

3.4 Finding a Discrete Solution Close to a Continuous Approximation

The next step is to find an exact solution $(\mathbb{P}(X^1), \dots, \mathbb{P}(X^K)) \in \mathbb{P}(\mathcal{K})$ which is the closest (in a suitable sense) to our approximate solution (Z^1, \dots, Z^K) .

Recall from Section 3.3 that every solution Z of problem $(*_2)$ yields a *family of solutions* of problem $(*_1)$; namely, all matrices of the form ZQ , where Q is a $K \times K$ matrix with nonzero and pairwise orthogonal columns.

Since the solutions ZQ of $(*_1)$ are all equivalent (they yield the same minimum for the normalized cut), it makes sense to look for a discrete solution X closest to one of these ZQ .

We view \mathcal{K} as a subset of $(\mathbb{R}\mathbb{P}^{N-1})^K$.

Because \mathcal{K} is closed under the antipodal map, it can be shown that for every j ($1 \leq j \leq K$), minimizing the distance $d(\mathbb{P}(X^j), \mathbb{P}(Z^j))$ on $\mathbb{R}\mathbb{P}^{N-1}$ is equivalent to minimizing $\|X^j - Z^j\|_2$, where X^j and Z^j are representatives of $\mathbb{P}(X^j)$ and $\mathbb{P}(Z^j)$ on the unit sphere (if we use the Riemannian metric on $\mathbb{R}\mathbb{P}^{N-1}$ induced by the Euclidean metric on \mathbb{R}^N).

Then, if we use the product distance on $(\mathbb{R}\mathbb{P}^{N-1})^K$ given by

$$\begin{aligned} d((\mathbb{P}(X^1), \dots, \mathbb{P}(X^K)), (\mathbb{P}(Z^1), \dots, \mathbb{P}(Z^K))) \\ = \sum_{j=1}^K d(\mathbb{P}(X^j), \mathbb{P}(Z^j)), \end{aligned}$$

minimizing the distance

$d((\mathbb{P}(X^1), \dots, \mathbb{P}(X^K)), (\mathbb{P}(Z^1), \dots, \mathbb{P}(Z^K)))$ in $(\mathbb{R}\mathbb{P}^{N-1})^K$ is equivalent to minimizing

$$\sum_{j=1}^K \|X^j - Z^j\|_2, \quad \text{subject to}$$

$$\|X^j\|_2 = \|Z^j\|_2 \quad (j = 1, \dots, K).$$

We are not aware of any optimization method to solve the above problem, which seems difficult to tackle due to constraints $\|X^j\|_2 = \|Z^j\|_2$ ($j = 1, \dots, K$).

Therefore, we drop these constraints and attempt to minimize

$$\|X - Z\|_F^2 = \sum_{j=1}^K \|X^j - Z^j\|_2^2,$$

the Frobenius norm of $X - Z$. This is implicitly the choice made by Yu.

Inspired by Yu [16] and the previous discussion, given a solution Z of problem $(*_2)$, we look for pairs (X, Q) with $X \in \mathcal{X}$ and where Q is a $K \times K$ matrix with nonzero and pairwise orthogonal columns, with $\|X\|_F = \|Z\|_F$, that minimize

$$\varphi(X, Q) = \|X - ZQ\|_F.$$

Yu [16] and Yu and Shi [17] consider the special case where $Q \in \mathbf{O}(K)$.

We consider the more general case where $Q = R\Lambda$, with $R \in \mathbf{O}(K)$ and Λ is a diagonal invertible matrix.

The key to minimizing $\|X - ZQ\|_F$ rests on the following result:

$$\|X - ZQ\|_F^2 = \|X\|_F^2 - 2\text{tr}(Q^\top Z^\top X) + \text{tr}(Z^\top ZQQ^\top).$$

Therefore, since $\|X\|_F = \|Z\|_F$ is fixed, minimizing $\|X - ZQ\|_F^2$ is equivalent to minimizing $-2\text{tr}(Q^\top Z^\top X) + \text{tr}(Z^\top ZQQ^\top)$.

This is a hard problem because it is a nonlinear optimization problem involving two matrix unknowns X and Q .

To simplify the problem, we proceed by *alternating steps* during which we *minimize* $\varphi(X, Q) = \|X - ZQ\|_F$ *with respect to* X *holding* Q *fixed*, and steps during which we *minimize* $\varphi(X, Q) = \|X - ZQ\|_F$ *with respect to* Q *holding* X *fixed*.

