A natural framework for sparse hierarchical clustering

Hongyang Zhang, Ruben H. Zamar

Department of Statistics, University of British Columbia, 3182-2207 Main Mall, Vancouver, British Columbia V6T 1Z4, Canada

Abstract

There has been a surge in the number of large and flat data sets – data sets containing a large number of features and a relatively small number of observations – due to the growing ability to collect and store information in medical research and other fields. Hierarchical clustering is a widely used clustering tool. In hierarchical clustering, large and flat data sets may allow for a better coverage of clustering features (features that help explain the true underlying clusters) but, such data sets usually include a large fraction of noise features (non-clustering features) that may hide the underlying clusters. Witten and Tibshirani (2010) proposed a sparse hierarchical clustering framework to cluster the observations using an adaptively chosen subset of the features, however, we show that this framework has some limitations when the data sets contain clustering features with complex structure. In this paper, another sparse hierarchical clustering (SHC) framework is proposed. We show that, using simulation studies and real data examples, the proposed framework produces superior feature selection and clustering performance comparing to the classical (of-the-shelf) hierarchical clustering and the existing sparse hierarchical clustering framework.

Keywords: Hierarchical clustering, Sparse data, High-dimensional data, Feature selection

1. Introduction

The performance of existing clustering algorithms can be distorted when the number of variables is large and many of them contain no information about the cluster structure. Furthermore, interpretability can be impeded when the clustering procedure uses a large number of variables. Thus, clustering algorithms that can simultaneously perform cluster analysis and feature selection are in demand.

Here we focus on hierarchical clustering, one of the most widely used clustering algorithms. Hierarchical clustering categorizes observations into a hierarchical set of groups organized in a tree structure called dendrogram. Hierarchical clustering has a broad range of applications such as microarray data analysis, digital imaging, stock prediction, text mining, etc.

There are several proposals for feature selection for other clustering methods such as K-means (e.g. Witten and Tibshirani (2010), Sun et al. (2012)) and model-based

clustering (e.g. Raftery and Dean (2006), Pan and Shen (2007), Wang and Zhu (2008), Xie et al. (2008)). However, there has been less research for the case of hierarchical clustering. A brief survey of such proposals is given below.

Let $\boldsymbol{X}$ be an $n \times p$ data matrix, with n observations and p features. Let $d_{i,i'} = d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ be a measure of dissimilarity between observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ $(1 \leq i, i' \leq n)$, which are the rows i and i' of the data matrix $\boldsymbol{X}$. We will assume that d is additive in the features: $d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = d_{i,i'} = \sum_{j=1}^{p} d_{i,i',j}$, where $d_{i,i',j}$ indicates the dissimilarity between observations i and i' along feature j. Unless specified otherwise, our examples and simulations take d equal to the squared Euclidean distance, $d_{i,i',j} = (X_{ij} - X_{i'j})^2$. However, other dissimilarity measures are possible, such as the absolute difference $d_{i,i',j} = |X_{ij} - X_{i'j}|$.

Friedman and Meulman (2004) proposed *clustering objects on subsets of attributes* (COSA). COSA employs a criterion, related to a weighted version of K-means clustering, to automatically detect subgroups of objects that preferentially cluster on subsets of the attribute variables rather than on all of them simultaneously. An extension of COSA for hierarchical clustering was also proposed. The algorithm is quite complex and requires multiple tuning parameters. Moreover, as noted by Witten and Tibshirani (2010), this proposal does not truly result in a sparse clustering because all the variables have nonzero weights.

Witten and Tibshirani (2010) proposed a new framework for sparse clustering that can be applied to procedures that optimize a criterion of the form

$$\max_{\Theta \in G} \left\{ \sum_{j=1}^{p} f_j(\mathbf{X}_j, \Theta) \right\}, \tag{1}$$

where $\mathbf{X}_j = (X_{1j}, X_{2j}, ..., X_{nj})^T \in \mathbb{R}^n$ denotes the observed j-th feature, each $f_j(\mathbf{X}_j, \Theta)$ is a function that solely depends on the j-th feature and Θ is a set of unknown parameters taking values on G. To introduce sparsity Witten and Tibshirani (2010) modified criterium (1) as follows:

$$\max_{\boldsymbol{w}, \Theta \in G} \left\{ \sum_{j=1}^{p} w_j f_j(\mathbf{X}_j, \Theta) \right\} \text{ subject to } ||\boldsymbol{w}||_2^2 \leq 1,\ ||\boldsymbol{w}||_1 \leq s \text{ and } w_j \geq 0. \tag{2}$$

Here $\boldsymbol{w} = (w_1, w_2, \cdots, w_p)$ is a vector of weights for each feature, $||\boldsymbol{w}||_2^2$ is squared L2-norm on $\boldsymbol{w}$, and $||\boldsymbol{w}||_1$ is L1-norm on $\boldsymbol{w}$. A feature with zero-weight is clearly not used in the criterion.

Hierarchical clustering does not optimize a criterium like (1) and, therefore, does not directly fit into Witten and Tibshirani (2010) sparse clustering framework (2). To overcome this difficulty they casted the dissimilarity matrix $\{d_{i,i'}\}_{n \times n}$ as the solution

of an optimization problem as follows:

$$\max_{\mathbf{U}\in\mathbb{R}^{n\times n}} \left\{ \sum_{j=1}^{p} \sum_{i,i'=1}^{n} d_{i,i',j} U_{i,i'} \right\} \text{ subject to } \sum_{i,i'=1}^{n} U_{i,i'}^2 \leq 1. \tag{3}$$

It can be shown that the solution $\widehat{U}_{i,i'}$ to (3) is proportional to the dissimilarity matrix, that is, $\widehat{U}_{i,i'} \propto d_{i,i'}$. The criterion in (3) is a special case of (1) when we let $f_j(\mathbf{X}_j, \Theta) = \sum_{i,i'=1}^{n} d_{i,i',j} U_{i,i'}$. Now sparse hierarchical clustering can be achieved by obtaining a sparse dissimilarity matrix. Now *the sparse hierarchical clustering criterion* can be defined as follows:

$$\max_{\boldsymbol{w},\mathbf{U}\in\mathbb{R}^{n\times n}} \left\{ \sum_{j=1}^{p} w_j \sum_{i,i'=1}^{n} d_{i,i',j} U_{i,i'} \right\} \text{ subject to } \sum_{i,i'=1}^{n} U_{i,i'}^2 \leq 1, ||\boldsymbol{w}||_2^2 \leq 1, \; ||\boldsymbol{w}||_1 \leq s. \tag{4}$$

The constraint $w_j \geq 0$ has been removed because $d_{i,i',j} \geq 0$ for all $1 \leq i, i' \leq n$ and $1 \leq j \leq p$. The solution to (4) can be obtained using sparse principal component (SPC) proposed in Witten et al. (2009) as follows: Let $\boldsymbol{u}$ be a vector of length n^2 that contains all elements in $(U_{i,i'})_{n\times n}$ and $\boldsymbol{D}$ be a $n^2 \times p$ matrix whose j-th column contains the n^2 elements of $\{d_{i,i',j}\}_{n\times n}$ – the dissimilarity matrix calculated from the j-th feature alone. Now the criterion in (4) is equivalent to the following:

$$\max_{\boldsymbol{w},\boldsymbol{u}} \left\{ \boldsymbol{u}^T \boldsymbol{D} \boldsymbol{w} \right\} \text{ subject to } ||\boldsymbol{u}||_2^2 \leq 1, ||\boldsymbol{w}||_2^2 \leq 1, ||\boldsymbol{w}||_1 \leq s. \tag{5}$$

