Modeling Recommender Systems with Regression Models

Presenter: Liang Hu

Outline

- Continuous and Categorical Data
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression

Outline

- Continuous and Categorical Data
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression

Continuous data

Continuous data represent measurements; their possible values cannot be counted and can only be described using intervals on the real number line.

- Blood pressure $\in [0,300]$

- Normalization
  - Standard score: $\bar{x} = \frac{x-\mu}{\sigma}$, where $\mu = \frac{\sum_{i=1}^{N} x_i}{N}$, $\sigma^2 = \frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N-1}$
    - We have $\bar{x} \sim N(0,1)$
    - Normalizing errors when population parameters are known. Works well for populations that are normally distributed
  - Min-Max scaling: $\bar{x} = \frac{x-x_{min}}{x_{max}-x_{min}}$
    - We have $\bar{x} \in [0,1]$
    - This is also called unity-based normalization.

Categorical data

Categorical data represent characteristics such as

- Gender {Male, Female, Unknown}
- Age group {0~10, 10~20, 20~30, 30~40, 40~50, 50~60, above 60}
- Types of movies {Action, Adventure, Animation, Comedy}

- However, some algorithms cannot work with categorical data directly
- One-Hot Encoding

Action	1	0	0	0
Adventure	0	1	0	0
Animation	0	0	1	0
Comedy	0	0	0	1

Example

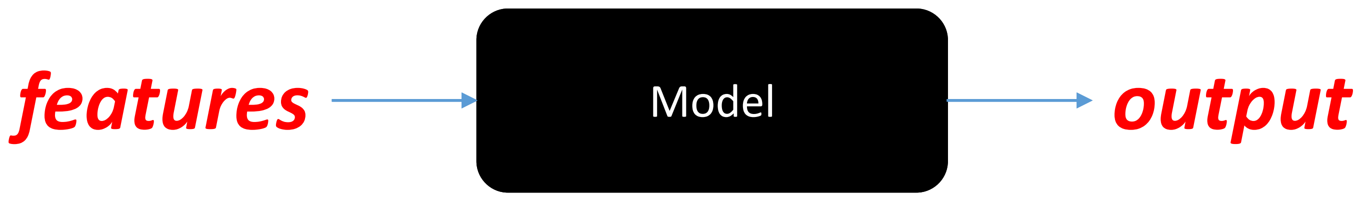
- A movie
  - Length: 120 Min
  - Adult Only: True
  - Type: Action
  - Country: U.S.

- Feature vector

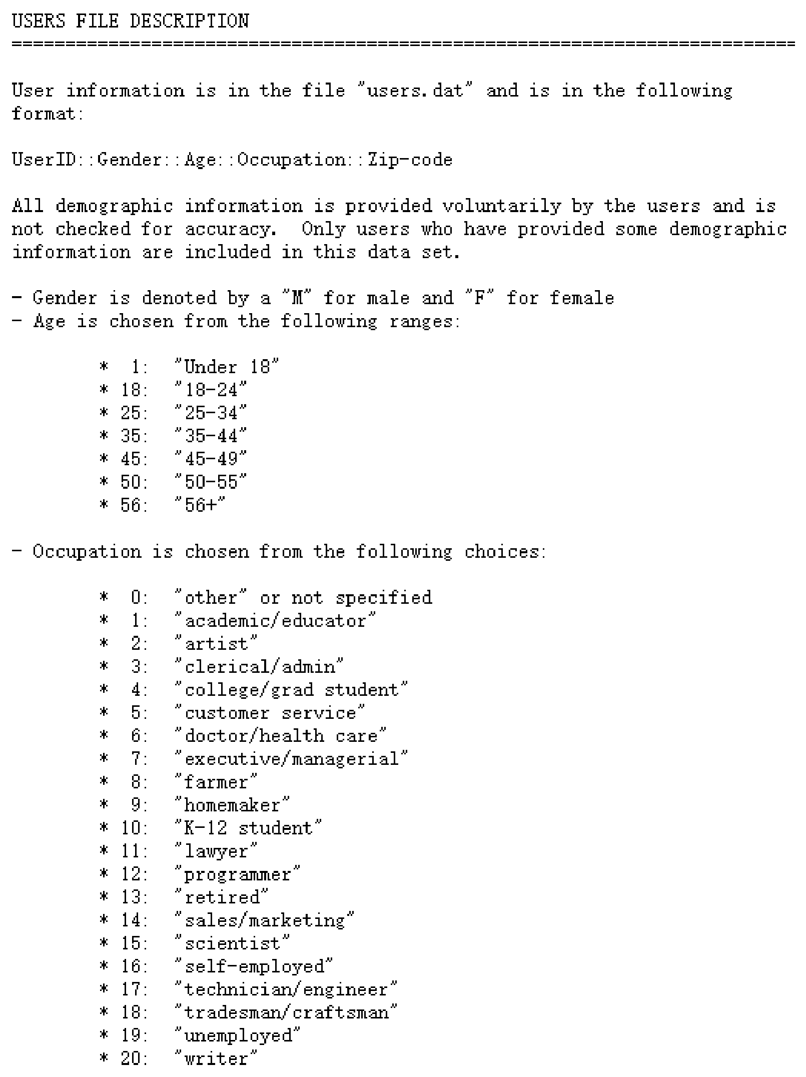
L	A	T				C				
0.2	1	1	0	0	0	0	1	0	0	0

Basic machine learning problem

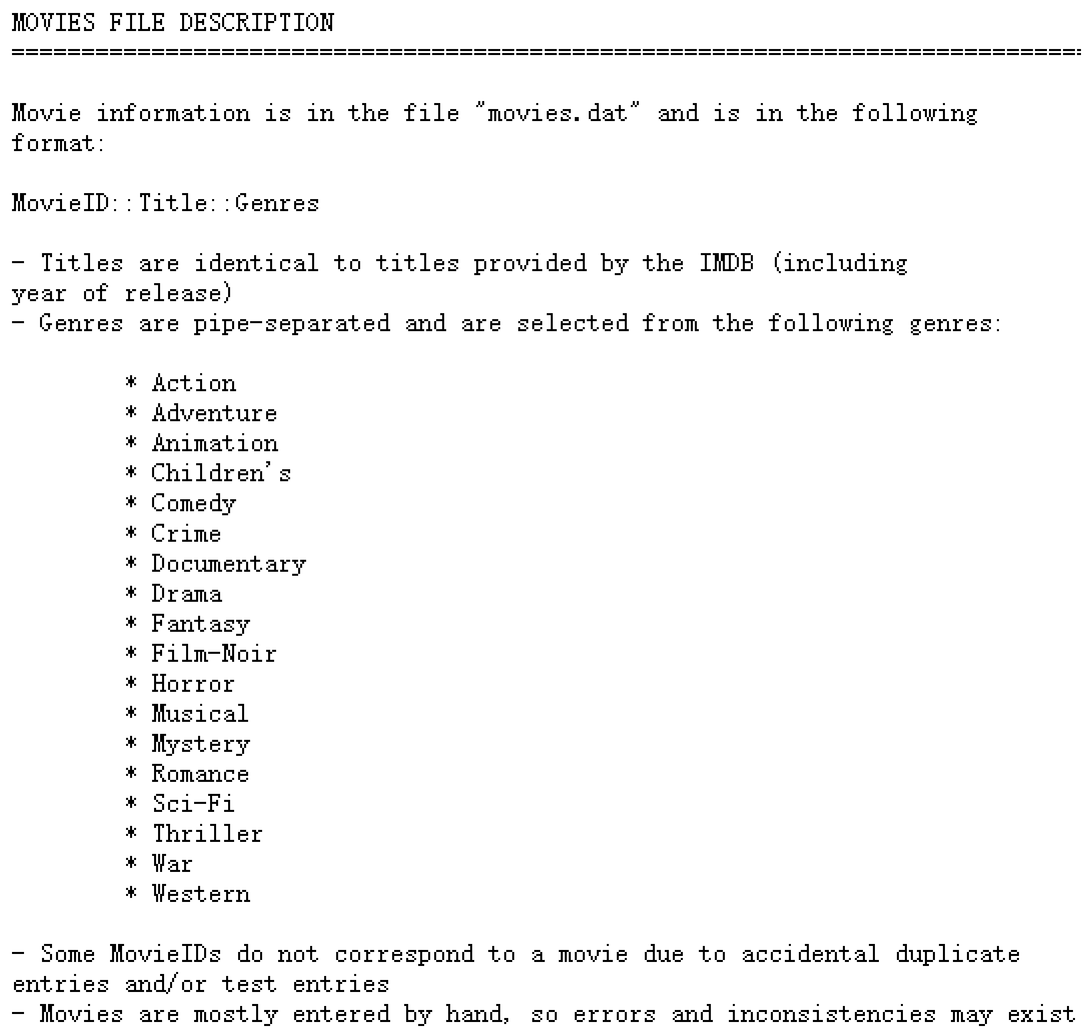
- Modeling the relationship between ***output*** and given ***input features***.



