Clustering Via Connections

by

Jonathan McFadden

6/13/2016 —

Table of Contents

1 Definitions

1.1 - Input Matrix

Let $\overline{\overline{L}} \in \mathbb{R}^{m \times n}$ be a matrix that represents how m terms describe a collection of n objects. We express this formally as either

$$L_{ij} \equiv \begin{cases} 1 & \text{if the } i^{th} \text{ term describes the } j^{th} \text{ object.} \\ 0 & \text{otherwise} \end{cases}$$

or

$$L_{ij} \equiv \begin{cases} n & \text{if the } i^{th} \text{ terms describes the } j^{th} \text{ object } n \text{ times} \\ 0 & \text{otherwise} \end{cases}$$

In both definitions for the L_{ij} we require that $i \in [1, m] \subset \mathbb{Z}^+$ and that $j \in [1, n] \subset \mathbb{Z}^+$.

1.2 - SVS Matrices

We will be using both the full SVS of $\bar{\bar{L}}$ and the k order reduced SVS of $\bar{\bar{L}}$. The order of the reduced SVS, k, must satisfy

$$k < r$$

where $r = \text{RANK}[\bar{\bar{L}}]$.

1.2.1 - Full SVS

Let the full singular value decomposition of $\bar{\bar{L}}$ be given by

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

where $\bar{\bar{U}} \in \mathbb{R}^{m \times r}$, $\bar{\bar{S}} \in \mathbb{R}^{r \times r}$, and $\bar{\bar{V}} \in \mathbb{R}^{n \times r}$, with r defined as above. We call $\bar{\bar{U}}$ the 'row matrix' of $\bar{\bar{L}}$, $\bar{\bar{S}}$ the 'singular value matrix' of $\bar{\bar{L}}$, and $\bar{\bar{V}}$ the 'column matrix' of $\bar{\bar{L}}$.

1.2.2 - Reduced SVS

Let the K order reduced singular

value decomposition of $\bar{\bar{L}}$ be given by

$$\bar{\bar{L}} \cong \bar{\bar{U}}_x \bar{\bar{S}}_x \bar{\bar{V}}_x^T$$

where $\bar{\bar{U}}_x \in \mathbb{R}^{m \times x}$, $\bar{\bar{S}}_x \in \mathbb{R}^{x \times x}$, and $\bar{\bar{V}}_x \in \mathbb{R}^{n \times x}$ with x defined as above. We call $\bar{\bar{U}}_x$ the 'reduced row matrix' of $\bar{\bar{L}}$, $\bar{\bar{S}}_x$ the 'reduced singular value matrix' of $\bar{\bar{L}}$, and $\bar{\bar{V}}_x$ the 'reduced column matrix' of $\bar{\bar{L}}$.

1.3 - Connection Matrices

From $\bar{\bar{L}}$, we will create two

'Connection Matrices'. These matrices will be used to provide information for clustering either the rows of $\bar{\bar{E}}$ or the columns of $\bar{\bar{E}}$.

1.3.1 - Row Connection Matrix

The first of the 'Connection Matrices' is the 'Row Connection Matrix'. This matrix will be used to provide information for clustering the rows of $\bar{\bar{E}}$. We will denote this matrix by $\bar{\bar{A}}_v \in \mathbb{R}^{m \times m}$ and define it, in terms of $\bar{\bar{E}}$, by either

$$\bar{\bar{A}}_v = \bar{\bar{L}}\ \bar{\bar{L}}^T$$

or

$$(\bar{\bar{A}}_v)_{ij} = \sum_{k=1}^{n} \left\{ \delta^*(\bar{\bar{L}}, i, k)\ \bar{L}_{jk} \right\}$$

where $i, j \in [1, m] \subset \mathbb{Z}^+$ and the function $\delta^*(\bar{\bar{L}}, l, \lambda)$ is defined

$$\delta^*(\bar{\bar{L}}, l, \lambda) \equiv \begin{cases} 1 & \text{if } \bar{\bar{L}}_{l\lambda} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

with $l \in [1, m] \in \mathbb{Z}^+$ and $\lambda \in [1, n] \subset \mathbb{Z}^+$ for the definition of $\delta^*(\bar{\bar{L}}, l, \lambda)$.