This reduces to applying SPC on *the transformed dissimilarity matrix*, $\boldsymbol{D}$. It can also be shown that the solution to (5) satisfies: $\widehat{\boldsymbol{u}}^T \propto \boldsymbol{D}\widehat{\boldsymbol{w}}$. As $\widehat{\boldsymbol{w}}$ is sparse, so is $\widehat{\boldsymbol{u}}$. Thus, by re-arranging the elements in $\widehat{\boldsymbol{u}}$ into a $n \times n$ dissimilarity matrix $\widehat{\boldsymbol{U}}$, we obtain a sparse dissimilarity matrix which only contains the information from a subset of selected features. Finally, sparse hierarchical clustering can be obtained by applying classical hierarchical clustering on the sparse dissimilarity matrix $\widehat{\mathbf{U}}$. Witten and Tibshirani (2010) showed, using a simulated dataset and a genomic dataset, that their proposed sparse hierarchical clustering results in more accurate identification of the underlying clusters and more interpretable results than standard hierarchical clustering and COSA when applied on datasets with noise features.

Criterion (3) was cleverly crafted to make hierarchical clustering fit in the general sparse framework. However, this procedure has some limitations. The reasoning for the choice of criterion (3) is not intuitive and this approach doesn't scale well with n. If we loosen the L_1 penalty on $\boldsymbol{w}$, i.e. set $s \to \infty$, the resulting clustering does not reduce to the classical hierarchical clustering. Furthermore, note that the sparse hierarchical clustering criterion is equivalent to applying SPC criterion to the $n^2 \times p$ transformed dissimilarity matrix $\boldsymbol{D}$, i.e. criterion (5), and the resulting "sparse" dissimilarity matrix $\widehat{\mathbf{U}}$ is calculated based on the loadings of the first sparse principal

component $\widehat{\mathbf{u}}$ of $\boldsymbol{D}$, which means the important variables are chosen according to the non-zero loadings of the first sparse principal component. However, as we may expect, the subset of features in $\mathbf{D}$, which correspond to the important clustering features may have a "complex structure" that could not be fully described using a single sparse principal component. A numerical illustration of this issue is shown below.

Example I: The data set X used in this example is generated as follows:

- X contains $n = 20$ observations with $p = 14$ features, i.e. $X_{n\times p} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)^T$, where $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,p})^T$, $1 \le i \le n$.
- The data are organized in four clusters of size 5.
- The first 4 features V_1, V_2, V_3, V_4 are clustering features generated as follows: $(x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})^T$ are generated from $N(\boldsymbol{\mu}(Y_i),\ 0.1 \times \boldsymbol{I}_4)$ $(1 \le i \le n)$ with

  $$\boldsymbol{\mu}(Y_i) = \mathbf{1}_4 \cdot I(Y_i = 1) + (-\mathbf{1}_2^T, \mathbf{1}_2^T)^T \cdot I(Y_i = 2) - \mathbf{1}_4 \cdot I(Y_i = 3) + (\mathbf{1}_2^T, -\mathbf{1}_2^T)^T \cdot I(Y_i = 4)$$

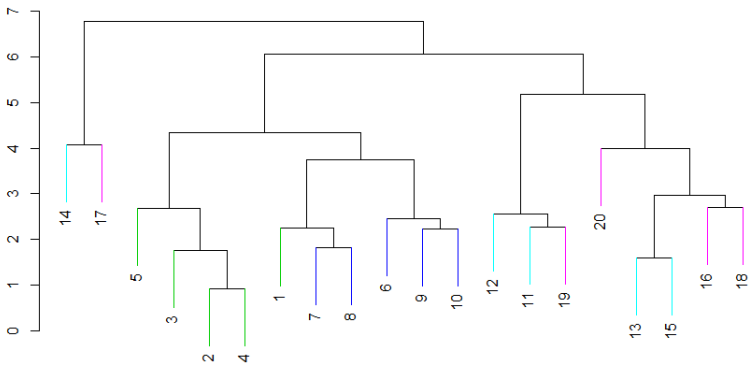
  where Y_i is the cluster indicator, $\boldsymbol{I}_k$ represents a $k \times k$ identity matrix and $\mathbf{1}_k$ represents a vector of k 1's.
- Ten noise features $V_5, V_6, \cdots, V_{14}$ are generated from $N(\mathbf{0}, \mathbf{I}_{10})$.

We apply Witten and Tibshirani's sparse hierarchical clustering framework to this data set. By gradually increasing the sparsity constraint, we obtain the following sequence of variables $\{V_{13}, V_{11}, V_6, V_2, V_1, V_{14}, V_3, V_8, V_4, V_7, V_5, V_{12}\}$ in which the first 3 chosen features are noise features. Regardless of the number of chosen features, say 4 (including only one clustering features) or 7 (4 noise features and 3 clustering features), the resulting dendrogram gives very mixed clusters. For example, the dendrogram generated by the first 7 features is presented in Figure 1.

We propose another framework for hierarchical clustering with feature selection capability to address the following questions simultaneously: "How many important features are there?" and "Which are these features?" A naive brute-force feature selection strategy for hierarchical clustering is: first consider all 2^p possible subsets of features, then by comparing the hierarchical clustering results obtained from these subsets, select the subset of features that gives the most satisfactory "performance". However, this approach is not feasible in practice because 2^p rapidly becomes too large as p increases. In addition, we need a comparison criterion to evaluate the different hierarchical clustering performances. To overcome these difficulties, we propose the following strategy:

1. Instead of all subsets of features, we consider several "candidate sizes" for the number of clustering features (i.e. the number of features we want to choose). For each candidate size, we select "candidate subsets" of potential clustering

Figure 1: Dendrogram generated with the first 7 features $\{V_{13}, V_{11}, V_6, V_2, V_1, V_{14}, V_3\}$ from Witten and Tibshirani's sparse hierarchical clustering framework



features based on the sparse loadings in SPC generated from the algorithm proposed by Witten et al. (2009). We apply the SPC algorithm directly on the original data matrix $\boldsymbol{X}$. This is more intuitive and less computationally burdensome than using the transformed dissimilarity matrix $\boldsymbol{D}$ as in Witten and Tibshirani (2010). Also, we take into account the sparse loadings in more than one SPC when selecting the features. Since the SPC algorithm is closely related to the sparse low-rank approximation, we refer to the number of SPCs as "rank" in the following sections.

2. In order to search for the "best" candidate subset, we propose the following comparison criterion:

   (a) To obtain the cluster labels from a dendrogram, we present a "multilayer" hierarchical clustering algorithm, inspired by the "multilayer" clustering proposed in Yan and Ye (2007), to produce cluster labels by adaptively cutting the dendrograms at different levels according to gap statistics.

