Social Network Analysis - Link Prediction

MSLab Wei-Ming

2014 tutorial

Schedule

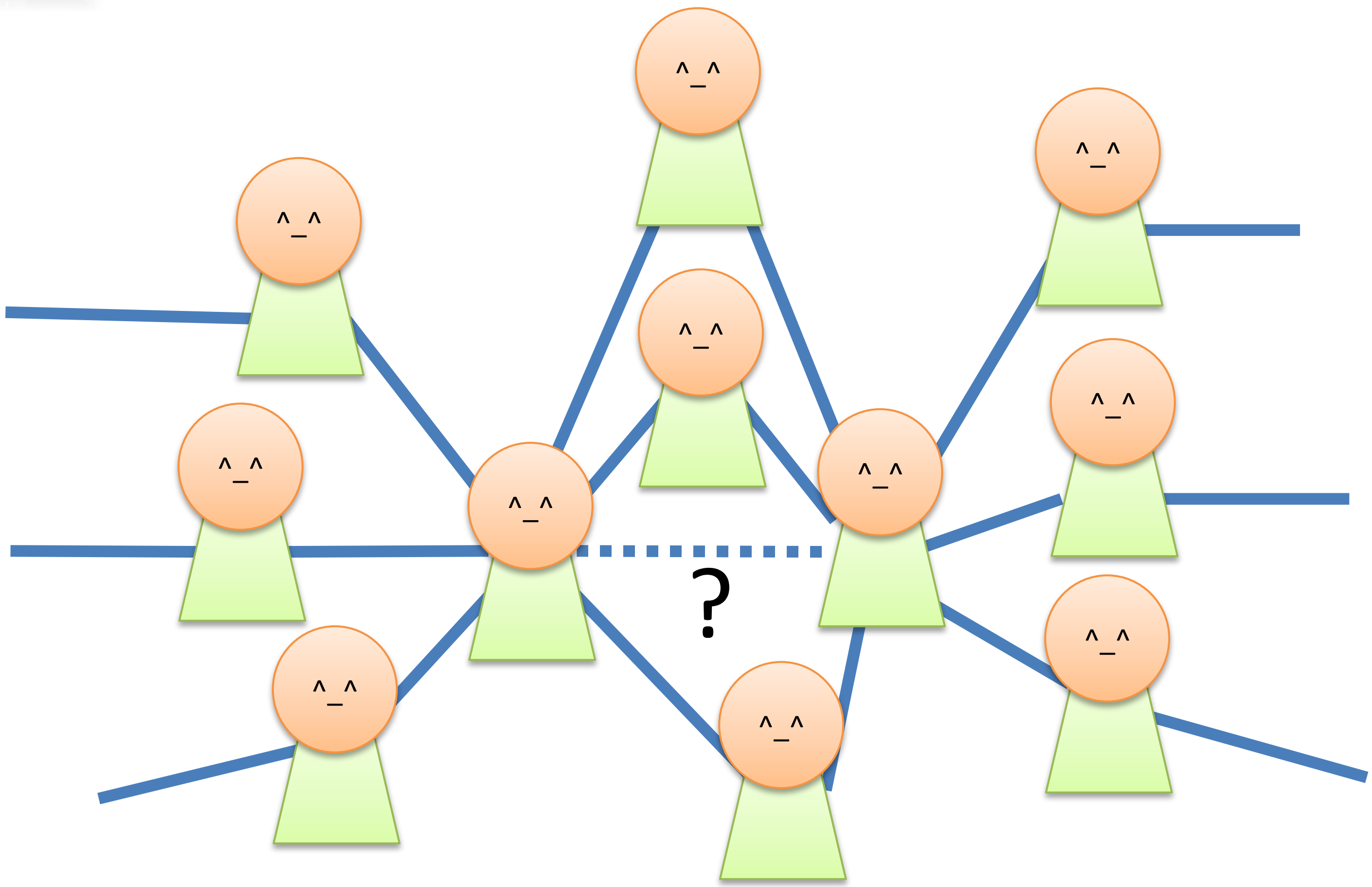
00:00 ~ 00:45	Introduction to final practice Lecture Reviewing sample codes
00:45 ~ 02:15	Practice and Lecture
02:15 ~ 02:30	Discussion and QA

Outline

- What is Link Prediction problem.
- Introduction to final practice
- How to make link prediction
  - What is machine learning?
  - Workflow of data-driven approach
  - Feature Extraction

Machine Discovery Social Network

What is Link Prediction problem?



What is Link Prediction problem?

