尊敬的吴老师，您好

我最近在论文工作上面有一点进展：一是借鉴神经网络的随机梯度下降，对原文中的训练方法做了一些改进，效率有所提升；二是成功定位了转发链长度预测的问题所在。

1. 训练方法改进

原文中的训练算法如Algorithm1所示，在更新I和S前，需要综合整个数据集的误差，计算出所有$I_u, S_u, u \in V$，再利用投影梯度法计算步长。这是典型的批量梯度下降（Batch Gradient Descent），该方法有几点不足：

1) 在求步长的过程中，需要反复对所有微博求目标值，这个过程非常耗时且存在冗余。

2) 如果目标函数存在多个局部极小值，可能会陷入局部最优。

3) 靠近极小值时下降缓慢

Algorithm 1 参数估计

Input: Collection of cascades observed in a given time period, the maximum epoch M, and regularization parameters γ_I and γ_S

Output: User-specific influence and susceptibility I, S

Construct diffusion network δ from cascades

Initialize parameters with random values, including I, S

Repeat

For $i = 1$ to n

Calculate gradients $\partial L/\partial I_u$ and $\partial L/\partial S_v$

Update I and S with PG method

Until maximum epoch M is reached or gradients vanish

我们考虑随机梯度下降法（Stochastic gradient descent），根据每个用户的误差增量计算得到$\frac{\partial \mathbb{L}}{\partial I_u}$和$\frac{\partial \mathbb{L}}{\partial S_u}$后，立即更新$I_u$和$S_u$。

对于用户u，影响力误差和敏感度误差如公式1所示：

$$
\begin{aligned}
I(u) &= -\sum_{v \in V} \sum_{D_{v,i} \in \Gamma(v)} \Phi_{u \in D_{v,i}} (n_{z_{v,i}, D_{v,i}} log\ p(z_{v,i} \mid z_{v,i-1}, D_{v,i}, \delta)) + \gamma_I \left\| I_u \right\|_F^2 \\
S(u) &= -\sum_{D_{u,i} \in \Gamma(u)} (n_{z_{u,i}, D_{u,i}} log\ p(z_{u,i} \mid z_{u,i-1}, D_{u,i}, \delta)) + \gamma_S \left\| S_u \right\|_F^2
\end{aligned}
\tag{1}
$$

梯度计算如公式2所示：

$$
\begin{aligned}
\frac{\partial \mathbb{L}}{\partial I_u} &= -\lambda \sum_{v \in V} S_v \sum_{D_{v,i} \in \Gamma(v)} \Phi_{u \in D_{v,i}} (n_{z_{v,i}=1, D_{v,i}} \frac{1 - p_{v, D_{v,i}}}{p_{v, D_{v,i}}} - n_{z_{v,i}=0, D_{v,i}}) + \gamma_I I_u \\
\frac{\partial \mathbb{L}}{\partial S_v} &= -\lambda \sum_{D_{v,i} \in \Gamma(v)} \sum_{u \in D_{v,i}} I_u (n_{z_{v,u}=1, D_{v,u}} \frac{1 - p_{v, D_{v,u}}}{p_{v, D_{v,u}}} - n_{z_{v,u}=0, D_{v,u}}) + \gamma_S S_v
\end{aligned}
\tag{2}
$$

在一个用户数量为199408，微博数量为395852的数据集上进行训练，两种算法的耗时对比如图1所示。 可以看出，SGD算法在700s时已接近最优值，速度远比BGD快。因此在这个训练

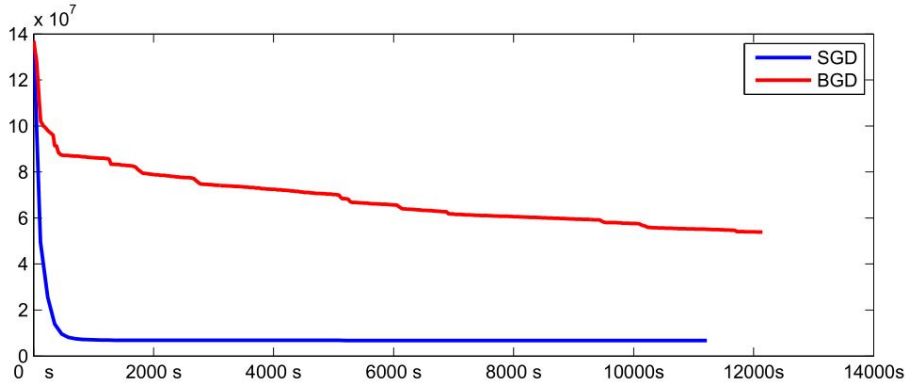

图 1: BGD与SGD性能对比

问题上，SGD要比BGD更合适。

2. 转发链长度预测问题定位

在解决掉训练时间过长的问题后，我又重新进行了转发链长度预测仿真，但效果相比以前只提高了20%左右。结果如图2所示：转发链长度为1的微博数量相差很大。

长度	预测	实际
1	41205	1737
2	23620	76042
3	13562	15517
4	8345	7129
5	5464	3939
6	3859	2563
7	2745	1805
8	2134	1267
9	1629	1085
10	1328	775

图 2: 转发链长度：预测与实际

我将所有用户的影响力和敏感度都设为1，即用户只要看到微博，必然转发。在这样的前提下，得到的转发链长度为1的微博数量都有15000条之多，远大于实际的1737。因此问题出在传播网络（Diffusion Network）上。

我们的程序输入是一条条关于用户转发行为的时间序列，需要根据这些数据**反推出用户间的follower-followee关系**。关于Diffusion Network怎么推导，在论文中作者一笔略过，但给出了数据集对应的Diffusion Network结果数据。但我用这个数据来训练却得到错误的结果。

下一步我想研究下Diffusion Network的有关内容，老师您觉得我的下一步方向有没有问题？

学生王超民，2016年9月11日