Example: movie rating prediction



```
USERS FILE DESCRIPTION
================================================================================

User information is in the file "users.dat" and is in the following
format:

UserID::Gender::Age::Occupation::Zip-code

All demographic information is provided voluntarily by the users and is
not checked for accuracy.  Only users who have provided some demographic
information are included in this data set.

- Gender is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:

        *  1:  "Under 18"
        * 18:  "18-24"
        * 25:  "25-34"
        * 35:  "35-44"
        * 45:  "45-49"
        * 50:  "50-55"
        * 56:  "56+"

- Occupation is chosen from the following choices:

        *  0:  "other" or not specified
        *  1:  "academic/educator"
        *  2:  "artist"
        *  3:  "clerical/admin"
        *  4:  "college/grad student"
        *  5:  "customer service"
        *  6:  "doctor/health care"
        *  7:  "executive/managerial"
        *  8:  "farmer"
        *  9:  "homemaker"
        * 10:  "K-12 student"
        * 11:  "lawyer"
        * 12:  "programmer"
        * 13:  "retired"
        * 14:  "sales/marketing"
        * 15:  "scientist"
        * 16:  "self-employed"
        * 17:  "technician/engineer"
        * 18:  "tradesman/craftsman"
        * 19:  "unemployed"
        * 20:  "writer"
```



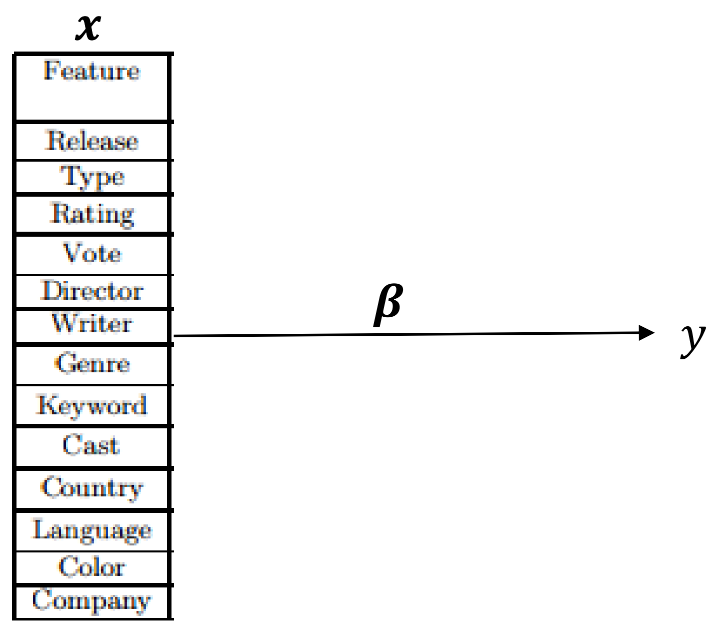
```
MOVIES FILE DESCRIPTION
================================================================================

Movie information is in the file "movies.dat" and is in the following
format:

MovieID::Title::Genres

- Titles are identical to titles provided by the IMDB (including
year of release)
- Genres are pipe-separated and are selected from the following genres:

        * Action
        * Adventure
        * Animation
        * Children's
        * Comedy
        * Crime
        * Documentary
        * Drama
        * Fantasy
        * Film-Noir
        * Horror
        * Musical
        * Mystery
        * Romance
        * Sci-Fi
        * Thriller
        * War
        * Western

- Some MovieIDs do not correspond to a movie due to accidental duplicate
entries and/or test entries
- Movies are mostly entered by hand, so errors and inconsistencies may exist
```

https://grouplens.org/datasets/movielens/

Rating prediction

- $\boldsymbol{\beta}$ is a set of parameters used to model the importance of features $\boldsymbol{x}$.
- y is the rating given by a user

$$y = f_{\boldsymbol{\beta}}(\boldsymbol{x})$$



Statistical estimation and inference focuses on $\boldsymbol{\beta}$

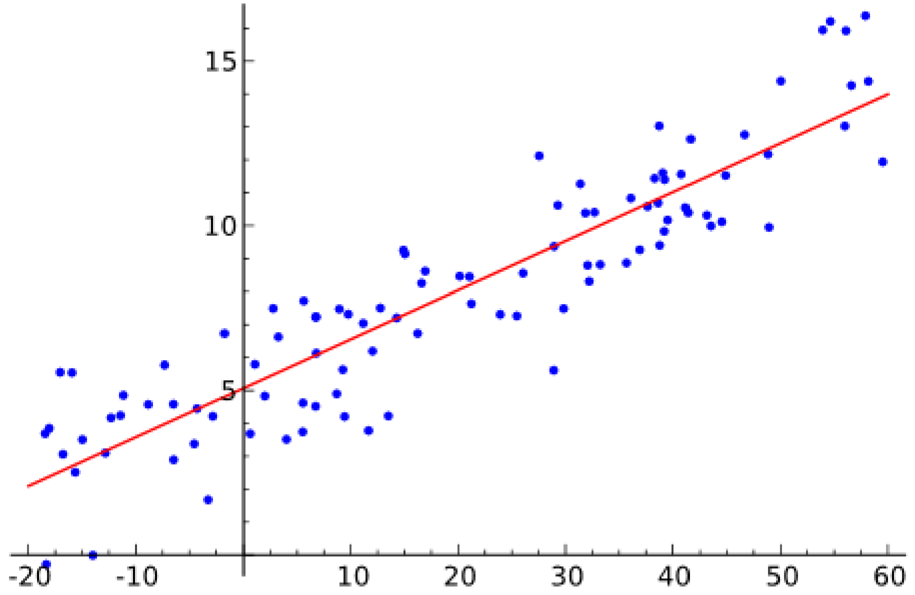
Outline

- Continuous and Categorical Data
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression

Ordinary linear regression

$$y = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\beta} + e$$

- y: is often called the *response variable, dependent variable or regressand*
- $\boldsymbol{x}$: is called *covariates, input variables, regressors, explanatory variables, predictor variables, or independent variables*
- $\boldsymbol{\beta}$: is called *effects*, or *regression coefficients*
- e: is called the *error term*, *disturbance term*, or *noise*



Linear regression model

- $y_{u,i}$**:** A rating on item i given by user u
- $\boldsymbol{x}_u$**:** A feature vector of user u
- $\boldsymbol{x}_i$**:** A feature vector of item i
- $\boldsymbol{\beta}$**:** Weights on features to estimate
- $e_{u,i}$**:** Error term
- $f(\cdot)$: Identity function

$$y_{u,i} = f_{\boldsymbol{\beta}}(\boldsymbol{x}_u, \boldsymbol{x}_i) = \boldsymbol{x}_u^{\mathrm{T}}\boldsymbol{\beta}_u + \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_i + e_{u,i} = \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta} + e_{u,i}$$

where $\boldsymbol{x}_{u,i} = [\boldsymbol{x}_u, \boldsymbol{x}_i], \boldsymbol{\beta} = [\boldsymbol{\beta}_u, \boldsymbol{\beta}_i]$

Rating prediction

- Given a user v, we can predict the rating on an item j

$$\hat{y}_{v,j} \approx \boldsymbol{x}_{v,j}^{\mathrm{T}} \boldsymbol{\beta}$$

- The key is to learn the model parameter $\boldsymbol{\beta}$

Objective

- $\boldsymbol{O} = \{\langle u, i \rangle\}$: a set of user-item interactions
- $y_{u,i}$ is the response, e.g. rating
- $\boldsymbol{x}_{u,i}$**:** A feature vector of user u and item i**.**
- $\boldsymbol{\beta}$**:** Weights on features to estimate
- $e_{u,i}$**:** Error
- The mean squared error (MSE) is

$$Loss = \frac{1}{|\boldsymbol{O}|} \sum_{\langle u,i \rangle \in \boldsymbol{O}} e_{u,i}^2 = \frac{1}{|\boldsymbol{O}|} \sum_{\langle u,i \rangle \in \boldsymbol{O}} \left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta} \right)^2$$

