Complexity of Low-Rank Matrix Approximations with Weights or Missing Data

Nicolas Gillis

University of Waterloo
ngillis@uwaterloo.ca
sites.google.com/site/nicolasgillis/

in collaboration with François Glineur
Université catholique de Louvain

13th Midwest Optimization Meeting & Workshop on Large Scale Optimization and Applications

Low-Rank Matrix Approximaton

We are interested in approximating a given m-by-n matrix M with the product of two matrices U and V:

$$M \approx UV = X,$$

where U has dimension m-by-$\mathbf{r}$ and V has dimension $\mathbf{r}$-by-n

Equivalently, X has dimension m-by-n and $\operatorname{rank}(X) \leq \mathbf{r}$.

If each column of M represents an element of a dataset, we have that

$$M(:, i) \approx \sum_{k=1}^{\mathbf{r}} U(:, k)\, V(k, i) \qquad \text{for all } i,$$

i.e., each column of M is reconstructed through a linear combination of the columns of U.

Low-Rank Matrix Approximaton

We are interested in approximating a given m-by-n matrix M with the product of two matrices U and V:

$$M \;\approx\; UV \;=\; X,$$

where U has dimension m-by-**r** and V has dimension **r**-by-n
Equivalently, X has dimension m-by-n and $\mathrm{rank}(X) \leq \mathbf{r}$.

If each column of M represents an element of a dataset, we have that

$$M(:,i) \approx \sum_{k=1}^{\mathbf{r}} U(:,k)\, V(k,i) \qquad \text{for all } i,$$

i.e., each column of M is reconstructed through a linear combination of the columns of U.

Low-Rank Matrix Approximaton

We are interested in approximating a given m-by-n matrix M with the product of two matrices U and V:

$$M \approx UV = X,$$

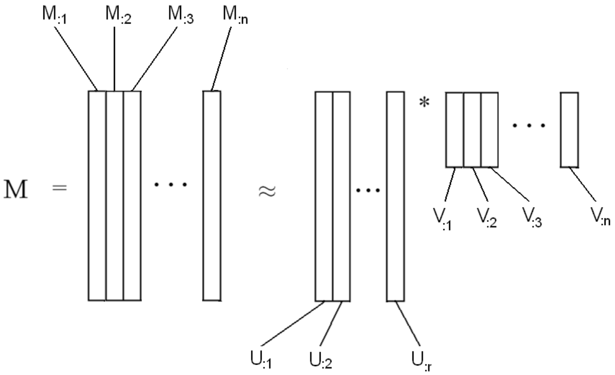
where U has dimension m-by-**r** and V has dimension **r**-by-n
Equivalently, X has dimension m-by-n and $\operatorname{rank}(X) \leq$ **r**.

If each column of M represents an element of a dataset, we have that

$$M(:,i) \approx \sum_{k=1}^{\mathbf{r}} U(:,k)\; V(k,i) \qquad \text{for all } i,$$

i.e., each column of M is reconstructed through a linear combination of the columns of U.

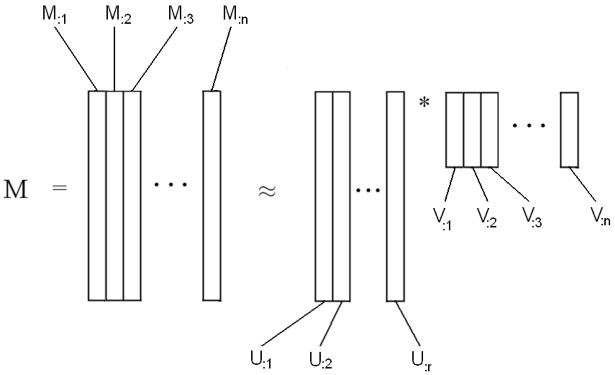
Linear dimensionality reduction



Matrix factorization allows to represent the columns of M in a smaller dimensional space defined by the columns of U. Coordinates of the columns of M in this space are given by the columns of V.

$\longrightarrow$ compression ($mn \gg mr + nr$), visualization, interpretation.

Linear dimensionality reduction



Matrix factorization allows to represent the columns of M in a smaller dimensional space defined by the columns of U. Coordinates of the columns of M in this space are given by the columns of V.
$\longrightarrow$ compression ($mn \gg mr + nr$), visualization, interpretation.

Matrix approximation and optimization

Low-rank matrix approximation can be formulated as unconstrained optimization problems, e.g., minimizing the sum of squared errors

$$\min_{U,V} \; ||M - UV||_F^2 = \sum_{ij} (M - UV)_{ij}^2.$$

This is a well-known problem with nice properties (e.g., all local minima are global) and which can be solved efficiently.

