Trust Prediction using Temporal Dynamics

SMAI Project

Mentor: Dr. Naresh Manwani
TA mentor: Krishna chaitanya pappu

Social trust Relation

- In open society based applications, we assume there are dependencies between users.
- Any user’s preference depends on the user’s trust on other users.
- Initial attempts assumes trust relations to be static.
- In real scenario, trust relations are dynamic and change with time.

Social Trust Prediction Model

A trust prediction framework is proposed based on low rank matrix factorization as:

$$\min_{U,V} \| G - UVU^{\mathrm{T}} \|_F^2$$

where $G = \sum G_i$, $U \in R^{n*d}$ is the user preference matrix and d is the number of facets of user preference. $V \in R^{d*d}$ captures the more compact correlations among U, such as G(i,j) as $U(i,:)\ V\ U^T(j,:)$,

Temporal Weight Matrix Factorization in trust prediction

- Social trust relation depend on time distance between the current time and the time on which trust relation was established.
- These trust relations decay with time.
- The rates of decay may be different for different for different users.
- Prediction error is related to the time distance.
- In prediction error we want to incorporate the effect of time on trust relations.

Continued...

- Let $g_{i,j}^{t}$ be the first timestamp when u_i trusted u_j .
- Earlier trust relations reflect users' previous preferences and should have less influence on the current trust prediction .
- Exponential time function $e^{-\eta_i (m - g_{ij}^{t})}$ is added. 'm' is current time.

In the error term, we multiply the time decay factor, the error in prediction of a trust relation from user u_i to u_j be.

$$e^{-\eta_i (m - g_{ij}^{t})} \| \boldsymbol{G}(i, j) - \boldsymbol{U}(i,:)\boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}(j,:) \|_2^2$$

Update Rule Derivation:

$$L = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \alpha\|\mathbf{U}\|_F^2 + \beta\|\mathbf{V}\|_F^2 - Tr(\Lambda_1\mathbf{U}^T) - Tr(\Lambda_2\mathbf{V}^T).$$

We have the following KKT condition,

$$\begin{aligned} \Lambda_1 \circ \mathbf{U} &= \mathbf{0}, \\ \Lambda_2 \circ \mathbf{V} &= \mathbf{0}, \end{aligned}$$

where $\circ$ denotes the Hadamard product. We then have

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{U}} &= \frac{\partial Tr(\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T - 2\mathbf{X}^T\mathbf{U}\mathbf{V}^T) + \alpha Tr(\mathbf{U}^T\mathbf{U}) - Tr(\Lambda_1\mathbf{U}^T)}{\partial \mathbf{U}} \\ &= 2(\mathbf{U}\mathbf{V}^T\mathbf{V} - \mathbf{X}\mathbf{V} + \alpha\mathbf{U}) - \Lambda_1, \\ \frac{\partial L}{\partial \mathbf{V}} &= \frac{\partial Tr(\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T - 2\mathbf{X}^T\mathbf{U}\mathbf{V}^T) + \beta Tr(\mathbf{V}^T\mathbf{V}) - Tr(\Lambda_2\mathbf{V}^T)}{\partial \mathbf{V}} \\ &= 2(\mathbf{V}\mathbf{U}^T\mathbf{U} - \mathbf{X}^T\mathbf{U} + \beta\mathbf{V}) - \Lambda_2. \end{aligned}$$

Let $\frac{\partial L}{\partial \mathbf{U}} = 0$ and $\frac{\partial L}{\partial \mathbf{V}} = 0$ as another KKT condition, we have

$$\begin{aligned} \Lambda_1 &= 2(\mathbf{U}\mathbf{V}^T\mathbf{V} - \mathbf{X}\mathbf{V} + \alpha\mathbf{U}), \\ \Lambda_2 &= 2(\mathbf{V}\mathbf{U}^T\mathbf{U} - \mathbf{X}^T\mathbf{U} + \beta\mathbf{V}). \end{aligned}$$

