MỤC LỤC

[1. Trình bày bài toán 2](#_Toc417142609)

[1.1. Đầu vào 2](#_Toc417142610)

[1.1.1. Dữ liệu do người dùng nhập 2](#_Toc417142611)

[1.1.2. Data dành cho Learning 2](#_Toc417142612)

[1.2. Đầu ra 2](#_Toc417142613)

[1.3. Những thao tác cần xử lý 2](#_Toc417142614)

[2. Thống kê và lựa chọn dữ liệu đầu vào 5](#_Toc417142615)

[2.1. Thống kê số lượng bạn bè của mỗi UID 5](#_Toc417142616)

[2.2. Lựa chọn các UID 6](#_Toc417142617)

[3. Xây dựng các feature 7](#_Toc417142618)

[3.1. Xây dựng các cặp UID để tính feature 7](#_Toc417142619)

[3.2. Tính các giá trị feature 7](#_Toc417142620)

[3.2.1. Tính x1 ( Tính số bạn chung giữa 2 UID) 7](#_Toc417142621)

[3.2.2. Tính x2 ( Tính số company chung giữa 2 UID) 8](#_Toc417142622)

[3.2.3. Tính x3 ( Tính số school chung giữa 2 UID) 8](#_Toc417142623)

[3.2.4. Tính x4 ( Tính số group chung giữa 2 UID) 8](#_Toc417142624)

[3.2.5. Tính Label y 8](#_Toc417142625)

[4. Logistic regression 9](#_Toc417142626)

[4.1. Biểu diễn hàm hypothesis 9](#_Toc417142627)

[4.2. Hàm Cost Function 9](#_Toc417142628)

[4.3. Gradient Descent 9](#_Toc417142629)

[4.4. Regularization 10](#_Toc417142630)

[4.5. Kết quả của việc học 10](#_Toc417142631)

[5. Đưa ra danh sách dự đoán 11](#_Toc417142632)

[6. Đánh giá hiệu quả dự đoán 12](#_Toc417142633)

# Trình bày bài toán

## Đầu vào

### Dữ liệu do người dùng nhập

* Người dùng nhập vào một User ID cần đưa ra danh sách gợi ý kết bạn

### Data dành cho Learning

* Friend list: Danh sách kết bạn của các UID
  + Mỗi dòng là một danh sách những UID đã kết bạn với UID đầu tiên trong danh sách
* Company list: Danh sách các Company ID của mỗi UID
  + Mỗi dòng có số đầu tiên là UID, các số tiếp theo là Company ID của người đó
* Group list: Danh sách các Group ID của mỗi UID
  + Mỗi dòng có số đầu tiên là UID, các số tiếp theo là Group ID của người đó
* School list: Danh sách các School ID của mỗi UID
  + Mỗi dòng có số đầu tiên là UID, các số tiếp theo là School ID của người đó
* Friend add: Nhật ký lưu lịch sử kết bạn của các UID từ 8-8-2013 đến 02-10-2013
  + Mỗi một dòng gồm 2 UID: UID thứ nhất kết bạn với UID thứ 2

## Đầu ra

* Từ User ID do người dùng nhập vào, cần phải đưa ra danh sách gợi ý kết bạn cho User ID đó.

