Lecture Notes 9: Matrix Completion and Clustering Algorithm

Professor: Zhihua Zhang *Scribe: Yao Cheng, Yang Zhang, Cheng Chen*

8.4 Solution of 1-norm penalty

8.4.1 Scalar case

We begin with the scalar case, that is, we try to solve the problem

$$\min_y \frac{1}{2}(a-y)^2 + \lambda|y|$$

Then, we can compute the subdifferential: $(y-a) + \lambda\partial|y|$.
Suppose that the solution of the problem is $y = \hat{y}$, then we can get $0 \in (\hat{y}-a) + \partial\lambda|\hat{y}|$, namely, $a - \hat{y} \in \lambda\partial|\hat{y}|$.
If $\hat{y} > 0$, then $\partial|\hat{y}| = 1$. We have $a < -\lambda$ and $\hat{y} = a + \lambda$.
If $\hat{y} < 0$, then $\partial|\hat{y}| = -1$. We have $a > \lambda$ and $\hat{y} = a - \lambda$.
If $\hat{y} = 0$, then $\partial|\hat{y}| = [-\lambda, \lambda]$. We have $|a| \leq \lambda$ and $\hat{y} = 0$.
Hence, the solution is

$$\hat{y} = \text{sign}(a)(|a| - \lambda)_+ = \begin{cases} a - \lambda & a > \lambda \\ 0 & |a| \leq \lambda \\ a + \lambda & a < -\lambda \end{cases}$$

where the sign $(\cdot)_+$ is defined as $(a)_+ = max\{a, 0\}$.

8.4.2 Matrix case

Now we consider the Lasso regularization. Here the problem is

$$\min_{\mathbf{Y}} \frac{1}{2}||\mathbf{A} - \mathbf{BY}||_F^2 + \lambda||\mathbf{Y}||_1$$

Since the columns of Y are independent, the optimization problem can be decoupled into a set of optimization problems of the form

$$\min_{\mathbf{y}_j} \frac{1}{2}(\mathbf{a}_j - B\mathbf{y}_j)_2^2 + \lambda||\mathbf{y}_j||_1$$

where $\mathbf{y}_j$ is the j-th column of Y and $\mathbf{a}_j$ is the j-th column of A. The optimization problem can be solved using coodinate descent algorithm at each element of $\mathbf{y}_j$, which has been discussed before.

8.4.3 Nuclear Norm case

Now we turn to the Nuclear Norm penalty. The problem is

$$\min_{\mathbf{X}} \frac{1}{2}||\mathbf{Y}-\mathbf{X}||_F^2 + \lambda||\mathbf{X}||_*$$

Consider the subdifferential at $\hat{X}$, we have

$$0 \in (\hat{\mathbf{X}}-\mathbf{Y}) + \lambda\partial||\hat{\mathbf{X}}||_*$$

$$\mathbf{Y}-\hat{\mathbf{X}} \in \lambda\partial||\hat{\mathbf{X}}||_*$$

Suppose $\mathbf{Y}$ is of rank r, we have

$$\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T = \mathbf{U}^{(1)}\mathbf{\Sigma}^{(1)}\mathbf{V}^{(1)^T} + \mathbf{U}^{(2)}\mathbf{\Sigma}^{(2)}\mathbf{V}^{(2)^T}$$

where $\mathbf{U}^{(1)}, \mathbf{V}^{(1)}$(resp. $\mathbf{U}^{(2)}, \mathbf{V}^{(2)}$) are the singular vectors associated with singular values greater than λ (resp. smaller than or equal to λ).
Let $\hat{\mathbf{X}} = \mathbf{U}^{(1)}(\mathbf{\Sigma}^{(1)} - \lambda\mathbf{I})\mathbf{V}^{(1)^T}$, we get

$$\mathbf{Y}-\hat{\mathbf{X}} = \lambda\mathbf{U}^{(1)}\mathbf{V}^{(1)^T} + \mathbf{U}^{(2)}\mathbf{\Sigma}^{(2)}\mathbf{V}^{(2)^T} = \lambda(\mathbf{U}^{(1)}\mathbf{V}^{(1)^T} + \mathbf{U}^{(2)}\lambda^{-1}\mathbf{\Sigma}^{(2)}\mathbf{V}^{(2)^T})$$