Now we combine Eq. 8 and Eq. 10, we have

Continuation:

An **auxiliary function** $G(\mathbf{U},\mathbf{U}^t)$ of function $L(\mathbf{U})$ is a function that satisfies

$$G(\mathbf{U},\mathbf{U}) = L(\mathbf{U}),\ \ G(\mathbf{U},\mathbf{U}^t) \geq L(\mathbf{U}). \tag{13}$$

Then, if we take $\mathbf{U}^{t+1}$ such that

$$\mathbf{U}^{t+1} = \arg\min_{U} G(\mathbf{U},\mathbf{U}^t), \tag{14}$$

we have

$$L(\mathbf{U}^{t+1}) \leq G(\mathbf{U}^{t+1},\mathbf{U}^t) \leq G(\mathbf{U}^t,\mathbf{U}^t \leq L(\mathbf{U}^t)). \tag{15}$$

This proves that $L(U)$ is monotonically decreasing.

Step 1 - Finding an appropriate auxiliary function needs to take advantage of two inequalities,

$$z \geq 1 + log z,\ \forall z > 0, \tag{16}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{k}\frac{(\mathbf{AS'B})(i,j)\mathbf{S}(i,j)^2}{\mathbf{S'}(i,j)} \geq Tr(\mathbf{S}^T\mathbf{ASB}),$$
$$\forall \mathbf{A} \in \mathbb{R}_+^{m\times m}, \mathbf{B} \in \mathbb{R}_+^{k\times k}, \mathbf{S'} \in \mathbb{R}_+^{m\times k}, \mathbf{S} \in \mathbb{R}_+^{m\times k}. \tag{17}$$

The proof for Eq. 17 can be found in [5] (Proposition 6).

After removing irrelevant terms, the objective function Eq. 6 in terms of $\mathbf{U}$ can be written as

$$\begin{aligned}&Tr(\mathbf{VU}^T\mathbf{UV}^T - 2\mathbf{X}^T\mathbf{UV}^T) + \alpha Tr(\mathbf{U}^T\mathbf{U})\\ =&Tr(\mathbf{U}^T\mathbf{UV}^T\mathbf{V} - 2\mathbf{U}^T\mathbf{XV}) + \alpha Tr(\mathbf{U}^T\mathbf{U})\end{aligned} \tag{18}$$

We now propose an auxiliary function

$$
\begin{aligned}
G(\mathbf{U}, \mathbf{U}^t) = -2 \sum_{i,j} (\mathbf{XV})(i,j) \mathbf{U}^t(i,j)(1 + log \frac{\mathbf{U}(i,j)}{\mathbf{U}^t(i,j)}) \\
+ \sum_{i,j} \frac{(\mathbf{U}^t \mathbf{V}^T \mathbf{V})(i,j) \mathbf{U}(i,j)^2}{\mathbf{U}^t(i,j)} + \alpha \sum_{i,j} \frac{\mathbf{U}^t(i,j) \mathbf{U}(i,j)^2}{\mathbf{U}^t(i,j)}.
\end{aligned} \tag{19}
$$

Combining the two inequalities Eq. 16, 17, it is straightforward to see that Eq. 19 is a legal auxiliary function for Eq. 18, i.e., the two conditions in Eq. 13 are satisfied. Now we procceed to find $\mathbf{U}^{t+1}$ that satisfies condition Eq. 14.