Matrix form

- Given N users and M items, the matrix form of errors can be written as follows:

$$\boxed{\overset{\boldsymbol{e}}{\begin{bmatrix} e_{1,1} \\ \vdots \\ e_{N,M} \end{bmatrix}}} = \begin{bmatrix} y_{1,1} - \mathbf{x}_{1,1}^{\mathrm{T}}\boldsymbol{\beta} \\ \vdots \\ y_{N,M} - \mathbf{x}_{N,M}^{\mathrm{T}}\boldsymbol{\beta} \end{bmatrix} = \boxed{\overset{\mathbf{y}}{\begin{bmatrix} y_{1,1} \\ \vdots \\ y_{N,M} \end{bmatrix}}} - \boxed{\overset{\boldsymbol{X^T}}{\begin{bmatrix} \mathbf{x}_{1,1}^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{N,M}^{\mathrm{T}} \end{bmatrix}}}\boldsymbol{\beta}$$

$$\Downarrow$$

$$\mathbf{e} = \mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}$$

$$\begin{aligned} Loss &= \frac{1}{|\boldsymbol{O}|}\sum_{i \in \boldsymbol{O}} e_{u,i}^2 \\ &= \frac{1}{|\boldsymbol{O}|}\mathbf{e}^{\mathrm{T}}\mathbf{e} \\ &= \frac{1}{|\boldsymbol{O}|}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right) \end{aligned}$$

https://mathworld.wolfram.com/MatrixTrace.html

Parameter estimation

- Compute the derivative

$$L = \frac{1}{|\mathbf{O}|}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right)$$

Using the chain rule (you can refer to *matrix cookbook*)

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \frac{2}{|\mathbf{O}|}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right)(-\mathbf{X})^{\mathrm{T}} = \frac{2}{|\mathbf{O}|}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} - \mathbf{X}^{\mathrm{T}}\mathbf{y}\right)$$

- Close form solution

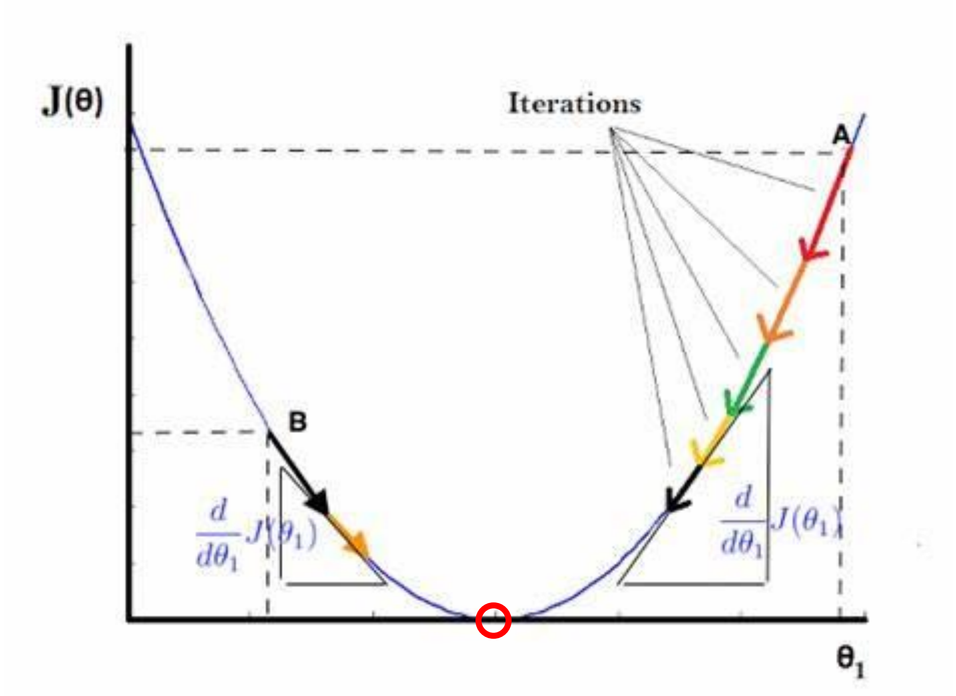
Minimize the loss, set the partial derivative w.r.t. $\boldsymbol{\beta}$ to $\mathbf{0}$

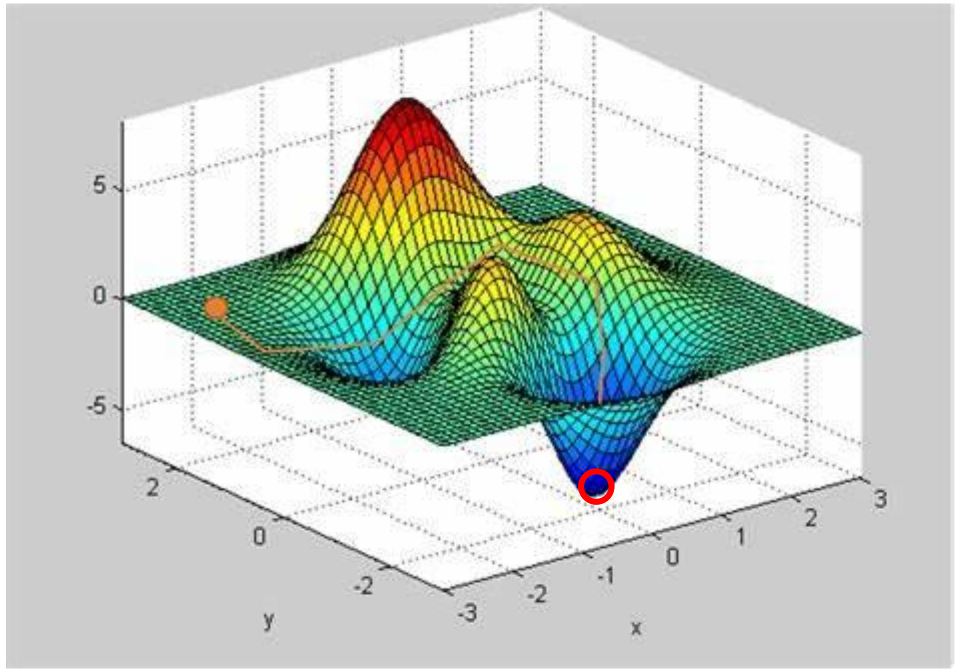
$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} - \mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{0} \;\Rightarrow\; \boldsymbol{\beta} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

Practice: using chain rule to derive $\frac{\partial L}{\partial \boldsymbol{\beta}}$

Gradient descent

- Gradient descent is an optimization algorithm used to find the values of parameters that minimizes a cost function.


J(θ)
Iterations
A
B
$\frac{d}{d\theta_1}J(\theta_1)$
$\frac{d}{d\theta_1}J(\theta_1)$
θ1



$$\boldsymbol{\beta}^{t+1} \leftarrow \boldsymbol{\beta}^{t} - \boldsymbol{\alpha}\frac{\partial L}{\partial \boldsymbol{\beta}}$$

Learning rate

- Learning rate controls how quickly or slowly a model learns a problem

$$\boldsymbol{\beta}^{t+1} \leftarrow \boldsymbol{\beta}^{t} - \boldsymbol{\alpha}\frac{\partial L}{\partial \boldsymbol{\beta}}$$

- Adaptive optimization algorithms

$$\boldsymbol{\beta}^{t+1} \leftarrow \boldsymbol{\beta}^{t} - f_{\alpha}\left(\frac{\partial L}{\partial \boldsymbol{\beta}}\right)$$

e.g. Adagrad, Adadelta, RMSprop, Adam

https://ruder.io/optimizing-gradient-descent/

Three types of gradient descent

- ***Batch gradient descent:*** calculates the error for all examples in the training dataset $\boldsymbol{O}$, and then updates the model parameters

$$Loss = \frac{1}{|\boldsymbol{O}|}\sum_{\langle u,i\rangle\in\boldsymbol{O}}\left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)^2$$

- ***Stochastic gradient descent (SGD):*** calculates the error for each example in the training dataset $\langle u, i\rangle \in \boldsymbol{O}$, and updates the model parameters