- Predict the information on edges.
- 1. **Link existence prediction**: to classify whether each edge is 0(not exist) or 1(exist)
- 2. **Link type classification**: to classify the type of edges (e.g. student-teacher, student-student …)
- 3. **Link regression**: to predict the weight of link (importance, rating …)

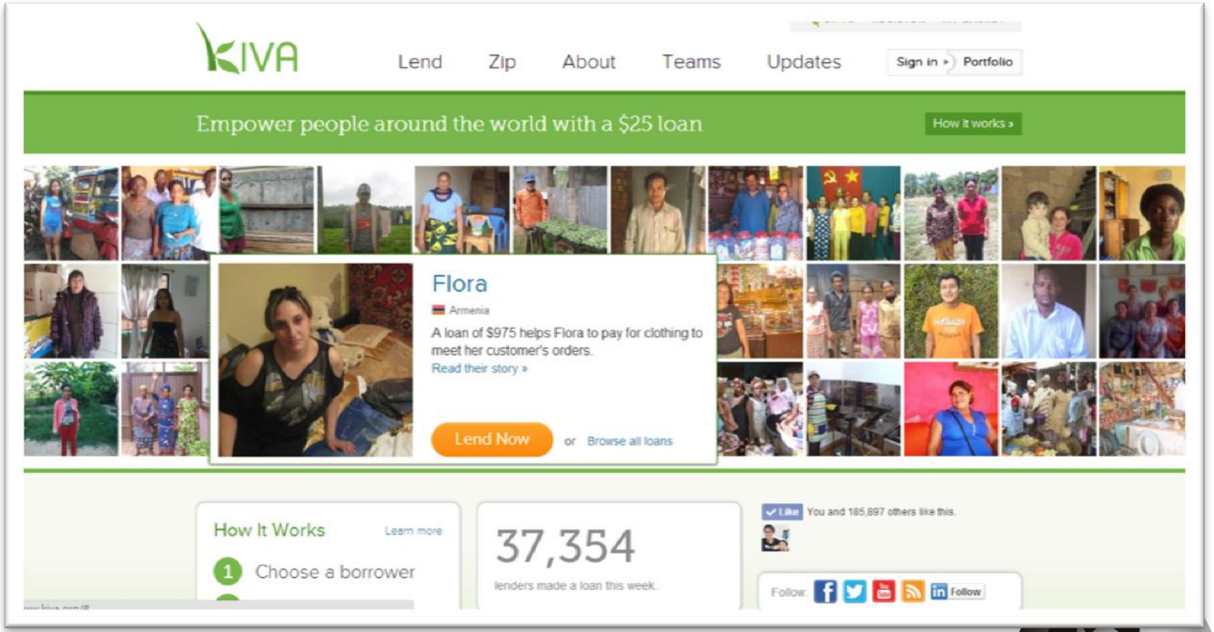
Outline

- What is Link Prediction problem.
- Introduction to final practice
- How to make link prediction
  - What is machine learning?
  - Workflow of data-driven approach
  - Feature Extraction

Machine Discovery Social Network

Data description(1)

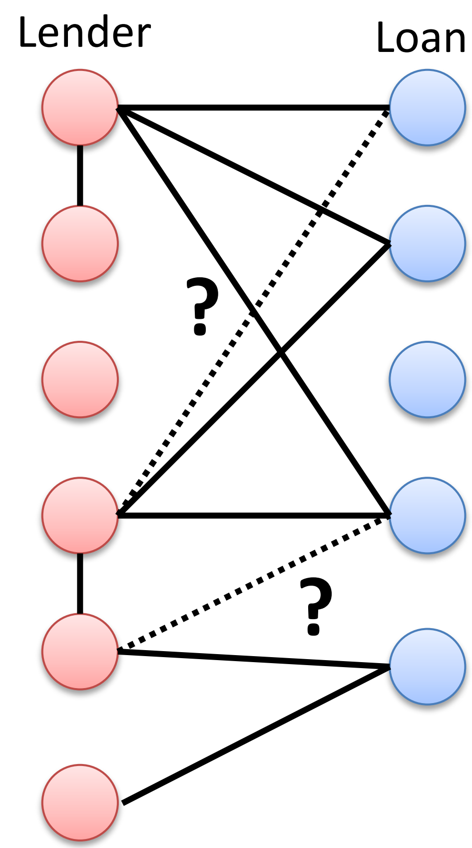
- Kiva (http://www.kiva.org/) is a non-profit organization which aims at providing a micro-loan crowding-sourcing platform to alleviate poverty.