Step 2 - Finding $\mathbf{U}^{t+1}$ can be achieved by obtaining the *global minima* of Eq. 19. First, we have

$$
\frac{\partial G(\mathbf{U}, \mathbf{U}^t)}{\partial \mathbf{U}(i,j)} = -2(\mathbf{XV})(i,j) \frac{\mathbf{U}^t(i,j)}{\mathbf{U}(i,j)} + 2 \frac{(\mathbf{U}^t \mathbf{V}^T \mathbf{V})(i,j) \mathbf{U}(i,j)}{\mathbf{U}^t(i,j)} + 2\alpha \mathbf{U}(i,j). \tag{20}
$$

Let $\frac{\partial G(\mathbf{U}, \mathbf{U}^t)}{\partial \mathbf{U}(i,j)} = 0$ we have

$$
(\mathbf{XV})(i,j) \frac{\mathbf{U}^t(i,j)}{\mathbf{U}^{t+1}(i,j)} = (\frac{(\mathbf{U}^t \mathbf{V}^T \mathbf{V})(i,j)}{\mathbf{U}^t(i,j)} + \alpha) \mathbf{U}^{t+1}(i,j), \tag{21}
$$

from which we directly have

$$
\mathbf{U}^{t+1}(i,j) = \mathbf{U}^t(i,j) \sqrt{\frac{(\mathbf{XV})(i,j)}{(\mathbf{U}^t \mathbf{V}^T \mathbf{V} + \alpha \mathbf{U}^t)(i,j)}}, \tag{22}
$$

$$\sum_i \sum_j e^{-n_i(m-g_{ij}^+)} \| G(i,j) - U(i,:) V U^T(j,:) \|^2$$

$$+ \alpha \sum_i \|n_i\|^2 + \beta \|U\|^2 + \gamma \|V\|^2$$

$U, V \geq 0$; $n_i \geq 0$.

$$= \| G(i,j) - U(i,:) V U^T(j,:) \|_2^2$$

$$= -\sum_j c_{ij} (m - g_{ij}^+) e^{-n_i(m-g_{ij}^+)} + 2\alpha n_i$$

$$2\alpha n_i + \sum c_{ij} g_{ij}^+ e^{-n_i(m-g_{ij}^+)}$$

$$- \sum_j c_{ij} m e^{-n_i(m-g_{ij}^+)}$$

$$\mathcal{L}(U) = Tr\left(G^T G + (U V^T U^T)(U V U^T) - 2 G^T U V U^T\right) + \beta Tr(U^T U)$$

$$\frac{\partial L}{\partial U} = U V^T U^T U (V + V^T) + U V U^T U (V^T + V) + 2\beta U - \frac{\partial [G^T U V U^T]}{\partial U}$$

$$= 2U \left[V^T U^T U V + V^T U^T U V^T + 2\beta I - G^T U V - G^T U V \right]$$

$$\Rightarrow U(i,k) \leftarrow U(i,k) \sqrt{\frac{a_i^T(k)}{[(U_{i,:}) A_i](k)}}$$

$$a_i = \sum_{j=1}^n b_{ij}^+ G(i,j) V U^T(j,:) + \sum_{j=1}^n b_{ji}^+ G(j,i) V^T U^T(j,:)$$

$$A_i = \sum_j b_{ij} V U^T(j,:) U(j,:) V^T + \sum_j b_{ji}^+ V^T U^T(j,:) U(j,:) V + \beta I$$

$$\frac{\partial L}{\partial V} = U^T U V^T U^T U + U^T U V U^T U - 2 G U^T U$$

$$\Rightarrow V(i,k) \leftarrow V(i,k) \sqrt{\frac{B(i,k)}{C(i,k)}}$$

$$B = \sum_i \sum_j b_{ij} G(i,j) U^T(i,:)$$

$$C = \sum_i \sum_j b_{ij} U^T(i,:) U(j,:) U^T($$

Algorithm:

Algorithm 1 TWMF for Trust Prediction

Input: $\{\boldsymbol{G}_1, \boldsymbol{G}_2, \cdots, \boldsymbol{G}_m\}$, α, β and γ.

