Machine Learning for IT company (Matrix completion & Active Learning)

Makoto Yamada
myamada@i.Kyoto-u.ac.jp

KYOTO UNIVERSITY
KYOTO JAPAN
FOUNDED 1897

Types of Data

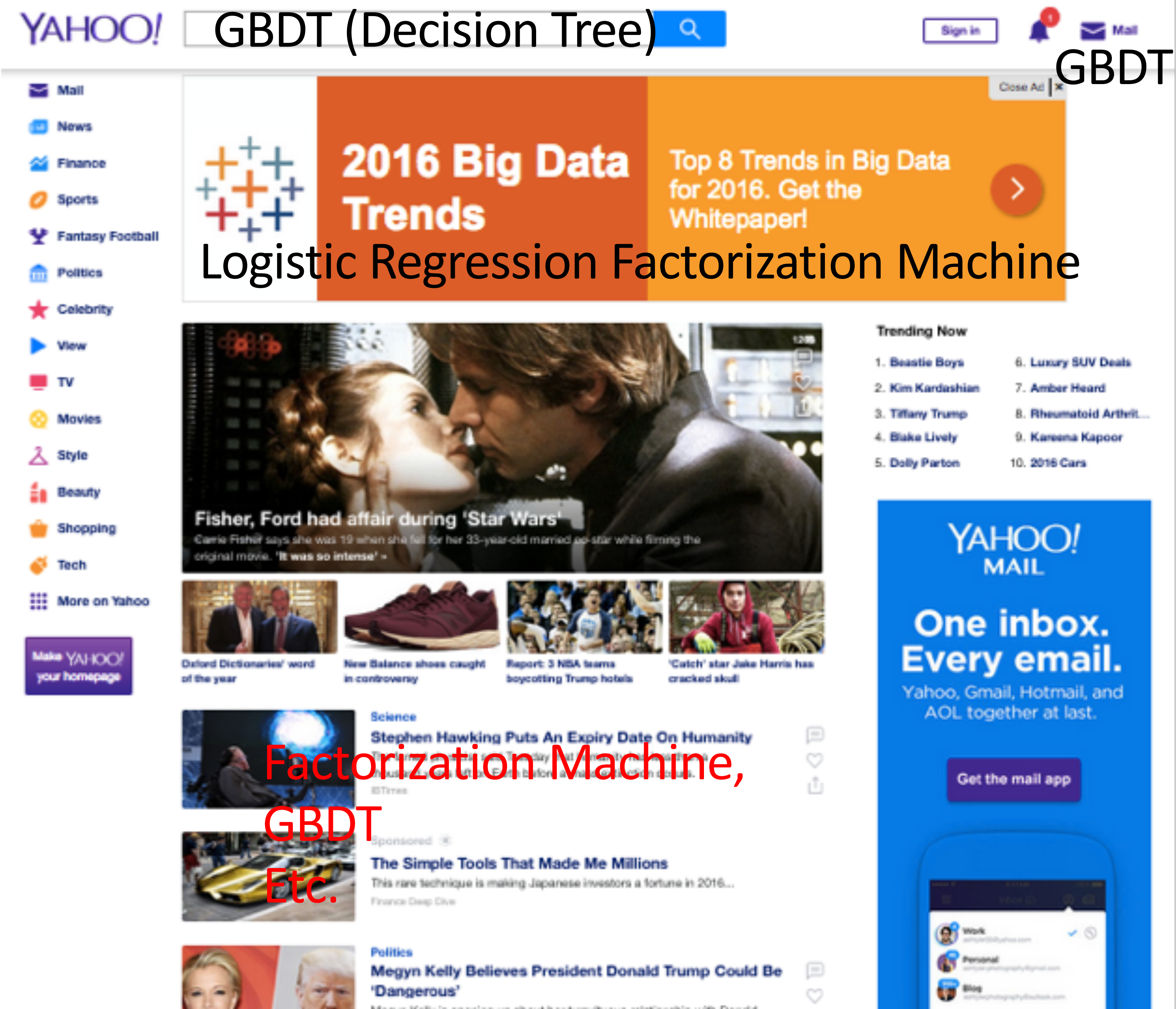
- Click data
  - News (Yahoo, Google)
  - E-commerce (Yahoo Auction, eBay, Alibaba, Amazon)
  - Ad recommendation (Google, Criteo, Cyber Agent)
  - Video sharing (Youtube, ニコニコ, SnapChat)
  - Image sharing (Facebook, Flickr, Picasa)
- Text data (web page, item description, etc.)

GBDT (Decision Tree)

GBDT

Logistic Regression Factorization Machine

Factorization Machine,
GBDT
Etc.



IBIS 2016

ibis 2016
ibis 2016 **ripley**
ibis 2016 **conference**
ibis 2016 **las vegas**
ripley ibis 2016

Query auto completion
GBDT (Decision Tree)
CRF

IBIS 2016. The premier BusinessObjects training and information exchange of the year hosted by InfoSol.