Since $\mathbf{\Sigma}^{(2)}$ corresponds to singular values less than λ, we have $\sigma_1(\lambda^{-1}\mathbf{\Sigma}^{(2)}) \leq 1$. Thus $\mathbf{Y}-\hat{\mathbf{X}} \in \lambda\partial||\hat{\mathbf{X}}||_*$.
Hence, the solution is

$$\hat{\mathbf{X}} = \mathbf{U}(\mathbf{\Sigma} - \lambda\mathbf{I})_+\mathbf{V}^T$$

8.4.4 Robust PCA

Robust PCA try to represent a matrix $\mathbf{Z}$ as the sum of a low rank matrix $\mathbf{Y}$ and a sparse matrix $\mathbf{X}$. The optimization problem of robust PCA is

$$\min_{\mathbf{X},\mathbf{Y}} \frac{1}{2}||\mathbf{Z}-\mathbf{Y}-\mathbf{X}||_F^2 + \lambda_1||\mathbf{Y}||_1 + \lambda_2||\mathbf{X}||_*$$

The algorithm is as follows:

1. Initialize $\mathbf{X}^{(0)}$ and $\mathbf{Y}^{(0)}$
2. Rounding:

   (a) Fix $\mathbf{Y}^{(k)}$, using algorithm in subsection 7.4.3 to gain $\mathbf{X}^{(k)}$.

   (b) Fix $\mathbf{X}^{(k)}$, using algorithm in subsection 7.4.2 to gain $\mathbf{Y}^{(k+1)}$.

8.4.5 Matrix Completion

The optimization problem of matrix completion is

$$\min_{\mathbf{X}} \frac{1}{2} \sum_{(i,j)\in\Omega} (\mathbf{Y}_{ij} - \mathbf{X}_{ij})^2 + \lambda ||\mathbf{X}||_*$$

The algorithm is as follows:

1. Assign 0 to unknown elements in $\mathbf{Y}$.
2. Rounding:
   (a) Using algorithm in subsection 7.4.3 to estimate $\mathbf{X}$.
   (b) Fill the unknown elements in $\mathbf{Y}$ with estimated values in $\mathbf{X}$.

9 Clustering Algorithm

Definition 9.1. *Given a set of n samples $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n\} \subset \mathbb{R}^p$, the clustering algorithm provides a partition of them in c classes: $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_c$. We use n_j to denote the number of samples in class $\mathcal{X}_j$ so that $n_j = |\mathcal{X}_j|$.*

9.1 K-means Algorithm

The K-means algorithm aims to solve the following problem:

$$\min \sum_{j=1}^{c} \sum_{i\in\mathcal{X}_j} ||x_i - m_j||^2$$

where $m_j = \frac{1}{n_j} \sum_{i\in\mathcal{X}_j} x_i$.
The K-means clustering algorithm is as follows:

1. Let $t = 0$, and initialize cluster centroids $m_j^{(0)}$, $j = 1, 2, ...c$.
2. Assign x_i to class k, where $k = \arg\min_j ||x_i - m_j^{(t)}||_2$, $i = 1, 2, \cdots, n$.
3. $m_j^{(t+1)} = \frac{1}{|\mathcal{X}_j|} \sum_{i\in\mathcal{X}_j} x_i$.
4. If the algorithm has not been convergence, let $t = t + 1$ and goto 2.

Remark 9.1. *In K-means algorithm, we implicitly assume that the probability distribution of samples in a cluster is Gaussian distribution, i.e.,* $p(x) \sim \sum_{j=1}^{c} w_j \mathcal{N}(x|0, \sigma^2 \mathbf{I}_p)$*, with* $w_j \in \{0, 1\}$*. We can generalize this assumption as follows:* $p(x) \sim \sum_{j=1}^{c} w_j \mathcal{N}(x|0, \sigma^2 \mathbf{I}_p)$*, with* $w_j \geq 0$ *and* $\sum_{j=1}^{c} w_j = 1$*. If we use EM algorithm to solve this Gaussian mixture model, we can get a soft version of K-means algorithm.*

9.2 Spectral clustering

9.2.1 Graph notation

Let $G = (V, E)$ be an undirected graph with vertex set $V = \{1, 2, ...n\}$. $\mathbf{W}$ is the affinity matrix (or similarity matrix) and satisfies the following conditions:

1. $\mathbf{W}_{n \times n}$ is symmetric.
2. $\mathbf{W}_{ij} \geq 0$ for $(i, j) \in E$.
3. $\mathbf{W}_{ij} = 0$ for $(i, j) \notin E$.