Output: $\hat{\boldsymbol{G}}$

1: $\boldsymbol{G} = \sum_{t=1}^{m} \boldsymbol{G}_t$

2: Initialize $\boldsymbol{V}$ randomly;

3: Initialize η_i randomly;

4: Initialize $\boldsymbol{U}(i,:)$ randomly

5: **while** not reach convergence or the maximal iteration

6: Update $\eta_i \leftarrow \eta_i \sqrt{\dfrac{\sum_{j=1}^{n} c_{ij} e^{-\eta_i (m-g_{ij}^t)}}{2\alpha\eta_i + \sum_{j=1}^{n} c_{ij} e^{-\eta_i (m-g_{ij}^t)}}}$;

7: Update $\boldsymbol{U}(i,k) \leftarrow \boldsymbol{U}(i,k) \sqrt{\dfrac{\boldsymbol{a}_i^{\mathrm{T}}(k)}{[\boldsymbol{U}(i,:)\boldsymbol{A}_i](k)}}$

8: $\boldsymbol{V}(i,k) \leftarrow \boldsymbol{V}(i,k) \sqrt{\dfrac{\boldsymbol{B}(i,k)}{\boldsymbol{C}(i,k)}}$

9: end **while**

10: $\hat{\boldsymbol{G}} = \boldsymbol{U}\boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}$

where $\boldsymbol{a}_i$ and $\boldsymbol{A}_i$ are defined as follows:

$$\boldsymbol{a}_i = \sum_{j=1}^{n} b_{ij}^t \boldsymbol{G}(i,j)\boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}(j,:) + \sum_{j=1}^{n} b_{ji}^t \boldsymbol{G}(j,i)\boldsymbol{V}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}(j,:)$$

$$\boldsymbol{A}_i = \sum_{j=1}^{n} b_{ij}^t \boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}(j,:)\boldsymbol{U}(j,:)\boldsymbol{V}^{\mathrm{T}}] + \sum_{j=1}^{n} b_{ji}^t \boldsymbol{V}^{\mathrm{T}}\boldsymbol{U}(j,:)^{\mathrm{T}}\boldsymbol{U}(j,:)\boldsymbol{V} + \beta\boldsymbol{I}$$

where $\boldsymbol{B}$ and $\boldsymbol{C}$ are defined as

$$\boldsymbol{B} = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^t \boldsymbol{G}(i,j)\boldsymbol{U}^{\mathrm{T}}(i,:)\boldsymbol{U}(j,:)$$

$$\boldsymbol{C} = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^t \boldsymbol{U}^{\mathrm{T}}(i,:)\boldsymbol{U}(j,:)\boldsymbol{V}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}(i,:)\boldsymbol{U}(j,:) + \gamma\boldsymbol{V}$$

where $c_{ij} = \| \boldsymbol{G}(i,j) - \boldsymbol{U}(i,:)\boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}(j,:) \|_2^2$