This *second step in which* X *is held fixed* has been studied, but it is still a hard problem for which no closed-form solution is known.

Consequently, we further simplify the problem.

Since Q is of the form $Q = R\Lambda$ where $R \in \mathbf{O}(K)$ and Λ is a diagonal invertible matrix, we minimize $\|X - ZR\Lambda\|_F$ in two stages.

1. We set $\Lambda = I$ and find $R \in \mathbf{O}(K)$ that minimizes $\|X - ZR\|_F$.
2. Given X , Z , and R , find a diagonal invertible matrix Λ that minimizes $\|X - ZR\Lambda\|_F$.

The matrix $R\Lambda$ is not a minimizer of $\|X - ZR\Lambda\|_F$ in general, but it is an improvement on R alone, and both stages can be solved quite easily.

In *stage 1*, the matrix $Q = R$ is orthogonal, so $QQ^\top = I$, and since Z and X are given, the problem reduces to minimizing $-2\text{tr}(Q^\top Z^\top X)$; that is, *maximizing* $\text{tr}(Q^\top Z^\top X)$.

To solve this problem, we need the following proposition.

Proposition 3.2. *For any $n \times n$ matrix A and any orthogonal matrix Q , we have*

$$\max\{\text{tr}(QA) \mid Q \in \mathbf{O}(n)\} = \sigma_1 + \cdots + \sigma_n,$$

where $\sigma_1 \geq \cdots \geq \sigma_n$ are the singular values of A . Furthermore, this maximum is achieved by $Q = VU^\top$, where $A = U\Sigma V^\top$ is any SVD for A .

As a corollary of Proposition 3.2 (with $A = Z^\top X$ and $Q = R^\top$), we get the following result (see Golub and Van Loan [8], Section 12.4.1):

Proposition 3.3. *For any two fixed $N \times K$ matrices X and Z , the minimum of the set*

$$\{\|X - ZR\|_F \mid R \in \mathbf{O}(K)\}$$

is achieved by $R = UV^\top$, for any SVD decomposition $U\Sigma V^\top = Z^\top X$ of $Z^\top X$.

The following proposition takes care of *stage 2*.

Proposition 3.4. *For any two fixed $N \times K$ matrices X and Z , where Z has no zero column, there is a unique diagonal matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_K)$ minimizing $\|X - Z\Lambda\|_F$ given by*

$$\lambda_j = \frac{(Z^\top X)_{jj}}{\|Z^j\|_2^2} \quad j = 1, \dots, K.$$

It should be noted that Proposition 3.4 does not guarantee that Λ is invertible. For example, for

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & -1 \end{pmatrix},$$

we have

$$Z^{\top} X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 0 \end{pmatrix},$$

so $\lambda_2 = 0$. When Proposition 3.4 yields a singular matrix, we skip stage 2 (we set $\Lambda = I$).

We now deal with *step 1, where $Q = R\Lambda$ is held fixed.*

For fixed Z and Q , we would like to find some $X \in \mathcal{K}$ with $\|X\|_F = \|Z\|_F$ so that $\|X - ZQ\|_F$ is minimal.

Without loss of generality, we may assume that the entries a_1, \dots, a_K occurring in the matrix X are positive and all equal to some common value $a \neq 0$.

Recall that a matrix $X \in \mathcal{X}$ has the property that every row contains exactly one nonzero entry, and that every column is nonzero.

To find $X \in \mathcal{K}$, first we find the shape \hat{X} of X , which is the matrix obtained from X by rescaling the columns of X so that \hat{X} has entries $+1, 0$.

The problem is to decide for each row, which column contains the nonzero entry.

After having found \widehat{X} , we rescale its columns so that $\|X\|_F = \|Z\|_F$.

