# Chapter 1

# Finding $\kappa$

Both the SVD Signs Algorithm and the SVD Gaps Algorithm require a value for $\kappa$ when we use the Reduced SVD (or Approximate SVD) of $\bar{\bar{\mathbf{L}}}$. Thus, we must have both an understanding of what $\kappa$ represents, as well as a process for reliably determining its value.

## 1.1 What is $\kappa$

To begin to understand what $\kappa$ is, let us define $r \in \mathbb{Z}^+$ as

$$r = \text{rank}\left[\bar{\bar{\mathbf{L}}}\right]$$

Next, recall the expression for the full Singular Value Decomposition of $\bar{\bar{\mathbf{L}}}$ from chapter 2 written below

$$\bar{\bar{\mathbf{L}}} = \bar{\bar{\mathbf{U}}}\ \bar{\bar{\mathbf{S}}}\ \bar{\bar{\mathbf{V}}}^{\intercal},$$

where $\bar{\bar{\mathbf{U}}} \in \mathbb{R}^{m\times r}$, $\bar{\bar{\mathbf{S}}} \in \mathbb{R}^{r\times r}$, and $\bar{\bar{\mathbf{V}}} \in \mathbb{R}^{n\times r}$. This is expensive to compute, in both time and memory; therefore, we seek something which can be computed for less cost.

Let us now create new versions of the $\bar{\bar{\mathbf{U}}}$, $\bar{\bar{\mathbf{S}}}$, and $\bar{\bar{\mathbf{V}}}$ matrices, denoted $\bar{\bar{\mathbf{U}}}_\xi$, $\bar{\bar{\mathbf{S}}}_\xi$, and $\bar{\bar{\mathbf{V}}}_\xi$, respectively. We will define these new versions of the $\bar{\bar{\mathbf{U}}}$, $\bar{\bar{\mathbf{S}}}$, and $\bar{\bar{\mathbf{V}}}$ matrices as such that

$$\bar{\bar{\mathbf{U}}}_\xi \in \mathbb{R}^{m\times\xi}$$

$$\bar{\bar{\mathbf{S}}}_\xi \in \mathbb{R}^{\xi\times\xi}$$

and

$$\bar{\bar{\mathbf{V}}}_\xi \in \mathbb{R}^{n\times\xi}$$

where $\xi \in \mathbb{Z}++ \ni \xi < r$. Since $\xi < r$, these new matrices are obviously less expensive to store. Additionally, note that if we compute the matrix product of $\bar{\bar{\mathbf{U}}}_\xi$, $\bar{\bar{\mathbf{S}}}_\xi$, and $\bar{\bar{\mathbf{V}}}_\xi$, we obtain the result

$$\left(\bar{\bar{\mathcal{L}}} = \bar{\bar{\mathbf{U}}}_\xi\, \bar{\bar{\mathbf{S}}}_\xi\, \bar{\bar{\mathbf{V}}}_\xi^{\intercal}\right) \in \mathbb{R}^{m\times n} \tag{1.1.1}$$

which is clearly such that $\bar{\bar{\mathcal{L}}} \in \mathbb{R}^{m\times n}$. Since $\bar{\bar{\mathbf{S}}}$ is the *Singular Value Matrix* of $\bar{\bar{\mathbf{L}}}$, its only non-zero elements are the $r$ singular values of $\bar{\bar{\mathbf{L}}}$. These elements are arranged along the diagonal of $\bar{\bar{\mathbf{S}}}$ and ordered from highest to lowest. That is to say, for the singular values of $\bar{\bar{\mathbf{L}}}$, $\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ we have $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_i \geq \cdots \geq \sigma_r$. Now, the singular values of $\bar{\bar{\mathbf{L}}}$ can be found by computing the eigenvalues of $\bar{\bar{\mathbf{L}}}^{\intercal}\bar{\bar{\mathbf{L}}}$ (denoted $\lambda_i^\star$), followed by taking the real component of the square root of each of these eigenvalues. That is to say, the value of each $\sigma_i$ can be found using the relation

$$\sigma_i = \mathrm{Re}\left[\sqrt{\lambda_i^\star}\right] \tag{1.1.2}$$

Thus the diagonal elements of $\bar{\bar{\mathbf{S}}}$, $\left\{\left(\bar{\bar{\mathbf{S}}}\right)_{11},\left(\bar{\bar{\mathbf{S}}}\right)_{22},\ldots,\left(\bar{\bar{\mathbf{S}}}\right)_{ii},\ldots,\left(\bar{\bar{\mathbf{S}}}\right)_{rr}\right\}$, are such that

$$\left(\bar{\bar{\mathbf{S}}}\right)_{11}\geq\left(\bar{\bar{\mathbf{S}}}\right)_{22}\geq\cdots\geq\left(\bar{\bar{\mathbf{S}}}\right)_{ii}\geq\cdots\geq\left(\bar{\bar{\mathbf{S}}}\right)_{rr}$$

Since our new $\bar{\bar{\mathbf{S}}}_\xi$ is constructed from $\bar{\bar{\mathbf{S}}}$, $\bar{\bar{\mathbf{S}}}_\xi$ also contains the singular values of $\bar{\bar{\mathbf{L}}}$ arranged along its diagonal by magnitude. Therefore, the elements of $\bar{\bar{\mathbf{S}}}_\xi$ must be such that

$$\left\{\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{11},\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{22},\ldots,\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{\iota\iota},\ldots,\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{\xi\xi}\right\}\subset\left\{\left(\bar{\bar{\mathbf{S}}}\right)_{11},\left(\bar{\bar{\mathbf{S}}}\right)_{22},\ldots,\left(\bar{\bar{\mathbf{S}}}\right)_{ii},\ldots,\left(\bar{\bar{\mathbf{S}}}\right)_{rr}\right\}$$