$b_{ij}^t = e^{-\eta_i (m-g_{ij}^t)}$ and $b_{ji}^t = e^{-\eta_j (m-g_{ji}^t)}$,

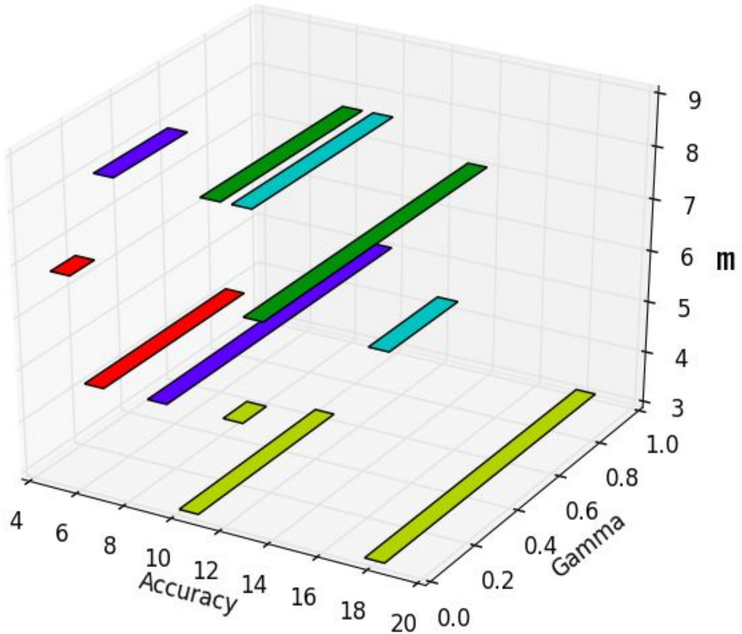
Experiment Settings

- Used dataset generated by www.epinions.com .
- We choose trust relations from t_i to t_{l+i-1} t as old trust relations 'O' with the time window size 'i'.
- 'N' is the new trust relations established at t_{l+i}.
- 'A' is the trust relation predicted excluding 'O'.
- We sort values in A and take top N pairs as C.

