0.056)	0.870 (0.046)	0.733 (0.028)	0.875 (0.087)	0.883 (0.067)
RBHC-nnca	0.469 (0.112)	0.290 (0.056)	0.865 (0.045)	0.736 (0.040)	0.875 (0.087)	0.883 (0.067)
BHC	0.265 (0.080)	0.134 (0.052)	0.907 (0.069)	0.894 (0.044)	0.863 (0.109)	0.860 (0.108)

methods were almost identical except for the multinomial distribution. BHC was best for the multinomial case where the hyperparameter tuning was relatively easy, but showed poor performance in Poisson case (we failed to find the best hyperparameter setting in that case). Hence, it would be a good choice to use RBHC-greedy or RBHC-nnca which do not need careful hyperparameter tuning, and RBHC-nnca may be the best choice considering its space and time complexity compared to RBHC-greedy.

4.2 Experiments on Real Data

We tested the agglomerative clustering algorithms on two types of real-world data. The first one was a subset of MNIST digit database (LeCun et al., 1998). We scaled down the original 28×28 to 7×7 and vectorized each image to be \mathbb{R}^{49} vector. Then we sampled 3,000 images from the classes 0, 3, 7 and 9. We clustered this dataset with Gaussian assumption. The second one was visual-word data extracted from Caltech101 database (Fei-Fei et al., 2004). We sampled 2,033 images from "Airplane", "Mortorbikes" and "Faces-easy" classes, and extracted SIFT features for image patches. Then we quantized those features into 1,000 visual words. We clustered the data with multinomial assumption. Table 3 shows the ARI values of agglomerative clustering algorithms. As in the synthetic experiments, the accuracy RBHC-greedy and RBHC-nnca were identical, and outperformed the traditional agglomerative clustering algorithms. BHC was best for Caltech101, where the multinomial distribution with easy hyperparameter tuning was assumed. However, BHC was even worse than Ward's

method for MNIST case, where we failed to tune 49×49 matrix Ψ .

Table 3: Average adjusted Rand index values for MNIST and Caltech101 datasets. Best ones are marked as bold face.

	MNIST	Caltech101
single linkage	0.000	0.000
complete linkage	0.187	0.000
Ward's method	0.485	0.465
RBHC-greedy	0.637	0.560
RBHC-nnca	0.637	0.560
BHC	0.253	0.646

5 CONCLUSIONS

In this paper we have presented a non-probabilistic counterpart of BHC, referred to as RBHC, using a small variance relaxation when underlying likelihoods are assumed to be conjugate exponential families. In contrast to the original BHC, RBHC does not require careful tuning of hyperparameters. We have also shown that the dissimilarity measure emerged in RBHC is reducible with high probability, so that the nearest neighbor chain algorithm was used to speed up the RBHC and to reduce the space complexity, leading to RBHC-nnca. Experiments on both synthetic and real-world datasets demonstrated the validity of RBHC.

Acknowledgements: This work was supported by National Research Foundation (NRF) of Korea (NRF-2013R1A2A2A01067464) and the IT R&D Program of MSIP/IITP (14-824-09-014, Machine Learning Center).

References

- C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.
- A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.
- L. M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.
- M. Bruynooghe. Classification ascendante hiérarchique, des grands ensembles de données: un algorithme rapide fondé sur la construction des voisinages réductibles. *Les Cahiers de l'Analyse de Données*, 3:7–33, 1978.
- W. H. E. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1:7–24, 1984.
- R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Proceedings of the IEEE CVPR-2004 Workshop on Generative-Model Based Vision*, 2004.
- T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.
- K. A. Heller and Z. Ghahrahmani. Bayesian hierarchical clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.
- L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.
- K. Jiang, B. Kulis, and M. I. Jordan. Small-variance asymptotics for exponential family Dirichlet process mixture models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, 2012.
- B. Kulis and M. I. Jordan. Revisiting k -means: New algorithms via Bayesian nonparametrics. In *Proceedings of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, UK, 2012.
- G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: 1 hierarchical systems. *The Computing Journal*, 9:373–380, 1967.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- J. Lee and S. Choi. Incremental tree-based inference with dependent normalized random measures. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Reykjavik, Iceland, 2014.
- M. Telgarsky and S. Dasgupta. Agglomerative Bregman clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, UK, 2012.
- H. M. Wallach, S. T. Jensen, L. Dicker, and K. A. Heller. An alternative prior process for nonparametric Bayesian clustering. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy, 2010.
- J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, pages 236–244, 1963.