In particular, truncating the singular value decomposition (SVD):

$$M \; = \; U \Sigma V^T \; = \; \sum_{i=1}^{\mathrm{rk}(M)} \sigma_i U_{:i} V_{:i}^T, \quad U^T U = I_m, V^T V = I_n,$$

gives an optimal rank-$\mathbf{r}$ solution

$$M_k = \mathrm{argmin}_{X, \mathrm{rank}(X) \leq \mathbf{r}} \, ||M - X||_F^2 = \sum_{i=1}^{\mathbf{r}} \sigma_i U_{:i} V_{:i}^T.$$

Matrix approximation and optimization

Low-rank matrix approximation can be formulated as unconstrained optimization problems, e.g., minimizing the sum of squared errors

$$\min_{U,V} ||M - UV||_F^2 = \sum_{ij} (M - UV)_{ij}^2.$$

This is a well-known problem with nice properties (e.g., all local minima are global) and which can be solved efficiently.

In particular, truncating the singular value decomposition (SVD):

$$M \;=\; U \Sigma V^T \;=\; \sum_{i=1}^{\mathrm{rk}(M)} \sigma_i U_{:i} V_{:i}^T, \quad U^T U = I_m, V^T V = I_n,$$

gives an optimal rank-**r** solution

$$M_k = \mathrm{argmin}_{X, \mathrm{rank}(X) \leq \mathbf{r}} ||M - X||_F^2 = \sum_{i=1}^{\mathbf{r}} \sigma_i U_{:i} V_{:i}^T.$$

Missing data

In some cases, some entries are missing/unknown.

For example, we would like to predict how much someone is going to like a movie based on its movie preferences :

$$\begin{array}{cc} & \text{Movies} \\ \text{Users} & \begin{bmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{bmatrix} \end{array}$$

Huge potential in electronic commerce sites (movies, books, music, ...).
Good recommendations will increase the propensity of a purchase.

Missing data

In some cases, some entries are missing/unknown.

For example, we would like to predict how much someone is going to like a movie based on its movie preferences :

$$
\begin{array}{cc}
 & \text{Movies} \\
\text{Users} & \begin{bmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{bmatrix}
\end{array}
$$

Huge potential in electronic commerce sites (movies, books, music, ...).
Good recommendations will increase the propensity of a purchase.

Collaborative Filtering for Recommendation Systems

Objective. Automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating).

Method. The behavior of users is modeled using linear combinations of 'feature' users (related to age, sex, culture, etc.)

$$\underbrace{M(i,:)}_{\text{user } i} \approx \sum_{k=1}^{r} \underbrace{U(i,k)}_{\text{weights}} \underbrace{V(k,:)}_{\text{feature user } k}$$

Equivalently, movies ratings are modeled as linear combinations of 'feature' movies (related to different types - child oriented, serious vs. escapist, thriller, romantic, actors, etc.).

$$\underbrace{M(:,j)}_{\text{movie } j} \approx \sum_{k=1}^{r} \underbrace{U(:,k)}_{\text{type } k} \underbrace{V(k,j)}_{\text{weights}}$$

Collaborative Filtering for Recommendation Systems

Objective. Automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating).

Method. The behavior of users is modeled using linear combinations of 'feature' users (related to age, sex, culture, etc.)

$$\underbrace{M(i,:)}_{\text{user } i} \approx \sum_{k=1}^{r} \underbrace{U(i,k)}_{\text{weights}} \quad \underbrace{V(k,:)}_{\text{feature user } k}$$

Equivalently, movies ratings are modeled as linear combinations of 'feature' movies (related to different types - child oriented, serious vs. escapist, thriller, romantic, actors, etc.).

$$\underbrace{M(:,j)}_{\text{movie } j} \approx \sum_{k=1}^{r} \underbrace{U(:,k)}_{\text{type } k} \quad \underbrace{V(k,j)}_{\text{weights}}$$

Collaborative Filtering for Recommendation Systems

Objective. Automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating).