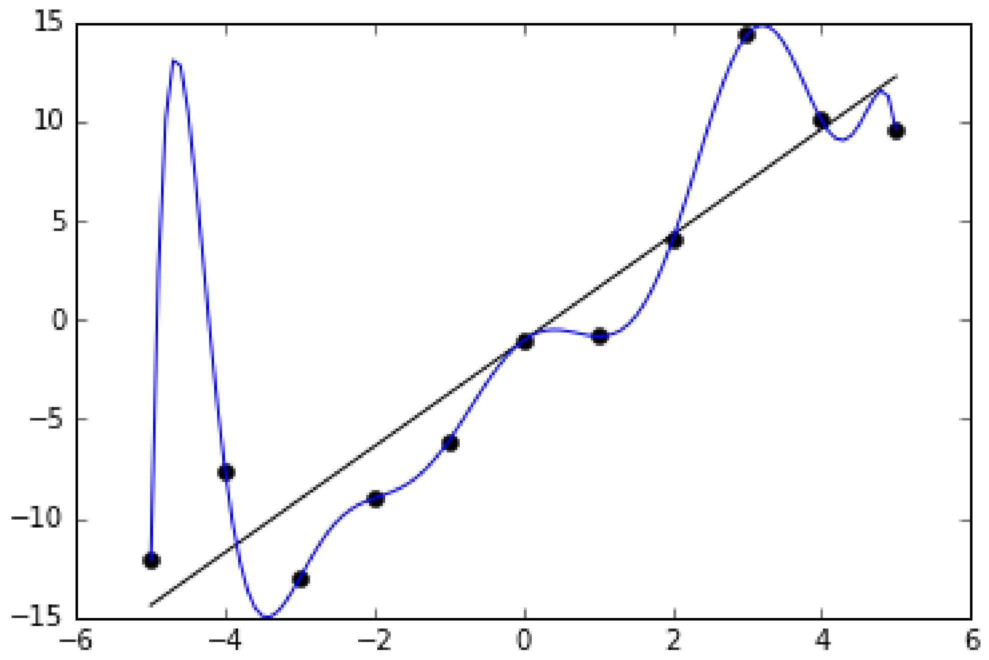
$$Loss = \left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)^2$$

- ***Mini-batch gradient descent:*** calculates the error for a mini-batch $\boldsymbol{B} \subset \boldsymbol{O}$ ($|\boldsymbol{B}|$ is small, e.g. 64 examples), and updates the model parameters

$$Loss = \frac{1}{|\boldsymbol{B}|}\sum_{\langle u,i\rangle\in\boldsymbol{B}}\left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)^2$$

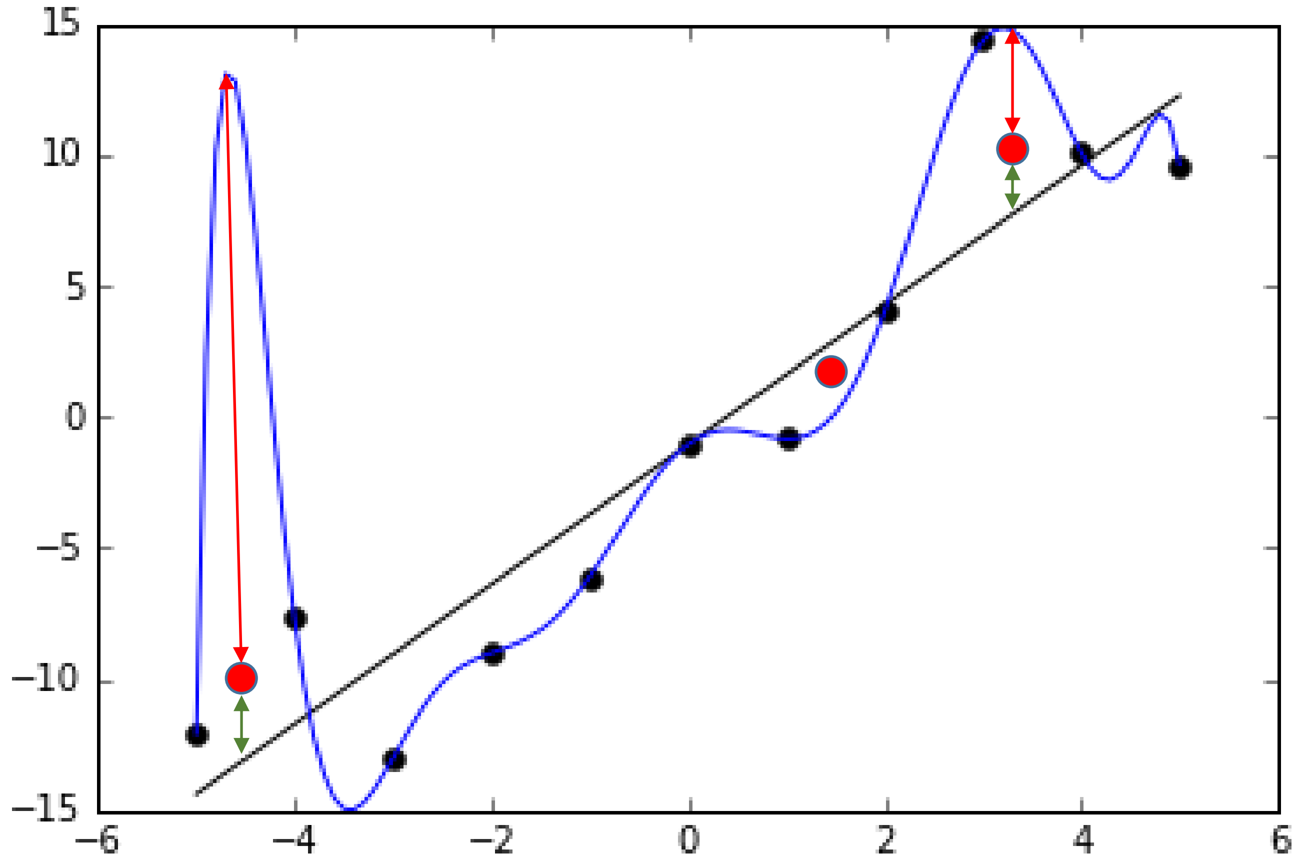
Overfitting

- In regression analysis, overfitting occurs frequently.
- Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.
- As an extreme example, if there are p variables in a linear regression with p data points, the fitted line can go exactly through every point.



Overfitting

- A model which has been overfit will generally have poor predictive performance, as it can exaggerate minor fluctuations in the data



Regularization

- Regularization discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.
  - L1/L2-norm
  - Early stopping
  - Dropout

Regularized linear regression

- Least-squares estimation with L2-norm regularization (ridge regression):

$$L = \frac{1}{|\mathbf{O}|}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}\right) + \lambda\|\boldsymbol{\beta}\|^2$$

$$\|\boldsymbol{\beta}\| = \sqrt{\sum_i \beta_i^2}, \qquad \|\boldsymbol{\beta}\|^2 = \sum_i \beta_i^2 = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}$$

- Least-squares estimation with L2-norm regularization:

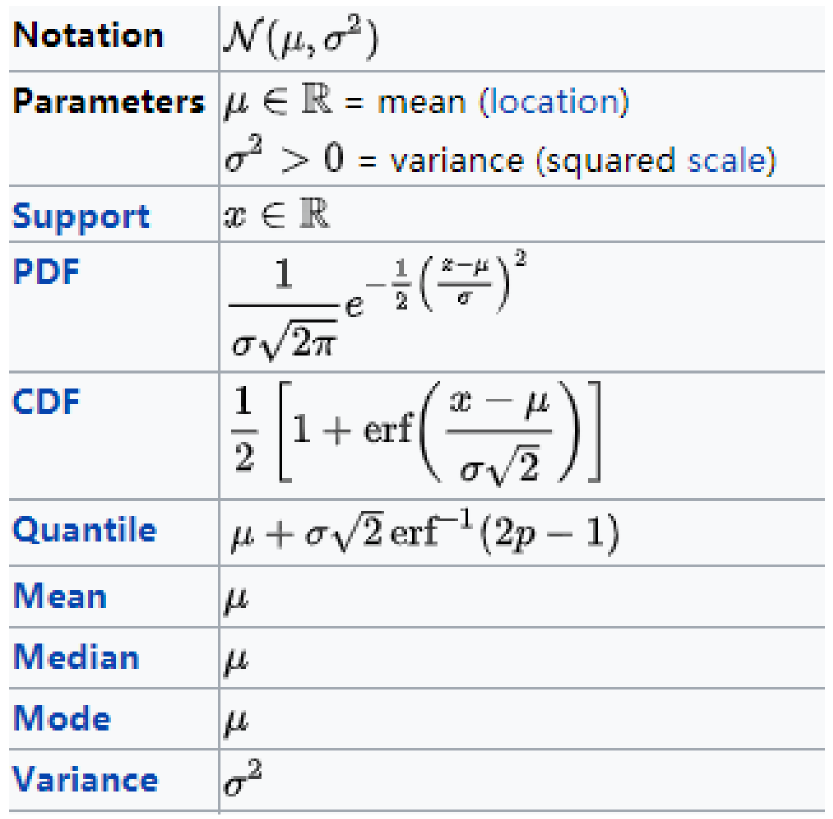
$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \frac{2}{|\mathbf{O}|}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta} - \mathbf{X}^{\mathrm{T}}\mathbf{y}\right)^{\mathrm{T}}$$