   (b) We then apply hierarchical clustering to each candidate subset and obtain the corresponding cluster labels produced by the multilayer hierarchical clustering algorithm. Their clustering performances are compared by a comparison criterion based on silhouette values.

In the following sections, we refer to our proposed Sparse Hierarchical Clustering framework as SHC, and refer to Sparse Hierarchical Clustering proposed by Witten and Tibshirani (2010) as WT-SHC.

The remainder of the paper is organized as follows: Section 2 introduces our proposed SHC framework. Section 3 presents some simulation studies to compare the performance of SHC and WT-SHC. Section 4 discusses the computational complexity and times for SHC. Section 5 shows some applications of SHC on four microarray data sets. Finally, Section 6 gives the conclusion and further remarks.

2. A natural framework for sparse hierarchical clustering

Following the general idea illustrated in Section 1, SHC resembles an election process: First, we consider several possible numbers of clustering variables, referred to as "candidate sizes". For each candidate size, using sparse principal components (SPC), with careful "rank" considerations, we can form several "candidate clustering features" of that size that convey most of the information contained in the data. The best candidate subset for each candidate size can then be determined using a clustering performance-based criterion. Our criterion is a combination of the gap statistics (Tibshirani et al., 2001) and silhouette (Rousseeuw, 1987). After the best candidate subset of clustering features *for each candidate size* is found, we conduct a "higher-level" comparison among them (across different candidate sizes), again using our performance criterion. Thus, we obtain the "global" best candidate subset of important features and its corresponding size.

In the remainder of this section, we discuss SHC framework in details.

2.1. Multilayer hierarchical clustering

We first describe the "multilayer" hierarchical clustering algorithm inspired by the "multilayer" clustering proposed in Yan and Ye (2007). Our procedure accepts splits of the nodes in a hierarchical clustering dengrogram recursively and stops the process when the gap statistics indicates so or the number of clusters exceeds a certain "reference number of clusters". As a result, it produces the clustering labels which are necessary for the comparison of clustering performance.

Algorithm 1: Multilayer hierarchical clustering

Step 1. Set the inputs and arguments:

(a) A dendrogram resulted from hierarchical clustering.

(b) Number of bootstrap samples for the gap statistics, with default $B = 50$.

(c) Reference number of clusters K (see Remark 2.1).

Step 2. Examine splits in the dendrogram using the gap statistics:

(a) Given the dendrogram, assign the root node as the current node.

(b) Apply the gap statistics to the observations within the current node:

- If the resulting number of clusters is greater than 1: accept the binary split of the current node into two nodes according to the dendrogram.
- If the resulting number of clusters is 1: set the current node a terminal leaf.

(c) If there are K resulting leaves in the dendrogram, or if there are less than K leaves but all of them are terminal leaves, go to Step 3.

(d) Among all the leaves that are not terminal leaves, assign the leaf node with the highest height in dendrogram as the current node.

(e) Repeat Step 2(b)-(d).

Step 3. Output:

(a) The resulting number of clusters, which is less than or equal to K.

(b) The cluster labels for all the observations.

Remark 2.1. *Partitions with different number of clusters are in general difficult to compare. A reference number K of clusters is necessary in our procedure because otherwise we may obtain partitions with different number of clusters for different candidate subsets. Notice that given the sequential nature of hierarchical clustering, if more than K clusters were optimal, a partition with K clusters should also be reasonably good.*

K may be chosen by users based on their subject area knowledge. If not specified, the default reference number of clusters K is set as follows:

1. *Modify Algorithm 1 as follows:*
   - *Delete Step 1(c).*
   - *Replace Step 2(c) by:* (c) if all the leaves are terminal leaves, go to Step 3.
2. *Conduct hierarchical clustering on the whole data $X_{n\times p}$ and apply the modified Algorithm 1 to the resulting dendrogram. Record the resulting number of clusters K'.*
3. *Define the "reference number of clusters" K by $K = \max\{2, K'\}$.*

2.2. Algorithms for selecting the best candidate subset given a fixed candidate size

In this section, we propose our main algorithm for generating and selecting the best candidate subset given a fixed candidate size. An algorithm for automated choice of the candidate size is presented in the next section.

Algorithm 2: Candidate subset selection with fixed candidate size

Step 1. Set the inputs and arguments:

(a) Data matrix $X_{n\times p}$.

(b) Minimum and maximum rank to be considered: R^{min}, R^{max}.

(c) Pre-fixed candidate size q, reference number of clusters K.

Step 2. For each rank k in $[R^{min}, R^{max}]$:

(a) Assign $\lambda_k^- = 1$ and $\lambda_k^+ = \sqrt{n}$ (λ_k^+ can be set smaller in practice).

(b) Apply SPC algorithm to $X_{n\times p}$ with tunning parameter $\lambda_k^* = (\lambda_k^- + \lambda_k^+)/2$.

(c) Record the number of variables having *non-zero loadings in at least one of the first k SPCs*, say q_k^*.

(d) If $q_k^* = q$, assign $\lambda_k = \lambda_k^*$, and denote the corresponding q variables $V_{k1}, \cdots, V_{kq}$. Then go to Step 2(f).

(e) If $q_k^* > q$, then set $\lambda_k^+ = \lambda_k^*$, otherwise set $\lambda_k^- = \lambda_k^*$. Repeat Step 2(b)-(e).

(f) Form the candidate subset which contains variables from (d), i.e. $\mathcal{C}_k = \{V_{k1}, \cdots, V_{kq}\}$.

(g) Conduct hierarchical clustering using only variables in $\mathcal{C}_k$. Denote the resulting dendrogram as $\mathcal{HC}_k$.

Step 3. Evaluate and select the best candidate subset among $\mathcal{C}_k$'s:

(a) Apply Algorithm 1 to each dendrogram $\mathcal{HC}_k$ ($R^{min} \le i \le R^{max}$) from Step 2, obtain the resulting dendrograms and cluster labels.

(b) Screen out the resulting dendrograms from Step 3(a) which have *less than K* leaves.

(c) If all the dendrograms are screened out from Step 3(b), report "Please choose another set of ranks", then break the algorithm.

(d) Suppose c ($0 < c < R^{max} - R^{min}$) dendrograms are left from Step 3(b), keep record of their corresponding ranks $\mathcal{R} = \{r_1, r_2, \cdots, r_c\}$, candidate subsets $\mathcal{C}_{r_j}$ ($1 \le j \le c$), and cluster labels $\mathcal{L}_{r_j}$ ($1 \le j \le c$).

(e) For every rank in $\mathcal{R}$, calculate the average silhouette value, say $\texttt{AvgSil}_{r_j}$ ($1 \le j \le c$) using: the candidate subset $\mathcal{C}_{r_j}$, and the cluster labels $\mathcal{L}_{r_j}$.

(f) Create a scatter plot of the average silhouette values $\texttt{AvgSil}_{r_j}$ versus their corresponding ranks r_j ($1 \le j \le c$).

(g) Among all the candidate subset corresponding to the local minimum average silhouette values, discard the one with the highest rank.

(h) Create a new scatter plot of left average silhouette values and ranks from Step 3(g).

(i) If a monotonically increasing or decreasing plot results, go to Step 3(j), otherwise, repeat Step 3(g)-(i).