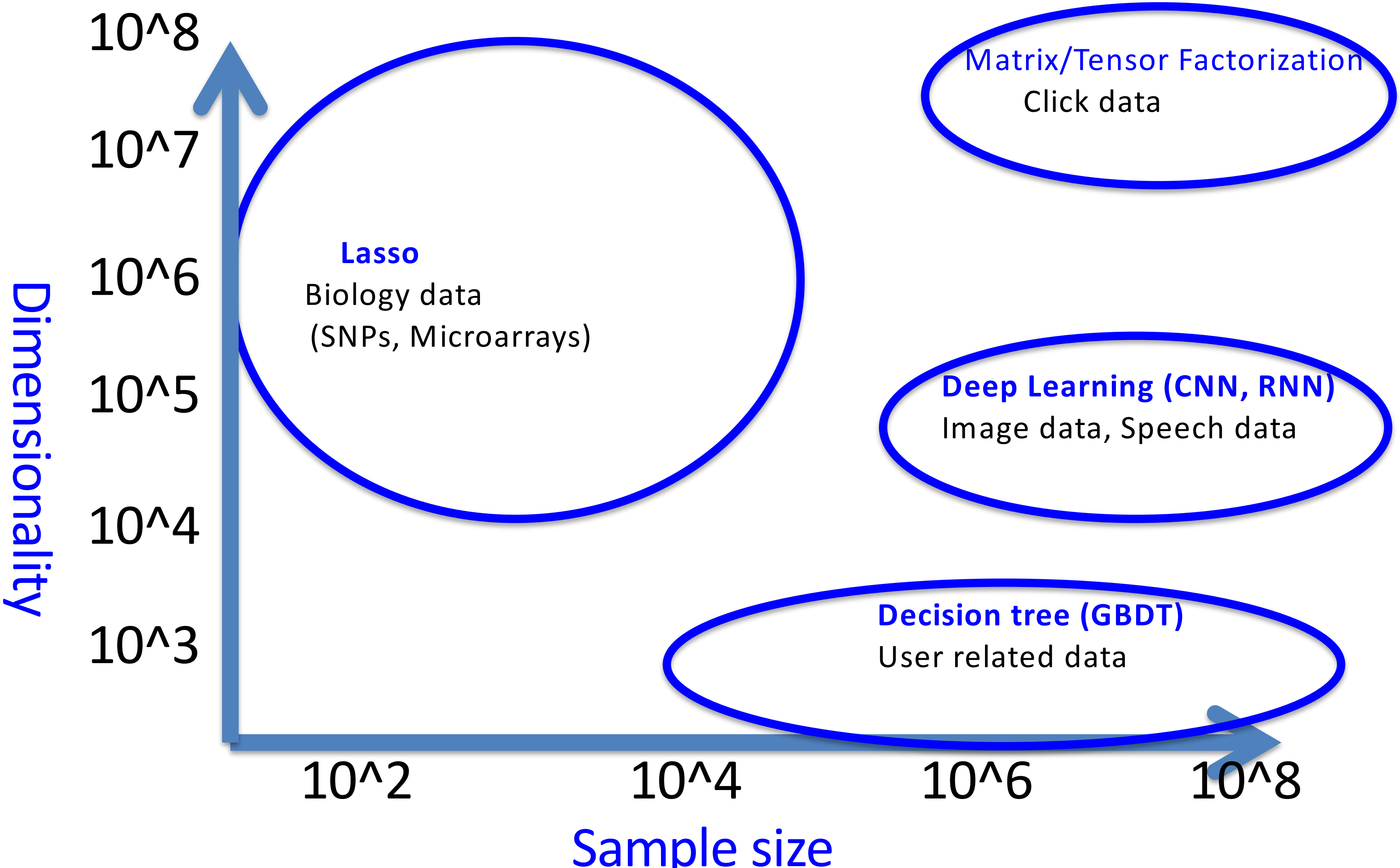
IBIS Academy 2016 - HOME
www.ibisacademy.com
IBIS Academy Attendees; **IBIS** Conference **2016**; History of the **IBIS** Academy; Dates to Remember. Dates to Remember. 23 Oct Registration Opens: Partners.

Web ranking
GBDT (Decision Tree)