1.3.2 - Column Connection Matrix

The second 'Connection Matrix' is the 'Column Connection Matrix'. This matrix will provide information when clustering the columns of $\bar{\bar{L}}$ and is denoted $\bar{\bar{A}}_V \in \mathbb{R}^{n \times n}$. We define this matrix, in terms of $\bar{\bar{L}}$, by either

$$\bar{\bar{A}}_V = \bar{\bar{L}}^T \bar{\bar{L}}$$

or

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

where $i, j \in [1, n] \subset \mathbb{Z}^+$ and $\delta^*(\Sigma, \lambda)$ is as defined above.

1.4 - Relation Sets

We will now define two sets of 2-tuples, with one for rows and the other for columns. These sets will describe the relation or relations any row/column element with any other row/column element as the strength of the relation or relations in question.

1.4.1 - Row Relation Set

Let $\mathbb{U}$ be a set of m 2-tuples such that

$$\mathbb{U} = \{\{\mathbb{M}_1, \mathbb{U}_1\}, \{\mathbb{M}_2, \mathbb{U}_2\}, \ldots, \{\mathbb{M}_m, \mathbb{U}_m\}\}$$

where the $\{\mathbb{M}_i, \mathbb{U}_i\}$ are 2-tuples representing all of the rows to which the i^{th} row is related, $\mathbb{U}_i$, and the strength of those relations, $\mathbb{M}_i$; with $i \in [1,m] \subset \mathbb{Z}^+$. Let us now define $\mathbb{U}_i$ and $\mathbb{M}_i$ concurrently as

$$\mathbb{U}_i = \{ j \in [1,m] \subset \mathbb{Z}^+ \mid (\bar{\bar{A}}_U)_{ij} \neq 0 \}$$

and

$$\mathbb{M}_i = \{(\bar{\bar{A}}_\sigma)_{ij} \mid j \in [1,m] \subset \mathbb{Z}^+ \text{ and } (\bar{\bar{A}}_\sigma)_{ij} \neq \Theta\}$$

These definitions require that

$$|\mathbb{M}_i| = |\mathbb{U}_i|$$

for all $i \in [1,m] \subset \mathbb{Z}^+$. Additionally, the strength of the connection to the l^{th} element in $\mathbb{U}_i$ is represented by the l^{th} element in $\mathbb{M}_i$.

1.4.2 - Column Relation Set

Let $\mathbb{V}$ be a set of n 2-tuples such that

$$\mathbb{V} = \{\{\mathbb{N}_1, \mathbb{V}_1\}, \{\mathbb{N}_2, \mathbb{V}_2\}, \ldots, \{\mathbb{N}_n, \mathbb{V}_n\}\}$$

where the $\{\mathbb{N}_j, \mathbb{V}_j\}$ are 2-tuples representing all of the columns to which the j^{th} column is related, $\mathbb{V}_j$, and the strength of those relations, $\mathbb{N}_j$; with $j \in [1, n] \subset \mathbb{Z}^+$. Let us now define $\mathbb{V}_j$ and $\mathbb{N}_j$ concurrently as

$$\mathbb{V}_j \equiv \{ i \in [1, n] \subset \mathbb{Z}^+ \mid (\bar{\bar{A}}_v)_{ji} \neq 0 \}$$

and

$$\mathbb{N}_j = \{ (\bar{\bar{A}}_v)_{ji} \mid i \in [1, n] \subset \mathbb{Z}^+ \text{ and } (\bar{\bar{A}}_v)_{ji} \neq 0 \}$$

These definitions require that

$$|\mathbb{N}_j| = |\mathbb{V}_j|$$

for all $j \in [1, n] \subset \mathbb{Z}^+$. Additionally, the strength of the connection to the l^{th} element in $\mathbb{V}_j$ is represented by the l^{th} element in $\mathbb{N}_j$.

1.5 - Cluster Sets

We now define two sets of sets, with one set representing the row clusters and the other representing the column clusters.

These sets consist of sets that represent the members of each cluster.