(j) If the average silhouette values are **decreasing** as the rank increases, then the smallest rank will be chosen. Go to Step 4.

(k) If the average silhouette values are **increasing** as the rank increases, then the smallest rank after the biggest increase in average silhouette values will be chosen.

Step 4. Outputs:

(a) The best candidate subset C_r and its corresponding rank r.

(b) Cluster labels $\mathcal{L}_r$ (with K clusters) and average silhouette value $\texttt{AvgSil}_r$.

Remark 2.2. *Different from WT-SHC, Algorithm 2 only uses the loadings in sparse principal components as indicators of potentially important variables, and we use the original variables instead of the weighted variables in the following hierarchical clustering.*

Remark 2.3. *Different from WT-SHC which only uses the first SPC to select features, Algorithm 2 compares candidate subsets for different ranks.* R^{min} *and* R^{max} *can take values within range* $1 \leq R^{min} \leq R^{max} \leq q$*, and we set the default values as* $R^{min} = 2$ *and* $R^{max} = \min\{8, q\}$ *since this setting yields a good balance between performance and the computational burden (*R^{min} *is not default to 1 since emperically the candidate subsets generated with more than one rank are always superior or at least as good).*

Remark 2.4. *When comparing the candidate subsets for different ranks, Algorithm 2 follows the following reasoning:*

1. *If we obtain less than K leaves in the resulting dendrogram when apply Algorithm 1 to a subset of features, then we consider it a "poor" subset of features and screens it out of the further comparison, since it failed to reveal enough number of underlying clusters in the data.*

2. *We evaluate the left over candidate subsets from (a) by plotting their corresponding average silhouette values against the ranks, then we propose the comparison criterion which follows the reasonings below:*

   (a) Higher average silhouette values are prefered since they implies that the clusters separate the data well.

   (b) Given "similar" average silhouette values, the subsets with lower rank are prefered since the subset corresponding to a higher rank may be more likely to contain noise variables.

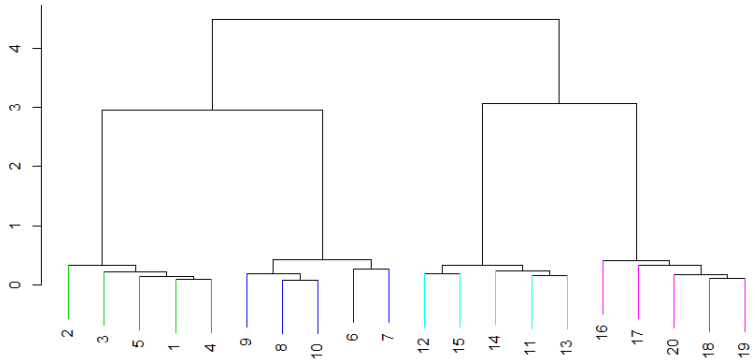
   (c) Among all the candidate subset corresponding to the local minimum average silhouette values, discard the one with the highest rank, since it will never be selected as the best subset due to i and ii.

(d) *If the average silhouette values are increasing as rank increases, then in order to incorporate both i and ii, we select the candidate subset corresponding to the smallest rank after the largest increase in average silhouette value, which means we prefer smaller ranks unless the increase in average silhouette values is relatively large.*

(e) *If the average silhouette values are decreasing as rank increases, then the best subset can be easily chosen as the candidate subset corresponding to the smallest rank.*

In the reminder of this section, we revisit Example I in Section 1. Suppose that the candidate size $q = 4$ is correctly specified, and we apply our proposed Algorithm 2 with default choice of reference number of clusters ($K = 2$ in this experiment). The resulting best candidate subset C_r with corresponding rank $r = 2$ contains the 4 true clustering features: V_1, V_2, V_3 and V_4, and the resulting dendrogram is presented in Figure 2. We can see that the 4 clusters are separated correctly according to the chosen subset of features.

Figure 2: Dendrogram generated from our proposed Algorithm 2 with four chosen features **V1**, **V2**, **V3** and **V4**



2.3. Algorithms for selecting the "global" best candidate subset when the candidate size is unknown

In the previous section, we present an algorithm to select the best candidate subset for a fixed candidate size. In this section, we introduce an algorithm for selecting the "global" best candidate subset when the candidate size is unknown. We propose to evaluate and compare the performance of the best candidate subsets with different

sizes. Similar to the comparison criterion in Algorithm 2, Step 3, the comparison criterion is based on the following ideas:

1. Candidate subsets with higher average silhouette values are preferred.
2. Candidate subsets with smaller sizes are preferred.

Detail implementation of these ideas are listed in Algorithm 3, Step 2(b)-(g).

Algorithm 3: Candidate subset selection with unknown candidate size

Step 1. Setting the inputs and arguments:

(a) Data matrix $X_{n\times p}$.

(b) Minimum and maximum rank to be considered: R^{min}, R^{max}.

(c) default list of candidate size $q_1, \cdots, q_d$.

(d) reference number of clusters K.

Step 2. selecting the "global" best candidate subset:

(a) Apply Algorithm 2 with every candidate size q_i in $q_1, q_2, \cdots, q_d$, obtain the corresponding:

- best candidate subset $C_{r_i}^{(q_i)}$,
- cluster labels $\mathcal{L}_{r_i}^{(q_i)}$,
- average silhouette value $\texttt{AvgSil}_{r_i}^{(q_i)}$.

(b) Create a scatter plot of the average silhouette values $\texttt{AvgSil}_{r_i}^{(q_i)}$ versus their corresponding ranks candidate size q_i $(1 \le i \le d)$.

(c) Discard the local minimum silhouette values and their candidate sizes.

(d) Create a new scatter plot of left average silhouette values and candidate sizes from Step 2(c).

(e) If a monotonically increasing or decreasing plot results, go to Step 3(f), otherwise, repeat Step 2(c)-(e).

(f) If the average silhouette values are **decreasing** as the size increases, then the smallest size will be chosen. Go to Step 4.

(g) If the average silhouette values are **increasing** as the size increases, then the smallest size after the biggest increase in average silhouette values will be chosen.

Step 3. Outputs:

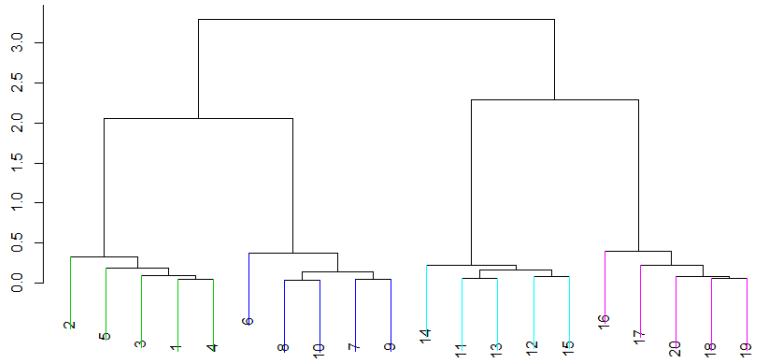
(a) The best candidate size as q_r.

(b) The "global" best candidate subset $C_r^{(q)}$, and the corresponding dendrogram $\mathcal{HC}_r^{(q)}$.