Additionally, since $\xi < r$, we have

$$\left|\left\{\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{11},\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{22},\ldots,\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{\iota\iota},\ldots,\left(\bar{\bar{\mathbf{S}}}_\xi\right)_{\xi\xi}\right\}\right|<\left|\left\{\left(\bar{\bar{\mathbf{S}}}\right)_{11},\left(\bar{\bar{\mathbf{S}}}\right)_{22},\ldots,\left(\bar{\bar{\mathbf{S}}}\right)_{ii},\ldots,\left(\bar{\bar{\mathbf{S}}}\right)_{rr}\right\}\right|$$

All that remains is to decide *which* elements of $\bar{\bar{\mathbf{S}}}$ to include in $\bar{\bar{\mathbf{S}}}_\xi$, subject to the criteria above. Let us

## 1.2 Choosing $\kappa$

From here, there are six possible ways to choose $\kappa$. The first five are quite simple and are

1. $\kappa \ni \forall i \in [1,k], \sigma_i > (\sigma_1/2)$
2. $\kappa \ni \forall i \in [1,k], \sigma_i > \sqrt{\sigma_1}$
3. $\kappa \ni \forall i \in [1,k], \sigma_i > \left(\frac{1}{2}\left((\sigma_1/2)+\sqrt{\sigma_1}\right)\right)$
4. $\kappa \ni \forall i \in [1,k], \sigma_i > \min\left[(\sigma_1/2), \sqrt{\sigma_1}\right]$
5. $\kappa \ni \forall i \in [1,k], \sigma_i > \max\left[(\sigma_1/2), \sqrt{\sigma_1}\right]$

The sixth option is considerably more complicated as it requires find the point (singular value index) where the rate of change in the difference between adjacent singular values, $\Delta\sigma_i$, changes most sharply. Where we define $\Delta\sigma_i$ as

$$\Delta\sigma_i = \begin{cases} 0 & \underline{\text{for}}\ i = 1 \\ \sigma_i - \sigma_{i+1} & \underline{\text{otherwise}} \end{cases} \tag{1.2.1}$$

This is to say, we seek the point at which the difference between adjacent $\Delta\sigma_i$ changes most abruptly. Thus, we seek the value for $j$, where $j \in (1, r]$, such that

$$\Delta\sigma_j - \Delta\sigma_{j+1}$$

is a maximum. Graphically, this appears as an 'elbow' in the graph of the singular values versus their associated indices. The index, $j$, at which this 'elbow' occurs represents the desired value for $\kappa$. If time permits, we will attempt to create an algorithm to automatically find the value for $\kappa$ using this method; however, for now, this will have to be done manually.

Finally, with the required value of $\kappa$ in hand, it is possible to proceed to each of the three clustering algorithm that will be employed, starting with the *SVD Signs* algorithm.

# Chapter 2

# SVD Signs

The SVD Signs Algorithm has algorithm has eight steps of its own and one step it shares with the SVD Gaps Algorithm. The step shared between the These steps are

1. Compute the $k$th order approximate SVD of $\bar{\bar{\mathbf{A}}}$.
2. Compare the row sign patterns.
3. Complete the row cluster's sets.
4. Create single sets for each row cluster.
5. Compare the column sign patterns.
6. Complete the column cluster's sets.
7. Create single sets for each column cluster.
8. Reorder $\bar{\bar{\mathbf{A}}}$ so that rows and columns from the same cluster have indices which are adjacent.

## 2.1 Choosing which SVD

For the first step, we simply use the expression from (1.2) with the value for $k$ found previously. Thus, we will proceed with

$$\bar{\bar{\mathbf{A}}} \approx \bar{\bar{\mathbf{U}}}_k \, \bar{\bar{\mathbf{S}}}_k \, \bar{\bar{\mathbf{V}}}_k^{\intercal} \tag{1.2}$$

from which we will use $\bar{\bar{\mathbf{U}}}_k$ for clustering rows and $\bar{\bar{\mathbf{V}}}_k$ for clustering columns.

Choosing which SVD section SVD Signs Algorithm info goes here

## 2.2 Compare Row Sign Patters

To simplify the comparison of sign patterns between the rows of $\bar{\bar{\mathbf{U}}}_k$, we will create a new matrix $\bar{\bar{\mathbf{U}}}_k^{(\text{sign})} \in \mathbb{R}^{m \times k}$. This new matrix will be based on $\bar{\bar{\mathbf{U}}}_k$, with the values for the elements of $\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}$ being determined by the following definition

$$\left(\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}\right)_{ij} = \begin{cases} 1 & \underline{\text{if}} \quad \left(\bar{\bar{\mathbf{U}}}_k\right)_{ij} > 0 \\ 0 & \underline{\text{if}} \quad \left(\bar{\bar{\mathbf{U}}}_k\right)_{ij} = 0 \\ -1 & \underline{\text{if}} \quad \left(\bar{\bar{\mathbf{U}}}_k\right)_{ij} < 0 \end{cases} \tag{2.2.1}$$

Here, the $\left(\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}\right)_{ij}$ are the elements of $\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}$, the $\left(\bar{\bar{\mathbf{U}}}_k\right)_{ij}$ are the elements of $\bar{\bar{\mathbf{U}}}_k$, and, for both $\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}$ and $\left(\bar{\bar{\mathbf{U}}}_k\right)_{ij}$, the indices $i$ and $j$ are such that $i \in [1, m]$ and $j \in [1, k]$.