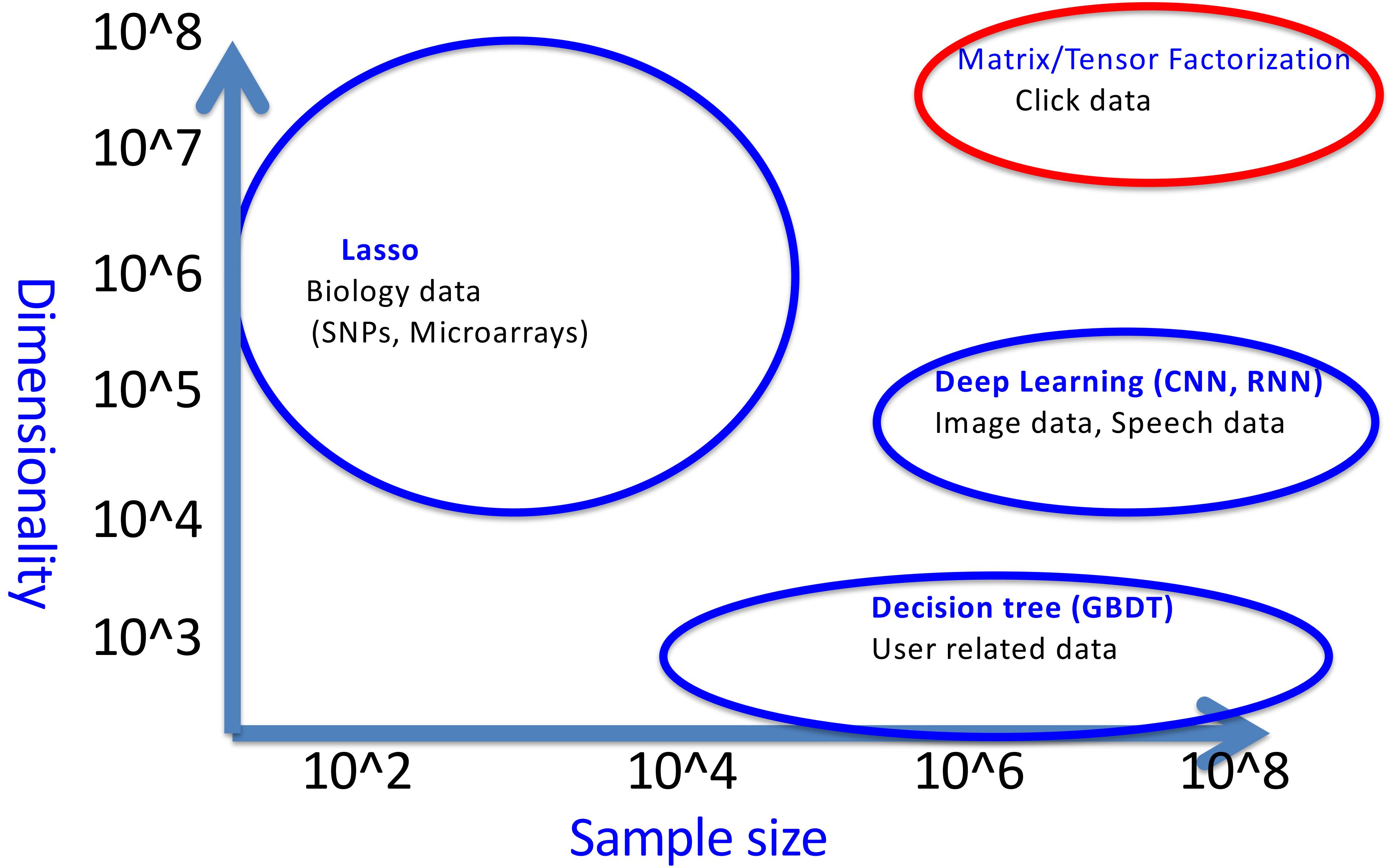
Ibis Demo Tour 2016 | Buy/Try
www.ibiscycles.com/buytry/ibis_demo_tour
Ibis Demo Tour **2016** | Buy/Try. **Ibis** Cycles Inc. designs, develops, sources, distributes, and markets the best bicycles and cycling related products in the world.

IBIS 2016 - Image Results

Data & Related methods

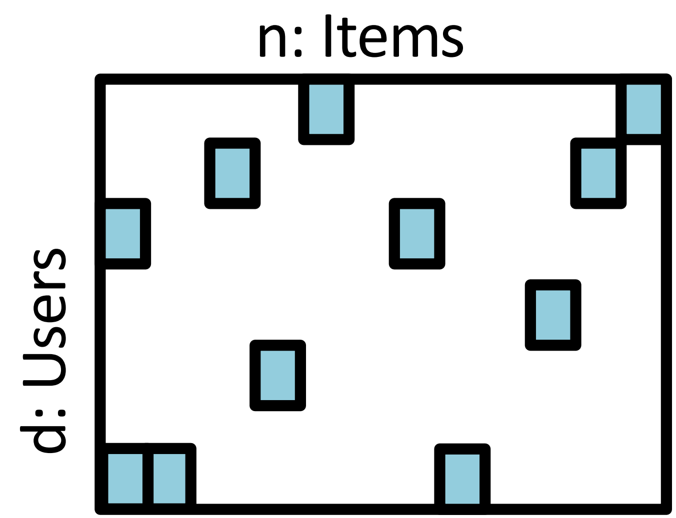


Data & Related methods



High-dimensional Sparse data

- Dimensionality and sample size are both large!
  - Text data
  - Click data
  - Link data