Data description(2)

- Task: to predict whether a lender will lend to certain loan or not
  - 2014/01/08 - 2014/01/15
  - #Lender: 15,870
  - #Loan: 4,872
  - #Transaction: 38026
    - Training: 26617
    - Testing: 11409 (+) and randomly sample 11409 (-)

train.csv test.csv



Data description(3)

- Lender's information:
  - lender_id
  - country_code
  - inviter_id
  - invitee_count
  - loan_count
- lender.csv
- Loan's information:
  - loan_id
  - Sector
  - Amount
  - Borrowers
  - Country
  - geo
- loan.csv

Machine Discovery
Social Network

Evaluation

- Accuracy
  - $Accuracy = \sum_{(u,i) \in Test} \frac{[y_{ui} = \hat{y}_{ui}]}{|Test|}$

- Baseline: all positive / all negative : 0.5
  - test.ans

Machine Discovery
Social Network

Outline

- What is Link Prediction problem.
- Introduction to final practice
- How to make link prediction
  - What is machine learning?
  - Workflow of data-driven approach
  - Feature Extraction

How to make link prediction

Two different strategies:

1. **Knowledge-driven strategy**: produce some rules for prediction
   - E.g.: if degree <10, then predict it as a college node.
   - Cons: Need domain experts. Could miss patterns that were unknown.
2. **Data-driven approach**: Machine learning approach
   - Suppose you are given a social network, while some nodes have labels (professors, department, keywords… etc.) and some don’t.
   - The goal is to predict the labels of some of them.

Outline

- What is Link Prediction problem.
- Introduction to final practice
- How to make link prediction
  - What is machine learning?
  - Workflow of data-driven approach
  - Feature Extraction

What is machine learning?

- How can you distinguish apples / orange?

- Color / shape …

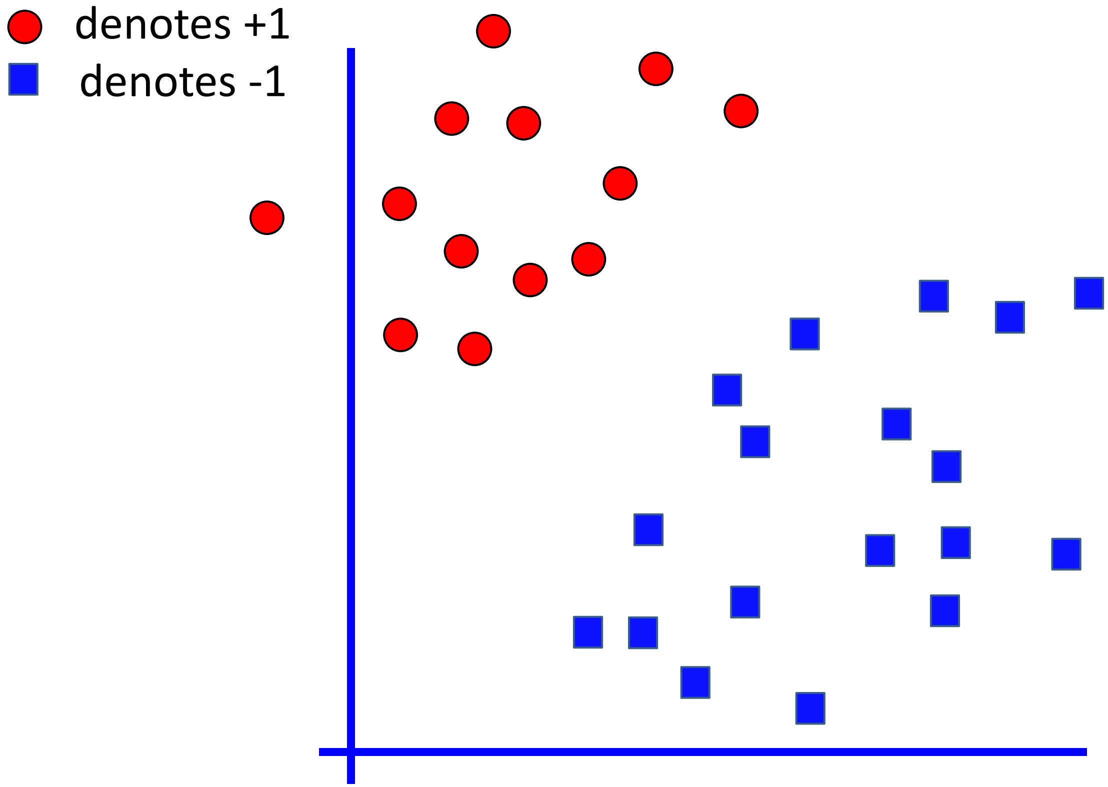
What is machine learning?

- But if we have 1 billion pictures with 200 types of fruit to be classified ?
- Let machine(computer) learn it automatically!