Since

$$\|X - ZQ\|_F^2 = \|X\|_F^2 - 2\text{tr}(Q^\top Z^\top X) + \text{tr}(Z^\top ZQQ^\top),$$

minimizing $\|X - ZQ\|_F$ is equivalent to *maximizing*

$$\text{tr}(Q^\top Z^\top X) = \text{tr}((ZQ)^\top X) = \text{tr}(X(ZQ)^\top),$$

and since the i th row of X contains a single nonzero entry a in column j_i ($1 \leq j_i \leq K$), if we write $Y = ZQ$, then

$$\text{tr}(XY^\top) = a \sum_{i=1}^N y_{ij_i}. \quad (*)$$

By (*), since $a > 0$, the quantity $\text{tr}(XY^\top)$ is maximized iff y_{ij_i} is maximized for $i = 1, \dots, N$; *this is achieved if for the i th row of X , we pick a column index ℓ such that $y_{i\ell}$ is maximum.*

To find the shape \widehat{X} of X , we first find a matrix \overline{X} by choosing a single nonzero entry $\overline{x}_{ij} = 1$ on row i in such a way that y_{ij} is maximum according to the following method.

If we let

$$\begin{aligned}\mu_i &= \max_{1 \leq j \leq K} y_{ij} \\ J_i &= \{j \in \{1, \dots, K\} \mid y_{ij} = \mu_i\},\end{aligned}$$

for $i = 1, \dots, N$, then

$$\overline{x}_{ij} = \begin{cases} +1 & \text{for some chosen } j \in J_i, \\ 0 & \text{otherwise.} \end{cases}$$

Of course, a single column index is chosen for each row. In our implementation, we pick the smallest index in J_i .

Unfortunately, the matrix \overline{X} may not be a correct solution, because the above prescription does not guarantee that every column of \overline{X} is nonzero.

When this happens, we *reassign certain nonzero entries in columns having “many” nonzero entries to zero columns, so that we get a matrix in \mathcal{K} .*

Suppose column j is zero. Then, we pick the leftmost index k of a column with a maximum number of 1, and if i the smallest index for which $\overline{X}_{ik} = 1$, then we set $\overline{X}_{ik} = 0$ and $\overline{X}_{ij} = 1$.

We repeat this reallocation scheme until every column is nonzero.

We obtain a new matrix \widehat{X} in \mathcal{X} , and finally we normalize \widehat{X} to obtain X , so that $\|X\|_F = \|Z\|_F$.

The above method is essentially the method described in Yu [16] and Yu and Shi [17].

The fact that \overline{X} may have zero columns is not addressed by Yu.

Furthermore, it is important to make sure that X has the same norm as Z , but this normalization step is not performed in the above works.

On the other hand, the rows of Z are normalized and the resulting matrix may no longer be a correct solution of the relaxed problem.

In practice, it appears to be a good approximation of a discrete solution; see option (3) of the initialization methods for Z described below.

Any matrix obtained by flipping the signs of some of the columns of a solution ZR of problem $(*_2)$ is still a solution.

Moreover, all entries in X are nonnegative. It follows that a “good” solution ZQ_p (that is, close to a discrete solution) should have the property that the average of each of its column is nonnegative.

We found that the following heuristic is quite helpful in finding a better discrete solution X .

Given a solution ZR of problem $(*_2)$, we compute ZQ_p , defined such that if the average of column $(ZR)^j$ is negative, then $(ZQ_p)^j = -(ZR)^j$, else $(ZQ_p)^j = (ZR)^j$.

It follows that the average of every column in ZQ_p is nonnegative. Then, we apply the above procedure to find discrete solutions X and X_p closest to ZR and ZQ_p respectively, and we pick the solution corresponding to $\min\{\|X - ZR\|_F, \|X_p - ZQ_p\|_F\}$.

Flipping signs of columns of ZR correspond to a diagonal matrix R_p with entries ± 1 , a very special kind of orthogonal matrix.

In summary, the procedure for finding a discrete X close to a continuous ZR also updates R to $Q_p = RR_p$.

This step appears to be very effective for finding a good initial X .

The method due to Yu and Shi (see Yu [16] and Yu and Shi [17]) to find $X \in \mathcal{K}$ and $Q = R\Lambda$ with $R \in \mathbf{O}(K)$ and Λ diagonal invertible that minimize

$\varphi(X, Q) = \|X - ZQ\|_F$ is to *alternate steps* during which *either Q is held fixed* (step PODX) *or X is held fixed* (step PODR), except that Yu and Shi consider the special case where $\Lambda = I$.