## Những thao tác cần xử lý

* Đầu tiên ta cần xây dựng các Feature cho Logistic regression. Để có các feature cho Logistic regression ta cần xây dựng một tập Data dành cho việc học. Vì số lượng Data của Input hay nói cách khác là có rất nhiều User ID, nên để xử lý trên máy tính đơn, ta không thể sử dụng được hết. Khi đó nếu lấy ngẫu nhiên từng phần của Data Input thì sẽ không thể bảo toàn được sự phân bố của Data ban đầu. Chẳng hạn như trong Friend list, ta chỉ lấy những danh sách của các UID có hơn 200 bạn, nhưng trong thực tế số lượng bạn của các UID phần lớn nằm trong khoảng từ 20 đến 30, khi đó sẽ ảnh hưởng rất nhiều đến kết quả của các Feature, dẫn đến giảm độ chính xác của dự đoán đưa ra. Vì vậy, cần phải **thống kê Data Input**, cụ thể đó là Friend list, đưa ra số lượng bạn của các UID, từ đó chọn các ID sao cho vẫn giữ được sự phân bố về số lượng bạn.
* Sau khi lựa chọn được các UID để xây dựng feature, ta sẽ tiến hành tính các feature cho mỗi UID. Với một cái nhìn trực quan ban đầu, ta thấy rằng khi hai người kết bạn với nhau, giữa họ có khả năng rất lớn sẽ có một mối liên quan đến nhau về bạn bè, công ty, group, trường học. Đó có thể là giữa họ có một hoặc một vài người bạn chung, họ làm chung công ty, chung nhóm hay chung trường học. Ngoài các trường hợp trên, nếu hai người kết bạn với nhau mà chưa có mối quan hệ nào trên mạng xã hội thì có lẽ họ chỉ quen nhau ngoài đời (cha mẹ với con cái, một người bạn mới quen trên xe buýt, ...), trường hợp đó có lẽ vượt ngoài khả năng của một hệ thống gợi ý kết bạn trên mạng xã hội. Như vậy, ta sẽ tìm các xây dựng các feature cho việc học dựa trên số bạn chung, trường học chung, group chung và company chung giữa 2 UID.
* Sau khi xây dựng được các feature, ta sẽ tiến hành học bằng Logistic regression. Khi đó, ta sẽ tính được của hàm:

()=g(

g(z)=

Trong đó: : Số bạn chung giữa 2 UID

: Số company chung giữa 2 UID

: Số School chung giữa 2 UID

: Số group chung giữa 2 UID

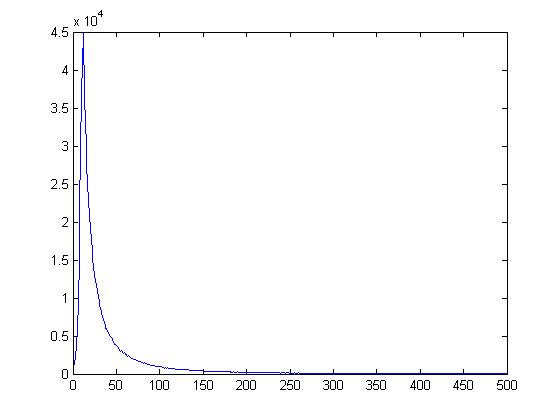
h: Xác suất kết bạn giữa 2 UID

* Cuối cùng với User ID ( UID1) do người dùng nhập vào ban đầu, ta sẽ tiến hành sinh danh sách dự đoán cho UID1 đó. Để làm được, ta sẽ tính ra danh sách L các UID có bạn chung với UID1. Từ danh sách đó, ta sẽ tính các giá trị , , , cho các cặp giữa UID1 và các UID trong danh sách L. Nếu xác suất kết bạn giữa một cặp nào đó h>0.5 ta sẽ đưa ra dự đoán UID1 sẽ kết bạn với UID tương ứng.
* Cuối cùng, để đánh giá hiệu quả của hệ thống, ta sẽ tính các giá trị precision và recall trung bình cho 100 danh sách gợi ý của 100 UID. Với các trường hợp độ dài d của danh sách gợi ý tăng từ 1 đến 100, ta sẽ chứng minh khi d tăng thì recall tăng và precision giảm.

# Thống kê và lựa chọn dữ liệu đầu vào

## Thống kê số lượng bạn bè của mỗi UID

* Như đã trình bày mục 1.3, để đảm bảo sự phân bố về số lượng bạn bè giữa các UID mà ta chọn, ta sẽ thống kê số lượng bạn bè cho tất cả các UID, rồi từ đó ta sẽ chọn ra các UID mà vẫn đảm bảo được sự phân bố số lượng bạn bè.
* Ta sẽ đọc file friendlist, với mỗi dòng của friendlist ta sẽ tính số bạn bè của UID tương ứng, lưu số đó vào một mảng s 2 chiều ( chiều thứ nhất là số lượng bàn bè, chiều thứ hai là số UID có số lượng bạn bè như vậy). Ta sẽ duyệt như thế đến hết file friendlist và sẽ được một mảng s lưu phân bố về số lượng bạn bè của các UID
* Từ mảng s ta sẽ vẽ được biểu đồ thống kê như sau:



Trong đó: Trục nằm ngang: Số lượng bạn bè

Trục nằm dọc: Số lượng UID có số lượng bạn bè tương ứng với trục ngang

## Lựa chọn các UID

* Từ mảng thống kê s đã xây dựng, ta sẽ chọn ra danh sách K chứa 100 UID sao cho vẫn đảm bảo được sự phân bố về số lượng bạn bè giữa các UID.