Method. The behavior of users is modeled using linear combinations of 'feature' users (related to age, sex, culture, etc.)

$$\underbrace{M(i,:)}_{\text{user } i} \approx \sum_{k=1}^{r} \underbrace{U(i,k)}_{\text{weights}} \quad \underbrace{V(k,:)}_{\text{feature user } k}$$

Equivalently, movies ratings are modeled as linear combinations of 'feature' movies (related to different types - child oriented, serious vs. escapist, thriller, romantic, actors, etc.).

$$\underbrace{M(:,j)}_{\text{movie } j} \approx \sum_{k=1}^{r} \underbrace{U(:,k)}_{\text{type } k} \quad \underbrace{V(k,j)}_{\text{weights}}$$

Example

$$M = \begin{pmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{pmatrix}$$

$$\approx \begin{pmatrix} 0.5 & 0.6 & -0.1 \\ 0.8 & -0.2 & -0.3 \\ 0.8 & -0.7 & 0.6 \\ -2 & 2.3 & 1.8 \\ -0.2 & 0.3 & 0.9 \\ 1 & -0.2 & -0.2 \end{pmatrix} \begin{pmatrix} 1.7 & 2.1 & 3.7 & 5 & 4.1 \\ 2.2 & 3.2 & 0.8 & 5 & 0.5 \\ 2 & 0.6 & 2.6 & 0.9 & 5 \end{pmatrix} = UV$$

$$= \begin{pmatrix} 2 & 2.9 & 2.1 & \mathbf{5.4} & \mathbf{1.9} \\ 0.3 & 0.9 & 2 & 2.7 & 1.7 \\ 1 & -0.2 & 4 & 1 & 5.9 \\ 5.3 & 4.2 & -0.9 & 3.1 & 2 \\ 2.1 & 1.1 & 1.8 & 1.3 & 2.8 \\ 0.9 & 1.3 & 3 & 3.8 & 3 \end{pmatrix}$$

Example

$$M = \begin{pmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{pmatrix}$$

$$\approx \begin{pmatrix} 0.5 & 0.6 & -0.1 \\ 0.8 & -0.2 & -0.3 \\ 0.8 & -0.7 & 0.6 \\ -2 & 2.3 & 1.8 \\ -0.2 & 0.3 & 0.9 \\ 1 & -0.2 & -0.2 \end{pmatrix} \begin{pmatrix} 1.7 & 2.1 & 3.7 & 5 & 4.1 \\ 2.2 & 3.2 & 0.8 & 5 & 0.5 \\ 2 & 0.6 & 2.6 & 0.9 & 5 \end{pmatrix} = UV$$

$$= \begin{pmatrix} 2 & 2.9 & 2.1 & 5.4 & 1.9 \\ 0.3 & 0.9 & 2 & 2.7 & 1.7 \\ 1 & -0.2 & 4 & 1 & 5.9 \\ 5.3 & 4.2 & -0.9 & 3.1 & 2 \\ 2.1 & 1.1 & 1.8 & 1.3 & 2.8 \\ 0.9 & 1.3 & 3 & 3.8 & 3 \end{pmatrix}$$

Example

$$M = \begin{pmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{pmatrix}$$

$$\approx \begin{pmatrix} 0.5 & 0.6 & -0.1 \\ 0.8 & -0.2 & -0.3 \\ 0.8 & -0.7 & 0.6 \\ -2 & 2.3 & 1.8 \\ -0.2 & 0.3 & 0.9 \\ 1 & -0.2 & -0.2 \end{pmatrix} \begin{pmatrix} 1.7 & 2.1 & 3.7 & 5 & 4.1 \\ 2.2 & 3.2 & 0.8 & 5 & 0.5 \\ 2 & 0.6 & 2.6 & 0.9 & 5 \end{pmatrix} = UV$$

$$= \begin{pmatrix} 2 & 2.9 & 2.1 & \mathbf{5.4} & \mathbf{1.9} \\ 0.3 & 0.9 & 2 & 2.7 & 1.7 \\ 1 & -0.2 & 4 & 1 & 5.9 \\ 5.3 & 4.2 & -0.9 & 3.1 & 2 \\ 2.1 & 1.1 & 1.8 & 1.3 & 2.8 \\ 0.9 & 1.3 & 3 & 3.8 & 3 \end{pmatrix}$$

For example, using a rank-2 factorization on the **Netflix dataset**, female vs. male and serious vs. escapist behaviors were extracted.

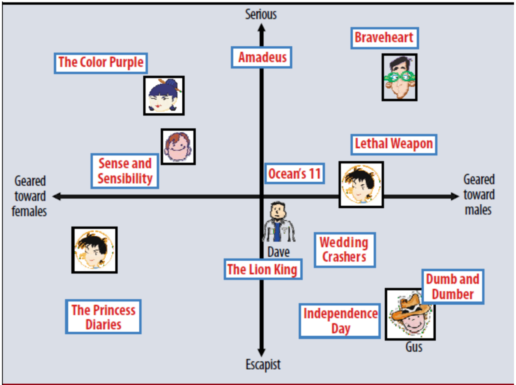


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Koren, Bell, Volinsky, *Matrix Factorization Techniques for Recommender Systems, 2009*. Winners of the Netflix prize 1,000,000$.