Text classification, Sentiment analysis: Logistic regression

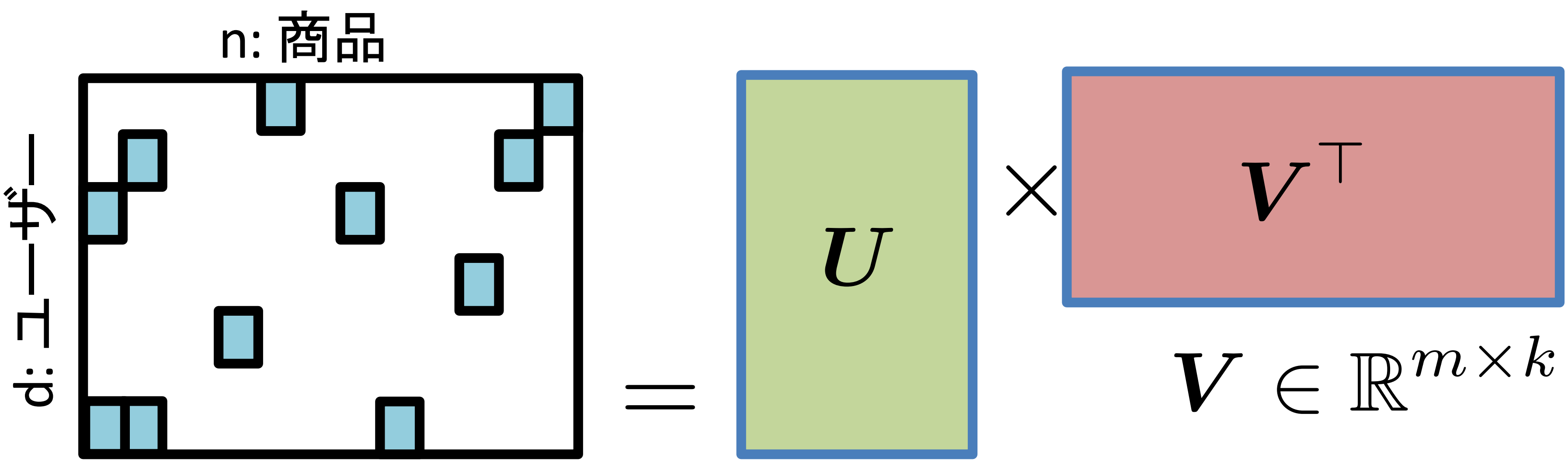
Ad recommendation, item recomendation: Matrix completion

Example of data format

- Movielens data (1M)
- 1::1193::5::978300760
- 1::661::3::978302109
- 1::914::3::978301968
- 1::3408::4::978300275
- 1::2355::5::978824291
- 1::1197::3::978302268
- 1::1287::5::978302039
- 1::2804::5::978300719
- 1::594::4::978302268
- 1::919::4::978301368
- 1::595::5::978824268

Collaborative filtering

- Recommending items using user click information (Amazon, Netflix, etc.)



$\boldsymbol{A} \in \mathbb{R}^{n \times m}$ $= \boldsymbol{U}\times\boldsymbol{V}^\top$

$\boldsymbol{U} \in \mathbb{R}^{n \times k}$

$\boldsymbol{V} \in \mathbb{R}^{m \times k}$

$\boldsymbol{A}$ is a sparse matrix

Singular Value Decomposition

- Fill un-observed element by 0 and do SVD.

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$$

- Use low-rankness to estimate un-observed elements

$$\widehat{\boldsymbol{A}} = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^\top$$

- This approach makes un-observed elements as 0. However, those elements are simply not observed (not zero!)

Alternating Least Squares (ALS)

- Fitting only observed entries.

$$\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{(i,j)\in\Omega} (a_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j)^2 + \lambda_1 \|\boldsymbol{U}\|_F^2 + \lambda_2 \|\boldsymbol{V}\|_F^2$$

- $\boldsymbol{U}$ and $\boldsymbol{V}$ can be alternatingly optimized.

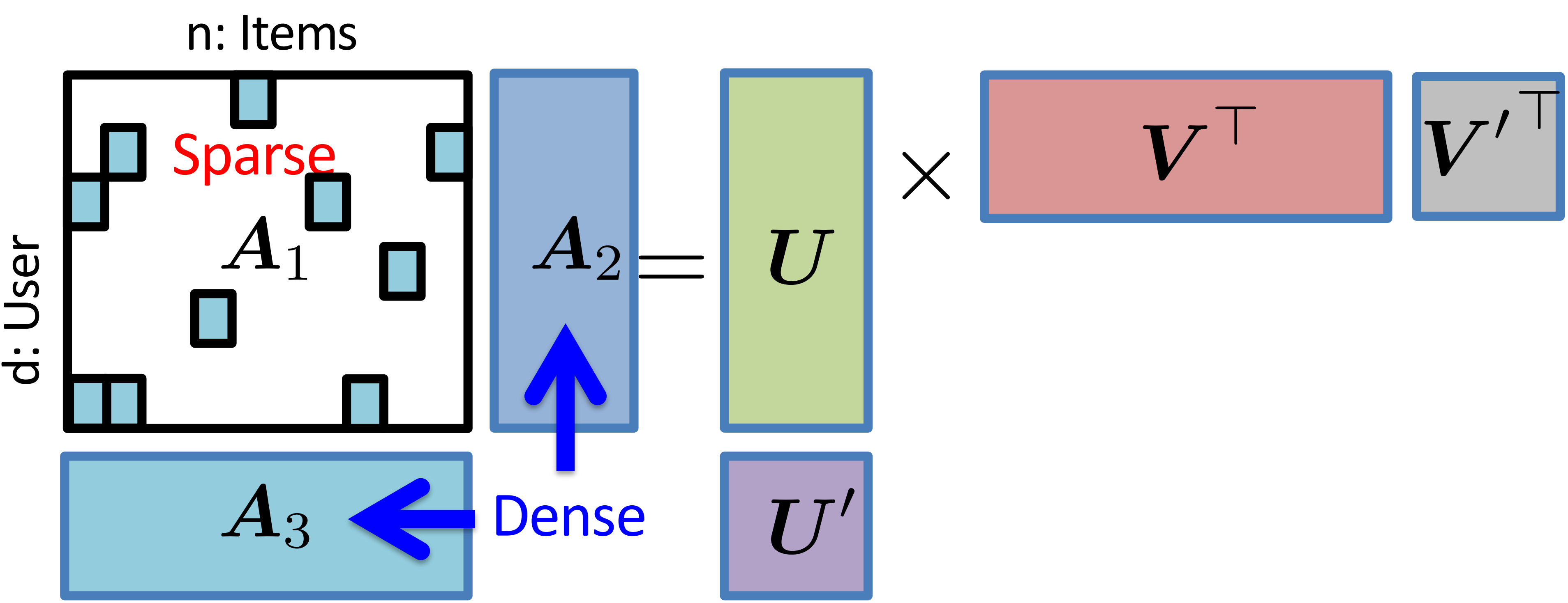
$$\boldsymbol{u}_i = \left(\sum_{(i,j)\in\Omega} \boldsymbol{v}_j \boldsymbol{v}_j^\top + \lambda_1 \boldsymbol{I} \right)^{-1} \sum_{(i,j)\in\Omega} a_{ij} \boldsymbol{v}_j$$