- y = f(x)
- X = { picture of fruit }
- Y = { type of fruit } e.g. ‘apple’, ‘orange’ …
- Given lots of (x, y) pairs, learn y=f(x)

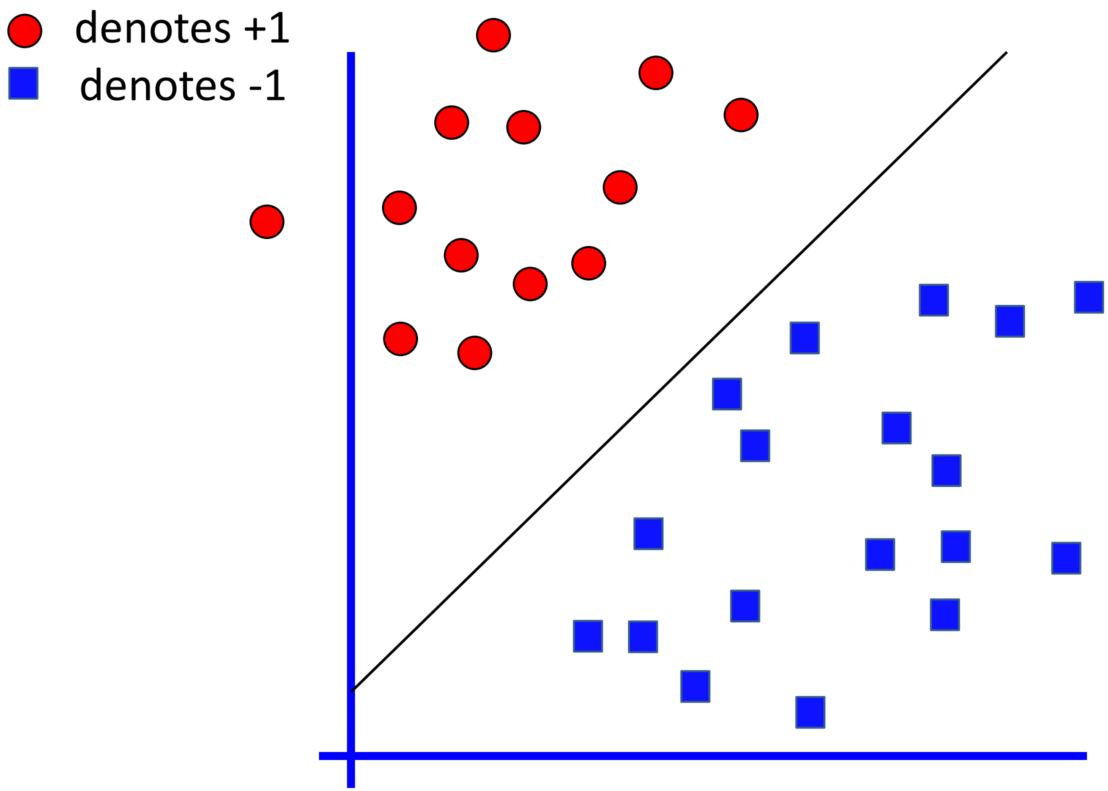
Machine Discovery
Social Network

Brief introduction to common classifier

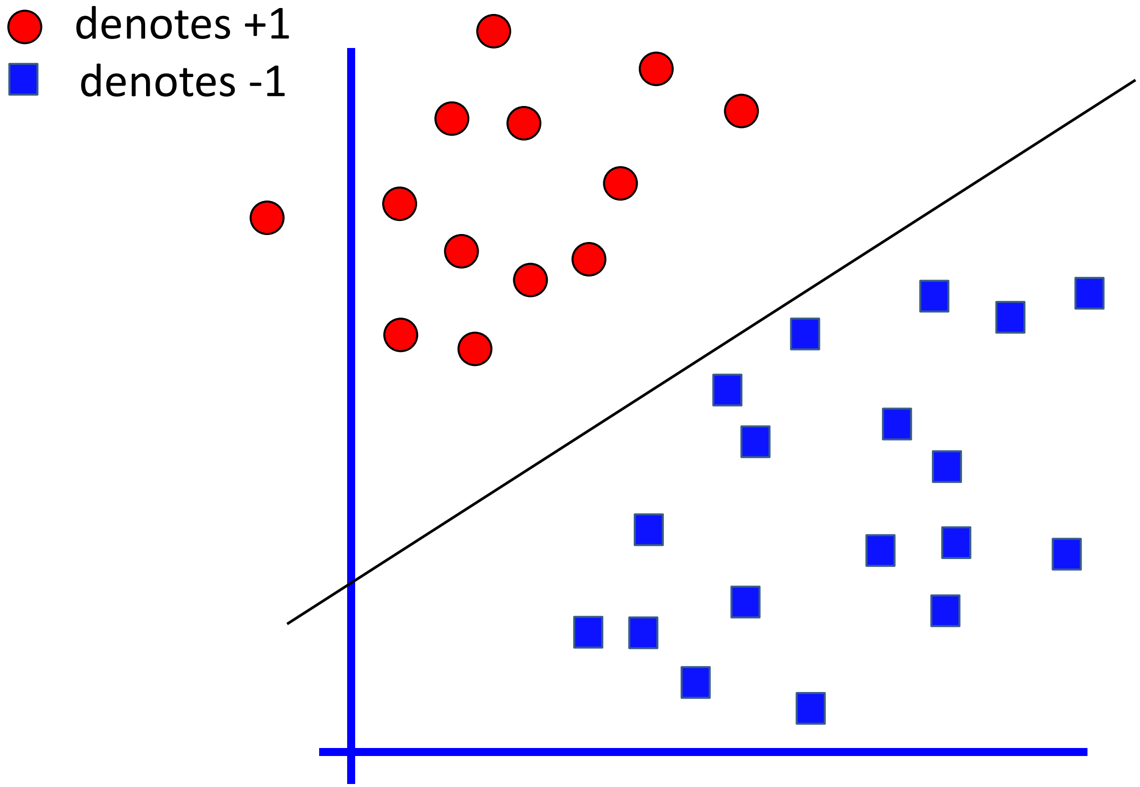


How can we classify this data?