With the proposed Algorithm 2-3, we are able to obtain the sparse hierarchical clustering with automatically selected "best" subset of features.

Again, we revisit Example I in Section 1, but with automatically selected candidate size q_r. We apply Algorithm 3 with default choice of reference number of clusters ($K = 2$ in this experiment) and list of candidate size $\{2, 3, \cdots, 8\}$. The resulting best candidate size is $q_r = 2$ with corresponding best candidate subset $C_r^{(q)} = \{\ V2, V3\ \}$ and rank $r = 2$. We can see that although the auto-selected candidate size q_r is 2 instead of the "true size" 4, the resulting dendrogram from the 2 chosen features **V2** and **V3** can still produce perfect clusters as presented in Figure 3.

Figure 3: Dendrogram generated from our proposed Algorithm 3 with 2 chosen features **V2** and **V3**



3. Simulation study

In this section, we conduct simulation study to evaluate and compare the performance of SHC, and WT-SHC proposed by Witten and Tibshirani (2010).

We simulate data sets from the following "sparse model":

- The data set X contains $n = 60$ observations and $p = 500, 1000, 1500$ features, i.e. $X_{n\times p} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)^T$, where $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,p})^T$.
- The data are organized in 3 clusters of size $n/3 = 20$.
- The first $p' = 50$ clustering features are generated from $N(\mu(Y_i), I_{p'})$ with

$$\mu(Y_i) = \mu\mathbf{1}_{p'} I(Y_i = 1) + (-\mu\mathbf{1}_{p'/2}^T, \mu\mathbf{1}_{p'/2}^T)^T I(Y_i = 2) - \mu\mathbf{1}_{p'} I(Y_i = 3)$$

where Y_i is the cluster indicator, $\mathbf{1}_k$ represents a vector of k ones and the magnitude of μ controls how far the clusters are separated from each other. We use $\mu = 0.8, 0.6$.

- The remaining $p - p'$ non-clustering features are generated from $N(\mathbf{0}, I_{p-p'})$.

Each combination of these settings is replicated 100 times.

Both SHC and WT-SHC with linkage "complete" are applied to each simulated data set (linkages "average" and "ward" are also considered, the results are similar to those from "complete" and hence omitted). The partitions are obtained by cutting the dendrogram into $K = 3$ clusters, and the clustering performances are measured by the classification error rate (CER) defined in Chipman and Tibshirani (2006). Notice that both SHC and WT-SHC need to define the sparsity constraint or candidate size of clustering features, say q, here we compare the two frameworks under either fixed or unknown candidate size. To be more specific, we consider situations when $q = 30, 50, 70,$ or "auto". When $q =$ "auto", we don't provide any pre-specified candidate sizes, instead, we let SHC and WT-SHC choose their candidate sizes automatically. In order to make a fair comparison, we set the default candidate size list for both methods as: $\{10, 20, \cdots, 100\}$.

The averages (and standard deviations) of CERs for the different settings are presented in Table 1. We can see that consistently, SHC produces much smaller average CER, which means SHC produces, in general, more accurate clusters than the WT-SHC in our simulation study. We should also notice that as the signal-to-noise ratio increases (e.g. μ decreases and/or p increases), the performance of both methods deteriorated, however, WT-SHC is affected more severely.

Table 1: Average (standard deviation) of CERs in different simulation settings

		$\mu = 0.8$		$\mu = 0.6$	
p	q	SHC	WT-SHC	SHC	WT-SHC
$p = 500$	$q = 30$	0.017 (0.021)	0.158 (0.101)	0.081 (0.043)	0.355 (0.104)
	$q = 50$	0.004 (0.012)	0.077 (0.090)	0.036 (0.034)	0.283 (0.114)
	$q = 70$	0.001 (0.005)	0.030 (0.053)	0.019 (0.025)	0.224 (0.125)
	$q =$ auto	0.034 (0.041)	0.234 (0.106)	0.061 (0.051)	0.397 (0.106)
$p = 1000$	$q = 30$	0.022 (0.029)	0.173 (0.101)	0.084 (0.042)	0.405 (0.096)
	$q = 50$	0.006 (0.013)	0.073 (0.084)	0.062 (0.050)	0.324 (0.115)
	$q = 70$	0.001 (0.004)	0.045 (0.073)	0.031 (0.034)	0.270 (0.112)
	$q =$ auto	0.033 (0.043)	0.275 (0.102)	0.071 (0.060)	0.400 (0.118)
$p = 1500$	$q = 30$	0.033 (0.034)	0.178 (0.118)	0.103 (0.069)	0.415 (0.100)
	$q = 50$	0.007 (0.018)	0.089 (0.099)	0.062 (0.046)	0.362 (0.109)
	$q = 70$	0.004 (0.011)	0.042 (0.067)	0.040 (0.032)	0.289 (0.117)
	$q =$ auto	0.023 (0.039)	0.278 (0.118)	0.071 (0.061)	0.430 (0.092)

Another important evaluation of sparse clustering methods is the selection rates

of the important clustering features. For example, we specify the true number of important variables $p' = 50$, suppose a sparse clustering method chooses p^* ($p^* \leq p'$) true clustering variables under a candidate size q, then the selection rate (SR) is calculated by the following formula:

$$\text{SR} = \begin{cases} p^*/q, & \text{if } q \leq p', \\ p^*/p', & \text{otherwise} \end{cases}$$

Table 2 presents the averages (and standard deviations) of SR's in settings where the candidate size q's are fixed. We can see that when candidate sizes are fixed, SHC framework produces larger average SR's, which indicates SHC gives, in general, a more accurate feature selection than WT-SHC in our simulation study. Again, the selection accuracies of both methods are affected by larger signal-to-noise ratio, and again WT-SHC is affected more severely.

Table 2: Average (standard deviation) of SR's in different simulation settings

		$\mu = 0.8$		$\mu = 0.6$	
p	q	SHC	WT-SHC	SHC	WT-SHC
$p = 500$	$q = 30$	0.914 (0.146)	0.815 (0.082)	0.807 (0.176)	0.456 (0.102)
	$q = 50$	0.820 (0.156)	0.662 (0.063)	0.718 (0.152)	0.391 (0.061)
	$q = 70$	0.967 (0.076)	0.741 (0.061)	0.833 (0.106)	0.474 (0.074)
$p = 1000$	$q = 30$	0.792 (0.220)	0.723 (0.097)	0.709 (0.208)	0.338 (0.110)
	$q = 50$	0.727 (0.181)	0.558 (0.070)	0.536 (0.151)	0.279 (0.069)
	$q = 70$	0.903 (0.144)	0.634 (0.066)	0.689 (0.146)	0.334 (0.067)
$p = 1500$	$q = 30$	0.702 (0.239)	0.649 (0.121)	0.643 (0.234)	0.263 (0.100)
	$q = 50$	0.653 (0.190)	0.524 (0.069)	0.501 (0.154)	0.236 (0.057)
	$q = 70$	0.829 (0.164)	0.571 (0.078)	0.577 (0.143)	0.274 (0.066)

Table 3 presents the averages (and standard deviations) of the auto-selected candidate sizes q and their corresponding SR's in settings where candidate sizes q's are auto-selected. We can see that SHC always results in larger auto-selected candidate sizes (which are closer to the true 50). It is not natural to compare SR's given different auto-selected candidate sizes, however, we may see that both methods result in relatively similar SR's when $\mu = 0.8$, but when $\mu = 0.6$, SR's drop severely and again the SR's from WT-SHC are more affected.