- Close form solution:

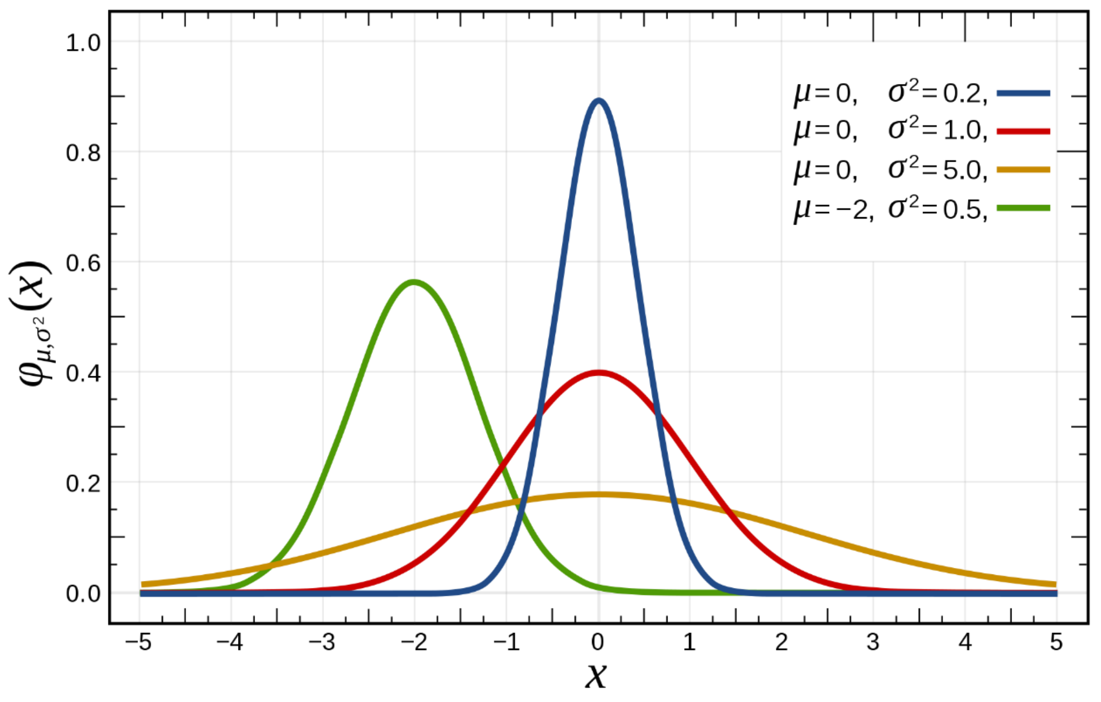
$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

Linear regression from probabilistic view

- Normal (Gaussian) Distribution (e.g. Height of people)



Notation	$\mathcal{N}(\mu, \sigma^2)$
Parameters	$\mu \in \mathbb{R}$ = mean (location) $\sigma^2 > 0$ = variance (squared scale)
Support	$x \in \mathbb{R}$
PDF	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
CDF	$\frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$
Quantile	$\mu + \sigma\sqrt{2}\operatorname{erf}^{-1}(2p-1)$
Mean	μ
Median	μ
Mode	μ
Variance	σ^2



Likelihood

- $\mathrm{P}(E|H)$ is the probability of observing E given H, and is called likelihood.

- Given a datapoint x, the log-likelihood of x with $N(\mu, \sigma^2)$ is

$$\log N(x|\mu,\sigma^2) = \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$= -\left(\frac{1}{2\sigma^2}(x-\mu)^2 + \log\sigma\sqrt{2\pi}\right)$$

Likelihood over user-item interactions

- Likelihood for a rating, given the attributes $\boldsymbol{x}_{u,i}$

$y_{u,i} = \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta} + e_{u,i}$ where $e_{u,i} \sim N(0, \sigma^2)$

$$e_{u,i} \sim N(0, \sigma^2) \quad i.e. \left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) \sim N(0, \sigma^2)$$

$$\Rightarrow y_{u,i} \sim \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta} + N(0, \sigma^2) \quad i.e. \, y_{u,i} \sim N\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}, \sigma^2\right)$$

- Likelihood for a set of user-item interactions $\boldsymbol{O}$

$$L = \prod_{u,i \in \boldsymbol{O}} N\left(y_{u,i} | \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}, \sigma^2\right) = \prod_{u,i \in \boldsymbol{O}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)^2}{2\sigma^2}}$$

Maximum Likelihood Estimation (MLE)

- Likelihood function:

$$f_{\boldsymbol{\beta}}(\boldsymbol{O}) = \log \prod_{u,i \in \boldsymbol{O}} N\left(y_{u,i} | \boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}, \sigma^2\right) = -\sum_{u,i \in \boldsymbol{O}} \left[\frac{1}{2\sigma^2} \left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)^2 + \log \sigma \sqrt{2\pi}\right]$$

- MLE: maximizing $f_{\boldsymbol{\beta}}(\boldsymbol{O})$ w.r.t. model parameters $\boldsymbol{\beta}$

$$\arg\max_{\boldsymbol{\beta}} f_{\boldsymbol{\beta}}(\boldsymbol{O}) = -\frac{1}{2\sigma^2} \sum \left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)^2 + Const$$

- So, estimating $\boldsymbol{\beta}$ with MLE is just to solve a least square problem.

The prior of model parameters

- Place a prior on $\boldsymbol{\beta} \sim \mathrm{P}(\theta)$
    - Here we also use normal distribution

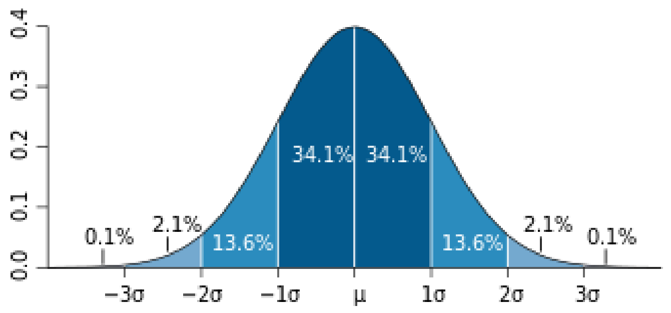
$$\beta_i \sim N(0, \sigma^2)$$

$$N(\beta_i | 0, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\beta_i{}^2}{2\sigma^2}}$$

- Log-prior:

$$L(\boldsymbol{\beta}) = \log \prod_i N(\beta_i | 0, \sigma^2) = -\frac{1}{2\sigma^2} \sum_i \beta_i{}^2 + Const = -\lambda \|\boldsymbol{\beta}\|^2 + Const$$

where $\lambda = \frac{1}{2\sigma^2}$



Bayes' theorem

- Joint probability:

$$P(y,\theta) = \underbrace{P(y|\theta)}_{likelihood}\ \underbrace{P(\theta)}_{prior} = P(\theta|y)P(y)$$

- The posterior distribution will yield:

$$P(\theta|y) = \frac{P(y,\theta)}{P(y)} = \frac{P(y|\theta)P(\theta)}{P(y)} \propto P(y|\theta)P(\theta)$$

Maximum A Posterior (MAP) Estimation

- The log posterior of parameters given the data is

$$P(\boldsymbol{\beta}|\boldsymbol{O}) \propto P(\boldsymbol{O}|\boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta})$$

$$\max \log P(\boldsymbol{\beta}|\boldsymbol{O}) = \max[\log P(\boldsymbol{O}|\boldsymbol{\beta}) + \log \boldsymbol{P}(\boldsymbol{\beta})]$$

$$= \max\left[\log \prod_{u,i\in\boldsymbol{O}} N\left(y_{u,i}|\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}, \sigma_R^2\right) + \log \prod_k N(\beta_k|0, \sigma_P^2)\right]$$

$$= \max\left[-\left(\frac{1}{2\sigma_R^2}\sum_{u,i\in\boldsymbol{O}}\left(y_{u,i} - \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)^2 + \lambda||\boldsymbol{\beta}||^2\right)\right]$$

- So, this is equivalent to solving a L2-norm regularized least square problem.