Example 9.1.

$$W = \begin{cases} -\exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) & if\ (i,j) \in E \\ 0 & otherwise \end{cases}$$

Definition 9.2. *Assume* A *and* B *are subsets of* V*, we define* $W(A, B) = \sum_{i \in A, j \in B} \mathbf{W}_{ij}$.

9.2.2 Graph cut point of view

The intuition of clustering is to separate points in different groups according to their similarities. For data given in form of a similarity graph, this problem can be restated as follows: we want to find a partition of the graph such that the edges between different groups have very low weights(which means that points in different clusters are dissimilar from each other) and the edges within a group have high weights(which means that points within the same cluster are similar to each other). In this subsection we will see how spectral clustering can be derived as an approximation to such graph partition problems.
Given a similarity graph with adjacency matrix $\mathbf{W}$, the simplest and most direct way to construct a partition of the graph is to solve the mincut problem. To define it, we divide V into c subsets $V_j (j = 1, 2, \cdots, c)$, which meet following conditions:

1. $V_i \bigcap V_j = \emptyset, i \neq j$.
2. $\bigcup_{j=1}^{c} V_j = V$.
3. $|V_j| = n_j,\ j = 1, 2, \cdots, c$.

the mincut approach simply consists in choosing a partition which minimize the cut criterion

$$\text{PCUT} = \sum_{j=1}^{c} \frac{W(V_j, V) - W(V_j, V_j)}{\sum_{i \in V_j} \pi_i}$$

where $\boldsymbol{\pi} = (\pi_1, \pi_2, ...\pi_n)^T$ is a user defined vector of weights and $\pi_j > 0$.
Let $\mathbf{D} = \text{diag}(\mathbf{W}\mathbf{1}_n)$, we define the Laplacian matrix as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Then, $\mathbf{L}$ has following propositions:

Proposition 9.1. *The Laplacian matrix* $\mathbf{L}$ *satisfies:*

1. $\mathbf{L}_{ii} > 0$, $\mathbf{L}_{ij} \leq 0 (i \neq j)$
2. $\mathbf{L}$ *is p.s.d.*

Here we give the proof of proposition 2.

Proof. First of all, we have $\mathbf{L}\mathbf{1}_n = (\mathbf{D} - \mathbf{W})\mathbf{1}_n = \mathbf{D}\mathbf{1}_n - \mathbf{W}\mathbf{1}_n = 0$, which indicates that the sum of all elements of any rows is zero. Since $\mathbf{L}_{ii} > 0$ and $\mathbf{L}_{ij} \leq 0 (i \neq j)$, we can get $|\mathbf{L}_{ii}| = \sum_{i \neq j} |\mathbf{L}_{ij}|$. According to Gerschgorin Circle Theorem, we can deduce that all eigenvalues of $\mathbf{L}$ is greater than or equal to zero, i.e. $\mathbf{L}$ is p.s.d. □

To rewrite the problem in a more convenient form, we further define the following notations:

- $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_n]^T$, where $\mathbf{e}_i \in \{0, 1\}^c$ is a binary vector, and $\mathbf{E}_{ij} = 1$ iff the i-th point belongs to subset V_j.
- $\boldsymbol{\Pi} = \text{diag}(\pi_1, \pi_2, ..., \pi_n)$

Then we have

$$\mathbf{E}^T\mathbf{E} = \begin{pmatrix} n_1 & & \\ & \ddots & \\ & & n_c \end{pmatrix}$$

$$\mathbf{E}^T\boldsymbol{\Pi}\mathbf{E} = \begin{pmatrix} \sum_{i \in V_1} \pi_i & & \\ & \ddots & \\ & & \sum_{i \in V_c} \pi_i \end{pmatrix}$$

$$\text{PCUT} = \text{tr}(\mathbf{E}^T\mathbf{L}\mathbf{E}(\mathbf{E}^T\boldsymbol{\Pi}\mathbf{E})^{-1})$$

And our goal is to solve the optimization problem

$$\min_{\mathbf{E}} \text{tr}(\mathbf{E}^T\mathbf{L}\mathbf{E}(\mathbf{E}^T\boldsymbol{\Pi}\mathbf{E})^{-1})$$

In practice, we would like to require the sets $V_1, V_2, ...V_C$ are "reasonably large". Two most common objective functions to encode this are RatioCut and the normalized cut(Ncut). In RationCut, we have$\boldsymbol{\Pi} = \mathbf{I}_n$, while in Ncut, we have $\boldsymbol{\Pi} = \mathbf{D}$.