4. Computational times and complexity

In this section, we investigate the computational times of SHC and WT-SHC. For WT-SHC, `HierarchicalSparseCluster` function from the R-package `sparcl` is used. We implement our proposed approach in `R`, and for the sparse PCA algorithm used in our framework, we use function `SPC` from the R-package `PMA`. We notice that

Table 3: Average (standard deviation) of SR's with auto-selected q

		$\mu = 0.8$		$\mu = 0.6$	
p	q	SHC	WT-SHC	SHC	WT-SHC
$p = 500$	q	23.1 (10.21)	17.78 (9.27)	37.50 (15.98)	16.09 (14.86)
	SR	0.907 (0.143)	0.905 (0.096)	0.767 (0.173)	0.585 (0.195)
$p = 1000$	q	27.80 (12.35)	16.02 (8.88)	41.20 (16.28)	19.76 (20.11)
	SR	0.800 (0.194)	0.853 (0.135)	0.605 (0.191)	0.403 (0.185)
$p = 1500$	q	35.00 (17.72)	15.89 (8.06)	44.70 (17.78)	15.32 (12.37)
	SR	0.740 (0.207)	0.786 (0.170)	0.550 (0.172)	0.351 (0.176)

SHC can be naturally fit into parallel computing since we can select the candidate subsets for different ranks separately. Therefore, we use the function `mcapply` from the R-package `multicore` to speed up the computation. Although our framework is currently implemented in R, we plan to implement our algorithm fully in `C++` to further boost the speed. In this section, we still simulate data sets using the "sparse model" introduced in Section 3. We fix the following settings in the sparse model: true number of important variables $p' = 50$, $\mu = 0.6$, then we simulate 50 data sets for each of the combinations of the following settings:

- $n = 60, 150, 300$.
- $p = 500, 1000, 2000$.

Both SHC and WT-SHC with linkage "complete" are applied to the simulated data sets. We consider the case when the candidate size is correctly specified $q = 50$ for simplicity. Also, for SHC, we consider different settings for the number of bootstraps in Algorithm 1, say $B = 50, 100, 150$. The resulting CER's under different settings are plot against their corresponding computational times (in seconds) in Figure 4.

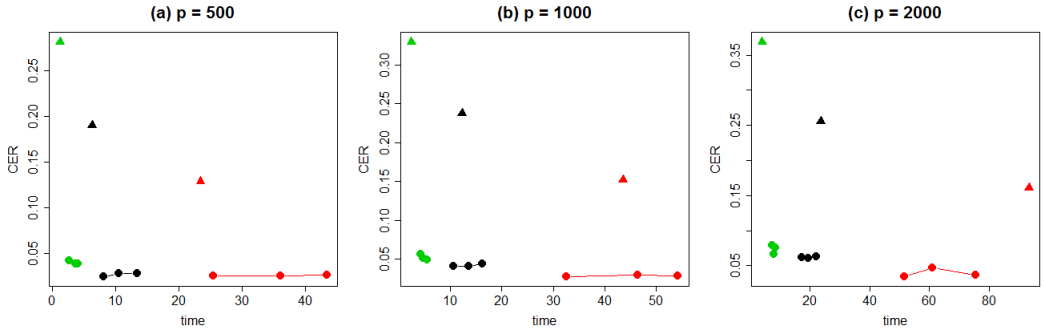


Figure 4: CERs vs. Times for different settings. (Different colors – green, black and red represent $n = 60, 150, 300$ respectively, triangle points represent WT-SHC and solid circle points represent SHC with different number of bootstraps B.)

From Figure 4, we may conclude the following:

- Increasing the number of bootstraps B in Algorithm 1 leads to longer computational times but little increase in cluster accuracy. In Algorithm 1, we set default $B = 50$.
- When n and p are relatively small, WT-SHC is less time consuming than SHC with $B = 50$. However, when n and p get larger, WT-SHC spends more time than SHC (even with $B = 150$). This can be explained by the computational complexities of the two approaches: $O(n^3qB + np)$ for SHC, and $O(n^3q + n^2p)$ for WT-SHC. Therefore, when n and p increase and $p >> n$, WT-SHC becomes more computationally demanding due to the n^2p term in the complexity.
- In every case, SHC gives more accurate clustering performance.

However, we need to notice that since hierarchical clustering do not scale when n is large, both SHC and WT-SHC do not scale well as a consequence.

5. Application to microarray data sets

In this section, we apply SHC to four benchmark microarray data sets, Lymphoma (Alizadeh et al., 2000), Leukemia (Golub et al., 1999), and two breast cancer data sets (Perou et al. (2000) and van't Veer et al. (2002)). For each data set, we perform four versions of hierarchical clustering with complete linkage on the observations:

1. Classical hierarchical clustering using all the features (i.e. genes).
2. WT-SHC with q_1 auto-selected genes.
3. SHC with q_2 auto-selected genes.
4. Hierarchical clustering using the q_2 genes with the highest marginal variance.

Note that in this section, we do not specify the reference number of clusters in SHC, and the default method (see Remark 2.1 for details) is used. Also, we choose the default list of candidate size to be $\{10, 20, 40, 60, 80, 100, 125, 150, 175, 200, 233, 266, 300, 350, 400, 500\}$, which is used by both SHC and WT-SHC when selecting the number of features.

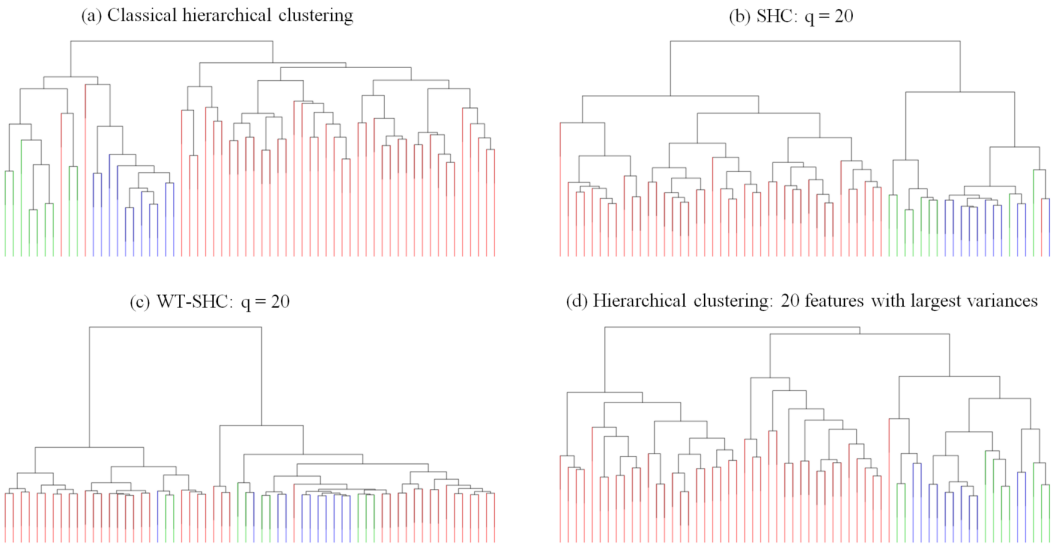
5.1. Lymphoma data set

In the lymphoma data set, the total sample size is 62 and the number of genes is 4026. Three types of most prevalent adult lymphoid malignancies were studied: 42 cases of diffuse large B-cell lymphoma (DLBCL), 9 samples of follicular lymphoma (FL), and 11 observations of B-cell chronic lymphocytic leukemia (CLL). A specialized cDNA microarray was used to measure the gene expression levels. The data set