Next, we will create a new $m$ by $m$ matrix to represent the connections between the rows of $\bar{\bar{\mathbf{U}}}_k$. This new matrix will be denoted by $\bar{\bar{\mathbf{U}}}_k^{(\text{conn})}$, with $\bar{\bar{\mathbf{U}}}_k^{(\text{conn})} \in \mathbb{R}^{m \times m}$ and having elements $\left(\bar{\bar{\mathbf{U}}}_k^{(\text{conn})}\right)_{ij}$, for $i, j \in [1, m]$. We will define each $\left(\bar{\bar{\mathbf{U}}}_k^{(\text{conn})}\right)_{ij}$ to be 1 if the $i$th row of $\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}$ is equivalent to the $j$th row of $\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}$. That is to say, formally, that

$$\left(\bar{\bar{\mathbf{U}}}_k^{(\text{conn})}\right)_{ij} = \begin{cases} 1 & \underline{\text{if}} \quad \forall l \in [1,k]\,, \left(\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}\right)_{il} = \left(\bar{\bar{\mathbf{U}}}_k^{(\text{sign})}\right)_{jl} \quad \underline{\text{holds}} \\ 0 & \underline{\text{otherwise}} \end{cases} \tag{2.2.2}$$

Additionally, we note that the definition in (1.7) implies that the relation

$$\left(\bar{\bar{\mathbf{U}}}_k^{(\text{conn})}\right)_{ii} = 1,\ \forall i \in [1,m]$$

must be valid as well.

The final step in comparing the sign patterns of the rows in $\bar{\bar{\mathbf{U}}}_k$ is to collect the connections between rows. These connections between rows, indicated by their sign patterns, will be collected into the set of sets $\mathbb{U}_k^{(\text{rel})}$ with $m$ elements such that $\mathbb{U}_k^{(\text{rel})}$ may be defined

$$\mathbb{U}_k^{(\text{rel})} = \left\{ \left(\mathbb{U}_k^{(\text{rel})}\right)_1, \left(\mathbb{U}_k^{(\text{rel})}\right)_2, \ldots, \left(\mathbb{U}_k^{(\text{rel})}\right)_m \right\}$$

where the elements of $\mathbb{U}_k^{(\text{rel})}$ are all sets in their own right and are denoted by the $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ for $i \in [1,m]$. Each of these $k$ elements in $\mathbb{U}_k^{(\text{rel})}$ is defined such that for $\forall i \in [1,m]$ the $i$th element, $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$, is the set

$$\left(\mathbb{U}_k^{(\text{rel})}\right)_i \equiv \left\{ j \,\middle|\, j \in [1,m] \in \left(\bar{\bar{\mathbf{U}}}_k^{(\text{conn})}\right)_{ij} \neq 0 \right\},\ \forall i \in [1,k] \tag{2.2.3}$$

Additionally, note that this definition must imply that $\{i\} \in \left(\mathbb{U}_k^{(\text{rel})}\right)_i,\ \forall i \in [1,m]$ is valid as well. The elements of each of the sets $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ as defined above should only be considered as the initial elements of each $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$. This is due to the fact that additional elements may be added to each $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$ as described in the next section.

## 2.3 Complete Row Cluster Sets

The elements of $\mathbb{U}_k^{(\mathrm{rel})}$ should contain the indices of all rows in the same cluster as each element. That is to say, each $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$ should be the set containing the indices of all the rows in the same cluster as row $i$, in addition to the index of row $i$; however, this is not guaranteed to be the case initially after the creation of the $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$ as described above. To illustrate this, consider the values for some $\bar{\bar{\mathbf{U}}}_k^{(\mathrm{conn})}$ as given below

$$\bar{\bar{\mathbf{U}}}_k^{(\mathrm{conn})} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{2.3.1}$$

Using the definition for the $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$ from (1.8), we obtain the following initial values for the $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$ generated from the $\bar{\bar{\mathbf{U}}}_k^{(\mathrm{conn})}$ expressed above

| $i$ | $\mathbb{U}_k^{(\mathrm{rel})}$ |
|---|---|
| 1 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_1 = \{1, 2\}$ |
| 2 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_2 = \{2\}$ |
| 3 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_3 = \{2, 3\}$ |
| 4 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_4 = \{4\}$ |
| 5 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_5 = \{5\}$ |
| 6 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_6 = \{5, 6\}$ |

***Table ??:*** *Example of initial values of* $\mathbb{U}_k^{(rel)}$

Clearly, rows 1, 2, and 3 belong in the same cluster; rows 5 and 6 also belong to the same cluster, but one different than the first; and row 4 belongs by itself. However, with the exception of rows 4 and 6, the $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ do not reflect this, at least not initially. Thus, each of the $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ must be completed by adding the other indices of rows which are in the same cluster as the row with which $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ is associated. This will be accomplished using the intersect, $\left(\mathbb{U}_k^{(\text{rel})}\right)_i \bigcap \left(\mathbb{U}_k^{(\text{rel})}\right)_j$, and union, $\left(\mathbb{U}_k^{(\text{rel})}\right)_i \bigcup \left(\mathbb{U}_k^{(\text{rel})}\right)_j$, set operations.

We will use the intersect set operation will indicate if the rows represented by $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ and $\left(\mathbb{U}_k^{(\text{rel})}\right)_j$ belong to the same cluster, as the operation will give

$$\left(\mathbb{U}_k^{(\text{rel})}\right)_i \bigcap \left(\mathbb{U}_k^{(\text{rel})}\right)_j = \emptyset$$

if rows $i$ and $j$ belong to different clusters. If this intersection yields a non-empty set, then we will use the second set operation, the union

$$\left(\mathbb{U}_k^{(\text{rel})}\right)_i \bigcup \left(\mathbb{U}_k^{(\text{rel})}\right)_j$$

to combine the elements of $\left(\mathbb{U}_k^{(\text{rel})}\right)_i$ and $\left(\mathbb{U}_k^{(\text{rel})}\right)_j$. This discussion allows us to layout the following process for completing all of the $\left(\mathbb{U}_k^{(\text{rel})}\right)_i \in \mathbb{U}_k^{(\text{rel})}$.