- *Thinking: How to find the optimal value of λ?*

Outline

- Continuous and Categorical Data
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression

Click-through rate (CTR)

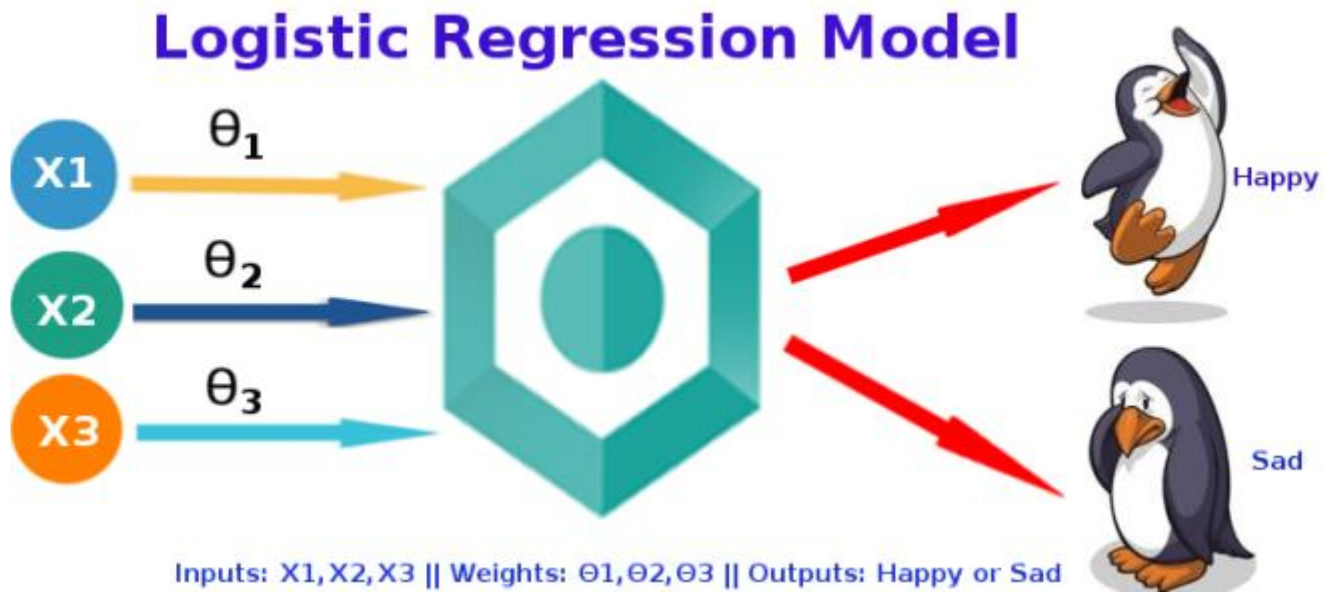
- Users click on a specific link to view a page, email, or advertisement.

- The CTR of an advertisement is the number of times a click is made on the ad, divided by the number of times the ad is “served”.

- $CTR = \frac{\#\, click-throughs}{\#\, served}$

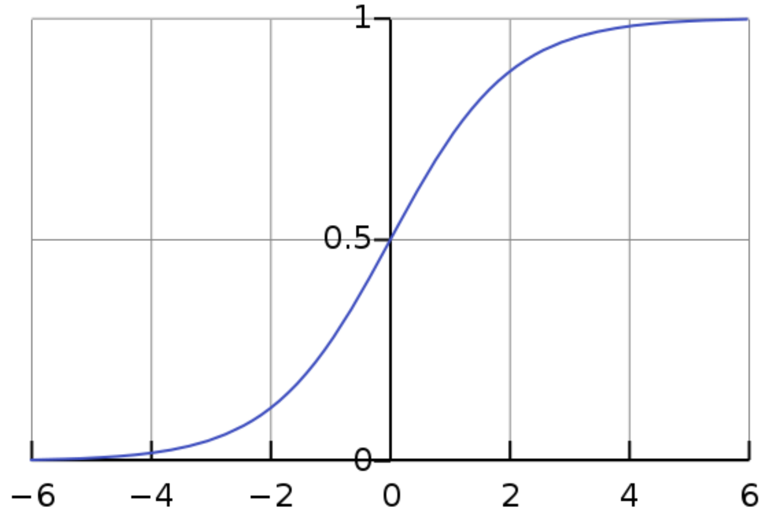
Logistic regression

- The logistic model (or logit model) is used to model the binary probability of a certain class such as pass/fail, win/lose, alive/dead.


Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1, X2, X3 || Weights: Θ1, Θ2, Θ3 || Outputs: Happy or Sad

The probability of a click-through

- $y_{u,i}$: User u clicked item i
- $\boldsymbol{x}_u$: A feature vector of user u
- $\boldsymbol{x}_i$: A feature vector of item i
- $\boldsymbol{\beta}$: Weights on features to estimate
- $\sigma(x) = \frac{1}{1+e^{-x}}$: Sigmoid function



$$\mathrm{P}\left(y_{u,i} = 1 | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right) = \frac{1}{1 + e^{-\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}}}$$

$$\mathrm{P}\left(y_{u,i} = 0 | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = 1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right) = \frac{e^{-\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}}}{1 + e^{-\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}}}$$

where $\boldsymbol{x}_{u,i} = [\boldsymbol{x}_u, \boldsymbol{x}_i]$

A unified representation

$$\begin{cases} \mathrm{P}\left(y_{u,i} = 1 | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right) = \dfrac{1}{1 + e^{-\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}}} \\ \mathrm{P}\left(y_{u,i} = 0 | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = 1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right) = \dfrac{e^{-\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}}}{1 + e^{-\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}}} \end{cases}$$

- A trick for unifying $\mathrm{P}\left(y_{u,i} = 1\right)$ and $\mathrm{P}\left(y_{u,i} = 0\right)$

$$\mathrm{P}\left(y_{u,i} | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)^{y_{u,i}} \left(1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)\right)^{1 - y_{u,i}}$$

Log-likelihood

$$L\left(y_{u,i} | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = \log \mathrm{P}\left(y_{u,i} | \boldsymbol{x}_{u,i}, \boldsymbol{\beta}\right) = \log \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)^{y_{u,i}} \left(1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)\right)^{1-y_{u,i}}$$

$$= y_{u,i} \log \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right) + \left(1 - y_{u,i}\right) \log \left(1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}} \boldsymbol{\beta}\right)\right)$$

which is often called binary cross-entropy: $H(P, Q) = -\sum_x P(x) \log Q(x)$

Gradient

$$L\left(y_{u,i}|\boldsymbol{x}_{u,i},\boldsymbol{\beta}\right) = \log \mathrm{P}\left(y_{u,i}|\boldsymbol{x}_{u,i},\boldsymbol{\beta}\right) = y_{u,i}\log\sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) + (1-y_{u,i})\log\left(1-\sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)\right)$$

Using the chain rule

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \left(\sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) - y_{u,i}\right)\boldsymbol{x}_{u,i}$$

Practice: using chain rule to derive $\frac{\partial L}{\partial \boldsymbol{\beta}}$

Mini-batch updating

- Likelihood for a mini-batch of user-item interactions $\mathbf{B} = \{\langle u, i\rangle\}$

$$L = \prod_{\langle u,i\rangle \in \boldsymbol{B}} \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)^{y_{u,i}} \left(1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)\right)^{1-y_{u,i}}$$

- Negative log-likelihood

$$L(\boldsymbol{B}|\boldsymbol{\beta}) = -\sum_{u,i\in \boldsymbol{B}} y_{u,i} \log \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) + \left(1 - y_{u,i}\right) \log\left(1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)\right)$$

- Parameter update

$$\boldsymbol{\beta}^{t+1} \leftarrow \boldsymbol{\beta}^{t} - f_{\boldsymbol{\alpha}}\left(\frac{\partial L(\boldsymbol{B}|\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} / |\mathbf{B}|\right)$$