Weighting

In other applications, it might be necessary to give different importances for each entry of the data matrix, e.g.,

- ◇ when the number of samples and/or the expected variance vary among the data;
- ◇ when one wants to emphasize a localized part of the data;
- ◇ . . .

[LPW97] Lu, Pei and Wang, *Weighted low-rank approximation of general complex matrices and its application in the design of 2-D digital filters*, IEEE Trans. Circuits Syst. I, Vol. 44, pp. 650–655, 1997.
[HVB07] Ho, Van Dooren and Blondel, *Weighted Nonnegative Matrix Factorization and Face Feature Extraction*, 2007.

Weighted Low-Rank Approximation (WLRA)

Giving different importances to the entries of M, we obtain the following optimization problem

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{r \times n}} ||M - UV||_W^2 = \sum_{ij} W_{ij}(M - UV)_{ij}^2,$$

where $W \geq 0$ is the weighting matrix. For missing data, $W_{ij} = 0$.

What can we say about this optimization problem? When can we expect to solve it up to global optimality? Is this a difficult problem in general?

Weighted Low-Rank Approximation (WLRA)

Giving different importances to the entries of M, we obtain the following optimization problem

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{r \times n}} ||M - UV||_W^2 = \sum_{ij} W_{ij}(M - UV)_{ij}^2,$$

where $W \geq 0$ is the weighting matrix. For missing data, $W_{ij} = 0$.

What can we say about this optimization problem? When can we expect to solve it up to global optimality? Is this a difficult problem in general?

Special case: $rank(W) = 1$

If the weight matrix W is rank-one, i.e., $W = xy^T \geq 0$,

$$\begin{aligned} ||M - UV^T||_W^2 &= \sum_{i,j} x_i y_j \, (M - UV^T)_{ij}^2 \\ &= \sum_{i,j} \Big(\underbrace{(\sqrt{W} \circ M)_{ij}}_{M'} - \underbrace{(\sqrt{x_i}\, U_{i:})}_{U'} \underbrace{(\sqrt{y_j}\, V_{j:}^T)}_{V'} \Big)^2, \end{aligned}$$

where $\circ$ is the component-wise product.

WLRA can be recovered from the SVD decomposition of $(\sqrt{W} \circ M)$.

As far as we know, none other (nontrivial) complexity result is known.

Special case: $rank(W) = 1$

If the weight matrix W is rank-one, i.e., $W = xy^T \geq 0$,

$$\begin{aligned} ||M - UV^T||_W^2 &= \sum_{i,j} x_i y_j \, (M - UV^T)_{ij}^2 \\ &= \sum_{i,j} \Big(\underbrace{(\sqrt{W} \circ M)_{ij}}_{M'} - \underbrace{(\sqrt{x_i}\, U_{i:})}_{U'} \underbrace{(\sqrt{y_j}\, V_{j:}^T)}_{V'} \Big)^2, \end{aligned}$$

where $\circ$ is the component-wise product.

WLRA can be recovered from the SVD decomposition of $(\sqrt{W} \circ M)$.

As far as we know, none other (nontrivial) complexity result is known.

Complexity of rank-one WLRA

Let consider the *simplest* case : $r = 1$.

$$\min_{u \in \mathbb{R}^m, v \in \mathbb{R}^n} ||M - uv^T||_W^2 = \sum_{ij} W_{ij}(M_{ij} - u_i v_j)^2.$$

Is the problem difficult?

Example.

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \text{and} \quad W = \begin{pmatrix} 1 & 100 & 2 \\ 100 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}.$$

Imposing $||u||_2 = 1$, WLRA has two degrees of freedom left :

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \sqrt{1 - u_1^2 - u_2^2} \end{pmatrix}, \; v^* = \text{argmin}_v \, ||M - uv^T||_W \text{ (LS)}.$$

Complexity of rank-one WLRA

Let consider the *simplest* case : $r = 1$.

$$\min_{u\in\mathbb{R}^m, v\in\mathbb{R}^n} ||M - uv^T||_W^2 = \sum_{ij} W_{ij}(M_{ij} - u_i v_j)^2.$$

Is the problem difficult?

Example.

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \text{and} \quad W = \begin{pmatrix} 1 & 100 & 2 \\ 100 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}.$$

Imposing $||u||_2 = 1$, WLRA has two degrees of freedom left :

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \sqrt{1 - u_1^2 - u_2^2} \end{pmatrix}, \; v^* = \operatorname{argmin}_v ||M - uv^T||_W \text{ (LS)}.$$

Complexity of rank-one WLRA

Let consider the *simplest* case : $r = 1$.

$$\min_{u \in \mathbb{R}^m, v \in \mathbb{R}^n} ||M - uv^T||_W^2 = \sum_{ij} W_{ij}(M_{ij} - u_i v_j)^2.$$

Is the problem difficult?