is provided by Dettling (2004) and available at `http://stat.ethz.ch/~dettling/bagboost.html`. Following the pre-processing steps in Dudoit et al. (2002), the data set is pre-processed by first setting a thresholding window $[100, 16000]$ and then excluding genes with $\max / \min \leq 5$ or $(\max - \min) \leq 500$. Finally a logarithmic transformation and standardization are applied. For the original lymphoma data set, some arrays contain genes with missing values. As suggested in Dudoit et al. (2002), a simple 5 nearest neighbor algorithm is employed to impute the missing values.

We apply the above mentioned four versions of hierarchical clustering to Lymphoma data set and the resulting dendrograms are shown in Figure 5. We can see that classical hierarchical clustering using all the features does well with only two misclassified cases, however, SHC achieves similar accuracy (3 misclassified cases) with only 20 auto-selected features. WT-SHC also selected 20 variables, but its performance is not satisfactory. Classical hierarchical clustering using 20 features with the highest marginal variances also gives mixed clusters. Thus, we may conclude that SHC performs the best in the sense of producing accurate and interpretable results.

Figure 5: Dendrograms for lymphoma data ($n = 62$, $p = 4026$)



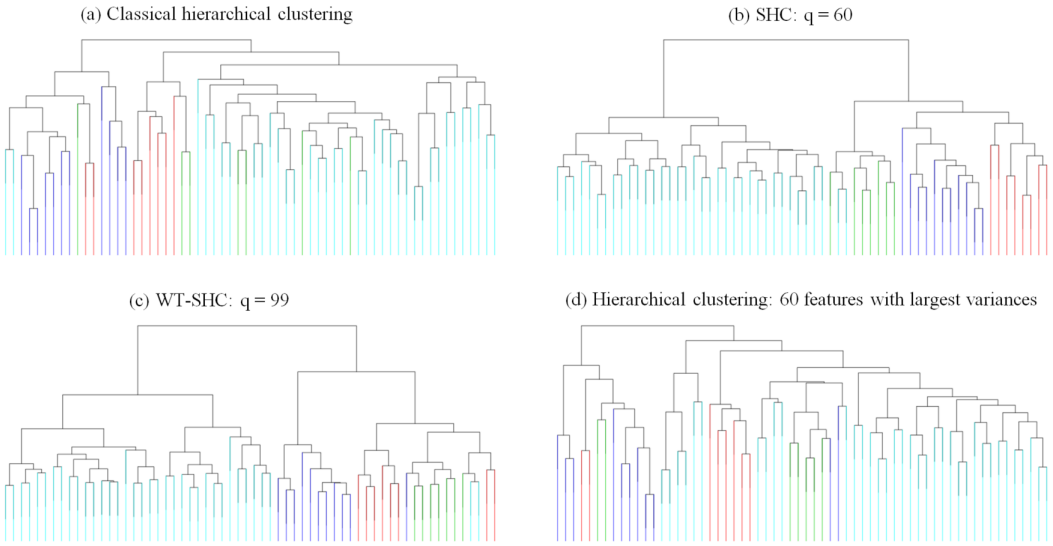
5.2. Breast cancer data set in Perou et al. (2000)

In this breast cancer data set, Perou et al. (2000) used 1753 gene expression microarrays to profile 65 surgical specimens of human breast tumors. Some of the samples were taken from the same tumor before and after chemotherapy. They identified four classes – basal-like, Erb-B2, normal breast-like, and ER+ – to which 62 of the 65 samples belong and it was determined that the remaining three observations did not belong to any of the four classes. Following Witten and Tibshirani (2010), we

focus only on the 62 samples. The data are publicly available at `http://genome-www.stanford.edu/breast_cancer/molecularportraits/download.shtml`.

The resulting dendrograms are shown in Figure 6. We can see that classical hierarchical clustering using all the features gives mixed clusters. WT-SHC with 100 auto-selected features performs well by misclassifying only 5 cases. SHC works even better with only 60 auto-selected features, and only 2 cases are misclassified. Standard hierarchical clustering using 60 features with the highest marginal variances produces clusters with low accuracy. Thus, we may conclude that, for this data set, SHC gives the best performance.

Figure 6: Dendrograms for breast cancer data ($n = 62$, $p = 1753$)



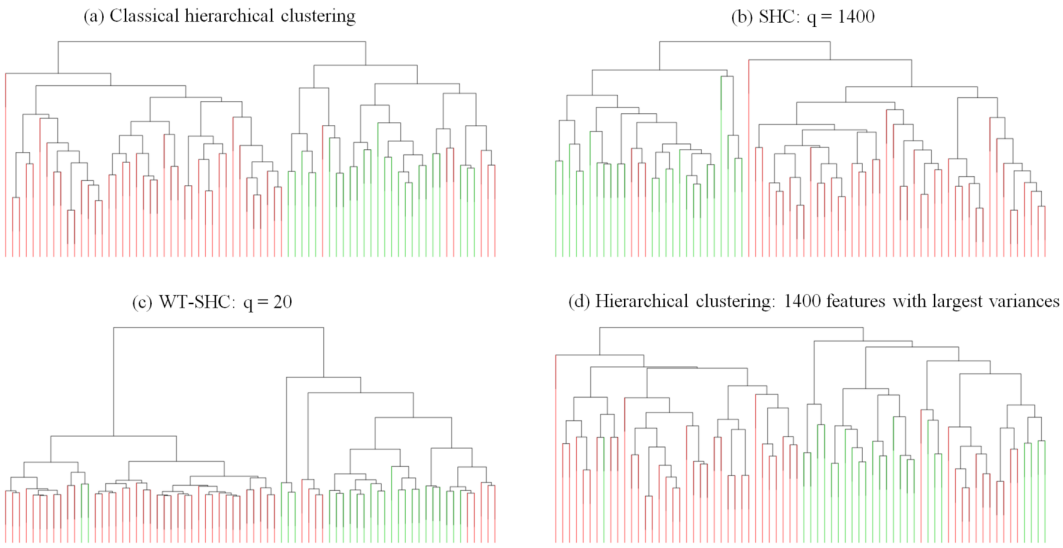
5.3. Leukemia data set

In the Leukemia data, Golub et al. (1999) used microarray gene expression data, which were measured by Affymetrix microarrays containing 6817 human genes, to identify two types of human acute leukemias: acute myeloid leukemia (AML) and acute lymphoblastic leukemia(ALL). This data set consists of 72 patients – 25 patients with AML and 47 patients with ALL. Distinguishing ALL from AML is clinically significant for successful treatment because those chemotherapy regimens for ALL patients are different from AML patients, in which case using ALL therapy for AML (and vice versa) cases may result in distinctly reduced cute rates and possible toxicities. The data set is also pre-processed by following the pre-processing steps in Dudoit et al. (2002), which is described briefly in Section 4.1.