1.5.1 - Row Cluster Set

The 'Row Cluster Set' is a set of u_m sets representing each of the u_m clusters. We denote this $\mathbb{U}^*$ and describe it by

$$\mathbb{U}^* = \{ U_1^*, U_2^*, \dots, U_{u_m}^* \}$$

where the U_μ^* are sets containing the members of the μ^{th} cluster, with $\mu \in [1, u_m] \subset \mathbb{Z}^+$. Each U_μ^* is constructed from the $\{M_i, U_i\}$ of

its (the V_v^*) constituent elements, via a process which is defined later.

1.5.2 - Column Cluster Set

The 'Column Cluster Set' is a set of v_n sets representing each of the v_n clusters. We will denote this $\mathbb{V}^*$ and describe it by

$$\mathbb{V}^* = \{ \mathbb{V}_1^*, \mathbb{V}_2^*, \ldots, \mathbb{V}_{v_n}^* \}$$

where the $\mathbb{V}_j^*$ are sets containing the members of the v^{th} cluster, with $v \in [1, v_n] \subset \mathbb{Z}^+$. Each of the $\mathbb{V}_v$ is constructed from

the $\{N_j, V_j\}$ of its (the V_j^*) constituent elements through a process which will be described later.

2 Disjoint Sets

When $\bar{\bar{L}}$ consists of a series of disjoint sets, the algorithm proceeds by following the steps

- 1) Compute $\bar{\bar{A}}_o$
- 2) Construct $\mathbb{U} = \{\{M_1, U_1\}, \ldots, \{M_m, U_m\}\}$
- 3) Complete the U_i
- 4) Construct the $\mathbb{U}^*$ set
- 5) Compute $\bar{\bar{A}}_v$
- 6) Construct $\mathbb{V} = \{\{N_1, V_1\}, \ldots, \{N_n, V_n\}\}$
- 7) Complete the V_j
- 8) Construct the $\mathbb{V}^*$ set
- 9) Reorder $\bar{\bar{L}}$ into a new matrix, $\bar{\bar{L}}^*$

2.1 - Compute $\bar{\bar{A}}_U$

Using the expression

$$\bar{\bar{A}}_U = \bar{\bar{L}}\,\bar{\bar{L}}^T$$

compute $\bar{\bar{A}}_U \in \mathbb{R}^{m \times m}$. Alternatively, the expression

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

may be used if different weighting is desired.

2.2 - Construct $\mathbb{U}$

We initially construct $\mathbb{U}$ by

the following routine

```
FOR (i = 1:m) {
    U_i = ∅;  M_i = ∅;

    FOR (j = 1:m) {
        IF ((Ā_v)_ij ≠ 0) {
            U_i = U_i ⋃ {j};
            M_i = M_i ⋃ {(Ā_v)_ij};
        } END
    } END

    U[i, 1] = M_i;  U[i, 2] = U_i;
} END
```

After completion of this routine, we make a copy of this initial $\mathbb{U}$, denoted $\mathbb{U}_0$.

2.3 - Complete the $\mathbb{U}$ Sets

The elements of $\mathbb{U}$ must now be 'completed' to include in-directly related elements. Taking advantage of the disjointedness of $\bar{E}$, this 'completion' is accomplished using the simple routine below

```
BOOLEAN  ISCHANGED = TRUE;

WHILE (ISCHANGED) {
    ISCHANGED = FALSE;

    FOR (i = 1:m) {
        Ui = U[i, 2];
        Mi = U[i, 1];

        FOR (j = 1:m) {
            Uj = U[j, 2];
            Mj = U[j, 1];
```

```
IF ( U_i ≠ U_j  &&  U_i ∩ U_j ≠ ∅ ){
    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) {
      U[j, 2] = U_j;
      U[j, 1] = M_j;
    } END
  } END

  IF (ISCHANGED) {
      U[i, 2] = U_i;
      U[i, 1] = M_i;
    } END
  } END
} END
```

2.4 - Construct the U^* Sets

To construct the U^* sets we look for unique U_i in U and store each unique U_i as its own set in U^*. This is done via the following routine

```
U*[1] = U[1,2];  clustCnt = 1;
For (i = 2 : m) {
    U_i = U[i,2];  Boolean isNew = True;

    For (j = 1 : clustCnt) {
        U*_j = U*[j];

        If (U*_j == U_i) {
            isNew = False;
            Break;
        } End
    } End

    If (isNew) {
        clustCnt = clustCnt + 1;
        U*[clustCnt] = U_i;
    } End
} End
```

2.5 - Compute $\bar{A}_V$

Using the expression

$$\bar{\bar{A}}_V = \bar{\bar{L}}^T \bar{\bar{L}}$$

compute $\bar{\bar{A}}_V \in \mathbb{R}^{n \times n}$. Alternatively, the expression

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

may be used if different weighting is desired.