- (1) In step PODX, the next discrete solution X^* is obtained from the previous pair (Q^*, Z) by computing \bar{X} and then $X^* = \hat{X}$ from $Y = ZQ^*$, as just explained above.
- (2) In step PODR, the next matrix $Q^* = R\Lambda$ is obtained from the previous pair (X^*, Z) by first computing

$$R = UV^\top,$$

for any SVD decomposition $U\Sigma V^\top$ of $Z^\top X^*$, and then computing Λ from X^* and ZR using Proposition 3.4. If Λ is singular, then set $\Lambda = I$.

We keep track of the progress of the procedure by computing $\varphi(X^*, Q^*) = \|X^* - ZQ^*\|_F$ after every step and checking that X^* or $\varphi(X^*, Q^*)$ stops changing, whichever comes first.

We observed that after a small number of steps, up to machine precision, $\varphi(X^*, Q^*)$ stops decreasing, and when this occurs the procedure halts (we also set a maximum number of steps in case $\varphi(X^*, Q^*)$ decreases for a very long time).

Moreover, looking for $Q = R\Lambda$ where $R \in \mathbf{O}(K)$ and Λ is obtained using the method of Proposition 3.4 speeds up the convergence and yields a better discrete solution X .

The process of searching for X and Q has an illuminating geometric interpretation in terms of graph drawings.

We may assume that the entries in the discrete solution X are 0 or 1.

Then the rows of the discrete solutions X correspond to the tips of the unit vectors along the coordinate axes in \mathbb{R}^K .

Every axis contains at least such a point, and the multiplicity of the point along the j th axis is the number of nodes in the j th block of the partition.

Similarly, the rows of Z are the nodes of a graph drawing of the weighted graph (V, W) .

Multiplying Z on the right by a $K \times K$ matrix Q (obtaining ZQ) is equivalent to multiplying Z^\top on the left by Q^\top (obtaining $Q^\top Z^\top$).

This means that the points in \mathbb{R}^K representing the rows of ZQ are obtained by applying the linear transformation Q^\top to the columns of Z^\top .

Thus, ZR amounts to applying the rigid motion R^\top to the graph drawing Z , and $Z\Lambda$ (where Λ is a diagonal invertible matrix) amounts to stretching or shrinking the graph drawing Z in the directions of the axes.

Then, in step 2 (PODR), we are trying to deform the graph drawing given by Z using a linear map $(R\Lambda)^\top$, so that the deformed graph drawing $ZR\Lambda$ is as close as possible to X (in the sense that $\|X - ZR\Lambda\|_F$ is minimized).

In step 1 (PODX), we are trying to approximate the deformed graph drawing $ZR\Lambda$ by a discrete graph drawing X (whose nodes are the tips of the unit vectors), so that $\|X - ZR\Lambda\|_F$ is minimized.

If we are willing to give up the requirement that the deformed Z is still a solution of problem $(*_1)$, we have quite a bit of freedom in step 2.

For example, we may allow normalizing the rows.

This seems reasonable to obtain an initial transformation Q .

However, we feel uncomfortable in allowing intermediate deformed Z that are not solutions of $(*_1)$ during the iteration process.

This point should be investigated further.

In some sense, we have less freedom in step 1, since the i th row of ZRA is assigned to the j th unit vector iff the index of the leftmost largest coordinate of this row is j .

If some axis has not been assigned any row of R , then we reallocate one of the points on an axis with a maximum number of points.

Figure 3.5 shows a graph (on the left) and the graph drawings X and $Z * R$ obtained by applying our method for three clusters.

The rows of X are represented by the red points along the axes, and the rows of $Z * R$ by the green points (on the right).

The original vertices corresponding to the rows of Z are represented in blue.

We can see how the two red points correspond to an edge, the three red points correspond to a triangle, and the four red points to a quadrangle.

These constitute the clusters.

