(i) 0.76, 0.32, 0.54.

(ii) Finally, we have $(1:0.86),(2:0.71),(3:1.00),(4:0.29),(5:1.00),(6:0.52),(7:0.14),(8:0.00),(9:0.38),(10:0.00)$, so nodes with indexes $[1,2,3,5,6]$ will belong to class "+" and nodes with indexes $[4,7,8,9,10]$ will belong to class "-".

(i)

$$\begin{aligned}
b_1(x_1) &= \frac{1}{Z}\phi_i(x_i)m_{21}(x_1) \\
&= \frac{1}{Z}\phi_1(x_1)\Sigma_{x_2}\phi_2(x_2)\psi_{21}(x_2,x_1)m_{32}(x_2)m_{42}(x_2) \\
&= \frac{1}{Z}\phi_1(x_1)\Sigma_{x_2}\phi_2(x_2)\psi_{21}(x_2,x_1)(\Sigma_{x_3}\phi_3(x_3)\psi_{32}(x_3,x_2)\Sigma_{x_4}\phi_4(x_4)\psi_{42}(x_4,x_2)) \\
&= \frac{1}{Z}\Sigma_{x_2}\Sigma_{x_3}\Sigma_{x_4}\phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\phi_4(x_4)\psi_{21}(x_2,x_1)\psi_{32}(x_3,x_2)\psi_{42}(x_4,x_2)
\end{aligned}$$

(ii) According to the properties of grapical model, we have

$$\begin{aligned}
p(x_1|y_1,y_2,y_3,y_4) &= \frac{1}{Z}\Sigma_{x_2}\Sigma_{x_3}\Sigma_{x_4}p(x_1,x_2,x_3,x_4|y_1,y_2,y_3,y_4) \\
&= \frac{1}{Z}\Sigma_{x_2}\Sigma_{x_3}\Sigma_{x_4}\phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\phi_4(x_4)\psi_{21}(x_2,x_1)\psi_{32}(x_3,x_2)\psi_{42}(x_4,x_2) \\
&= b_1(x_1)
\end{aligned}$$

BTW, if you are not familiar with grapical model, you can refer to the lecture *Probabilistic Graphical Models* or *Graphical Models and Belief Propagation* in MIT 6.869 Advances in Computer Vision.

(iii) The results are as below, for which the caculation is done by hand with cmd assitance of Python,

$$b_1(x_1) = \frac{1}{Z}\begin{bmatrix}1.42681\\1.299\end{bmatrix}$$

$$b_2(x_2) = \frac{1}{Z}\begin{bmatrix}2.4871\\0.24681\end{bmatrix}$$

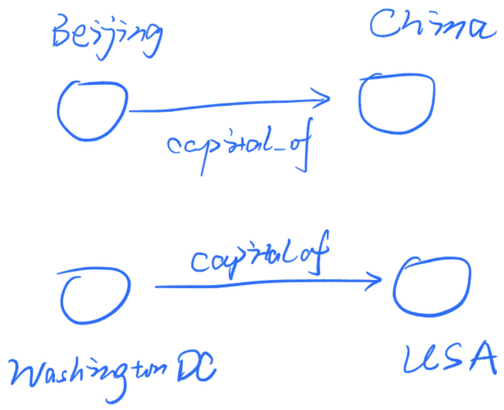
$$b_3(x_3) = \frac{1}{Z}\begin{bmatrix}0.418\\2.300881\end{bmatrix}$$

$$b_4(x_4) = \frac{1}{Z}\begin{bmatrix}0.231781\\2.4871\end{bmatrix}$$

$$b_5(x_5) = \frac{1}{Z}\begin{bmatrix}1.12009\\0.31009\end{bmatrix}$$

As we can see, x_2, y_4 is influenced by y_2, y_4 most respectively. ψ_{12}, ψ_{34} indicate that node 1 and node 2, and node 3 and node 4 tend to be the same slightly. And ψ_{23}, ψ_{35} indicate that node 2 and node 3, and node 3 and node 5 tend to be the opposite with high possibility. So the results are as expected.

Take this graph as an example,



Let all of **e** to be

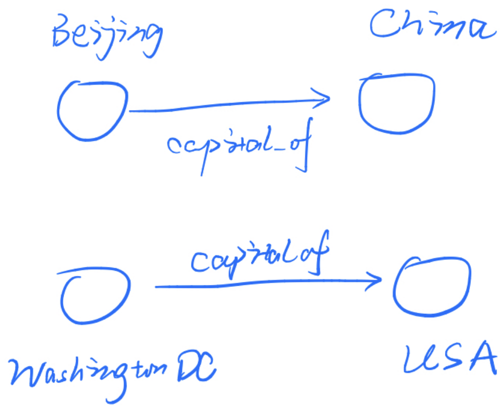
$$\begin{bmatrix}1.0\\0.0\end{bmatrix}$$

and ℓ to be

$$\begin{bmatrix}0.0\\0.0\end{bmatrix}$$

We would make the objective to be 0, but obviously the embeddings make no sense.