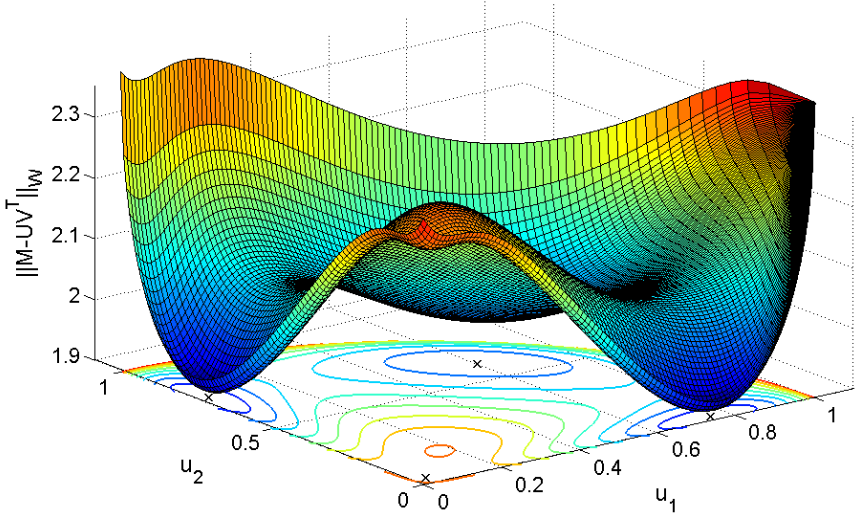
Example.

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \text{and} \quad W = \begin{pmatrix} 1 & 100 & 2 \\ 100 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}.$$

Imposing $||u||_2 = 1$, WLRA has two degrees of freedom left :

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \sqrt{1 - u_1^2 - u_2^2} \end{pmatrix}, \; v^* = \operatorname{argmin}_v ||M - uv^T||_W \text{ (LS)}.$$

Local minima of rank-one WLRA



Complexity of WLRA

In order to prove its *NP-hardness*, we use a reduction from the maximum-edge biclique problem (MBP):

Given a bipartite graph $G_b = (V_1 \cup V_2, E \in (V_1 \times V_2))$,
Find the maximum-edge complete bipartite subgraph (biclique).

Applications: text mining, web community discovery, collaborative filtering

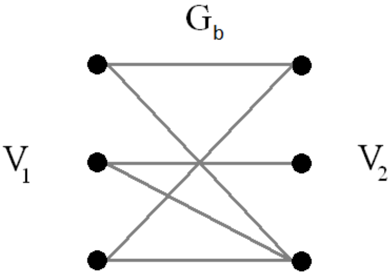
[Peet03] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3): 651-654, 2003.

Complexity of WLRA

In order to prove its *NP-hardness*, we use a reduction from the maximum-edge biclique problem (MBP):

Given a bipartite graph $G_b = (V_1 \cup V_2, E \in (V_1 \times V_2))$,

Find the maximum-edge complete bipartite subgraph (biclique).



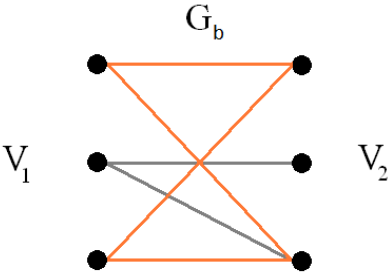
Applications: text mining, web community discovery, collaborative filtering

[Peet03] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3): 651-654, 2003.

Complexity of WLRA

In order to prove its *NP-hardness*, we use a reduction from the maximum-edge biclique problem (MBP):

Given a bipartite graph $G_b = (V_1 \cup V_2, E \in (V_1 \times V_2))$,
Find the maximum-edge complete bipartite subgraph (biclique).



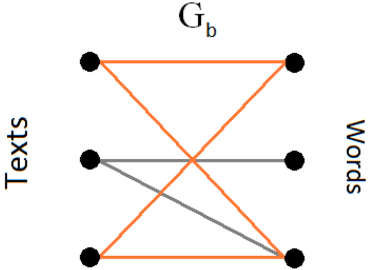
Applications: text mining, web community discovery, collaborative filtering

[Peet03] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3): 651-654, 2003.