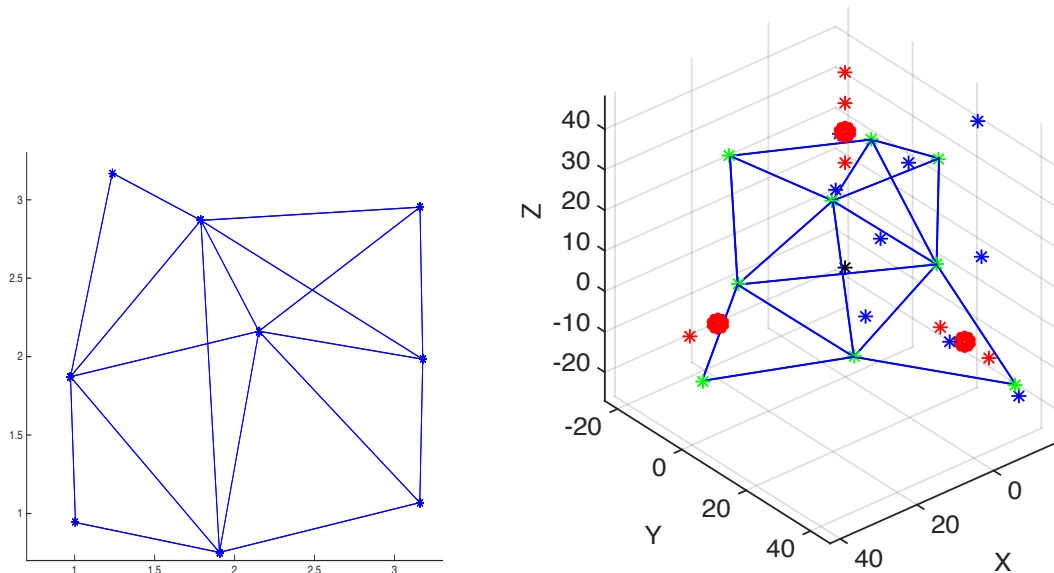


Figure 3.5: A graph and its drawing to find 3 clusters.

It remains to initialize Q^* to start the process, and then steps (1) and (2) are iterated, starting with step (1).

Actually, what we really need is a “good” initial X^* , but to find it, we need an initial R^* .

Method 1. One method is to use an orthogonal matrix denoted R_1 , such that distinct columns of ZR_1 are *simultaneously orthogonal and D-orthogonal*.

The matrix R_1 can be found by diagonalizing $Z^\top Z$ as $Z^\top Z = R_1 \Sigma R_1^\top$, as we explained at the end of Section 3.3. We write $Z_2 = ZR_1$.

Method 2. The method advocated by Yu [16] is to *pick K rows of Z that are as orthogonal to each other as possible* and to make a matrix R whose columns consist of these rows normalized to have unit length.

The intuition behind this method is that if a continuous solution Z can be sent close to a discrete solution X by a rigid motion, then many rows of Z viewed as vectors in \mathbb{R}^K should be nearly orthogonal.

This way, ZR should contain at least K rows well aligned with the canonical basis vectors, and these rows are good candidates for some of the rows of the discrete solution X .

The algorithm given in Yu [16] needs a small correction, because rows are not removed from Z when they are added to R , which may cause the same row to be added several times to R .

Given the $N \times K$ matrix Z (whose columns all have the same norm), we compute a matrix R whose columns are certain rows of Z .

We use a vector $c \in \mathbb{R}^N$ to keep track of the inner products of all rows of Z with the columns R^1, \dots, R^{k-1} that have been constructed so far, and initially when $k = 1$, we set $c = 0$.

The first column R^1 of R is any chosen row of Z .

Next, for $k = 2, \dots, K$, we compute all the inner products of R^{k-1} with all rows in Z , which are recorded in the vector ZR^{k-1} , and we update c as follows:

$$c = c + \mathbf{abs}(ZR^{k-1}).$$

We take the absolute values of the entries in ZR^{k-1} so that the i th entry in c is a score of how orthogonal is the i th row of Z to R^1, \dots, R^{k-1} .

Then, we choose R^k as any row Z_i of Z for which c_i is minimal (the customary (and ambiguous) $i = \arg \min c$), and we delete this row from Z . The process is repeated (with the updated Z) until $k = K$.

At the end of the above process, we normalize the columns of R , to obtain a matrix that we denote R_2 .

After some experimentation, we found that to obtain a better initial X^* , it is may desirable to start from a variant of the continuous solution Z obtained by solving problem $(*_2)$.

We have implemented three methods.

1. We attempt to rescale the columns of Z by some diagonal invertible matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_K)$, so that the rows of $Z\Lambda$ sum to 1 as much as possible in the least-squares sense.