$$\boldsymbol{v}_j = \left(\sum_{(i,j)\in\Omega} \boldsymbol{u}_i \boldsymbol{u}_i^\top + \lambda_2 \boldsymbol{I} \right)^{-1} \sum_{(i,j)\in\Omega} a_{ij} \boldsymbol{u}_i$$

Advanced topic: Cold start problems

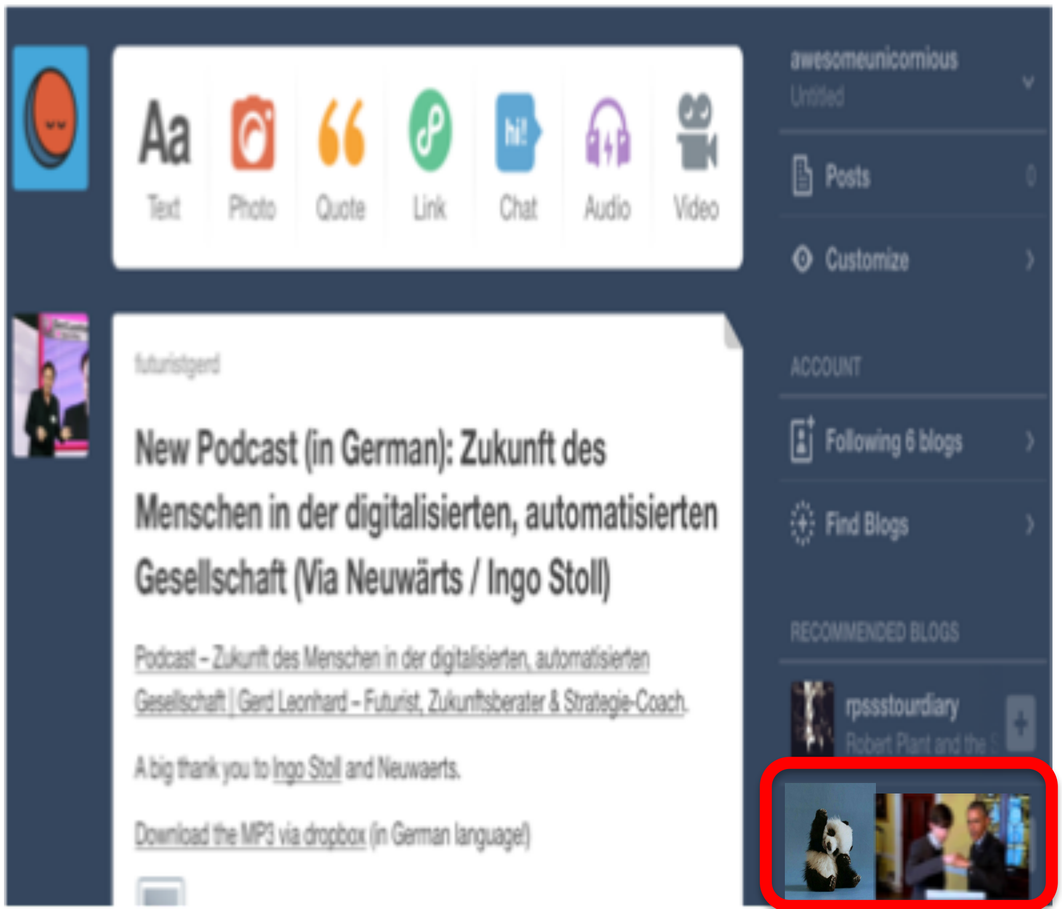
Singh, A. & Gordon, G, KDD 2008

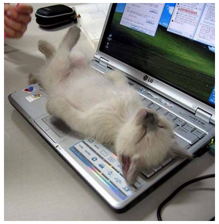
- Cold start: Matrix $\boldsymbol{A}$ is very sparse. Some row (user) or column (item) can be completely missing.

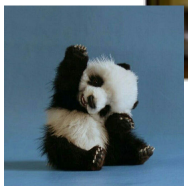


Tumblr Blog recommendation

- Which blog we should recommend to users?