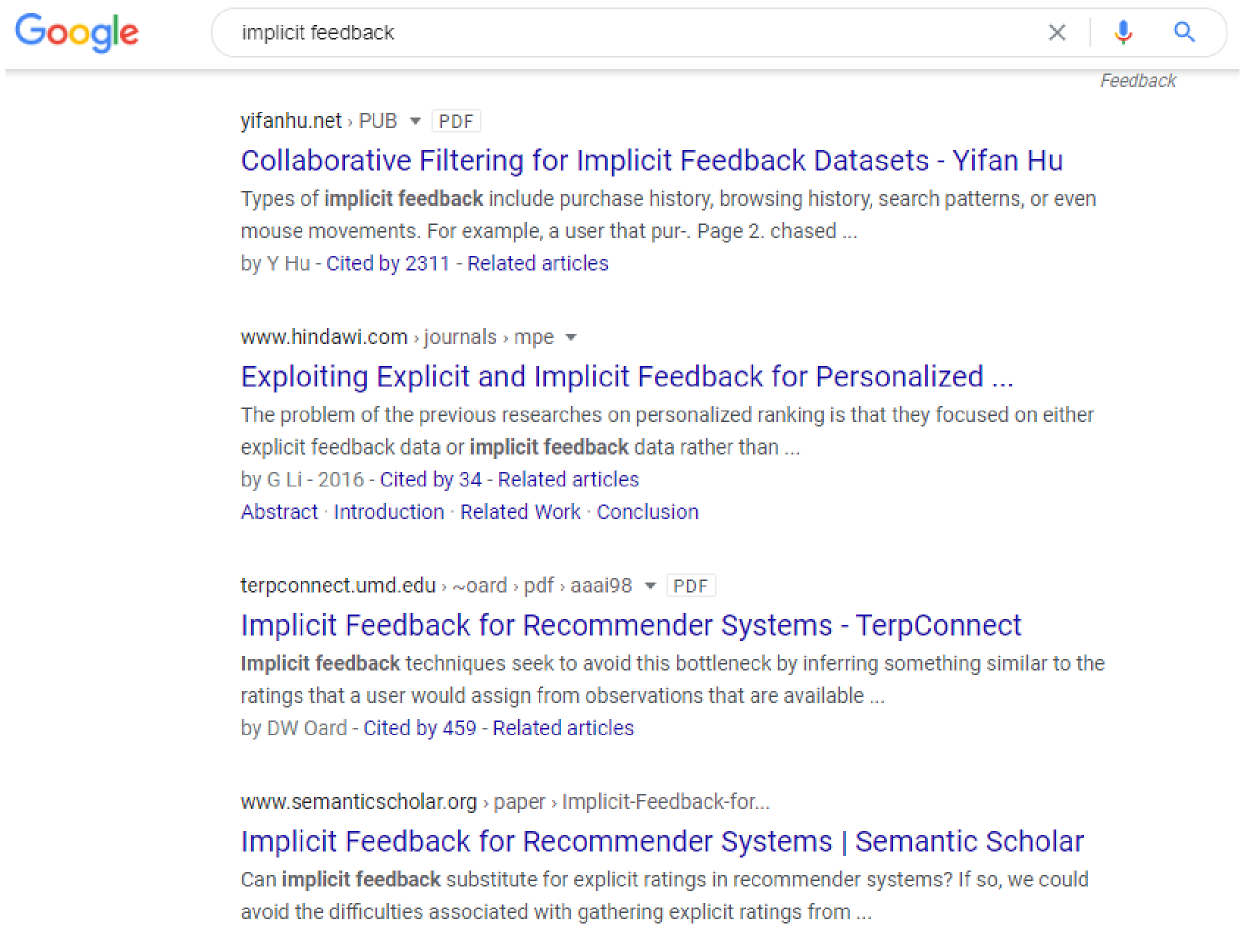
Prediction

- Once we obtain the learned model parameters $\boldsymbol{\beta}$, the probability of user v clicking on an item j is estimated as:

$$\mathrm{P}\left(y_{v,j} = 1 | \boldsymbol{x}_{v,j}, \boldsymbol{\beta}\right) = \sigma\left(\boldsymbol{x}_{v,j}^{\mathrm{T}} \boldsymbol{\beta}\right) = \frac{1}{1 + e^{-\boldsymbol{x}_{v,j}^{\mathrm{T}} \boldsymbol{\beta}}}$$

Explicit feedback is hard to obtain

- Quite often, we cannot always obtain binary labels, such as pass/fail, like/unlike
- For example, an App can only record which items user clicked


implicit feedback

Feedback

yifanhu.net › PUB PDF
Collaborative Filtering for Implicit Feedback Datasets - Yifan Hu
Types of **implicit feedback** include purchase history, browsing history, search patterns, or even mouse movements. For example, a user that pur-. Page 2. chased ...
by Y Hu - Cited by 2311 - Related articles

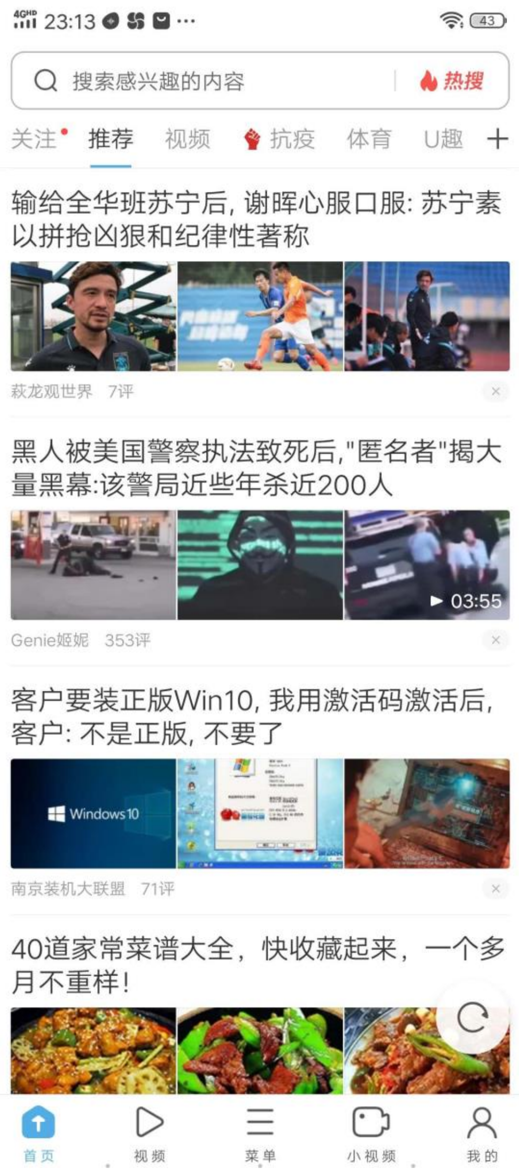
www.hindawi.com › journals › mpe
Exploiting Explicit and Implicit Feedback for Personalized ...
The problem of the previous researches on personalized ranking is that they focused on either explicit feedback data or **implicit feedback** data rather than ...
by G Li - 2016 - Cited by 34 - Related articles
Abstract · Introduction · Related Work · Conclusion

terpconnect.umd.edu › ~oard › pdf › aaai98 PDF
Implicit Feedback for Recommender Systems - TerpConnect
Implicit feedback techniques seek to avoid this bottleneck by inferring something similar to the ratings that a user would assign from observations that are available ...
by DW Oard - Cited by 459 - Related articles

www.semanticscholar.org › paper › Implicit-Feedback-for...
Implicit Feedback for Recommender Systems | Semantic Scholar
Can **implicit feedback** substitute for explicit ratings in recommender systems? If so, we could avoid the difficulties associated with gathering explicit ratings from ...

Implicit feedback

- Implicit feedback is inferred from user behavior, such as
  - docs they DO/DON’T click-through for viewing
  - the duration of time spent viewing a doc