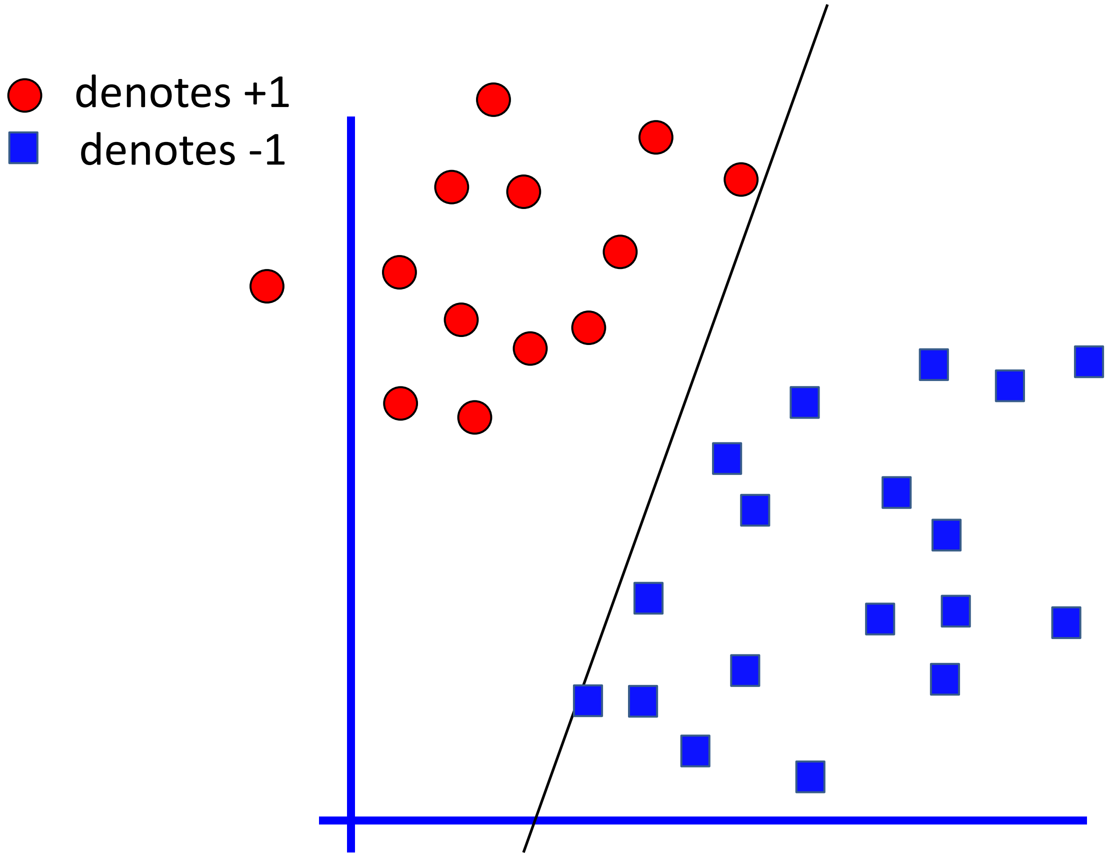
Brief introduction to common classifier



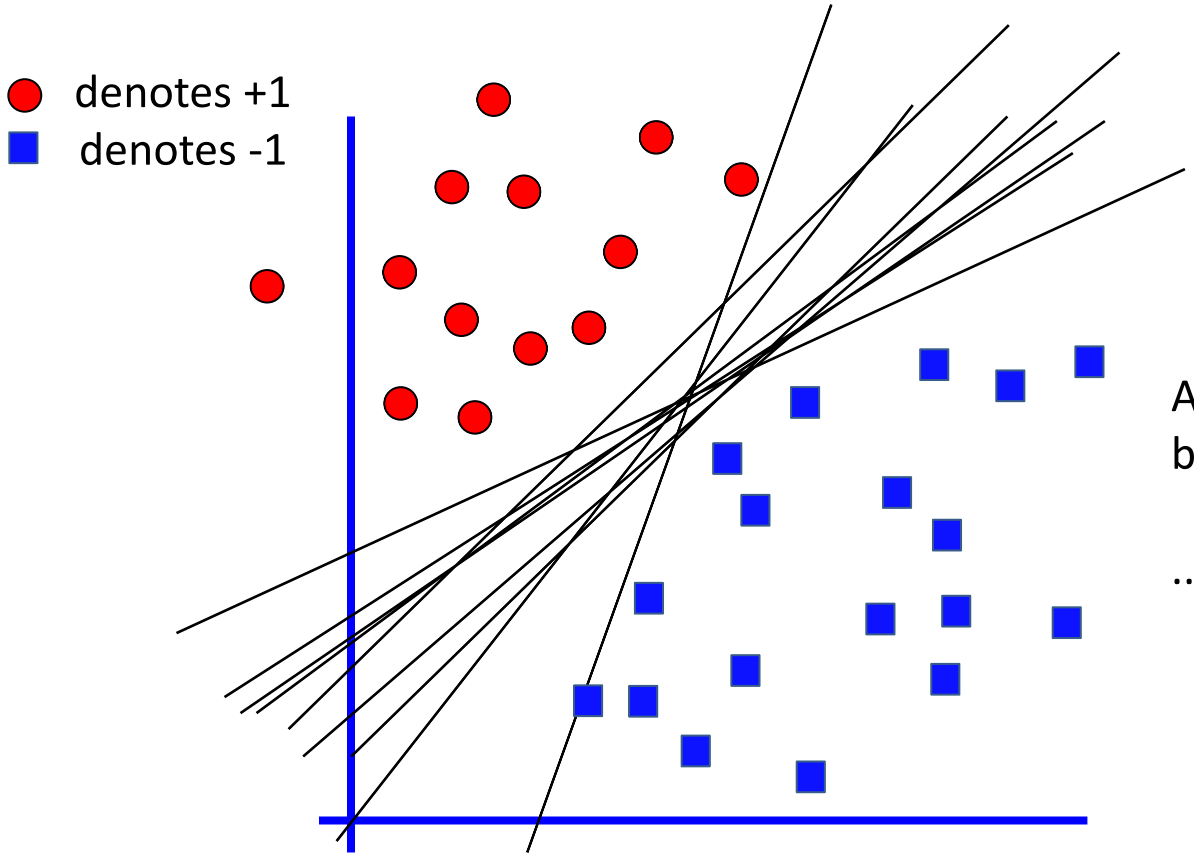
Brief introduction to common classifier



Brief introduction to common classifier



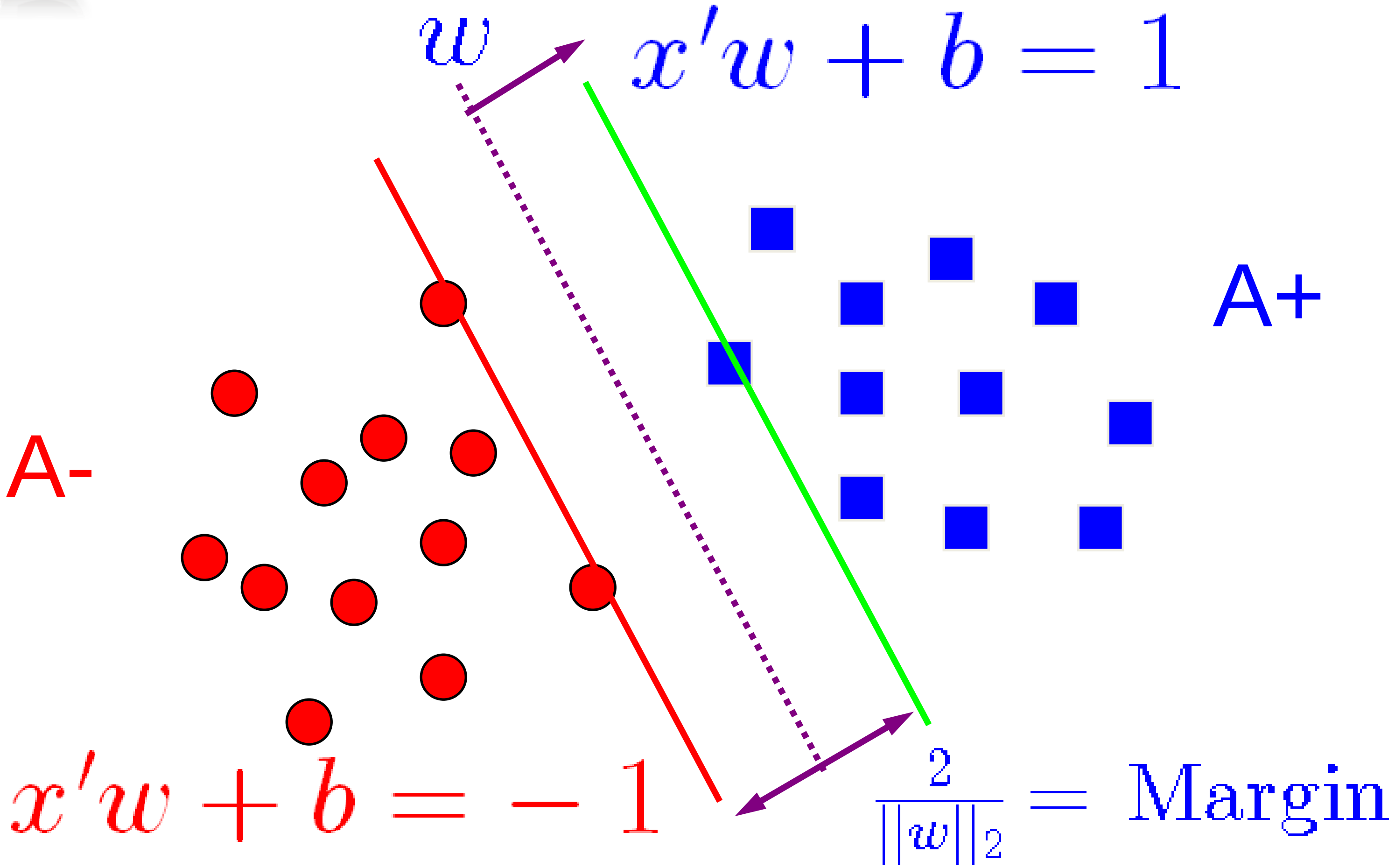
Brief introduction to common classifier



Any of these would be fine..

..but which is best?

Support Vector Machine



Package of SVM

- Liblinear
  - http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- LibSVM
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Outline