Applying this algorithm to the example in ***Table ??***, we have

**Data:** Initial values for the $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i \in \mathbb{U}_k^{(\mathrm{rel})}$

**Result:** Completed values for the $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i \in \mathbb{U}_k^{(\mathrm{rel})}$, fully representing the cluster of each row by containing all elements of that cluster.

```
begin
    for  i = 1 : m do
        for  j = 1 : m do
            if  (U_k^(rel))_i ∩ (U_k^(rel))_j ≠ ∅ then
                (U_k^(rel))_i = (U_k^(rel))_i ∪ (U_k^(rel))_j ;
                (U_k^(rel))_j = (U_k^(rel))_j ∪ (U_k^(rel))_i ;
            end
        end
    end
end
```

**Algorithm 1:** Completing the sets $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$, $\forall i \in [i, m]$

| $i$ | $\mathbb{U}_k^{(\mathrm{rel})}$ | $i=1$ $j=2$ | $i=1$ $j=3$ | $i=2$ $j=2$ | $i=5$ $j=6$ |
|---|---|---|---|---|---|
| 1 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_1 = \{1,2\}$ | | $=\{1,2,3\}$ | | |
| 2 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_2 = \{2\}$ | $=\{1,2\}$ | | $=\{1,2,3\}$ | |
| 3 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_3 = \{2,3\}$ | | $=\{1,2,3\}$ | | |
| 4 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_4 = \{4\}$ | | | | |
| 5 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_5 = \{5\}$ | | | | $=\{5,6\}$ |
| 6 | $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_6 = \{5,6\}$ | | | | |

***Table ??:*** *Example of algorithm on the elements of* $\mathbb{U}_k^{(rel)}$*,*

*with steps and elements without change omitted.*

## 2.4 Single Sets for each Row Cluster

Now that each set in $\mathbb{U}_k^{(\mathrm{rel})}$ contains all of the elements in the same cluster as the row represented by that set, the next step is to uniquely identify each cluster by numbering them. Additionally, we wish to associate the index number for each cluster with the set listing all of the elements of the cluster in question. To do this, we create a new set of sets,

$\mathbb{U}_k^{(\text{clust})}$, defined as

$$\mathbb{U}_k^{(\text{clust})} = \left\{ \left(\mathbb{U}_k^{(\text{clust})}\right)_1, \left(\mathbb{U}_k^{(\text{clust})}\right)_2, \ldots, \left(\mathbb{U}_k^{(\text{clust})}\right)_c \right\}$$

whose elements are sets of 2-tuples and with its cardinality, $\left|\mathbb{U}_k^{(\text{clust})}\right| = c$, being the number of unique clusters. These sets of 2-tuples that comprise $\mathbb{U}_k^{(\text{clust})}$ are denoted $\left(\mathbb{U}_k^{(\text{clust})}\right)_i$ and are defined

$$\left(\mathbb{U}_k^{(\text{clust})}\right)_i = \left\{ i, \left(\mathbb{U}_k^{(\text{rel})}\right)_j \right\} \tag{2.4.1}$$

The first element of these 2-tuples, $i$, is the index of the cluster in question and the second element of these 2-tuples, $\left(\mathbb{U}_k^{(\text{rel})}\right)_j$, is the set of all elements in that cluster, with the index $j$ being the lowest index of $\mathbb{U}_k^{(\text{rel})}$ where that set occurs. To build $\mathbb{U}_k^{(\text{clust})}$, we use the following process

**Data:** Initial values for the $\left(\mathbb{U}_k^{(\text{rel})}\right)_i \in \mathbb{U}_k^{(\text{rel})}$
**Result:** Completed $\mathbb{U}_k^{(\text{clust})}$.
**begin**
  $\left(\mathbb{U}_k^{(\text{clust})}\right)_1 \longleftarrow \left\{1, \left(\mathbb{U}_k^{(\text{rel})}\right)_1\right\}$;
  $c_0 \longleftarrow 1$;
  **for** $i = 2 : m$ **do**
    **for** $j = 1 : c_0$ **do**
      **if** $\left(\mathbb{U}_k^{(rel)}\right)_i \bigcap \left(\mathbb{U}_k^{(clust)}\right)_j = \emptyset$ **then**
        $c_0 \longleftarrow (c_0 + 1)$;
        $\left(\mathbb{U}_k^{(\text{clust})}\right)_{c_0} \longleftarrow \left\{c_0, \left(\mathbb{U}_k^{(\text{rel})}\right)_i\right\}$;
      **end**
    **end**
  **end**
**end**

**Algorithm 2:** Creating the sets $\left(\mathbb{U}_k^{(\text{clust})}\right)_i \in \mathbb{U}_k^{(\text{clust})}$, $\forall i \in [i, c]$.

Later, we will use the elements of $\mathbb{U}_k^{(\text{clust})}$ generated by this process to reorder the rows of

$\bar{\bar{\mathbf{A}}}$. However, we will next repeat the proceeding three steps for each of the $n$ columns in $\bar{\bar{\mathbf{A}}}$.

## 2.5 Compare Column Sign Patters

To simplify the comparison of sign patterns between the columns of $\bar{\bar{\mathbf{V}}}_k^{\intercal} = \bar{\bar{\mathcal{V}}}_k$, we will create a new matrix $\bar{\bar{\mathcal{V}}}_k^{(\text{sign})} \in \mathbb{R}^{k \times n}$. This new matrix will be based on $\bar{\bar{\mathcal{V}}}_k$, with the values for the elements of $\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}$ being determined by the following definition

$$\left(\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}\right)_{ij} = \begin{cases} 1 & \underline{\text{if}} \quad \left(\bar{\bar{\mathbf{V}}}_k^{\intercal}\right)_{ij} > 0 \\ 0 & \underline{\text{if}} \quad \left(\bar{\bar{\mathbf{V}}}_k^{\intercal}\right)_{ij} = 0 \\ -1 & \underline{\text{if}} \quad \left(\bar{\bar{\mathbf{V}}}_k^{\intercal}\right)_{ij} < 0 \end{cases} \tag{2.5.1}$$

Here, the $\left(\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}\right)_{ij}$ are the elements of $\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}$, the $\left(\bar{\bar{\mathbf{V}}}_k^{\intercal}\right)_{ij}$ are the elements of $\bar{\bar{\mathbf{V}}}_k^{\intercal}$, and, for both $\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}$ and $\left(\bar{\bar{\mathbf{V}}}_k^{\intercal}\right)_{ij}$, the indices $i$ and $j$ are such that $i \in [1, k]$ and $j \in [1, n]$.