Since the vector of sums of rows of $Z\Lambda$ is $Z\Lambda\mathbf{1}_K = Z\lambda$, with $\lambda^\top = (\lambda_1, \dots, \lambda_K)$, the least-squares problem is to minimize

$$\|Z\lambda - \mathbf{1}_N\|_2^2,$$

and since Z has rank K , the solution is $\lambda = (Z^\top Z)^{-1}Z^\top \mathbf{1}_N$, and thus,

$$\Lambda = \text{diag}((Z^\top Z)^{-1}Z^\top \mathbf{1}_N).$$

The matrix Λ is singular if some of the columns of Z sum to 0.

This happens for regular graphs, where the degree matrix is a multiple of the identity.

There are also cases where some of the λ_j are very small, so we use a tolerance factor to prevent this, and in case of failure, we set $\Lambda = I$.

In case of failure, we may also use ZR_1 instead of Z , where R_1 is the orthogonal matrix that makes ZR_1 both D -orthogonal and orthogonal.

2. We attempt to rescale the columns of Z by some diagonal invertible matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_K)$, so that the rows of $Z\Lambda$ have unit length as much as possible in the least-squares sense.

Since the square-norm of the i th row of $Z\Lambda$ is

$$\sum_{j=1}^K z_{ij}^2 \lambda_j^2,$$

if we write $Z \circ Z$ for the matrix (z_{ij}^2) of square entries of elements in Z (the Hadamard product of Z with itself), the least-squares problem is to minimize

$$\|Z \circ Z \lambda^2 - \mathbf{1}_N\|_2^2,$$

where $(\lambda^2)^\top = (\lambda_1^2, \dots, \lambda_K^2)$.

The matrix $Z \circ Z$ may not have rank K , so the least-squares solution for λ^2 is given by the pseudo-inverse of $Z \circ Z$, as

$$\lambda^2 = (Z \circ Z)^+ \mathbf{1}_N.$$

There is no guarantee that the vector on the right-hand side has all positive entries, so the method may fail.

It may also fail when some of the λ_j are very small.

We use a tolerance factor to prevent this, and in case of failure, we set $\Lambda = I$.

3. We use a method more drastic than (2), which consists in normalizing the rows of Z . Thus, we form the matrix

$$NZ = \text{diag}((ZZ^\top)_{11}^{-1/2}, \dots, (ZZ^\top)_{NN}^{-1/2}),$$

and we return $NZ * Z$.

Unlike the methods used in (1) and (2), this method does not guarantee that $NZ * Z$ is a solution of problem $(*_1)$.

However, since the rows of Z can be interpreted as vectors in \mathbb{R}^K that should align as much as possible with the canonical basis vectors of \mathbb{R}^K , this method makes sense as a way to come closer to a discrete solution.

In fact, we found that it does well in most cases.

We implemented a computer program that prompts the user for various options.

To avoid confusion, let us denote the original solution of problem $(*_2)$ by Z_1 , and let $Z_2 = Z_1 R_1$, as obtained by initialization method 1.

The four options are:

1. Use the original solution Z_1 of problem $(*_2)$, as well as Z_2 .
2. Apply method 1 to Z_1 and Z_2 .
3. Apply method 2 to Z_1 and Z_2 .
4. Apply method 3 to Z_1 and Z_2 .

Then, for each of these options, if we denote by Z_{init_1} and Z_{init_2} the solutions returned by the method, our program computes initial solutions X_1, X_2, X_3, X_4 as follows:

1. Use Z_{init_1} and $R = I$.
2. Use Z_{init_1} and $R = R2a$, the matrix given by initialization method 2.
3. Use Z_{init_2} and $R = I$.
4. Use Z_{init_2} and $R = R2b$, the matrix given by initialization method 2.

After this, the program picks the discrete solution $X = X_i$ which corresponds to the minimum of

$$\begin{aligned} &\|X_1 - Z_{init_1}\|, \|X_2 - Z_{init_1} * R2a\|, \\ &\|X_3 - Z_{init_2}\|, \|X_4 - Z_{init_2} * R2b\|. \end{aligned}$$

Our experience is that options (3) and (4) tend to give better results.

However, it is harder to say whether any of the X_i does a better job than the others, although (2) and (4) seem to do slightly better than (1) and (3).

We also give the user the option in step PODR to only compute R and set $\Lambda = I$.

It appears that the more general method is hardly more expensive (because finding Λ is cheap) and always gives better results.

We also found that we obtain better results if we rescale Z (and X) so that $\|Z\|_F = 100$.