- What is Link Prediction problem.
- Introduction to final practice
- How to make link prediction
  - What is machine learning?
  - Workflow of data-driven approach
  - Feature Extraction

Workflow - Data-driven approach(1)

- 1. Determine **what is** considered to be the **'instance' for classification**
  - Node or link?
  - Multi-class or single class?

- 2. **Obtaining features for the instance**
  - Topological features (e.g. degree, centrality)
  - Attributes of instances (e.g. time info, relation type of edges) Attributes of instances (e.g. time info, relation type of edges)
  - Social features (the information about neighbors)

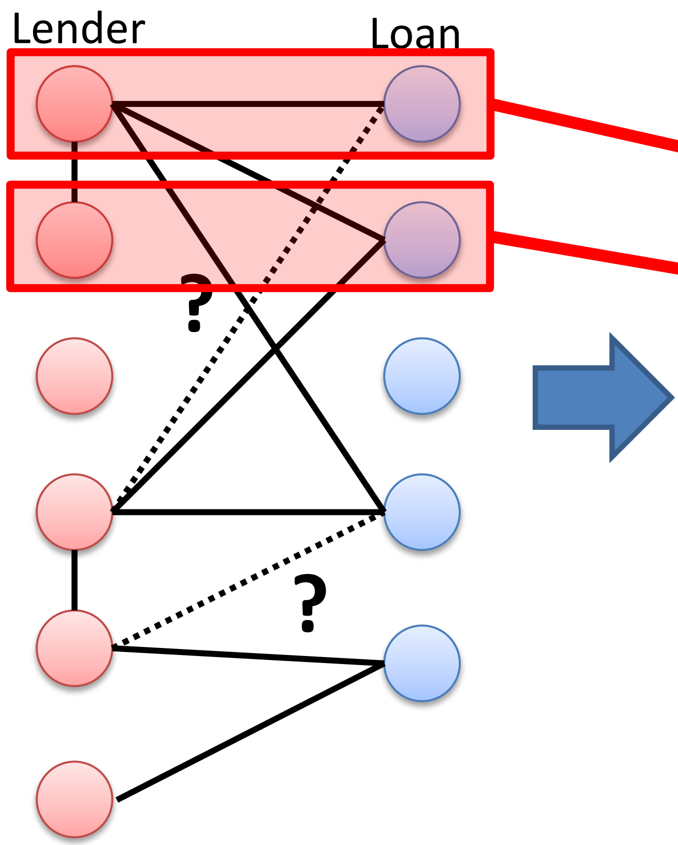
Machine Discovery
Social Network

Workflow - Data-driven approach(2)

- 3. **Determine a classifier to use**.
  - E.g. Liblinear, LibSVM, Weka ...

- 4. **Train a classifier and evaluate the results using held-out data** (i.e. data not used for training). If the performance is not satisfiable, go back to (2) and (3).