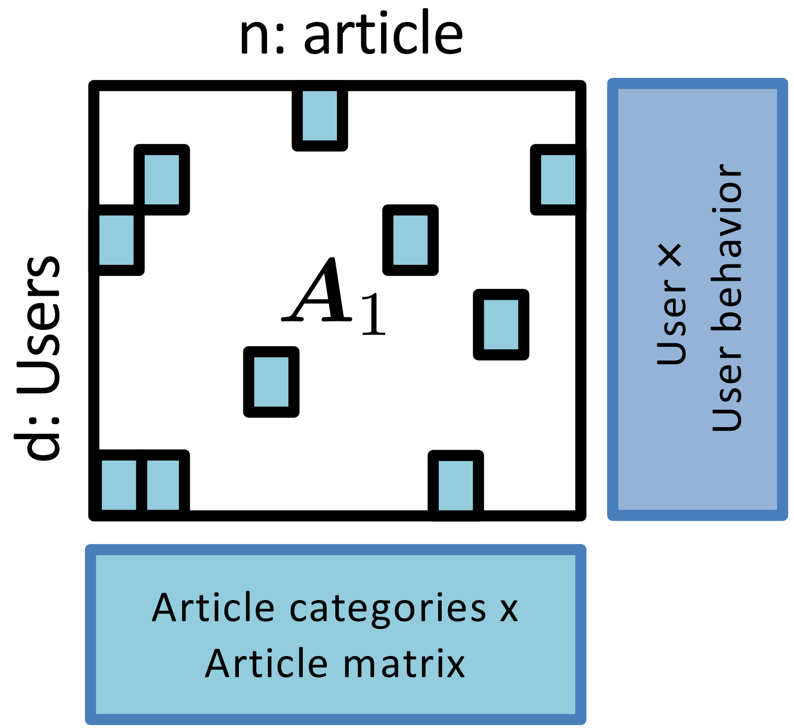






News recommendation

- News recommendation
  - #of users ～180,000
  - #of articles ～750
  - #of categories 34
  - #of Rating(1.4 million)



Method	News-Cold-Start	News-No-Cold-Start
CMF–Hazans	0.27408 ± 0.00016	0.21559 ± 0.00143
SMF	0.29051 ± 0.00074	0.21488 ± 0.00076

Factorization Machine

Rendle, ICDM 2010

- Generalized version of matrix completion
  - Matrix completion + User bias + Item bias
  - We can easily add user information
- Idea (super simple)
  - Solve matrix completion problems by regression
  - (i,j)-th rating input and output can be written as

$$\boldsymbol{x}_i = [\overbrace{0 \cdots 0 \quad \underbrace{1}_{k\text{-th user}} \quad 0 \cdots 0}^{|U|} \; \overbrace{0 \cdots 0 \quad \underbrace{1}_{k'\text{-th item}} \quad 0 \cdots 0}^{|I|}]^\top \in \mathbb{R}^d,$$

$$y_i = [\boldsymbol{A}]_{k,k'}.$$

Factorization Machine

Rendle, ICDM 2010

- Regression model

$$f(\boldsymbol{x};\boldsymbol{w},\boldsymbol{G}) = w_0 + \boldsymbol{w}_0^\top \boldsymbol{x} + \sum_{\ell=1}^{d} \sum_{\ell'=\ell+1}^{d} \boldsymbol{g}_\ell^\top \boldsymbol{g}_{\ell'} x_\ell x_{\ell'},$$

- FM is equivalent to matrix completion

$$\widehat{\boldsymbol{A}}_{k,k'} = w_0 + [\boldsymbol{w}_0]_k + [\boldsymbol{w}_0]_{|U|+k'} + \boldsymbol{g}_k^\top \boldsymbol{g}_{|U|+k'},$$

- We can also handle the cold start problems by simply concatenating user and item information.

Factorization Machine

- Optimization problem:

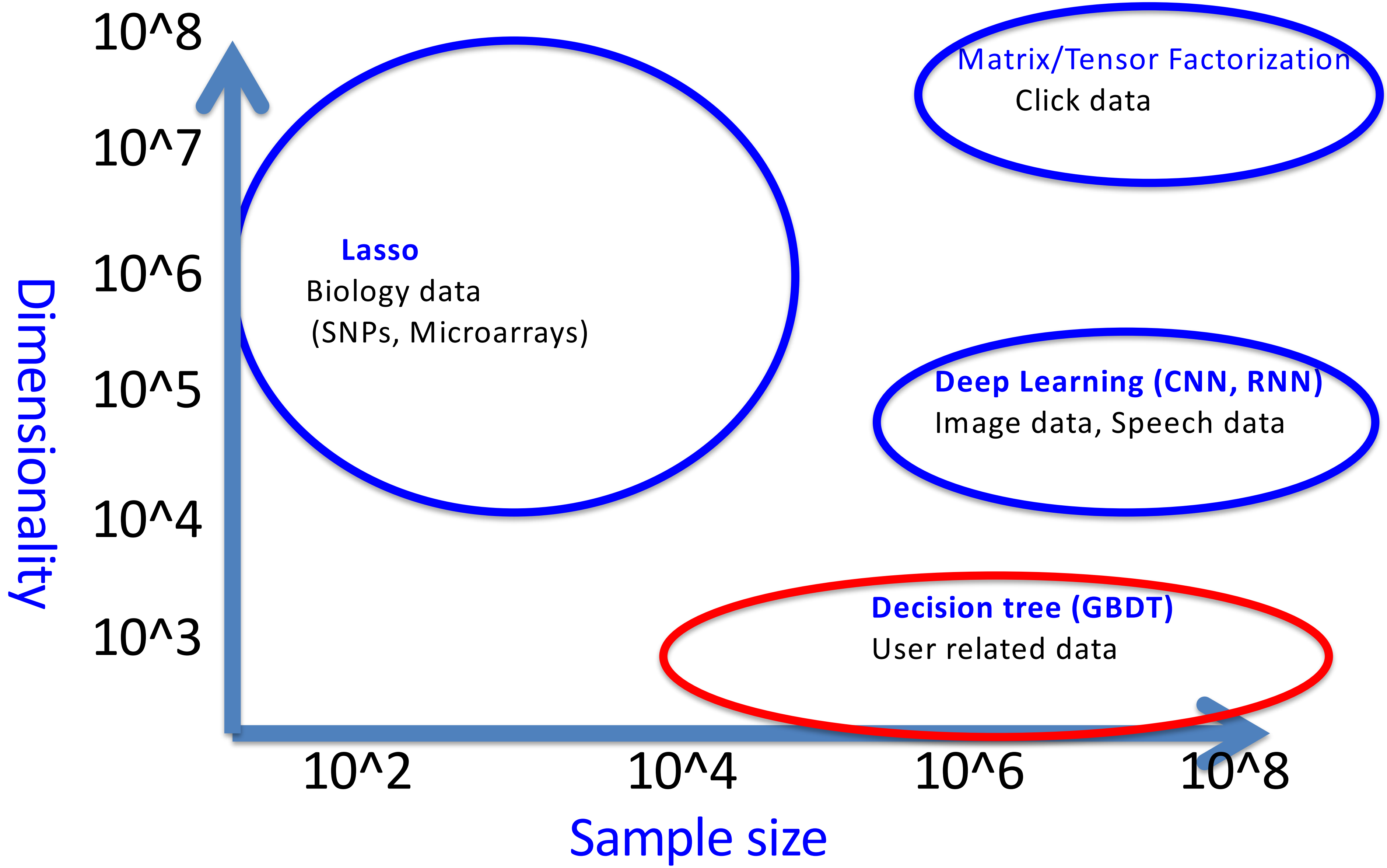
$$\min_{w_0, \boldsymbol{w}, \boldsymbol{G}} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i; w_0, \boldsymbol{w}_0, \boldsymbol{G}))^2$$
$$+\lambda_1 \|w_0\|_2^2 + \lambda_2 \|\boldsymbol{w}_0\|_2^2 + \lambda_3 \|\boldsymbol{G}\|_F^2,$$

- Alternating Least Squares,SGD, Markov Chain Monte Carlo (MCMC).
- Convex optimization version (Blondel, ECML 2015, Yamada KDD 2017)

Factorization Machine Usage

- URL: http://www.libfm.org/
- Movielens data: https://grouplens.org/datasets/movielens/
- ./triple_format_to_libfm.pl -in ml-1m/ratings.dat -target 2 -delete_column 3 -separator "::”
- ./libFM -task r -train ratings.dat.libfm -test ratings.dat.libfm -dim '1,1,8'

Data & Related methods



Low dimensionality & large sample

- The number of samples is larger than that of dimension (n >> d)
  - Images
  - Speech
  - User related data

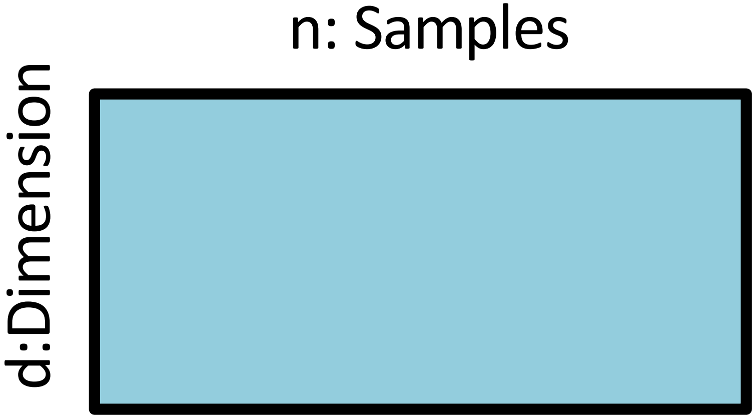


Image and speech recognition: Deep Learning

Spam detection, Web ranking: GBDT (xgboost), Logistic regression

Yahoo Auction

Auction Fraud Detection

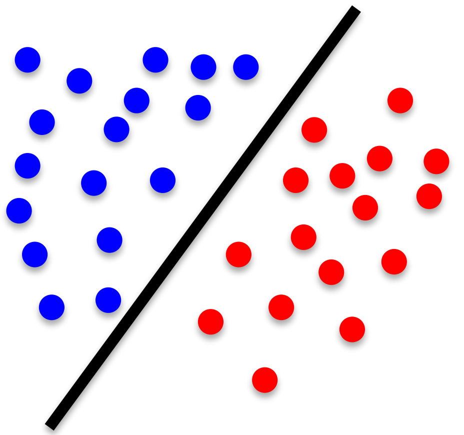
- Example of frauds
  - Selling fake items
  - Do not send items
  - Do a big frauds after gathering trust scores.
  - Etc.
- Detecting fraud is a very important to make users happy!
- Challenge
  - The fraud types changes over season
    - Active learning, transfer learning, etc.
  - Big data (the number of samples can be hundred million of items)