9.2.3 Relaxation Approach

The mincut problem discribed above, unfortunately, is NP hard. Therefore, we introduce Spectral clustering to solve the relaxed version of those problems.
We relax $\mathbf{E}_{n\times c}$ into a real matrix $\mathbf{Y}_{n\times(c-1)}$, and we have the following proposition:

Proposition 9.2. *Let* $\mathbf{Y}$ *be an* $n \times (c-1)$ *and real matrix such that*

(a) *the columns of* $\mathbf{Y}$ *are piecewise constant with respect to* $\mathbf{E}$*, i.e.* $\mathbf{Y} = \mathbf{E}_{n\times c}\mathbf{\Psi}_{c\times(c-1)}$

(b) $\mathbf{Y}^T\mathbf{\Pi}\mathbf{Y} = \mathbf{I}_{c-1}$

(c) $\mathbf{Y}^T\mathbf{\Pi}\mathbf{1}_n = 0$

Then, $\text{PCUT} = \text{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})$.

With this proposition, we can minimize PCUT by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{Y}} \text{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) \\ s.t. \quad &\mathbf{Y}^T\mathbf{\Pi}\mathbf{Y} = \mathbf{I}_{c-1} \\ &\mathbf{Y}^T\mathbf{\Pi}\mathbf{1}_n = 0 \end{aligned}$$

Now we prove Proposition 8.2.

Proof. Since $\mathbf{Y}^T\mathbf{\Pi}\mathbf{Y} = \mathbf{I}_{c-1}$, we can rewrite the equation as $\mathbf{Y}^T\mathbf{\Pi}^{\frac{1}{2}}\mathbf{\Pi}^{\frac{1}{2}}\mathbf{Y} = \mathbf{I}_{c-1}$. Assume that $\mathbf{Y}_0 = \mathbf{\Pi}^{\frac{1}{2}}\mathbf{Y}$, we get $\mathbf{Y}_0^T\mathbf{Y}_0 = \mathbf{I}_{c-1}$.
Let $\mathbf{\Psi}_0 = [\mathbf{\Psi}, \alpha\mathbf{I}_c]$ and $\mathbf{Z} = [\mathbf{Y}, \alpha\mathbf{\Pi}^{\frac{1}{2}}\mathbf{1}_n]$, where $\alpha = \frac{1}{\sqrt{\mathbf{1}_n^T\mathbf{\Pi}\mathbf{1}_n}}$, then

$$\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Z} = [\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Y}_0, \alpha\mathbf{1}_n] = [\mathbf{Y}, \alpha\mathbf{1}_n] = [\mathbf{E}\mathbf{\Psi}, \alpha\mathbf{E}\mathbf{1}_c] = E[\mathbf{\Psi}, \alpha\mathbf{I}_c] = \mathbf{E}\mathbf{\Psi}_0$$

Hence, we have

$$\mathbf{Z} = \mathbf{\Pi}^{\frac{1}{2}}\mathbf{E}\mathbf{\Psi}_0$$

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{\Psi}_0^T\mathbf{E}^T{\mathbf{\Pi}^{\frac{1}{2}}}^T\mathbf{\Pi}^{\frac{1}{2}}\mathbf{E}\mathbf{\Psi}_0 = \mathbf{\Psi}_0^T\mathbf{E}^T\mathbf{\Pi}\mathbf{E}\mathbf{\Psi}_0$$

What's more,

$$\mathbf{Z}^T\mathbf{Z} = \begin{bmatrix} \mathbf{Y}_0^T \\ \alpha\mathbf{1}_n^T\mathbf{\Pi}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_0 & \alpha\mathbf{\Pi}^{\frac{1}{2}}\mathbf{1}_n^T \end{bmatrix} = \mathbf{I}_c$$

Thus,

$$\mathbf{\Psi}_0^T\mathbf{E}^T\mathbf{\Pi}\mathbf{E}\mathbf{\Psi}_0 = \mathbf{I}_c$$

$$\mathbf{E}^T\mathbf{\Pi}\mathbf{E} = \mathbf{\Psi}_0^{-T}\mathbf{\Psi}_0^{-1} = (\mathbf{\Psi}_0\mathbf{\Psi}_0^T)^{-1}$$

$$\mathbf{\Psi}_0\mathbf{\Psi}_0^T = (\mathbf{E}^T\mathbf{\Pi}\mathbf{E})^{-1}$$

Since $\mathbf{Z}\mathbf{Z}^T = \mathbf{Y}_0\mathbf{Y}_0^T + \alpha^2\mathbf{\Pi}^{\frac{1}{2}}\mathbf{1}_n\mathbf{1}_n^T\mathbf{\Pi}^{\frac{1}{2}}$, we have