# Xây dựng các feature

## Xây dựng các cặp UID để tính feature

* Từ danh sách K chứa 100 UID, với mỗi UID ta sẽ xây dựng danh sách các UID có ít nhất một bạn chung với mỗi UID trong K. Tại sao lại như vậy? Vì giữa 2 người có mỗi liên hệ về bạn bè thì rất có thể họ sẽ ở chung công ty, chung nhóm hoặc chung trường học và quan trọng hơn, họ rất có thể sẽ kết bạn với nhau. Hay hiểu theo một cách máy móc hơn, những giá trị feature giữa họ có khả năng khác 0, khi đó sẽ không có những trường hợp mà feature dùng để học toàn là số 0 ( trường hợp này thì feature học thật vô dụng).
* Ta sẽ xây dựng một file ( gọi là file **inputFeature**) mà mỗi dòng là một danh sách UID. Trong đó, UID đầu tiên của mỗi dòng sẽ là một UID trong K, và các UID sau nó sẽ là các UID có ít nhất một bạn chung với nó. Để xây dựng được danh sách này:
  + Với mỗi UID trong K ( gọi là uk), ta sẽ tìm ra danh sách các bạn bè của uk ( gọi là L1).
    - Xét từng bạn bè của uk ( gọi là ui), ta sẽ tìm ra danh sách bạn bè ui (gọi là Li).
      * Xét từng UID trong Li, loại bỏ các UID là bạn bè của uk khỏi Li
    - Sau khi duyệt hết các uk, ta sẽ nối các danh sách Li lại và sẽ được một danh sách mà mỗi dòng là một danh sách UID. Trong đó, UID đầu tiên của mỗi dòng sẽ là một UID trong K, và các UID sau nó sẽ là các UID có ít nhất một bạn chung với nó.

## Tính các giá trị feature

### Tính x1 ( Tính số bạn chung giữa 2 UID)

* Ta có 2 UID: u1 và u2
* Tìm ra danh sách bạn bè của u1: F1
* Tìm ra danh sách bạn bè của u2: F2
* Tìm giao của F1 với F2 và trả về số phần tử giao

### Tính x2 ( Tính số company chung giữa 2 UID)

* Ta có 2 UID: u1 và u2
* Tìm ra danh sách các company của u1: C1
* Tìm ra danh sách các company của u2: C2
* Tìm giao của C1 với C2 và trả về số phần tử giao

### Tính x3 ( Tính số school chung giữa 2 UID)

* Ta có 2 UID: u1 và u2
* Tìm ra danh sách các school của u1: S1
* Tìm ra danh sách các school của u2: S2
* Tìm giao của S1 với S2 và trả về số phần tử giao

### Tính x4 ( Tính số group chung giữa 2 UID)

* Ta có 2 UID: u1 và u2
* Tìm ra danh sách các group của u1: G1
* Tìm ra danh sách các group của u2: G2
* Tìm giao của G1 với G2 và trả về số phần tử giao

### Tính Label y

* Label y chính là label xác định xem trong nhật ký kết bạn friendadd: 2 UID u1 và u2 có kết bạn với nhau không.
* Công việc này được thực hiện khá đơn giản: Ta sẽ tìm dòng chứa 2 UID u1 và u2 trong file friendadd, nếu tìm được ta sẽ gán label y=1 còn nếu không, ta sẽ gán nhãn label y=0

# Logistic regression

## Biểu diễn hàm hypothesis

* Như ở mục 1.3 đã trình bày, chúng ta có hàm hypothesis cho bài toán như sau:

()=g(

g(z)=

Trong đó: : Số bạn chung giữa 2 UID

: Số company chung giữa 2 UID

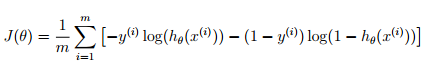
: Số School chung giữa 2 UID

: Số group chung giữa 2 UID

h: Xác suất kết bạn giữa 2 UID

## Hàm Cost Function

* Ta sẽ sử dụng hàm Cost Function sau:



* Trong đó: : là giá trị các feature thứ i trong Data Input dành cho learn

là giá trị label thứ i trong Data Input dành cho learn

## Gradient Descent

* Để cực tiểu hóa hàm Cost Function, ta sẽ dùng phương pháp Gradient Descent
* Ta có đạo hàm của Cost Function theo



* Thuật toán Gradient Descent:

Repeat{

:=-

( Cập nhật đồng thời các với j=0, 1, ...,n)

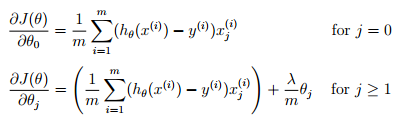
}

## Regularization

* Để tránh trường hợp Over fitting, ta sẽ sử dụng kỹ thuật Regularization:
* Khi đó Hàm Cost Function sẽ là:



* Và đạo hàm của Cost Function theo :



## Kết quả của việc học

* Sau khi học xong, ta sẽ được các giá trị và từ đó có một hàm hypothesis h đầy đủ tham số.
* Với mỗi cặp UID, ta sẽ tính các feature và thay vào h để tính được xác suất kết bạn giữa 2 UID đó.

# Đưa ra danh sách dự đoán

* Ban đầu người dùng nhập vào User ID cần sinh danh sách gợi ý ( gọi là UID1).
* Ta sẽ sinh ra một danh sách với phần tử đầu tiên là UID1, các phần tử tiếp theo sẽ là các UID mà có ít nhất một bạn chung với UID1.
* Từ danh sách trên ta sẽ ghép cặp UID1 với các UID còn lại, tính các giá trị feature , , , giữa chúng rồi thay vào hàm hypothesis h để tính được xác suất kết bạn. Nếu h>0.5 ta sẽ đưa ra trong danh sách gợi ý UID tương ứng.