Next, we will create a new $n$ by $n$ matrix to represent the connections between the rows of $\bar{\bar{\mathbf{V}}}_k^{\intercal}$. This new matrix will be denoted by $\bar{\bar{\mathcal{V}}}_k^{(\text{conn})}$, with $\bar{\bar{\mathcal{V}}}_k^{(\text{conn})} \in \mathbb{R}^{n \times n}$ and having elements $\left(\bar{\bar{\mathcal{V}}}_k^{(\text{conn})}\right)_{ij}$, for $i, j \in [1, n]$. We will define each $\left(\bar{\bar{\mathcal{V}}}_k^{(\text{conn})}\right)_{ij}$ to be 1 if the $i$th row of $\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}$ is equivalent to the $j$th row of $\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}$. That is to say, formally, that

$$\left(\bar{\bar{\mathcal{V}}}_k^{(\text{conn})}\right)_{ij} = \begin{cases} 1 & \underline{\text{if}} \quad \forall l \in [1, k]\,, \left(\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}\right)_{li} = \left(\bar{\bar{\mathcal{V}}}_k^{(\text{sign})}\right)_{lj} \quad \underline{\text{holds}} \\ 0 & \underline{\text{otherwise}} \end{cases} \tag{2.5.2}$$

Additionally, we note that the definition in (1.7) implies that the relation

$$\left(\bar{\bar{\mathcal{V}}}_k^{(\text{conn})}\right)_{ii} = 1, \ \forall i \in [1, n]$$

must be valid as well.

The final step in comparing the sign patterns of the rows in $\bar{\bar{\mathbf{V}}}_k^{\intercal}$ is to collect the connections between rows. These connections between rows, indicated by their sign patterns, will be collected into the set of sets $\mathbb{V}_k^{(\mathrm{rel})}$ with $m$ elements such that $\mathbb{V}_k^{(\mathrm{rel})}$ may be defined

$$\mathbb{V}_k^{(\mathrm{rel})} = \left\{ \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_1, \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_2, \ldots, \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_n \right\}$$

where the elements of $\mathbb{V}_k^{(\mathrm{rel})}$ are all sets in their own right and are denoted by the $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ for $i \in [1, n]$. Each of these $k$ elements in $\mathbb{V}_k^{(\mathrm{rel})}$ is defined such that for $\forall i \in [1, n]$ the $i$th element, $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$, is the set

$$\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \equiv \left\{ j \;\middle|\; j \in [1, m] \in \left(\bar{\bar{\mathcal{V}}}_k^{(\mathrm{conn})}\right)_{ij} \neq 0 \right\}, \; \forall i \in [1, k] \tag{2.5.3}$$

Additionally, note that this definition must imply that $\{i\} \in \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$, $\forall i \in [1, n]$ is valid as well. The elements of each of the sets $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ as defined above should only be considered as the initial elements of each $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$. This is due to the fact that additional elements may be added to each $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ as described in the next section.

## 2.6 Complete Column Cluster Sets

The elements of $\mathbb{V}_k^{(\mathrm{rel})}$ should contain the indices of all columns in the same cluster as each element. That is to say, each $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ should be the set containing the indices of all the columns in the same cluster as column $i$, in addition to the index of column $i$; however, this is not guaranteed to be the case initially after the creation of the $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ as described above. The same illustration we gave for the $\left(\mathbb{U}_k^{(\mathrm{rel})}\right)_i$ **Section 1.2.2** illustrates this point as well. Similar to the case with the rows, each of the $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ must be completed by adding the other indices of columns which are in the same cluster as the column with which $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ is associated. This will be accomplished using the intersection, $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \bigcap \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j$, and union, $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \bigcup \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j$, set operations.

We will use the intersect set operation will indicate if the columns represented by $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ and $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j$ belong to the same cluster, as the operation will give

$$\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \bigcap \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j = \emptyset$$

if columns $i$ and $j$ belong to different clusters. If this intersection yields a non-empty set, then we will use the second set operation, the union

$$\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \bigcup \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j$$

to combine the elements of $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ and $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j$. This discussion allows us to layout the following process for completing all of the $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \in \mathbb{V}_k^{(\mathrm{rel})}$.

**Data:** Initial values for the $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \in \mathbb{V}_k^{(\mathrm{rel})}$

**Result:** Completed values for the $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \in \mathbb{V}_k^{(\mathrm{rel})}$, fully representing the cluster of each column by containing all elements of that cluster.

**begin**
  **for** $i = 1:n$ **do**
    **for** $j = 1:n$ **do**
      **if** $\left(\mathbb{V}_k^{(rel)}\right)_i \bigcap \left(\mathbb{V}_k^{(rel)}\right)_j \neq \emptyset$ **then**
        $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i = \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i \bigcup \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j$ ;
        $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j = \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_j \bigcup \left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$ ;
      **end**
    **end**
  **end**
**end**

**Algorithm 3:** Completing the sets $\left(\mathbb{V}_k^{(\mathrm{rel})}\right)_i$, $\forall i \in [i, n]$

## 2.7 Single Sets for each Column Cluster

Now that each set in $\mathbb{V}_k^{(\text{rel})}$ contains all of the elements in the same cluster as the column represented by that set, the next step is to uniquely identify each cluster by numbering them. Additionally, we wish to associate the index number for each cluster with the set listing all of the elements of the cluster in question. To do this, we create a new set of sets, $\mathbb{V}_k^{(\text{clust})}$, defined as

$$\mathbb{V}_k^{(\text{clust})} = \left\{ \left(\mathbb{V}_k^{(\text{clust})}\right)_1, \left(\mathbb{V}_k^{(\text{clust})}\right)_2, \ldots, \left(\mathbb{V}_k^{(\text{clust})}\right)_c \right\}$$

whose elements are sets of 2-tuples and with its cardinality, $\left|\mathbb{V}_k^{(\text{clust})}\right| = c$, being the number of unique clusters. These sets of 2-tuples that comprise $\mathbb{V}_k^{(\text{clust})}$ are denoted $\left(\mathbb{V}_k^{(\text{clust})}\right)_i$ and are defined

$$\left(\mathbb{V}_k^{(\text{clust})}\right)_i = \left\{ i, \left(\mathbb{V}_k^{(\text{rel})}\right)_j \right\} \tag{2.7.1}$$