2.6 - Construct $\mathbb{V}$

We initially construct $\mathbb{V}$ by the

following routine

```
FOR (i = 1:n) {
|   V_i = Φ;   N_i = Φ;
|
|   FOR (j = 1:n) {
|   |  IF ((Ā_v)_ij ≠ 0) {
|   |  :
|   |  :   V_i = V_i ∪ {j};
|   |  :
|   |  :   N_i = N_i ∪ {(Ā_v)_ij};
|   |  :
|   |  }END
|   }END
|
|   V[i, 1] = N_i;   V[i, 2] = V_i;
|
}END
```

After the completion of this routine, we make a copy of this initial $\mathbb{V}$, denoted $\mathbb{V}_0$.

2.7 - Complete the $\mathbb{V}$ Sets

The elements of $\mathbb{V}$ must now be 'completed' to include indirectly related elements. Taking advantage of $\bar{\mathbb{L}}$'s disjointedness, this 'completion' can be done using the simple routine below

```
BOOLEAN  ISCHANGED = TRUE;

WHILE (ISCHANGED) {
    ISCHANGED = FALSE;

    FOR (i=1:n) {
        Vi = V[i, 2];
        Ni = V[i, 1];

        FOR (j=1:n) {
            Vj = V[j, 2];
            Nj = V[j, 1];
```

```
        IF(V_i ≠ V_j && V_i ∩ V_j ≠ ∅){
            V_i = V_i ∪ {V_j \ V_i};
            N_i = N_i ∪ {N_j \ N_i};
            V_j = V_j ∪ {V_i \ V_j};
            N_j = N_j ∪ {N_i \ N_j};
            ISCHANGED = TRUE;
        }END

        IF(ISCHANGED){
            V[j,2] = V_i;
            V[j,1] = N_j;
        }END

    }END

    IF(ISCHANGED){
        V[i,2] = V_i;
        V[i,1] = N_i;
    }END

}END
}END
```

2.8 - Construct the $\mathbb{V}^*$ Sets

To construct the sets in $\mathbb{V}^*$, we look for unique $\mathbb{V}_j$ in $\mathbb{V}$, then store each unique $\mathbb{V}_j$ as its own set in $\mathbb{V}^*$. This is done via the following routine

```
V*[1] = V[1,2]; CLUSTCNT = 1;
FOR (i = 2:n) {
    BOOLEAN  ISNEW = TRUE;
    Vi = V[i, 2];

    FOR (j = 1: CLUSTCNT) {
    |  V*j = V*[j];
    |
    |  IF (V*j == Vi) {
    |  |  ISNEW = FALSE; BREAK;
    |  } END
    } END

    IF (ISNEW) {
    |  CLUSTCNT++; V*[CLUSTCNT] = Vi;
    } END
} END
```

2.9 - Reorder $\bar{\bar{L}}$ into $\bar{\bar{L}}^*$

The last step is to reorder $\bar{\bar{L}}$ into the new, clustered matrix $\bar{\bar{L}}^* \in \mathbb{R}^{m \times n}$. This is done using U^* to cluster the rows, during reordering, and V^* to cluster the columns, during reordering

2.9.1 - Reorder Rows

To reorder the rows of $\bar{\bar{L}}$, we consider $\bar{\bar{L}}$ to consist of m row vectors, denoted $\vec{l}_{v_i}^{\,T} \in \mathbb{R}^n$. That is to say that we may express the i^{th} row of $\bar{\bar{L}}$, $(\bar{\bar{L}})_{i*}$, in terms of $\vec{l}_{v_i}$ by the relation

$$(\overline{\overline{L}})_{i*} = \vec{l}_{U_i}$$

Now, let us define $\mu_i \in \mathbb{Z}^+$ to be the number of elements in each of the $\mathbb{U}_i^* \in \mathbb{U}^*$; and $\mu^* \in \mathbb{Z}^+$ to be the number of $\mathbb{U}_i^*$ in $\mathbb{U}^*$. That is to say