Complexity of WLRA

In order to prove its *NP-hardness*, we use a reduction from the maximum-edge biclique problem (MBP):

Given a bipartite graph $G_b = (V_1 \cup V_2, E \in (V_1 \times V_2))$,
Find the maximum-edge complete bipartite subgraph (biclique).



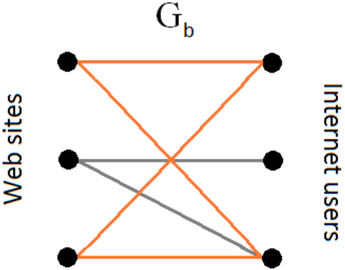
Applications: text mining, web community discovery, collaborative filtering

[Peet03] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3): 651-654, 2003.

Complexity of WLRA

In order to prove its *NP-hardness*, we use a reduction from the maximum-edge biclique problem (MBP):

Given a bipartite graph $G_b = (V_1 \cup V_2, E \in (V_1 \times V_2))$,
Find the maximum-edge complete bipartite subgraph (biclique).



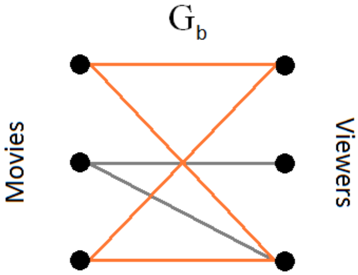
Applications: text mining, web community discovery, collaborative filtering

[Peet03] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3): 651-654, 2003.

Complexity of WLRA

In order to prove its *NP-hardness*, we use a reduction from the maximum-edge biclique problem (MBP):

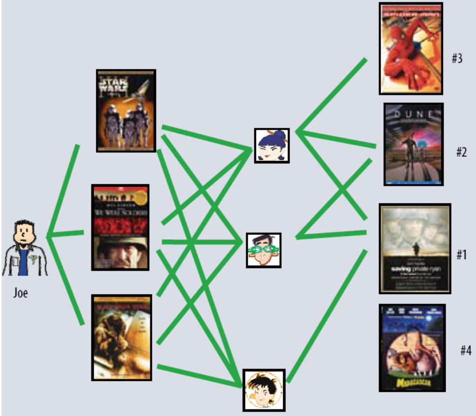
Given a bipartite graph $G_b = (V_1 \cup V_2, E \in (V_1 \times V_2))$,
Find the maximum-edge complete bipartite subgraph (biclique).



Applications: text mining, web community discovery, collaborative filtering

[Peet03] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3): 651-654, 2003.

Link with Neighborhood Methods...

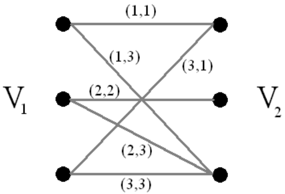


To make a prediction for Joe, the system finds similar users who also like the movies he likes, and then determines which other movies they liked. In this case, all three liked Saving Private Ryan, so that is the first recommendation. Two of them liked Dune, so that is next, and so on.

Reduction to WLRA

Let $M \in \{0,1\}^{m \times n}$ be the biadjacency matrix of the graph G,

$$\begin{array}{cc} & V_2 \\ V_1 & \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \quad \equiv$$



With (u, v) binary variables to indicate which vertices belong to the solution

$$u = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T, \quad v = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T,$$

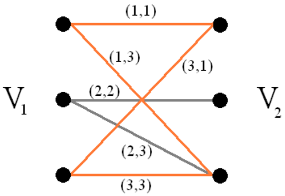
$$uv^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

bicliques of G can be represented as binary rank-one matrices.

Reduction to WLRA

Let $M \in \{0,1\}^{m \times n}$ be the biadjacency matrix of the graph G,

$$\begin{array}{cc} & V_2 \\ V_1 & \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \quad \equiv$$



With (u, v) binary variables to indicate which vertices belong to the solution

$$u = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T, \quad v = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T,$$

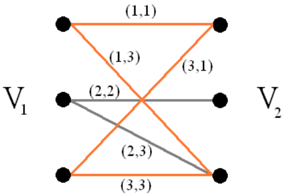
$$uv^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

bicliques of G can be represented as binary rank-one matrices.

Reduction to WLRA

Let $M \in \{0,1\}^{m \times n}$ be the biadjacency matrix of the graph G,

$$\begin{array}{cc} & V_2 \\ V_1 & \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \equiv$$



With (u, v) binary variables to indicate which vertices belong to the solution

$$u = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T, \quad v = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T,$$

$$uv^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

bicliques of G can be represented as binary rank-one matrices.