Take this graph as an example,



Let all of **e** to be

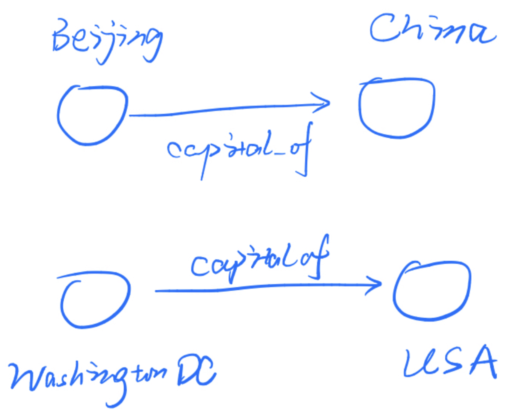
$$\begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}$$

and ℓ to be

$$\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

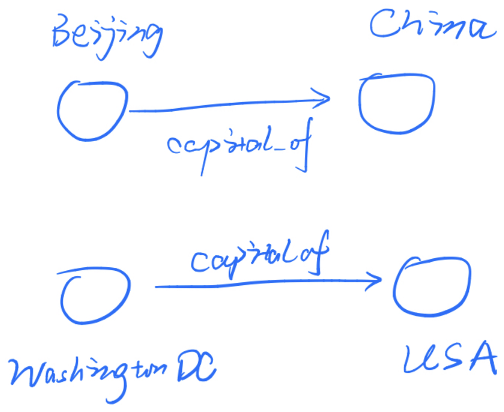
We would make the objective to be 0, but obviously the embeddings make no sense.

Take this graph as an example,



When we don't normalize the entity embeddings, the algorithm tends to trivially change the $||e||_2$ to optimize to the objective. In this example, in this case, one possible generation is that $||e||_2$ for *China* and *Beijing* tends to be very small and $||e||_2$ for *USA* and *Washington, DC* tends to be very large, or vice versa.

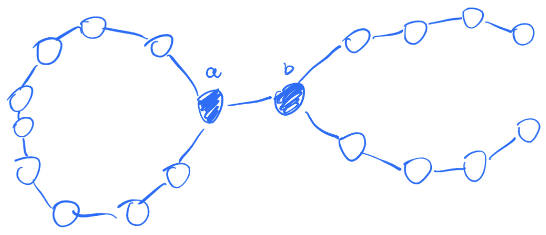
Take this graph as an example,



after we add a new node *Earth* and the relation from **China** and *USA*, $a_country_of$, it's impossible to find a proper embedding of $a_country_of$ and *Earth* to meet with the conditions in a 2-d embedding space, since *Beijing* $\leftarrow$ *China* and *Washington, DC* $\leftarrow$ *USA* are two parallel lines (vectors) in the embedding space.

(i) Three layers, because these 2 red nodes have the same 2-hop neighbor structure.

(ii) As the example in the below which shows the trickiest case,



node a which is expected to classify as *postive* and node b which is expected to classify as *negtive* have the same 4-hop neighbor structure. Only with more than 5 times message passing can a model classify them correctly.

(i) The entry (i, j) in transition matrix is,
$\frac{1}{|D_{i,i}|}$ for a connected point couple node i and node j
0 otherwise, which is

$$D^{-1}A$$

(ii) The entry (i, j) in transition matrix is,
$\frac{1}{2}$ when $i = j$,
$\frac{1}{2*|D_{i,i}|}$ for a connected point couple node i and node j,
0 otherwise, which is

$$\frac{1}{2}D^{-1}A + \frac{1}{2}I$$

After l times message passing, the hidden representation should be

$$h^{(l)} = (D^{-1}A)^l h^{(1)}$$

Use L_r to represent $D^{-1}A$. Firstly, it's easy to see that 1 and $[1, ..., 1]^T$ is one of the eigenvalue and eigenvector of matrix L_r. Consider a new vector y, which is not a multiple of $[1, ..., 1]^T$, and define S as a set of nodes whose value is $max(|y_i|)$.

Then, for some node $j \in S$ and a neighbor $i \notin S$, which must exist because the graph is connected and every node has a degree not less than 2, the value of $|L_r y_j|$ will be less than $|y_j|$ and thus we need the corresponding $|\lambda|$ to be less than 1. So we can conclude that **the absolute values of all the eigenvalues of L_r is less or equal than** 1.

So L_r^l tends to converge to some point and $h^{(l)}$ will converge as $l \to \inf$.

For a more formal prove of GCN's over smoothing effect, you can refer to Li et. al.'s work in AAAI2018, *Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning.*

(i) The update ruls is simple:

- set the embedding of points with maximum reachability (embedding) to 1
- perform average aggregration of neighbor nodes(or every node)

(ii)
- message function: inform the neighbor nodes to update their embeddings when current node is visited
- aggregration function: $h_i^{(l+1)} = \frac{1}{|N_i|} \sum_{j \in N_i} h_j^{(l)}$

,k/98J́¿K.

Information sheet
CS224W: Machine Learning with Graphs

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via GradeScope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

Late Homework Policy Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: ______________________________

Email: ____________________ **SUID:** ____________________

Discussion Group: ______________________________

I acknowledge and accept the Honor Code.

(Signed) ______________________________