# PopCore--Prepare

- Get familiar with the whole project

overall structure

record fields

kNN, k-means clustering

# PopCore--TIMELINE

- Try to find TIMELINE as a factor

*facebook_like.csv:*

user_id, item, release_time, like_time

*facebookfriend_like.csv:*

user_id, friend_id, item, release_time, like_time

# PopCore--TIMELINE

a). Time difference between like_time and release_time

- each item

has different like_time for different users. Define time_diff = like_time-release_time, then for each item, we have

<release_time, avg(time_diff)>

*draw the figure of the data, and do k-means clustering of similar items*

# PopCore--TIMELINE

- each user

would like different items at different time, i.e. has a list of <item_release_time, like_time>, define the time_diff as before, then for each user, we have

<user_id, avg(time_diff)>

*draw the figure of the data, and do k-means clustering of similar users*

# PopCore--TIMELINE

- Filtering the data

Quite a lot of items were released in early days while Facebook launched in 2004 and user can like the item after that. In that case, those items differs heavily in the releasing date while the liking date may be similar (e.g. user like the item on the day he joined facebook)

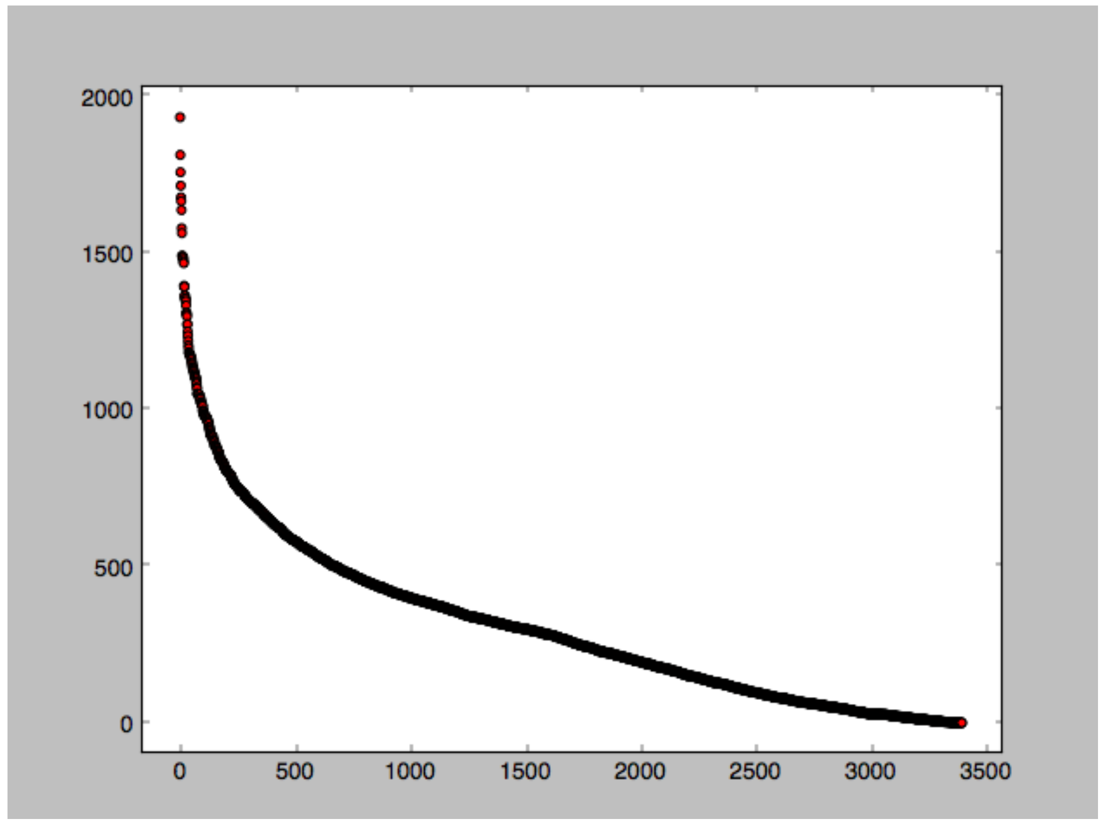
Use the items that were released after the users' join date (say the first like_time). Then we can analyze how users will response the newly released items.

# PopCore--TIMELINE

- Plotting

We need to find how frequently a user would like an item as well as how users' behaviors differ for the same item.