Reduction to WLRA

Let $M \in \{0,1\}^{m \times n}$ be the biadjacency matrix of the graph G, then

$$\max_{u \in \{0,1\}^m, v \in \{0,1\}^n} \quad \sum_{ij} (uv^T)_{ij}$$

$$(uv^T)_{ij} = 0 \text{ for } i,j \text{ such that } M_{ij} = 0,$$

is an exact formulation of the biclique problem. Noting that

$$\sum_{ij} (uv^T)_{ij} = \sum_{ij} (uv^T)_{ij}^2 = \sum_{ij} M_{ij} (uv^T)_{ij} = 2 \sum_{ij} M_{ij} (uv^T)_{ij} - (uv)_{ij}^2$$

and since M is binary, we have the equivalence with

$$\min_{u \in \mathbb{R}^m, v \in \mathbb{R}^n} \quad ||M||_F^2 - 2 \sum_{ij} M_{ij} (uv^T)_{ij} + (uv)_{ij}^2 = ||M - uv^T||_F^2$$

$$(uv^T)_{ij} = 0 \text{ for } i,j \text{ such that } M_{ij} = 0.$$

Hence, the biclique problem is equivalent to finding the best rank-one approximation of M, where zeros of M must be approximated by zeros.

Reduction to WLRA

Let $M \in \{0,1\}^{m\times n}$ be the biadjacency matrix of the graph G, then

$$\max_{u\in\{0,1\}^m, v\in\{0,1\}^n} \quad \sum_{ij} (uv^T)_{ij}$$

$$(uv^T)_{ij} = 0 \text{ for } i,j \text{ such that } M_{ij} = 0,$$

is an exact formulation of the biclique problem. Noting that

$$\sum_{ij}(uv^T)_{ij} = \sum_{ij}(uv^T)^2_{ij} = \sum_{ij} M_{ij}(uv^T)_{ij} = 2\sum_{ij} M_{ij}(uv^T)_{ij} - (uv)^2_{ij}$$

and since M is binary, we have the equivalence with

$$\min_{u\in\mathbb{R}^m, v\in\mathbb{R}^n} \quad ||M||_F^2 - 2\sum_{ij} M_{ij}(uv^T)_{ij} + (uv)^2_{ij} = ||M - uv^T||_F^2$$

$$(uv^T)_{ij} = 0 \text{ for } i,j \text{ such that } M_{ij} = 0.$$

Hence, the biclique problem is equivalent to finding the best rank-one approximation of M, where zeros of M must be approximated by zeros.

Reduction to WLRA

Let $M \in \{0,1\}^{m\times n}$ be the biadjacency matrix of the graph G, then

$$\max_{u\in\{0,1\}^m, v\in\{0,1\}^n} \quad \sum_{ij} (uv^T)_{ij}$$

$$(uv^T)_{ij} = 0 \text{ for } i,j \text{ such that } M_{ij} = 0,$$

is an exact formulation of the biclique problem. Noting that

$$\sum_{ij} (uv^T)_{ij} = \sum_{ij} (uv^T)^2_{ij} = \sum_{ij} M_{ij}(uv^T)_{ij} = 2\sum_{ij} M_{ij}(uv^T)_{ij} - (uv)^2_{ij}$$

and since M is binary, we have the equivalence with

$$\min_{u\in\mathbb{R}^m, v\in\mathbb{R}^n} \quad ||M||_F^2 - 2\sum_{ij} M_{ij}(uv^T)_{ij} + (uv)^2_{ij} = ||M - uv^T||_F^2$$

$$(uv^T)_{ij} = 0 \text{ for } i,j \text{ such that } M_{ij} = 0.$$

Hence, the biclique problem is equivalent to finding the best rank-one approximation of M, where zeros of M must be approximated by zeros.

Reduction to WLRA

Because zeros have to be approximated by zeros, we give them more importance using a weight $d \gg 1$:

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad W = \begin{pmatrix} 1 & d & 1 \\ d & 1 & 1 \\ 1 & d & 1 \end{pmatrix},$$

and the corresponding rank-one WLRA problem is:

$$\min_{u \in \mathbb{R}^m, v \in \mathbb{R}^n} ||M - uv^T||_W^2.$$

Theorem. *For $d \geq (2|E|)^6$, rounding optimal solutions of rank-one WLRA generate optimal solutions of the biclique problem.*

Corollary. *Weighted low-rank approximation is NP-hard.*

Reduction to WLRA

Because zeros have to be approximated by zeros, we give them more importance using a weight $d \gg 1$:

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad W = \begin{pmatrix} 1 & d & 1 \\ d & 1 & 1 \\ 1 & d & 1 \end{pmatrix},$$

and the corresponding rank-one WLRA problem is:

$$\min_{u \in \mathbb{R}^m, v \in \mathbb{R}^n} ||M - uv^T||_W^2.$$

Theorem. *For $d \geq (2|E|)^6$, rounding optimal solutions of rank-one WLRA generate optimal solutions of the biclique problem.*