# Đánh giá hiệu quả dự đoán

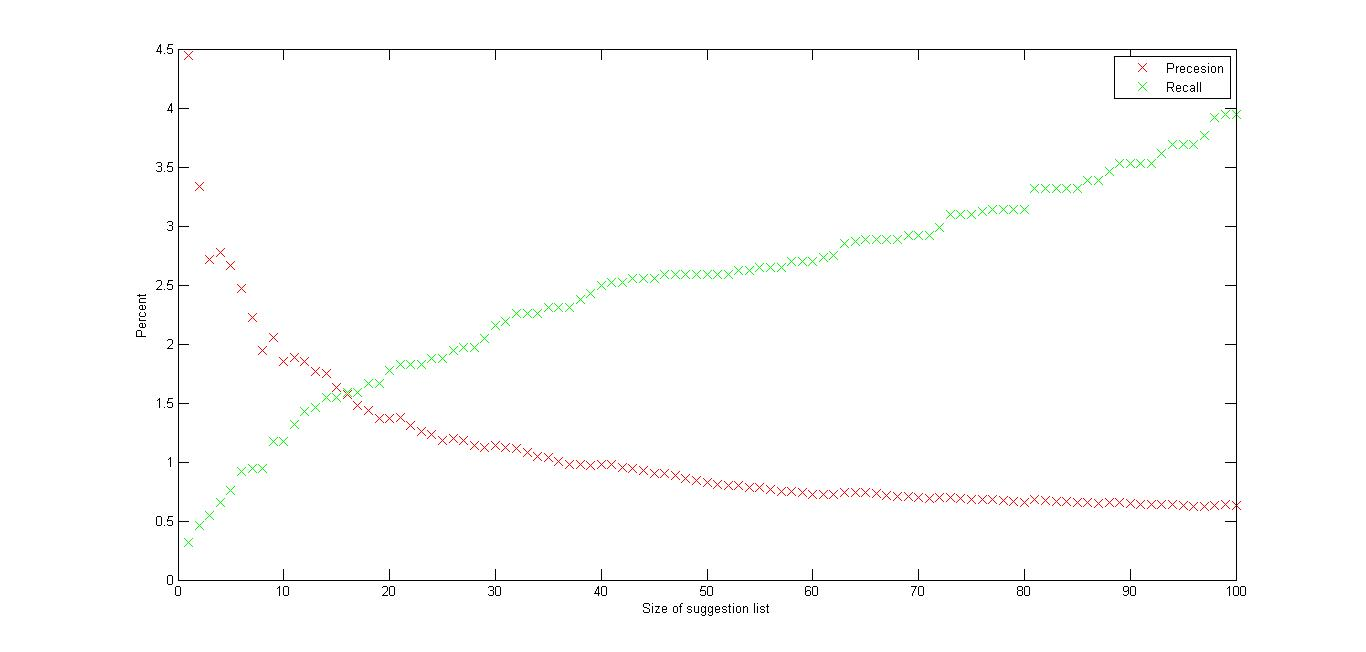
* Để đánh giá hiệu quả của hệ thống, ta sẽ tính các giá trị precision và recall trung bình cho 100 danh sách gợi ý của 100 UID. Với các trường hợp độ dài d của danh sách gợi ý tăng từ 1 đến 100, ta sẽ chứng minh khi d tăng thì recall tăng và precision giảm.
* Precision của một danh sách gợi ý cho một User ID sẽ bằng thương số giữa “True Positive” ( Số lượng kết bạn thực sự trong danh sách mình gợi ý) với tổng giữa “True Positive” và “False Positive” ( Số dự đoán mà mình đưa ra không chính xác, tức là mình dự đoán rằng có kết bạn nhưng tế lại không kết bạn) ( tổng giữa “True Positive” với “False Positive” có thể hiểu là độ dài danh sách dự đoán của chúng ta dành cho UID1). Như vậy Precision chính là độ chính xác của danh sách mà chúng ta dự đoán cho UID1 ( Tỉ lệ giữa số dự đoán đúng so với độ dài danh sách dự đoán)

Precision=

* Recall của một danh sách gợi ý cho một User ID sẽ bằng thương số giữa “True Positive” ( Số lượng kết bạn thực sự trong danh sách mình gợi ý) với tổng của “True Positive” và “False Negative” (Số lượng bạn mà mình đã bỏ sót trong danh sách gợi ý, vì thực tế các cặp đó đã kết bạn với nhau mà ta không đưa ra dự đoán) ( tổng giữa “True Positive” với “False Negative” có thể hiểu tổng số lượng kết bạn thực tế của UID1). Như vậy Recall chính là tỉ số giữa số lượng dự đoán đúng chia cho số kết bạn thực tế của UID1.

Recall=

* Ta có Biểu đồ thể hiện sự phụ thuộc của Precision và Recall với độ dài danh sách dự đoán:



* Để giải thích cho sự phụ thuộc trên, ta có thể phát biểu như sau:
  + Khi độ dài danh sách gợi ý ngắn: Khi đó tổng ( True Positive + False Positive) nhỏ, đồng thời danh sách gợi ý sẽ đưa ra các UID theo thứ tự giảm dần của h, nên Precision sẽ cao. Còn tổng ( True Positive + False Negative) không phụ thuộc vào độ dài danh sách gợi ý, nó là cố định vì nó là số kết bạn thực tế của UID1, nên khi ta dự đoán ít recall sẽ nhỏ.
  + Khi độ dài danh sách gợi ý dài: Khi đó tổng ( True Positive + False Positive) lớn, đồng thời các UID đưa ra càng về sau càng có h ( xác suất kết bạn giảm), khi đó tổng ( True Positive + False Positive) tăng đều còn số lượng dự đoán đúng (True Positive) tăng chậm dẫn đến Precision sẽ nhỏ dần. Còn tổng ( True Positive + False Negative) vẫn không thay đổi mà số lượng dự đoán đúng (True Positive) tăng lên ( dù chậm) sẽ khiến Recall tăng.