The first element of these 2-tuples, $i$, is the index of the cluster in question and the second element of these 2-tuples, $\left(\mathbb{V}_k^{(\text{rel})}\right)_j$, is the set of all elements in that cluster, with the index $j$ being the lowest index of $\mathbb{V}_k^{(\text{rel})}$ where that set occurs. To build $\mathbb{V}_k^{(\text{clust})}$, we use the following process

**Data:** Initial values for the $\left(\mathbb{V}_k^{(\text{rel})}\right)_i \in \mathbb{V}_k^{(\text{rel})}$
**Result:** Completed $\mathbb{V}_k^{(\text{clust})}$.
**begin**
$\left(\mathbb{V}_k^{(\text{clust})}\right)_1 \longleftarrow \left\{1, \left(\mathbb{V}_k^{(\text{rel})}\right)_1\right\}$;
$c_0 \longleftarrow 1$;
**for** $i = 2 : n$ **do**
**for** $j = 1 : c_0$ **do**
**if** $\left(\mathbb{V}_k^{\textit{(rel)}}\right)_i \bigcap \left(\mathbb{V}_k^{\textit{(clust)}}\right)_j = \emptyset$ **then**
$c_0 \longleftarrow (c_0 + 1)$;
$\left(\mathbb{V}_k^{(\text{clust})}\right)_{c_0} \longleftarrow \left\{c_0, \left(\mathbb{V}_k^{(\text{rel})}\right)_i\right\}$;
**end**
**end**
**end**
**end**

**Algorithm 4:** Creating the sets $\left(\mathbb{V}_k^{(\text{clust})}\right)_i \in \mathbb{V}_k^{(\text{clust})}$, $\forall i \in [i, c]$.

## 2.8 Reordering $\bar{\bar{\mathbf{L}}}$ into a new Matrix

The last step is to finally reorder $\bar{\bar{\mathbf{L}}}$ into the clustered matrix $\bar{\bar{\mathbf{L}}}^{\star} \in \mathbb{R}^{m \times n}$, which will be our result. To do this, we first use the information about the row clusters contained in $\mathbb{U}^{\,(\text{clust})}$ or $\mathbb{U}_k^{(\text{clust})}$ to reorder the rows of $\bar{\bar{\mathbf{L}}}$ into $\bar{\bar{\mathbf{L}}}^{\star}$. Similarly, we use the information about the column clusters contained in $\mathbb{V}^{\,(\text{clust})}$ or $\mathbb{V}_k^{(\text{clust})}$ to reorder the columns of $\bar{\bar{\mathbf{L}}}$ into $\bar{\bar{\mathbf{L}}}^{\star}$.

Additionally, we must use the information in $\mathbb{U}^{\,(\text{clust})}$ or $\mathbb{U}_k^{(\text{clust})}$ to ensure that the *term* label associated with each row (*representing Terms*) in $\bar{\bar{\mathbf{L}}}$ is moved so that its position corresponds to the new position of its associated row in $\bar{\bar{\mathbf{L}}}^{\star}$. In similar fashion, the information in $\mathbb{V}^{\,(\text{clust})}$ or $\mathbb{V}_k^{(\text{clust})}$ is used to ensure that the *object* label associated with each column (*representing Objects*) in $\bar{\bar{\mathbf{L}}}$ is moved so that its position corresponds to the new position of its associated column in $\bar{\bar{\mathbf{L}}}^{\star}$.

### 2.8.1 Reorder Rows

Process for reordering rows goes here.

### 2.8.2 Reorder Columns

Process for reordering columns goes here.

### 2.8.3 Reorder *Term*/Row Labels

Process for reordering *Term*/Row Labels goes here.

### 2.8.4 Reorder *Object*/Column Labels

Process for reordering *Object*/Column Labels goes here.

# Chapter 3

# SVD Gaps

Intro SVD Gaps Algorithm info goes here

# Chapter 4

# CbC Disjoint Sets

When $\bar{\bar{\mathbf{L}}}$ consists of a series of disjoint sets, our clustering algorithm proceeds according to the following simple steps

1. Compute $\bar{\bar{\mathbf{A}}}_U$
2. Construct $\mathbb{U} = \Big\{ \{\mathbb{U}_1, \mathbb{M}_1\}, \{\mathbb{U}_2, \mathbb{M}_2\}, \ldots, \{\mathbb{U}_m, \mathbb{M}_m\} \Big\}$
3. Complete the $\big\{\mathbb{U}_i, \mathbb{M}_i\big\} \in \mathbb{U}$
4. Construct the $\mathbb{U}^\star$ set of sets
5. Compute $\bar{\bar{\mathbf{A}}}_V$
6. Construct $\mathbb{V} = \Big\{ \{\mathbb{V}_1, \mathbb{N}_1\}, \{\mathbb{V}_2, \mathbb{N}_2\}, \ldots, \{\mathbb{V}_n, \mathbb{N}_n\} \Big\}$
7. Complete the $\big\{\mathbb{V}_j, \mathbb{N}_j\big\} \in \mathbb{V}$
8. Construct the $\mathbb{V}^\star$ set of sets
9. Reorder $\bar{\bar{\mathbf{L}}}$ in the a new, temporary matrix, $\bar{\bar{\mathbf{L}}}_0 \in \mathbb{R}^{m \times n}$.
10. Reorder the temporary matrix, $\bar{\bar{\mathbf{L}}}_0$ into the final matrix $\bar{\bar{\mathbf{L}}}^\star \in \mathbb{R}^{m \times n}$.