Corollary. *Weighted low-rank approximation is NP-hard.*

Missing Data (i.e., W binary)

The construction works as follows:

$$M = \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & d \end{array}\right) \quad \text{and} \quad W = \left(\begin{array}{ccc|cc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

Theorem. *For $d \geq (2|E|)^4$, rounding optimal solutions of rank-one WLRA generate optimal solutions of the biclique problem.*

Corollary. *Low-rank matrix approximation with missing data is NP-hard.*

Missing Data (i.e., W binary)

The construction works as follows:

$$M = \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & d \end{array}\right) \text{ and } W = \left(\begin{array}{ccc|cc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

Theorem. *For $d \geq (2|E|)^4$, rounding optimal solutions of rank-one WLRA generate optimal solutions of the biclique problem.*

Corollary. *Low-rank matrix approximation with missing data is NP-hard.*

Missing Data (i.e., W binary)

The construction works as follows:

$$M = \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & d \end{array}\right) \quad \text{and} \quad W = \left(\begin{array}{ccc|cc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

Theorem. *For $d \geq (2|E|)^4$, rounding optimal solutions of rank-one WLRA generate optimal solutions of the biclique problem.*

Corollary. *Low-rank matrix approximation with missing data is NP-hard.*

Missing Data (i.e., W binary)

The construction works as follows:

$$M = \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & d \end{array}\right) \text{ and } W = \left(\begin{array}{ccc|cc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

Theorem. *For $d \geq (2|E|)^4$, rounding optimal solutions of rank-one WLRA generate optimal solutions of the biclique problem.*

Corollary. *Low-rank matrix approximation with missing data is NP-hard.*

Complexity of Weighted Low-Rank Approximations

$$\min_{U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{r\times n}} \quad ||M - UV||_W^2 = \sum_{ij} W_{ij}(M - UV)_{ij}^2$$

(WLRA)

- If rank(W) $= 1$, solvable in polynomial time
- If rank(W) is free, even the rank-one problem is NP-hard
- Open questions:
  - Complexity for rank(W) fixed (e.g., rank(W)$= 2$)?
  - Approximability results (i.e., up to a multiplicative constant factor)?
  - Complexity given additional assumptions on the data matrix? For example, in some cases (sufficiently numerous entries, well-distributed, low level of noise), the original uncorrupted low-rank matrix can be recovered accurately, with a technique based on convex optimization (nuclear norm minimization).
    Candès, Plan, *Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements*, arXiv:1001.0339v1.

Complexity of Weighted Low-Rank Approximations

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{r \times n}} \quad ||M - UV||_W^2 = \sum_{ij} W_{ij}(M - UV)_{ij}^2$$

(WLRA)

- If rank(W) $= 1$, solvable in polynomial time
- If rank(W) is free, even the rank-one problem is NP-hard
- Open questions:
  - Complexity for rank(W) fixed (e.g., rank(W)$= 2$)?
  - Approximability results (i.e., up to a multiplicative constant factor)?
  - Complexity given additional assumptions on the data matrix? For example, in some cases (sufficiently numerous entries, well-distributed, low level of noise), the original uncorrupted low-rank matrix can be recovered accurately, with a technique based on convex optimization (nuclear norm minimization).
    Candès, Plan, *Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements*, arXiv:1001.0339v1.

Complexity of Weighted Low-Rank Approximations

$$\min_{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{r\times n}} \quad ||M - UV||_W^2 = \sum_{ij} W_{ij}(M-UV)_{ij}^2 \qquad \text{(WLRA)}$$

- If rank$(W) = 1$, solvable in polynomial time
- If rank(W) is free, even the rank-one problem is NP-hard
- Open questions:
  - Complexity for rank(W) fixed (e.g., rank$(W)= 2$)?
  - Approximability results (i.e., up to a multiplicative constant factor)?
  - Complexity given additional assumptions on the data matrix?
    For example, in some cases (sufficiently numerous entries, well-distributed, low level of noise), the original uncorrupted low-rank matrix can be recovered accurately, with a technique based on convex optimization (nuclear norm minimization).
    Candès, Plan, *Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements*, arXiv:1001.0339v1.

Reference. G., Glineur, *Low-Rank Matrix Approximation with Weights or Missing Data is NP-hard*, to appear in SIAM J. Mat. Anal. Appl.

Talk and paper available on `sites.google.com/site/nicolasgillis/`

Thank you for your attention!