If we apply the method (using method 3 to find the initial R) to the graph associated with the the matrix W_1 shown in Figure 3.6 for $K = 4$ clusters, the algorithm converges in 3 steps and we find the clusters shown in Figure 3.7.

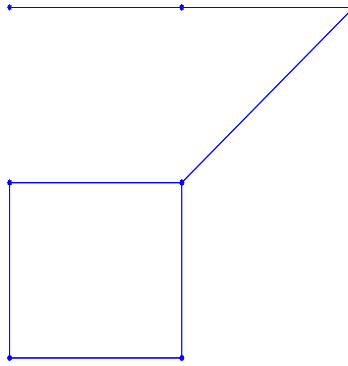


Figure 3.6: Underlying graph of the matrix W_1 .

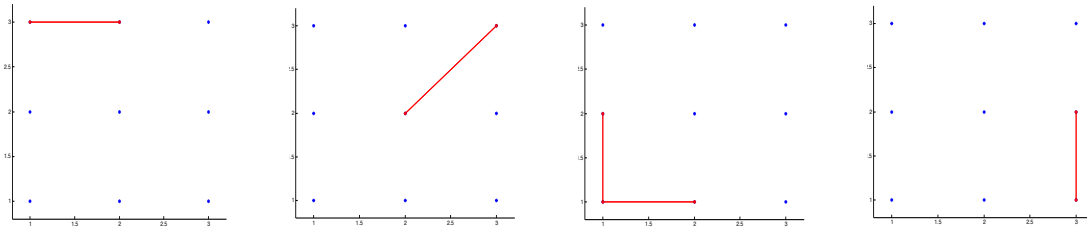


Figure 3.7: Four blocks of a normalized cut for the graph associated with W_1 .

The solution Z of the relaxed problem is

$$Z = \begin{pmatrix} -21.3146 & -0.0000 & 19.4684 & -15.4303 \\ -4.1289 & 0.0000 & 16.7503 & -15.4303 \\ -21.3146 & 32.7327 & -19.4684 & -15.4303 \\ -4.1289 & -0.0000 & 16.7503 & -15.4303 \\ 19.7150 & 0.0000 & 9.3547 & -15.4303 \\ -4.1289 & 23.1455 & -16.7503 & -15.4303 \\ -21.3146 & -32.7327 & -19.4684 & -15.4303 \\ -4.1289 & -23.1455 & -16.7503 & -15.4303 \\ 19.7150 & -0.0000 & -9.3547 & -15.4303 \end{pmatrix}.$$

We find the following sequence for $Q, Z * Q, X$:

$$Q = \begin{pmatrix} 0 & 0.6109 & -0.3446 & -0.7128 \\ -1.0000 & 0.0000 & 0.0000 & -0.0000 \\ 0.0000 & 0.5724 & 0.8142 & 0.0969 \\ -0.0000 & 0.5470 & -0.4672 & 0.6947 \end{pmatrix},$$

which is the initial Q obtained by method 1;

$$Z * Q = \begin{pmatrix} 0.0000 & -10.3162 & 30.4065 & 6.3600 \\ 0.0000 & -1.3742 & 22.2703 & -6.1531 \\ -32.7327 & -32.6044 & -1.2967 & 2.5884 \\ 0.0000 & -1.3742 & 22.2703 & -6.1531 \\ 0.0000 & 8.9576 & 8.0309 & -23.8653 \\ -23.1455 & -20.5505 & -5.0065 & -9.3982 \\ 32.7327 & -32.6044 & -1.2967 & 2.5884 \\ 23.1455 & -20.5505 & -5.0065 & -9.3982 \\ -0.0000 & -1.7520 & -7.2027 & -25.6776 \end{pmatrix}$$

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} ;$$

$$Q = \begin{pmatrix} -0.0803 & 0.8633 & -0.4518 & -0.2102 \\ -0.6485 & 0.1929 & 0.1482 & 0.7213 \\ -0.5424 & 0.0876 & 0.5546 & -0.6250 \\ -0.5281 & -0.4581 & -0.6829 & -0.2119 \end{pmatrix}$$

$$Z * Q = \begin{pmatrix} -0.6994 & -9.6267 & 30.9638 & -4.4169 \\ -0.6051 & 4.9713 & 21.6922 & -6.3311 \\ -0.8081 & -6.7218 & 14.2223 & 43.5287 \\ -0.6051 & 4.9713 & 21.6922 & -6.3311 \\ 1.4913 & 24.9075 & 6.8186 & -6.7218 \\ 2.5548 & 6.5028 & 6.5445 & 31.3015 \\ 41.6456 & -19.3507 & 4.5190 & -3.6915 \\ 32.5742 & -2.4272 & -0.3168 & -2.0882 \\ 11.6387 & 23.2692 & -3.5570 & 4.9716 \end{pmatrix}$$

