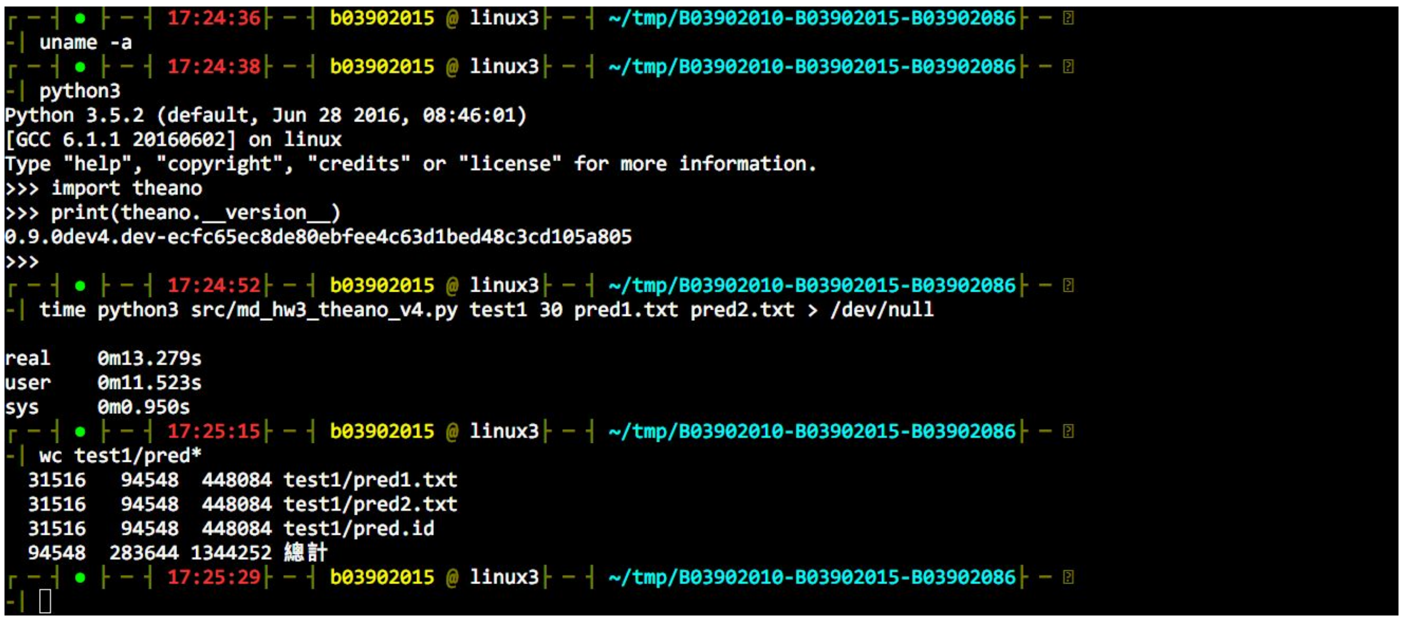
Machine Discovery Homework2 Report

Team members

- B03902010 耿宗揚
    - Model Design
    - Program Optimization
- B03902015 簡瑋德
    - Model Design
    - Report
- B03902086 李鈺昇
    - Feature Extraction Testing
    - Report

Environmental Settings

- Linux linux3 4.8.4-1-ARCH #1 SMP PREEMPT
- Python 3.5.2
- theano 0.9.0
- Screenshot



```
┌─┤ ● ├─┤ 17:24:36├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ▯
-| uname -a
┌─┤ ● ├─┤ 17:24:38├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ▯
-| python3
Python 3.5.2 (default, Jun 28 2016, 08:46:01)
[GCC 6.1.1 20160602] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import theano
>>> print(theano.__version__)
0.9.0dev4.dev-ecfc65ec8de80ebfee4c63d1bed48c3cd105a805
>>>
┌─┤ ● ├─┤ 17:24:52├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ▯
-| time python3 src/md_hw3_theano_v4.py test1 30 pred1.txt pred2.txt > /dev/null

real    0m13.279s
user    0m11.523s
sys     0m0.950s
┌─┤ ● ├─┤ 17:25:15├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ▯
-| wc test1/pred*
  31516   94548  448084 test1/pred1.txt
  31516   94548  448084 test1/pred2.txt
  31516   94548  448084 test1/pred.id
  94548  283644 1344252 總計
┌─┤ ● ├─┤ 17:25:29├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ▯
-| ▯
```

Model

- Assumptions
    - For each pair (u, i), there's a feature vector $v \in R^8$ representing the attributes of the pair
    - $P(\text{pair}(u, i) = \text{true} | w, b) = \sigma(w^T v - b)$, where $w \in R^8$ is the weight vector, $b \in R$ is the bias and $\sigma(s) = \frac{1}{1+e^{-s}}$ is the sigmoid active function
    - About 50% of the pairs in the prediction file are true, while the others are false

- Features

  We only consider those items with nonzero link counts, and omit the others.