Prediction Accuracy (PA) = $|N \cap C| / |N|$

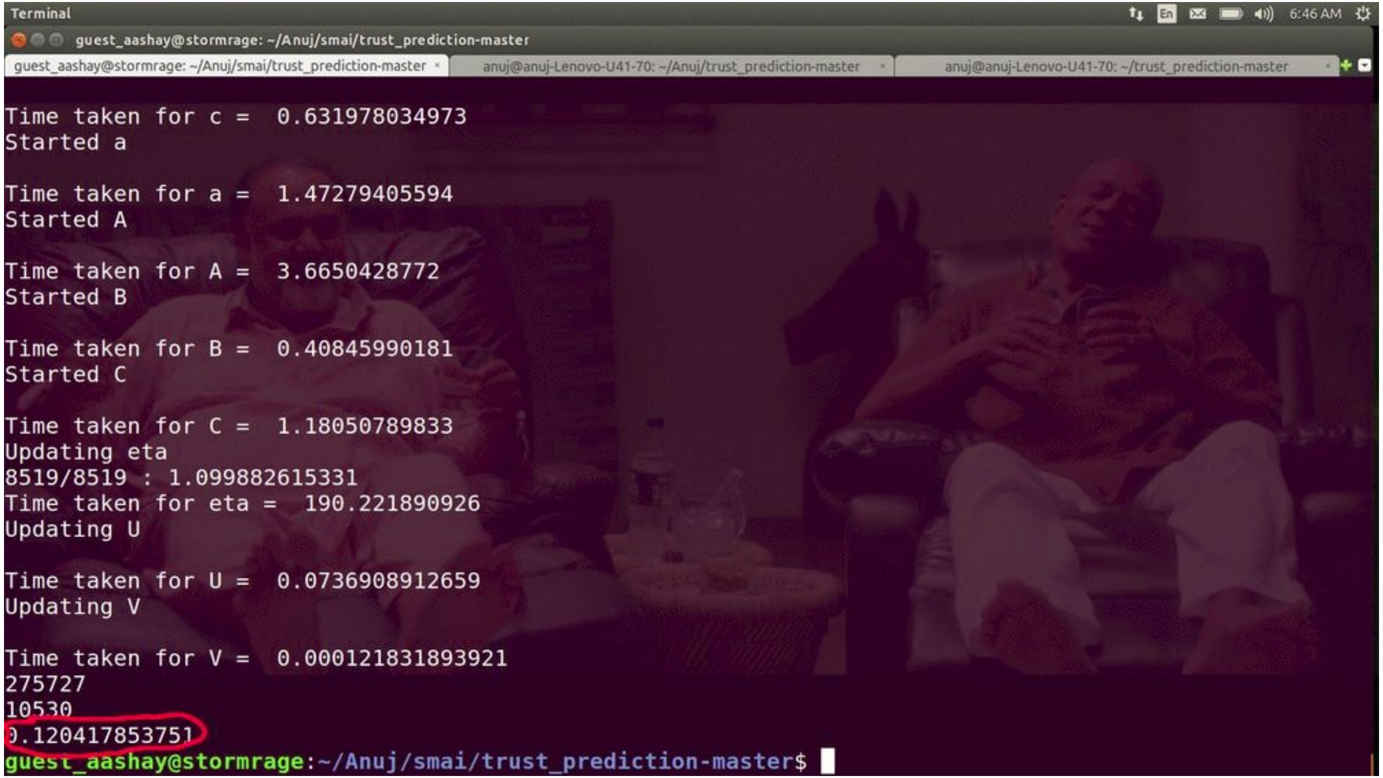
Results

- The following is the graph of The obtained results (accuracy) with varying values of m (time till which the trust relation is known) and gamma (varied from 0.1 to 1).
- The bars of same colors are shows the results obtained with same number of epochs.
- The average time taken for one epoch is about 210 seconds.



(Epocs,colour) - (10 - Red), (5,blue), (6,green), (4,Yellow), (3,Cyan)

Continued ..


Terminal
6:46 AM
guest_aashay@stormrage: ~/Anuj/smai/trust_prediction-master
guest_aashay@stormrage: ~/Anuj/smai/trust_prediction-master
anuj@anuj-Lenovo-U41-70: ~/Anuj/trust_prediction-master
anuj@anuj-Lenovo-U41-70: ~/trust_prediction-master
Time taken for c = 0.631978034973
Started a
Time taken for a = 1.47279405594
Started A
Time taken for A = 3.6650428772
Started B
Time taken for B = 0.40845990181
Started C
Time taken for C = 1.18050789833
Updating eta
8519/8519 : 1.099882615331
Time taken for eta = 190.221890926
Updating U
Time taken for U = 0.0736908912659
Updating V
Time taken for V = 0.000121831893921
275727
10530
0.120417853751
guest_aashay@stormrage:~/Anuj/smai/trust_prediction-master$

Continued ..

Terminal

guest_aashay@stormrage: ~/Anuj/smai/trust_prediction-master

guest_aashay@stormrage: ~/Anuj/smai/trust_prediction-master | anuj@anuj-Lenovo-U41-70: ~/Anuj/trust_prediction-master | anuj@anuj-Lenovo-U41-70: ~/trust_prediction-master

```
Time taken for V =  0.000141143798828
epochs # 2
Started c

Time taken for c =  0.648800849915
Started a

Time taken for a =  1.4781460762
Started A

Time taken for A =  3.63032507896
Started B

Time taken for B =  0.429671049118
Started C

Time taken for C =  1.22097206116
Updating eta
8519/8519 : 1.099882615331
Time taken for eta =  196.742071867
Updating U

Time taken for U =  0.0756840705872
Updating V

Time taken for V =  0.000118017196655
epochs # 3
```

Implementation (Matrix optimizations)

Matrix optimizations :-

① $\boxed{c = (G_m - UVU^T)^2}$

where $G_m = \sum_{i=1}^{m} g_i$ and g_i is the social trust matrix at time i.

U, V are variable matrices defined later.

② $\boxed{b_{ij} = e^{-\eta_i [m - g^t_{ij}]}}$ where η_i = decay rate of ith user.

g^t_{ij} = first time instance when user 'i' trust 'j'.

③ $\boxed{a = \left[(b \cdot G_m) \times (V \times U^T)^T + (b \cdot G_m)^T \times (V^T \times U^T)^T \right]}$

[•] = dot product of matrices

[x] = Matrix multiplication.

social trust matrix at time i.

U, V are variable matrices defined later.

② $\boxed{b_{ij} = e^{-\eta_i [m - g^t_{ij}]}}$ where η_i = decay rate of ith user.

g^t_{ij} = first time instance when user 'i' trusts 'j'.

③ $\boxed{a = \left[(b \cdot G_m) \times (V \times U^T)^T + (b \cdot G_m)^T \times (V^T \times U^T)^T \right]}$

[•] = dot product of matrices

[x] = Matrix multiplication.

④ $\boxed{B = U^T \times (b \cdot G_m) \times U}$

Continuation ..