Sort the time_diff in descending order, and hash the user_id or item_id
x-axis: hashed user_id
y-axis: time_diff (days)



# PopCore--TIMELINE

b). Time difference between like_time of a pair of friends

*Background: A user will influence or be influenced by his friends' like or dislike. We can use this as one factor to do recommendation.

We may say user A likes item1, then his friend B likes item 1 within a time period. It is probably that B likes this item because A likes it.

Based on this idea, we can find how a user can influence his friends' behavior.

# PopCore--TIMELINE

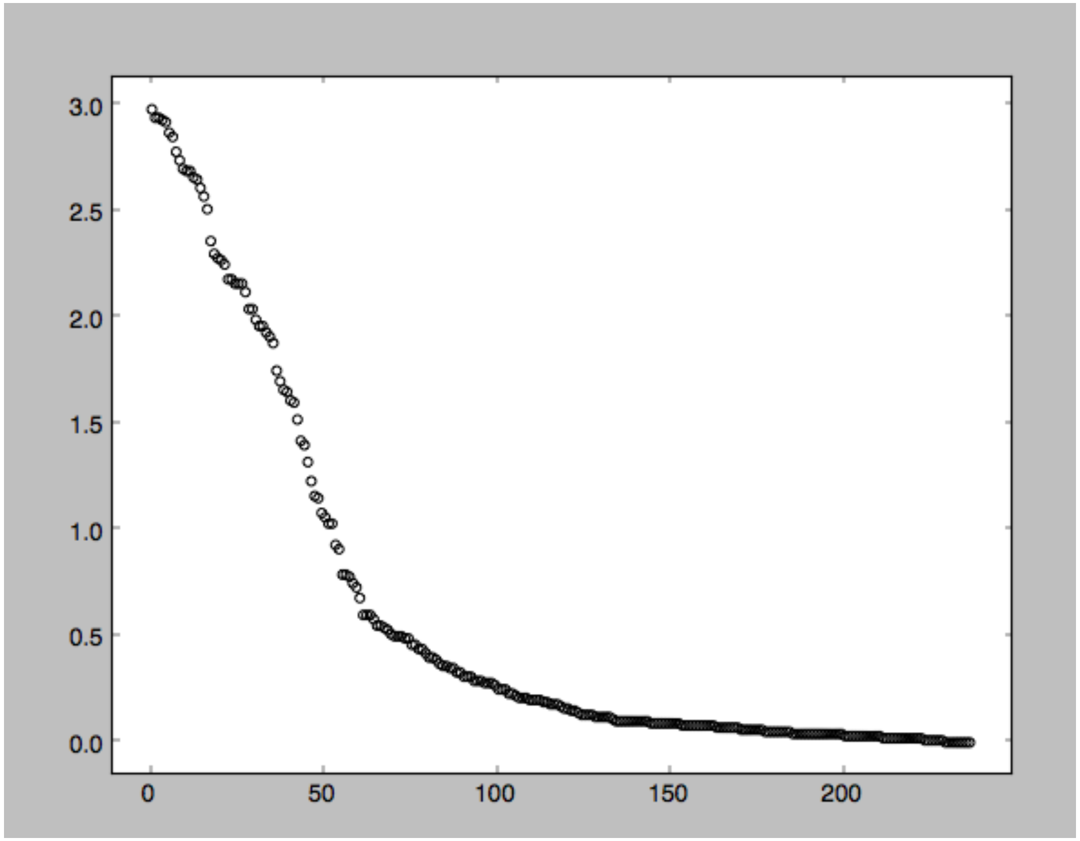
1. We can find all the pairs of friends, and lists of their liking items.

We may not need to filter the data this time because we only depend on the time_diff between the friends' like_time regardless of the item release_time.

# PopCore--TIMELINE

Sort the time_diff in descending order, and hash the friend_pair_id

x-axis: hashed friend_pair_id
y-axis: time_diff (days)



# PopCore--TIMELINE

2. For each user, get a list like [uid] : {<f1,1/t1>,...,<fn, 1/tn>}, inverse the time difference since the larger the time difference, the smaller probability one would influence another.Finally output dictionary: [uid]: {<f1,p1>,...,<fn,pn>}

Within the friends, we can normalize the probability as
Pi = Pi/\Sigma(Pj)

# PopCore--TIMELINE

3. For each friend pair, the items in common:
    required common items/ total common items

* it is not feasible since the total common items for each pair are almost 1 or 2 due to the limit of data we have