Goal



- +1 1:0.34 2:0.96 3:-0.5
- -1 1:0.9 2: 0.1 3: 0.3
- +1 1:0.28 2:0.77 3:-0.6
- -1 1:1.0 2: 0.2 3: 0.4
- …

Outline

- What is Link Prediction problem.
- Introduction to final practice
- How to make link prediction
  - What is machine learning?
  - Workflow of data-driven approach
  - Feature Extraction

Feature extraction(1)

- Topological feature
  - Shortes path length
  - Edge Embeddedness
    - Number of common neighbors of node u and v
  - Jaccard's coefficient
  - Adamic/Adar
  - Preferential attachment
  - Katz score
  - Hitting time
    - expected umber of steps of random walk from x to y.

Notation

- G: $graph$
- $\Gamma(x)$: $the\ set\ of\ node\ x's\ neighbors$
- $|\Gamma(x)|$: $the\ degree\ of\ node\ x$
- $Length(p)$: $length\ of\ path\ p$

- $score(x, y)$: $the\ feature\ score\ of\ node\ x$ $and\ node\ y$

Shortest-Path

- The length of shortes path from node x to node y

- $score(x, y) =$
  $(-1) \times Length(shortest\_path(x, y))$

Machine Discovery Social Network

Edge Embeddedness

- The number of common neighbors of node x and y

$$score(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

Jaccard's coefficient

$$score(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

- Measure how likely a neighbor of x is to be a neighbor of y and vice versa to be a neighbor of y and vice versa

Adamic/Adar

$$score(x,y) = \Sigma_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|}$$

- Assigns large weight to common neighbors z of x and y which themselves have few neighbors $|\Gamma(z)|$

Preferential Attachment

$$score(x, y) = |\Gamma(x)| \times |\Gamma(y)|$$

- Researchers found empirical evidence to suggest that co-authorship is correlated with the product of the neighborhood sizes

Feature extraction(2)

- Latent topological feature
  - Graph factorization

Feature extraction(3)

- Content-based feature
  - Totally depends on your DATA
  - There some small tips

- Dummy variable (indicator variable)
- Scaling

Dummy variable

node	A	B	C
Type	‘person’	‘animal’	‘Item’

- how about A=1, B=2, C=3 ?
  - But $y = w^T x,\ the\ value\ of\ x\ matters$
- A=> 1:1 2:0 3:0
- B=> 1:0 2:1 3:0
- C=> 1:0 2:0 3:1

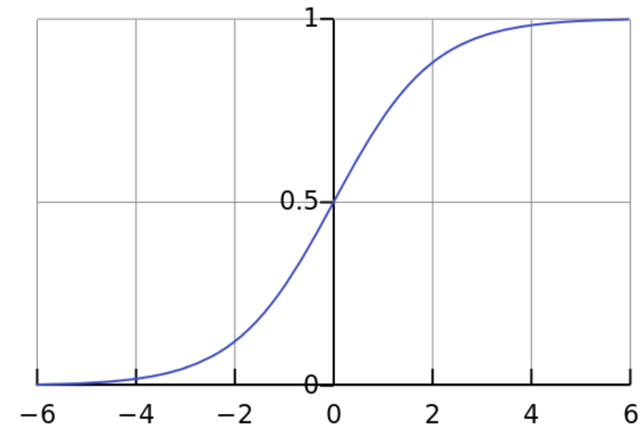
Scaling

- Since $y = w^T x$ …
- if $x = (10^{40}, 10^{20} \ \ldots)$
  - $w^T x$ could be overfloat !!

  - $x_1 = (10^{40}, 10^{20}, \ldots)$
  - $x_2 = (10^{39}, 10^{15}, \ldots)$
  - $x_3 = (10^{38}, 10^{10}, \ldots)$
  - ….

[-1, 1] [-1, 1]

- Divided by MAX
- Sigmoid function



Coding Time！

- https://github.com/barry800414/sample-codes/archive/master.zip

Task 1

- task1/extract_feature.py - Line 22
- Please find shortest path from node x to node y

- Hint: Please browse networkx documents

Task 2

- task2/extract_feature.py – Line 40
- Please find the number of common neighbors of node x and node y

- Hint: Please browse networkx documents

Task 3

- task3/extract_feature.py – Line 51
- Calculate jaccards_coefficients for node x and node y

- Hint: python has built-in type : set()
- There are “union” and “intersect” operation

Task 4

- task4/extract_feature.py – Line 62
- Complete adamic_adar_score function

Task 5

- task5/convert_feature.py – Line 17
- Normalize the column by the maximum of absolute value in a column

- Hint: python has built in max function

Task 6

- task6/convert_feature.py – Line 36
- Convert categorical feature to dummy / indicator variable

Machine Discovery Social Network

Task 7

- Using liblinear to train a model by training data, and predict the value on testing data
- See the accuracy value

QA & Discussion