⑤ $tmp1[i] = U[i]^T \times U[i]$

$tmp2 =$ reshaped $tmp1\ [\ (n)\times(d)\times(d) \longrightarrow (n)\times(d\times d)\ ]$

$tmp3[i] = [(b[i])^T \times tmp2].\text{reshape}([d, d])$

where d = no of faceits features

$\boxed{C \mathrel{+}= (V^T U[i]^T \times U[i])^T \times tmp3[i]}$ $\forall i$

$\boxed{C \mathrel{+}= \gamma V}$ $\gamma \Rightarrow$ a variable that is varied in our model

⑥ $\boxed{A = \beta I}$ $\beta \Rightarrow$ fixed variable

$tmp1[i] = U[i]^T \times U[i]$

$tmp2 =$ reshaped $tmp1\ [\ (n)\times(d)\times(d) \longrightarrow (n)\times(d\times d)\ ]$

$tmp3 = (b \times tmp2)$ reshaped $([d, d])$

$tmp4 = (b^T \times tmp2)$ reshaped $([d, d])$

$\boxed{A[i] \mathrel{+}= V \times tmp3[i] \times V^T + V^T \times tmp4[i] \times V}$ $\forall i$

Dimensions of matrices :- n = # of users

$\boxed{C \mathrel{+}= \gamma V}$ $\gamma \Rightarrow$ a variable that is varied in our model

⑥ $\boxed{A = \beta I}$ $\beta \Rightarrow$ fixed variable

$tmp1[i] = U[i]^T \times U[i]$

$tmp2 =$ reshaped $tmp1\ [\ (n)\times(d)\times(d) \longrightarrow (n)\times(d\times d)\ ]$

$tmp3 = (b \times tmp2)$ reshaped $([d, d])$

$tmp4 = (b^T \times tmp2)$ reshaped $([d, d])$

$\boxed{A[i] \mathrel{+}= V \times tmp3[i] \times V^T + V^T \times tmp4[i] \times V}$ $\forall i$

Dimensions of matrices :-

$C \Rightarrow n\times n$

$b \Rightarrow n\times n$

$a(i) \Rightarrow d\times 1$ $\forall\ i=[1,n]$

$A(i) \Rightarrow d\times d$ $\forall\ i=[1,n]$

$B \Rightarrow d\times d$

$c \Rightarrow d\times d$

$g \Rightarrow n\times n$

$G \Rightarrow n\times n$

$N \Rightarrow n\times d$

n = # of users.

m = time till which trust matrices are known.

d = # of faceit features

$U \Rightarrow n\times d$

$V \Rightarrow d\times d$

Predicted trust Matrix :-

$G' = UVU^T$

References:

- https://tcs.epfl.ch/files/content/sites/tcs/files/Lec2-Fall14-Ver2.pdf
- https://www.cse.msu.edu/~tangjili/publication/hTrust.pdf
- https://link.springer.com/content/pdf/10.1007%2Fs11859-014-1027-z.pdf
- https://arxiv.org/pdf/1507.00333.pdf

Minimizing the Error in prediction

Given the error function mentioned before, we write its lagrange to minimize the error as follows.

$$\min_{U,V,\eta_i} \sum_{i=1}^{n}\sum_{j=1}^{n} e^{-\eta_i}(m - g_{ij}^{t}) \| G(i,j) - U(i,:)VU^{T}(j,:) \|_2^2 + \alpha \sum_{i=1}^{n} \| \eta_i \|_2^2$$

$$+ \beta \| U \|_F^2 + \gamma \| V \|_F^2 \qquad s.t.\ U, V \geq 0, \quad \eta_i \geq 0, \ \forall i \in [1, n]$$

Presented By: Team 30

Amandeep Shahi (201501087)

Anuj Goyal (201501129)

Swapnil Gupta (201501175)

Aayush Tiwari (201531031)

Github : https://github.com/amanshahi/trust_prediction

Thank You!