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} ;$$

$$Q = \begin{pmatrix} -0.3201 & 0.7992 & -0.3953 & -0.3201 \\ -0.7071 & -0.0000 & 0.0000 & 0.7071 \\ -0.4914 & -0.0385 & 0.7181 & -0.4914 \\ -0.3951 & -0.5998 & -0.5728 & -0.3951 \end{pmatrix}$$

$$Z * Q = \begin{pmatrix} 3.3532 & -8.5296 & 31.2440 & 3.3532 \\ -0.8129 & 5.3103 & 22.4987 & -0.8129 \\ -0.6599 & -7.0310 & 3.2844 & 45.6311 \\ -0.8129 & 5.3103 & 22.4987 & -0.8129 \\ -4.8123 & 24.6517 & 7.7629 & -4.8123 \\ -0.7181 & 6.5997 & -1.5571 & 32.0146 \\ 45.6311 & -7.0310 & 3.2844 & -0.6599 \\ 32.0146 & 6.5997 & -1.5571 & -0.7181 \\ 4.3810 & 25.3718 & -5.6719 & 4.3810 \end{pmatrix}$$

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} .$$

During the next round, the exact same matrices are obtained and the algorithm stops.

Comparison of the matrices $Z * Q$ and X makes it clear that X is obtained from $Z * Q$ by retaining on every row the leftmost largest value and setting the others to 0 (non-maximum suppression).

In this example, the columns of all X were nonzero, but this may happen, for example when we apply the algorithm to the graph of Figure 3.6 to find $K = 5$ clusters shown in Figure 3.8.

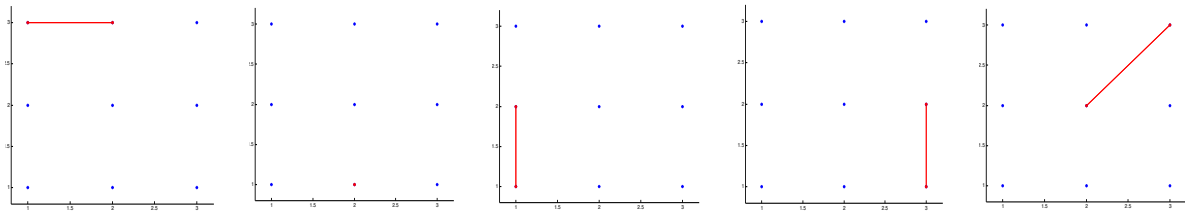


Figure 3.8: Five blocks of a normalized cut for the graph associated with W_1 .

We find that the initial value for $Z * Q$ is

$$Z * Q = \begin{pmatrix} -5.7716 & -27.5934 & 0.0000 & -9.3618 & -0.0000 \\ 5.5839 & -20.2099 & -29.7044 & -1.2471 & -0.0000 \\ -2.3489 & 1.1767 & -0.0000 & -29.5880 & -29.7044 \\ 5.5839 & -20.2099 & 29.7044 & -1.2471 & 0.0000 \\ 21.6574 & -7.2879 & 0.0000 & 8.1289 & 0.0000 \\ 8.5287 & 4.5433 & -0.0000 & -18.6493 & -21.0042 \\ -2.3489 & 1.1767 & -0.0000 & -29.5880 & 29.7044 \\ 8.5287 & 4.5433 & -0.0000 & -18.6493 & 21.0042 \\ 23.3020 & 6.5363 & -0.0000 & -1.5900 & -0.0000 \end{pmatrix}$$

The matrix $X1$ given by the above method in which we pick the leftmost largest entry on every row has a fourth row equal to 0.

The matrix $X1$ is repaired by migrating a 1 from the second entry of the first column, which contains the largest number of 1's, yielding the matrix $X2$; see below.

$$X1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad X2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