The resulting dendrograms are shown in Figure 7. We can see that classical hierarchical clustering using all the features gives 6 misclassified cases. WT-SHC only

selected 20 features, but produces 11 misclassified cases, which means the candidate size is selected poorly. SHC selects 1400 features and produces only 2 misclassified cases. Note that the number of features is not included in the default list, since all the candidate subsets with candidate sizes in the default list produce only 1 cluster, which is not useful. So we expand the default list to $\{100, 200, \cdots, 1400, 1500\}$. Classical hierarchical clustering using 1400 features with the highest marginal variances produces mixed clusters. Thus, we may conclude that, for this data set, SHC gives relatively better clustering accuracy, however, the number of features is larger than expected.

Figure 7: Dendrograms for leukemia data ($n = 72$, $p = 3571$)



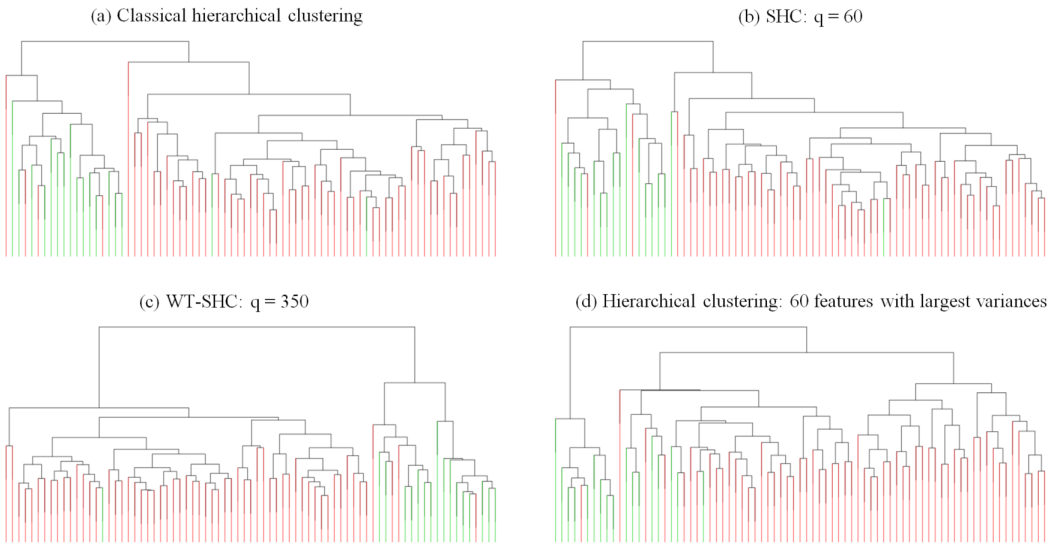
5.4. Breast cancer data set in van't Veer et al. (2002)

van't Veer et al. (2002) analyzed a data set consisting of 4751 gene expressions for 78 primary breast tumors. The data set is publicly available at: `http://www.nature.com/nature/journal/v415/n6871/suppinfo/415530a.html`. These authors applied a supervised classification technique and found subset of 70 genes that helps to discriminate patients that develop distant metastasis within 5 years from others.

We then apply the four versions of hierarchical clustering and the resulting dendrograms are shown in Figure 8. We can see that classical hierarchical clustering using all the features misclassifies 6 cases. SHC achieves slightly better accuracy (4 misclassified cases) with only 60 auto-selected features. Note that the number of features chosen is close to that found in van't Veer et al. (2002). WT-SHC achieves the same accuracy with 350 features. Classical hierarchical clustering using 60 features

with the highest marginal variances produces clusters with relatively low accuracy (9 misclassified cases). Thus, we may conclude that, for this data set, SHC performs the best in the sense of producing accurate and interpretable results (i.e. less features).

Figure 8: Dendrograms for breast cancer data ($n = 77$, $p = 4751$)



6. Conclusion

We propose a framework for sparse hierarchical clustering (SHC), which selects important clustering features for hierarchical clustering automatically and produces the corresponding sparse hierarchical clustering. As demonstrated in simulation studies and real data examples, SHC gives superior feature selection and clustering performance comparing with the classical hierarchical clustering and WT-SHC.

References

References

Alizadeh, Ash A; Eisen, Michael B; Davis, R Eric; Ma, Chi; Lossos, Izidore S; Rosenwald, Andreas; Boldrick, Jennifer C; Sabet, Hajeer; Tran, Truc; Yu, Xin, and others, . Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

Chipman, Hugh and Tibshirani, Robert. Hybrid hierarchical clustering with applications to microarray data. *Biostatistics*, 7(2):286–301, 2006.

Dettling, Marcel. Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593, 2004.

Dudoit, Sandrine; Fridlyand, Jane, and Speed, Terence P. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

Friedman, Jerome H and Meulman, Jacqueline J. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):815–849, 2004.

Golub, Todd R; Slonim, Donna K; Tamayo, Pablo; Huard, Christine; Gaasenbeek, Michelle; Mesirov, Jill P; Coller, Hilary; Loh, Mignon L; Downing, James R; Caligiuri, Mark A, and others, . Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

Hardin, Johanna; Mitani, Aya; Hicks, Leanne, and VanKoten, Brian. A robust measure of correlation between two genes on a microarray. *BMC bioinformatics*, 8(1): 220, 2007.

Pan, Wei and Shen, Xiaotong. Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research*, 8:1145–1164, 2007.

Perou, Charles M; Sørlie, Therese; Eisen, Michael B; van de Rijn, Matt; Jeffrey, Stefanie S; Rees, Christian A; Pollack, Jonathan R; Ross, Douglas T; Johnsen, Hilde; Akslen, Lars A, and others, . Molecular portraits of human breast tumours. *Nature*, 406(6797):747–752, 2000.

Raftery, Adrian E and Dean, Nema. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

Rousseeuw, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Sun, Wei; Wang, Junhui; Fang, Yixin, and others, . Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6:148–167, 2012.

Tibshirani, Robert; Walther, Guenther, and Hastie, Trevor. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

van't Veer, Laura J; Dai, Hongyue; Van De Vijver, Marc J; He, Yudong D; Hart, Augustinus AM; Mao, Mao; Peterse, Hans L; van der Kooy, Karin; Marton, Matthew J; Witteveen, Anke T, and others, . Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415(6871):530–536, 2002.

Wang, Sijian and Zhu, Ji. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64(2):440–448, 2008.

Witten, Daniela M and Tibshirani, Robert. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

Witten, Daniela M; Tibshirani, Robert, and Hastie, Trevor. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

Xie, Benhuai; Pan, Wei, and Shen, Xiaotong. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, 2:168, 2008.

Yan, Mingjin and Ye, Keying. Determining the number of clusters using the weighted gap statistic. *Biometrics*, 63(4):1031–1037, 2007.