The description of our algorithm for the case when $\bar{\bar{\mathbf{L}}}$ is composed of non-disjoint sets is given later in chapter 3.

## 4.1 Compute $\bar{\bar{\mathbf{A}}}_U$

We use the expression

$$\bar{\bar{\mathbf{A}}}_U = \bar{\bar{\mathbf{L}}}\,\bar{\bar{\mathbf{L}}}^{\intercal} \tag{1.4.1}$$

to compute $\bar{\bar{\mathbf{A}}}_U \in \mathbb{R}^{m\times m}$.

If a different weighting of the relations between the elements of $\bar{\bar{\mathbf{L}}}$ is desired, we can use the alternative expression

$$\left(\bar{\bar{\mathbf{A}}}_U\right)_{ij} = \sum_{k=1}^{n} \left\{ \delta^{\star}\left(\bar{\bar{\mathbf{L}}}, i, k\right) \bar{\bar{\mathbf{L}}}_{jk} \right\} \tag{1.4.1a}$$

where $i, j \in [1, m] \subset \mathbb{Z}^{+}$.

## 4.2 Construct $\mathbb{U}$

We initially construct $\mathbb{U}$ by following the sub-routine (sub algorithm) given below

**Data:** Connection Matrix for rows, $\bar{\bar{\mathbf{A}}}_U$, and the number of rows in $\bar{\bar{\mathbf{L}}}$, $m$.
**Result:** Initial value for each of the $\left\{\mathbb{U}_i, \mathbb{M}_i\right\} \in \mathbb{U}$.
**begin**
    **for** $i = 1 : m$ **do**
        $\mathbb{U}_i = \emptyset$; $\mathbb{M}_i = \emptyset$ ;
        **for** $j = 1 : m$ **do**
            **if** $\left(\bar{\bar{\mathbf{A}}}_U\right)_{ij} \neq 0$ **then**
                $\mathbb{U}_i = \mathbb{U}_i \bigcup \{j\}$ ;
                $\mathbb{M}_i = \mathbb{M}_i \bigcup \left\{ \left(\bar{\bar{\mathbf{A}}}_U\right)_{ij} \right\}$ ;
            **end**
        **end**
        $\mathbb{U}[i, 1] = \mathbb{U}_i$; $\mathbb{U}[i, 2] = \mathbb{M}_i$;
    **end**
**end**

**Algorithm 5:** Computing the initial value for each of the $\left\{\mathbb{U}_i, \mathbb{M}_i\right\} \in \mathbb{U}$.

After the completion of this sub-routine, we will make a copy of this initial state of $\mathbb{U}$. We denote this copy of this initial state as $\mathbb{U}_0$.

## 4.3 Complete the $\mathbb{U}$ Sets

The elements of $\mathbb{U}$ must now be "*completed*" so that they include any indirectly related elements. Taking advantage of the disjointedness of $\bar{\bar{\mathbf{L}}}$, the "*completion*" of the elements in $\mathbb{U}$ can be accomplished using the simple subroutine below

**Data:** Initial values for the elements of $\mathbb{U}$, $\left\{\mathbb{U}_i, \mathbb{M}_i\right\}$, and the number of rows in $\bar{\bar{\mathbf{L}}}$, $m$.
**Result:** The final values for each of the $\left\{\mathbb{U}_i, \mathbb{M}_i\right\} \in \mathbb{U}$.

```
begin
    boolean isChanged = true ;
    while isChanged do
        isChanged = false ;
        for i = 1 : m do
            U_i = U[i,1]; M_i = U[i,2] ;
            for j = 1 : m do
                U_j = U[j,1]; M_j = U[j,2] ;
                if U_i ≠ U_j && U_i ⋂ U_j ≠ ∅ then
                    U_i = U_i ⋃ {U_j U_i} ;
                    M_i = M_i ⋃ {M_j M_i} ;
                    U_j = U_j ⋃ {U_i U_j} ;
                    M_j = M_j ⋃ {M_i M_j} ;
                    isChanged = true ;
                end
                if isChanged then
                    U[j,1] = U_j; U[j,2] = M_j ;
                end
            end
            if isChanged then
                U[i,1] = U_i; U[i,2] = M_i ;
            end
        end
    end
end
```

**Algorithm 6:** Computing the final value for each $\left\{\mathbb{U}_i, \mathbb{M}_i\right\} \in \mathbb{U}$.

## 4.4 Construct the $\mathbb{U}^\star$ Sets

To construct the sets of $\mathbb{U}^\star$, we look for unique $\mathbb{U}_i \in \mathbb{U}$ and then store each unique $\mathbb{U}_i$ as its own set in $\mathbb{U}^\star$. We accomplish this by using the following subroutine

**Data:** The final values for the elements of $\mathbb{U}$, $\{\mathbb{U}_i, \mathbb{M}_i\}$, and the number of rows in $\bar{\bar{\mathbf{L}}}$, $m$.
**Result:** The sets of $\mathbb{U}^\star$, with each set in $\mathbb{U}^\star$ representing a row cluster and containing its elements.

```
begin
    int nClust = 1; U* [nClust] = U [1, 1] ;
    for  i = 2 : m do
        boolean isNew = true; U_i = U [i, 1] ;
        for  j = 1 : nClust do
            U*_j = U* [j] ;
            if  U*_j = U_i then
                isNew = false ;
                break ;
            end
        end
        if  isNew then
            nClust ++ ;
            U* [nClust] = U_i ;
        end
    end
end
```

**Algorithm 7:** Compute the set of all row clusters sets, $\mathbb{U}^\star$.

## 4.5 Compute $\bar{\bar{\mathbf{A}}}_V$

We use the expression

$$\bar{\bar{\mathbf{A}}}_V = \bar{\bar{\mathbf{L}}}^\intercal \, \bar{\bar{\mathbf{L}}} \tag{1.4.2}$$

to compute $\bar{\bar{\mathbf{A}}}_V \in \mathbb{R}^{n \times n}$.