$$\mu_i = |\mathbb{U}_i^*|$$

and

$$\mu^* = |\mathbb{U}^*|$$

Using these definitions, we will reorder $\overline{\overline{L}}$, by rows, into the

temporary matrix $\bar{\bar{\Sigma}}_0 \in \mathbb{R}^{m \times n}$ via the routine below

```
TERMLIST* = [];
t = 1;
FOR (i = 1 : μ*) {
    U*_i = U*[i];
    FOR (j = 1 : μ_i) {
        k = U*_i [j];
        (Σ̄_0)_{t*} = (Σ̄)_{k*};
        TERMLIST*[t] = TERMLIST[k];
        t++;
    } END
} END
```

where the 'TERMLIST[]' variable is an array, of size m, which contains the labels for each of the

terms corresponding to the original ordering. Similarly, the 'TERMLIST*[]' variable is an array, of size m, containing the labels for each of the corresponding to the new ordering. We will next use our temporary, row-reordered matrix, $\bar{\bar{L}}_o$, to create the final, completely reordered matrix, $\bar{\bar{L}}^*$.

2.9.2 - Reorder Columns

Since $\bar{\bar{L}}$ and $\bar{\bar{L}}_o$ have the same column order and $\bar{\bar{L}}_o$ is already row-reordered, we may construct $\bar{\bar{L}}^*$ from $\bar{\bar{L}}_o$ by reordering the

columns of $\bar{\bar{L}}_0$ according to $\mathbb{V}^*$. As with the row-reordering, we consider $\bar{\bar{L}}_0$ to consist of n column vectors, denoted $\vec{l}_{v_j} \in \mathbb{R}^m$. That is to say that we may express the j^{th} column of $\bar{\bar{L}}_0$, $(\bar{\bar{L}}_0)_{*j}$, in terms of $\vec{l}_{v_j}$ by the relation

$$(\bar{\bar{L}}_0)_{*j} = \vec{l}_{v_j}$$

Now, let us define $v_j \in \mathbb{Z}^+$ to be the number of elements in each of the $\mathbb{V}_j^* \in \mathbb{V}$, and $v^* \in \mathbb{Z}^+$ to be the number of $\mathbb{V}_j^*$ in $\mathbb{V}^*$. That is to say that

$$\nu_j = |\mathbb{V}_j|$$

and

$$\nu^* = |\mathbb{V}^*|$$

Using these definitions, we will reorder $\bar{\bar{L}}_o$, by columns, into the final matrix $\bar{\bar{L}}^* \in \mathbb{R}^{m \times n}$ via the routine below

```
OBJLIST* = [];  t = 1;
FOR (j = 1 : ν*) {
    V*_j = V*[j];

    FOR (i = 1 : ν_j) {
        k = V*_j[i];  (L̄̄*)_*t = (L̄̄_o)_*k;
        OBJLIST*[t] = OBJLIST[k];  t++;
    } END

} END
```

Where the 'OBJLIST[]' variable is an array, of size n, containing the labels for each of the objects in an order corresponding to the original column ordering of $\bar{L}$ and, by extension, $\bar{L}_0$. Similarly, the 'OBJLIST'[]' variable is an array, also of size n, which contains the labels for each of the objects but ordered to correspond to the new column ordering.

4 Other Algorithms

Our algorithm will be benchmarked against two other algorithms, each with two versions. These algorithms and their versions are

- 1) SVS Signs
  - +A) of order $k < r$
  - +B) of full rank $(k = r)$
- 2) SVS Gaps
  - +A) of order $k < r$
  - +B) of full rank $(k = r)$

with $r = \text{RANK}[\bar{L}]$.

4.1 - SVD Signs of order k

For

5 Test Data

We will use three types of data to test the algorithm. These are

- 1) Contrived Data
- 2) Random Data
- 3) 'Real-World' Data

The contrived data will be used to ensure that the algorithm produces the expected results from a known and solved dataset. On the other hand, the random data serves to verify that this algorithm produces similar results

compared to the other algorithms being used as benchmarks.