Formulation

- Classification problem (Fraud or non-Fraud)
  - Build a classifier using a labeled data

	Gender	Age	…	Locatio n	Label
User 1	Male	25	…	Tokyo	+1
User 2	Female	20	…	Kyoto	-1
…			…		
User n	Male	36	…	Tokyo	-1

Normal user



Fraud user

Supervised Learning (review)

- Input and output: $\boldsymbol{x} \in \mathbb{R}^d, \ y \in \mathbb{R}$
- Training samples: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y)$
- Goal: Training classifier from the training samples.
- Model (Linear model)

$$f(\boldsymbol{x}; \boldsymbol{w}) = w_1 x_1 + w_2 + x_2 + \ldots w_d x_d = \boldsymbol{w}^\top \boldsymbol{x}$$

- Model parameter

$$\boldsymbol{w} = [w_1, w_2, \ldots, w_d]^\top \in \mathbb{R}^d$$

Prediction

- How to prediction (User probability)

$$p(y=+1|\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$

$$p(y=-1|\boldsymbol{x}) = \frac{\exp(-\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$

- The probability should be sum to 1.

$$p(y=+1|\boldsymbol{x}) + p(y=-1|\boldsymbol{x}) = \frac{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})} = 1$$

Parameter training (Review)

- We train model parameter $\boldsymbol{w}$ from data

- How to estimate?
  - The positive class probability is high if the data is normal, and the negative class probability is high if the data is fraud.
- Likelihood function: $L(\boldsymbol{w}) = \prod_{i=1}^{n} p(y_i | \boldsymbol{x}_i; \boldsymbol{w})$

- Log likelihood function:

$$L(\boldsymbol{w}) = \log \prod_{i=1}^{n} p(y_i | \boldsymbol{x}_i; \boldsymbol{w}) = \sum_{i=1}^{n} \log p(y_i | \boldsymbol{x}_i; \boldsymbol{w})$$

Parameter training

- Optimization

$$\max_{\boldsymbol{w}} \quad L(\boldsymbol{w}) \rightarrow \min_{\boldsymbol{w}} - \sum_{i=1}^{n} \log p(y_i | \boldsymbol{x}_i; \boldsymbol{w})$$

- We can use a gradient descent.

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \nabla_{\boldsymbol{w}} (-L(\boldsymbol{w}))$$

Fraud Detection

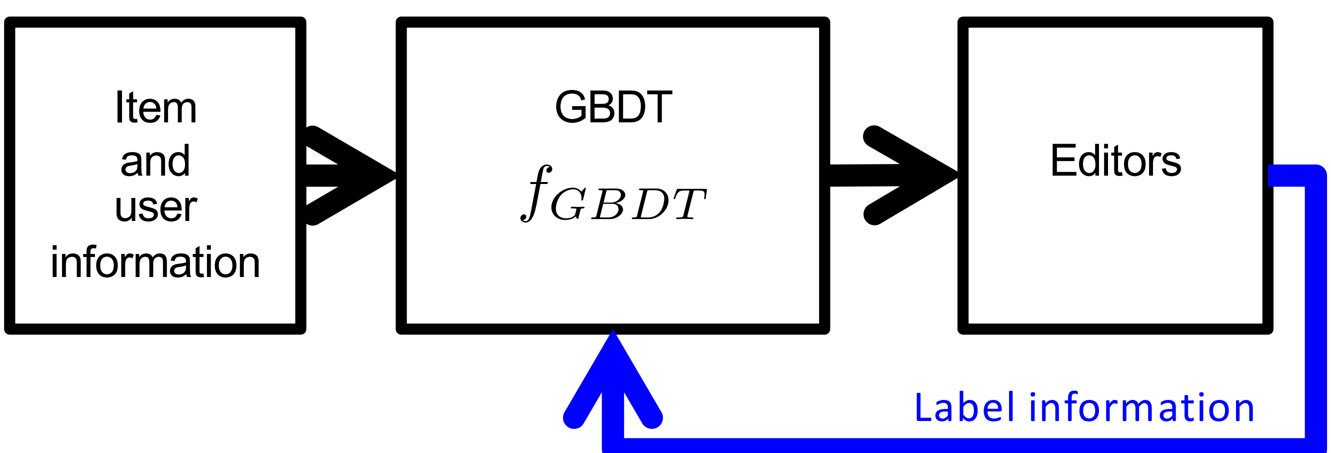
- How to find fraud users?

Practical issues

- Various types of frauds
  - Selling fake items
  - Do not send items
  - Do a big frauds after gathering trust scores.
  - Etc.
- Detecting fraud is a very important to make users happy!
- Challenge
  - The fraud types changes over season
    - Active learning, transfer learning, etc.
  - Big data (the number of samples can be hundred million of items)
- Can we automatically erase user account?

One solution: Using Active learning

- Key idea: We ask human editor to judge fraud or not → Feedback the result to machine learning model (Active learning)
- Use supervised learning (GBDT, xgboost)
  - Semi-supervised and unsupervised method tends not to work for real problems.
- Feature engineering is super important!



Results

- Detection results (rule based approach is a baseline)