If a different weighting of the relations between the elements of $\bar{\bar{\mathbf{L}}}$ is desired, we can use the alternative expression

$$\left(\bar{\bar{\mathbf{A}}}_V\right)_{ij} = \sum_{k=1}^{m} \left\{ \delta^\star \left(\bar{\bar{\mathbf{L}}}, k, i\right) \bar{\bar{\mathbf{L}}}_{kj} \right\} \tag{1.4.2a}$$

where $i, j \in [1, n] \subset \mathbb{Z}^+$.

## 4.6 Construct $\mathbb{V}$

We initially construct $\mathbb{V}$ by following the sub-routine (sub algorithm) given below

**Data:** Connection Matrix for columns, $\bar{\bar{\mathbf{A}}}_V$, and the number of columns in $\bar{\bar{\mathbf{L}}}$, $n$.
**Result:** Initial value for each of the $\{\mathbb{V}_j, \mathbb{N}_j\} \in \mathbb{V}$.

```
begin
    for j = 1 : n do
        V_j = ∅; N_j = ∅ ;
        for i = 1 : n do
            if (A_V)_ji ≠ 0 then
                V_j = V_j ⋃ {i} ;
                N_j = N_j ⋃ { (A_V)_ji } ;
            end
        end
        V[j,1] = V_j; V[j,2] = N_j;
    end
end
```

**Algorithm 8:** Computing the initial value for each of the $\{\mathbb{V}_j, \mathbb{N}_j\} \in \mathbb{V}$.

After the completion of this sub-routine, we will make a copy of this initial state of $\mathbb{V}$. We denote this copy of this initial state as $\mathbb{V}_0$.

## 4.7 Complete the $\mathbb{V}$ Sets

The elements of $\mathbb{V}$ must now be "*completed*" so that they include any indirectly related elements. Taking advantage of the disjointedness of $\bar{\bar{\mathbf{L}}}$, the "*completion*" of the elements in $\mathbb{V}$ can be accomplished using the simple subroutine below

**Data:** Initial values for the elements of $\mathbb{V}$, $\left\{\mathbb{V}_j, \mathbb{N}_j\right\}$, and the number of columns in $\bar{\bar{\mathbf{L}}}$, $n$.
**Result:** The final values for each of the $\left\{\mathbb{V}_j, \mathbb{N}_j\right\} \in \mathbb{V}$.
**begin**
  **boolean** isChanged = **true** ;
  **while** *isChanged* **do**
    isChanged = **false** ;
    **for** $j = 1 : n$ **do**
      $\mathbb{V}_j = \mathbb{V}[j,1]$; $\mathbb{N}_j = \mathbb{V}[j,2]$ ;
      **for** $i = 1 : n$ **do**
        $\mathbb{V}_i = \mathbb{V}[i,1]$; $\mathbb{N}_i = \mathbb{V}[i,2]$ ;
        **if** $\mathbb{V}_j \neq \mathbb{V}_i$ *&&* $\mathbb{V}_j \bigcap \mathbb{V}_i \neq \emptyset$ **then**
          $\mathbb{V}_j = \mathbb{V}_j \bigcup \{\mathbb{V}_i\ \mathbb{V}_j\}$ ;
          $\mathbb{N}_j = \mathbb{N}_j \bigcup \{\mathbb{N}_i\ \mathbb{N}_j\}$ ;
          $\mathbb{V}_i = \mathbb{V}_i \bigcup \{\mathbb{V}_j\ \mathbb{V}_i\}$ ;
          $\mathbb{N}_i = \mathbb{N}_i \bigcup \{\mathbb{N}_j\ \mathbb{N}_i\}$ ;
          isChanged = **true** ;
        **end**
        **if** *isChanged* **then**
          $\mathbb{V}[i,1] = \mathbb{V}_i$; $\mathbb{V}[i,2] = \mathbb{N}_i$ ;
        **end**
      **end**
      **if** *isChanged* **then**
        $\mathbb{V}[j,1] = \mathbb{V}_j$; $\mathbb{V}[j,2] = \mathbb{N}_j$ ;
      **end**
    **end**
  **end**
**end**

**Algorithm 9:** Computing the final value for each $\left\{\mathbb{V}_j, \mathbb{N}_j\right\} \in \mathbb{V}$.

## 4.8 Construct the $\mathbb{V}^{\star}$ Sets

To construct the sets of $\mathbb{V}^{\star}$, we look for unique $\mathbb{V}_i \in \mathbb{V}$ and then store each unique $\mathbb{V}_i$ as its own set in $\mathbb{V}^{\star}$. We accomplish this by using the following subroutine

**Data:** The final values for the elements of $\mathbb{V}$, $\left\{\mathbb{V}_i, \mathbb{M}_i\right\}$, and the number of columns in $\bar{\bar{\mathbf{L}}}$, $n$.
**Result:** The sets of $\mathbb{V}^{\star}$, with each set in $\mathbb{V}^{\star}$ representing a column cluster and containing its elements.

```
begin
    int nClust = 1; V*[nClust] = V[1,1] ;
    for i = 2 : n do
        boolean isNew = true; V_i = V[i,1] ;
        for j = 1 : nClust do
            V*_j = V*[j] ;
            if V*_j = V_i then
                isNew = false ;
                break ;
            end
        end
        if isNew then
            nClust ++ ;
            V*[nClust] = V_i ;
        end
    end
end
```

**Algorithm 10:** Compute the set of all column clusters sets, $\mathbb{V}^{\star}$.

# Chapter 5

# CbC Non-Disjoint Sets

Connection by Clustering on Non-Disjoint Algorithm info goes here