$$\begin{aligned}
\mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) &= \mathrm{tr}(\mathbf{Y}_0^T\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Y}_0) \\
&= \mathrm{tr}(\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Y}_0\mathbf{Y}_0^T) \\
&= \mathrm{tr}(\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}(\mathbf{Z}\mathbf{Z}^T - \alpha^2\mathbf{\Pi}^{\frac{1}{2}}\mathbf{1}_n\mathbf{1}_n^T\mathbf{\Pi}^{\frac{1}{2}})) \\
&= \mathrm{tr}(\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Z}\mathbf{Z}^T) - \mathrm{tr}(\alpha^2\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{1}_n\mathbf{1}_n^T\mathbf{\Pi}^{\frac{1}{2}}) \\
&= \mathrm{tr}(\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Z}\mathbf{Z}^T) \qquad \text{(using proposition 8.1)} \\
&= \mathrm{tr}(\mathbf{Z}^T\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Z}) \\
&= \mathrm{tr}(\mathbf{\Psi}_0^T\mathbf{E}^T\mathbf{L}\mathbf{E}\mathbf{\Psi}_0) \\
&= \mathrm{tr}(\mathbf{E}^T\mathbf{L}\mathbf{E}\mathbf{\Psi}_0\mathbf{\Psi}_0^T) \\
&= \mathrm{tr}(\mathbf{E}^T\mathbf{L}\mathbf{E}(\mathbf{E}^T\mathbf{\Pi}\mathbf{E})^{-1})
\end{aligned}$$

Hence,

$$\mathrm{PCUT} = \mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})$$

□

Now our goal is to solve the following optimization problem:

$$\begin{aligned}
&\min_{\mathbf{Y}_0} \mathrm{tr}(\mathbf{Y}_0^T\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Y}_0) \\
s.t. \quad &\mathbf{Y}_0^T\mathbf{Y}_0 = \mathbf{I}_{c-1} \text{ and } \mathbf{Y}_0^T\mathbf{\Pi}^{\frac{1}{2}}\mathbf{1}_n = 0
\end{aligned} \tag{$*$}$$

We can use the following theorem to solve this optimization problem.

Theorem 9.1. *Suppose that* $\mathbf{L}$ *is a real symmetric matrix such that* $\mathbf{L}\mathbf{1}_n = 0$ *and suppose that the diagonal elements of* $\mathbf{\Pi}$ *are positive. Let* $\boldsymbol{\mu}_1 = \alpha\mathbf{\Pi}^{\frac{1}{2}}\mathbf{1}_n$ *be the eigenvector associated with the eigenvalue* $\gamma_1 = 0$ *of* $\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}$*,where* $\alpha^2 = \frac{1}{\mathbf{1}_n^T\mathbf{\Pi}\mathbf{1}_n}$*. Let the remaining eigenvalues of* $\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}$ *be assigned so that* $\gamma_2 \leq ... \leq \gamma_n$*, and let the corresponding orthogonal eigenvectors be denoted by* $\boldsymbol{\mu}_i$*,* $i = 2, 3, \cdots, n$*. Then the solution of* $(*)$ *is* $\hat{\mathbf{Y}}_0 = \mathbf{U}\mathbf{Q}$ *where* $\mathbf{U} = [\boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_c]$ *and* $\mathbf{Q}$ *is an arbitrary* $(c-1) \times (c-1)$ *orthonormal matrix with* $\min \mathrm{tr}(\mathbf{Y}_0^T\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{Y}_0) = \prod_{i=2}^{c}\gamma_i$*. Furthermore, if* $\gamma_c < \gamma_{c+1}$*, then* $\hat{\mathbf{Y}}_0$ *is a strict local minimization of the problem.*

Especially,in the Ncut case, we have $\mathbf{\Pi} = \mathbf{D}$. Then

$$\mathbf{\Pi}^{-\frac{1}{2}}\mathbf{L}\mathbf{\Pi}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$$

As the smallest k eigenvectors of $\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ is the same as the largest k eigenvectors of $\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, we only need to perform eigenvalue decomposition on $\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$. In addition, note that $\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ is similar to $\mathbf{D}^{-1}\mathbf{W}$, and $\mathbf{D}^{-1}\mathbf{W}$ is the transition matrix of the random walk because the sum of each row is 1. Thus we can also view the clustering problem from the random walk aspect.