4. Influence proportion:
    for each user, # influenced items / total # of user items

# PopCore--TIMELINE

5. Jaccard Coefficient:

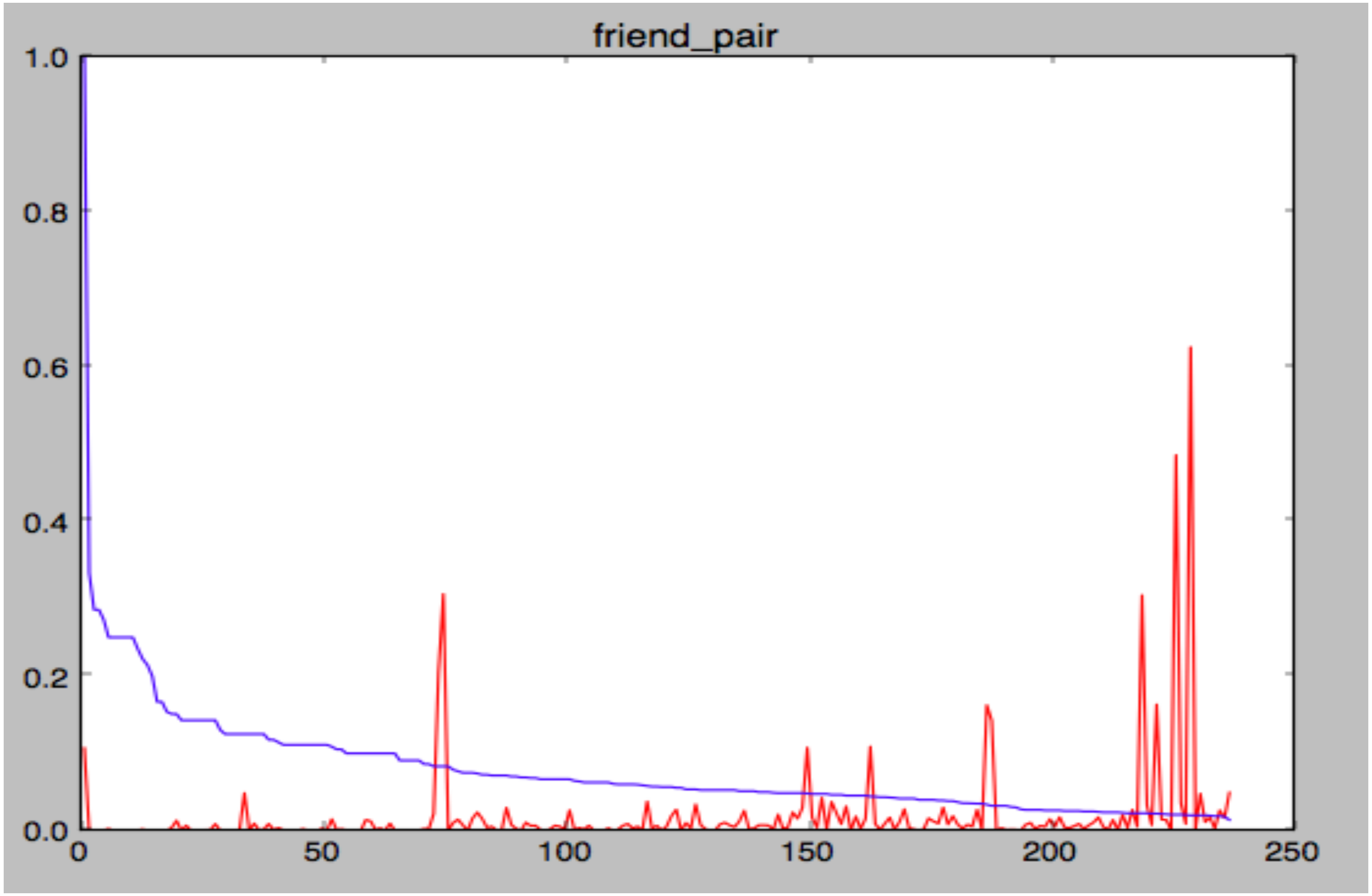
for each friend pair: (U1 and U2) / (U1 or U2)

we also have average time_diff for each friend pair <pair_i, t_i>, normalize the time_diff as

    t_i = t_i / sqrt(\Sigma t_j^2)

# PopCore--TIMELINE

x-axis: hashed friend_pair_id; red is normalized score, and blue is Jaccard Coefficient



# PopCore

I think TIMELINE can be an important factor to recommendation, and we can assign a weight to this factor when we doing our normal recommendation algorithm.

The larger the Jaccard Coefficient, the smaller the normalized score.