    - Item count ratio of user: $\frac{\#\text{ items owned by user}}{\text{total link counts of all items}}$
    - $1[\text{user is one of the owners of item}]$
    - $1[\text{user is a friend to any owner of item}]$
    - $1[\text{any owner of item is a friend to user}]$
    - $1[\text{the item belongs to any category of some item the user owns}]$
    - Ratio of users that like this item: $\frac{\text{the likes of the item}}{\text{number of users}}$
    - $\frac{\#\text{ people who are friends to the user}}{\#\text{ users}}$
    - $\frac{\#\text{ people whom the user is a friend to}}{\#\text{ users}}$

  * We consider friendship to be directed, so the 3rd and 4th features are considered separately. Likewise, the 7th and 8th features are considered separately.

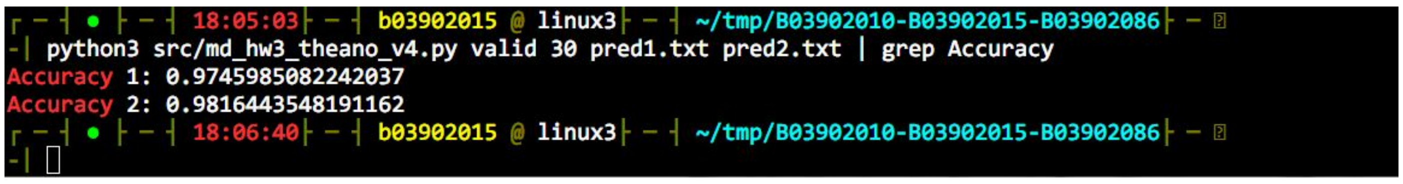
- Parameters Initialization
  Let $w_0 = [1, 1, \ldots, 1]^T$ with length 8 (number of features) and $b_0 = 1$.
  This is due to our belief that all the features are in positive correlation with the predicted probability.

- Cost/Loss Function
  - We want to maximize the probability difference between the pairs that seem to be true and the other pairs
  - $L = -\sum_{p \in D} P(p = \mathrm{true} | w, b) + \sum_{p \in D'} P(p = \mathrm{true} | w, b)$, where D is the set of pairs with higher probability, and $D' = \{\text{pairs in prediction file}\} \setminus D$ and $|D| \approx |D'|$

- Updating Parameters
  - We use gradient descent algorithm to update the parameters
  - In each interation, we calculate the cost function for all the pairs in prediction file with parameters w and b
  - Update $w' = w - \eta \frac{\delta L(D, D', w, b)}{\delta w}$ and $b' = b - \eta \frac{\delta L(D, D', w, b)}{\delta b}$, where η is the learning rate (value = 0.0005)

Performance

- Iteration = 30
- Threshold = The Median of $P(p_1), P(p_2), \ldots, P(p_n)$, where $p_1, p_2, \ldots, p_n$ ∈ prediction file
- Case 1
  - $p = \mathrm{true}$ if and only if $P(p) > \mathrm{Threshold}$
  - Accuracy = 97.5%
- Case 2
  - $p = \mathrm{true}$ if and only if $P(p) \geq \mathrm{Threshold}$
  - Accuracy = 98.1%
- ScreenShot



```
┌─┤ ● ├─┤ 18:05:03├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ⍰
└│ python3 src/md_hw3_theano_v4.py valid 30 pred1.txt pred2.txt | grep Accuracy
Accuracy 1: 0.9745985082242037
Accuracy 2: 0.9816443548191162
┌─┤ ● ├─┤ 18:06:40├─┤ b03902015 @ linux3├─┤ ~/tmp/B03902010-B03902015-B03902086├─ ⍰
└│ ▯
```

References

- theano (http://deeplearning.net/software/theano/): A Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently
- ACM 2013 paper (http://dl.acm.org/citation.cfm?id=2487614): Unsupervised link prediction using aggregative statistics on heterogeneous social networks