输给全华班苏宁后, 谢晖心服口服: 苏宁素以拼抢凶狠和纪律性著称

萩龙观世界 7评

黑人被美国警察执法致死后,"匿名者"揭大量黑幕:该警局近些年杀近200人

03:55

Genie姬妮 353评

客户要装正版Win10, 我用激活码激活后, 客户: 不是正版, 不要了

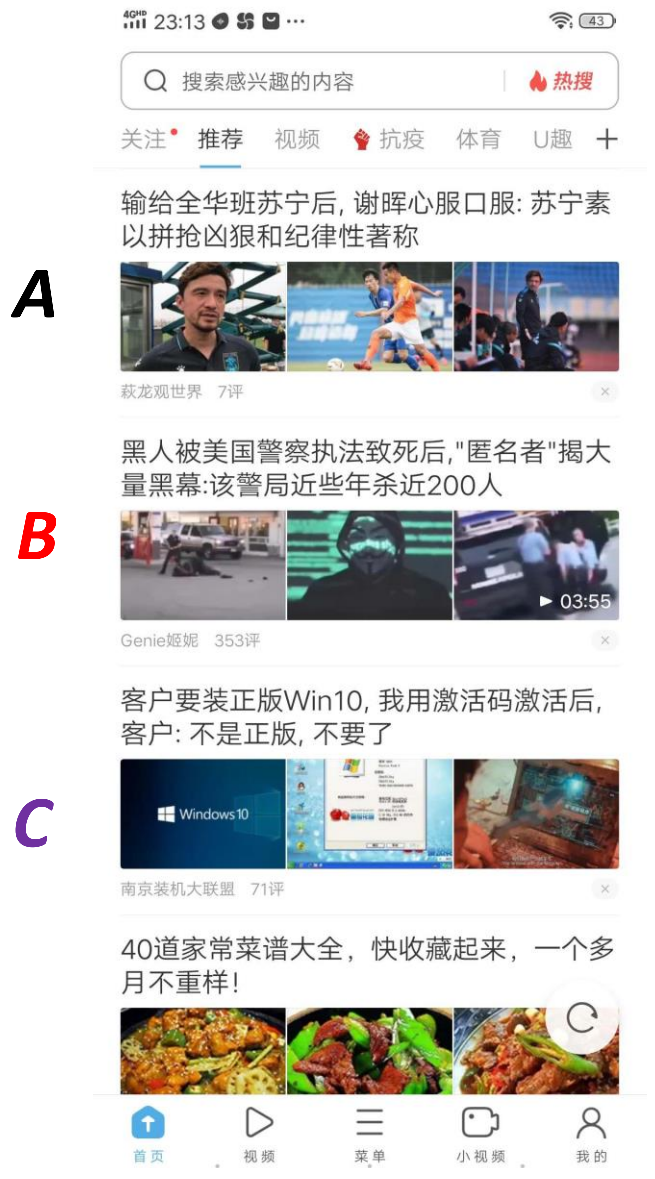
Windows 10

南京装机大联盟 71评

40道家常菜谱大全，快收藏起来，一个多月不重样!

Preference order

- We can infer the preference order according to implicit feedback context, such as
  - $A \succ B$: Doc A is selected for viewing while Doc B is not selected
  - $C \succ B$: the duration of time spent viewing Doc C is longer than that spent viewing Doc B



输给全华班苏宁后，谢晖心服口服：苏宁素以拼抢凶狠和纪律性著称

荻龙观世界 7评

黑人被美国警察执法致死后，"匿名者"揭大量黑幕:该警局近些年杀近200人

Genie姬妮 353评

客户要装正版Win10，我用激活码激活后，客户：不是正版，不要了

南京装机大联盟 71评

40道家常菜谱大全，快收藏起来，一个多月不重样！

Quantify preference order

- Let $U\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right)$ denote some utility function to measure the utility of user-item interaction $\langle u, i\rangle$,
- $\langle u, i\rangle \succ \langle u, j\rangle$ denote that user u prefers i to j, then we have

$$\langle u, i\rangle \succ \langle u, j\rangle \Leftrightarrow U\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) > U\left(\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}\right) \Leftrightarrow U\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) - U\left(\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}\right) > 0$$

Likelihood of preference order

- We label $\langle u,i\rangle \succ \langle u,j\rangle$ as 1, and $\langle u,i\rangle \preccurlyeq \langle u,j\rangle$ as 0, i.e.

$$y_{u,i,j} = \begin{cases} 1 & \langle u,i\rangle \succ \langle u,j\rangle \\ 0 & \langle u,i\rangle \preccurlyeq \langle u,j\rangle \end{cases}$$

- Infinite margin optimization:
    - Let $m = U\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) - U\left(\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}\right)$

$$\mathrm{P}\left(y_{u,i,j} = 1 | \boldsymbol{x}_{u,i}, \boldsymbol{x}_{u,j}, \boldsymbol{\beta}\right) = \sigma(m) = \frac{1}{1+e^{-m}}$$

$$\mathrm{P}\left(y_{u,i,j} = 0 | \boldsymbol{x}_{u,i}, \boldsymbol{x}_{u,j}\boldsymbol{\beta}\right) = 1 - \sigma(m) = \frac{e^{-m}}{1+e^{-m}}$$

- Log-likelihood:

$$\mathrm{L}_{\boldsymbol{\beta}}\left(y_{u,i,j} | \boldsymbol{x}_{u,i}, \boldsymbol{x}_{u,j}\right) = y_{u,i,j} \log \sigma(m) + \left(1 - y_{u,i,j}\right) \log\left(1 - \sigma(m)\right)$$

Thinking: What's the form of gradient for $\mathrm{L}_{\boldsymbol{\beta}}$*?*

BPR: Bayesian Personalized Ranking

- Log-likelihood:

$$\mathrm{L}_{\boldsymbol{\beta}}\left(y_{u,i,j} | \boldsymbol{x}_{u,i}, \boldsymbol{x}_{u,j}\right) = y_{u,i,j} \log \sigma(m) + \left(1 - y_{u,i,j}\right) \log\left(1 - \sigma(m)\right) + \lambda\boldsymbol{\beta}$$

Where $m = U\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) - U\left(\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}\right)$

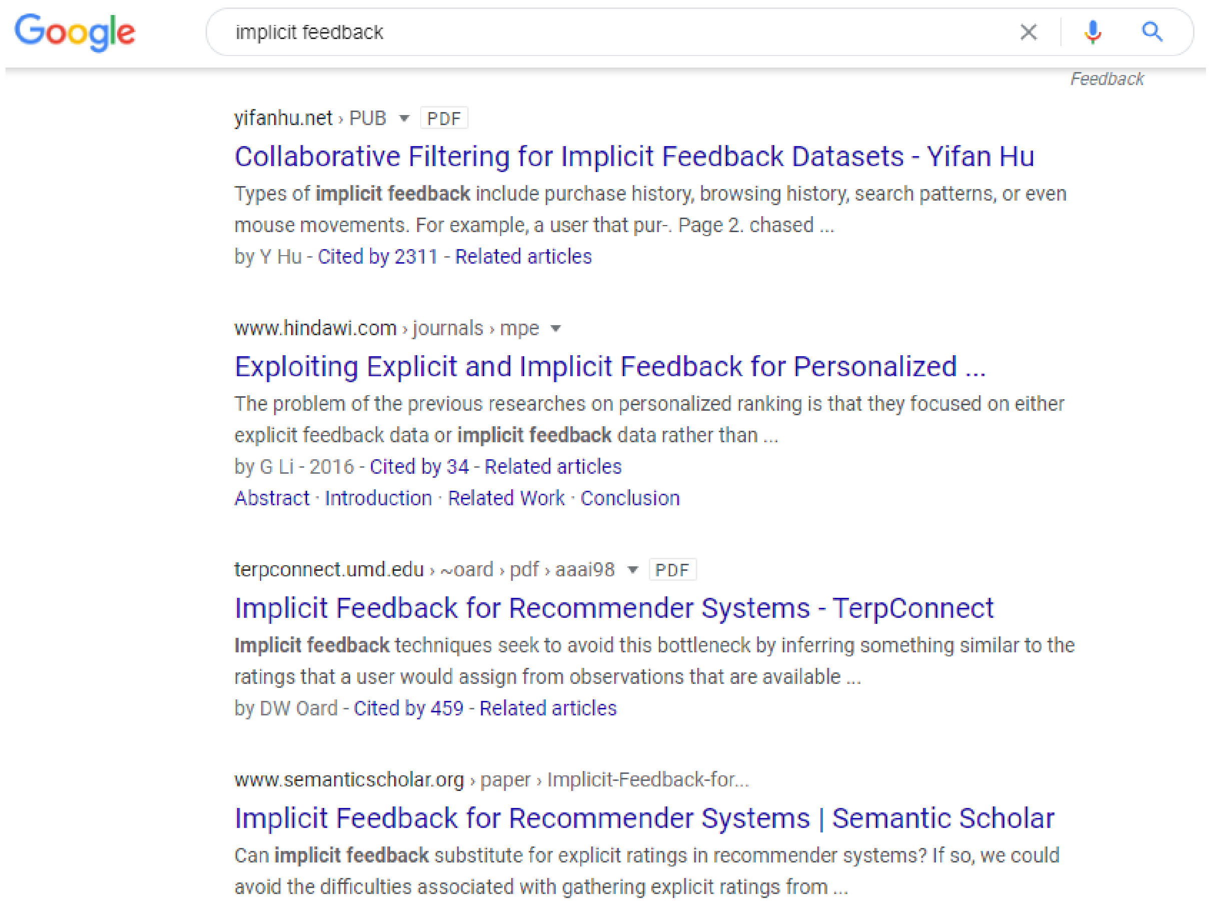
- If we use a linear utility function: $U\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}\right) = \boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}$

$$\mathrm{L}_{\boldsymbol{\beta}}\left(y_{u,i,j} | \boldsymbol{x}_{u,i}, \boldsymbol{x}_{u,j}\right) = y_{u,i,j} \log \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta} - \boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}\right) + \left(1 - y_{u,i,j}\right) \log\left(1 - \sigma\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta} - \boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}\right)\right) + \lambda\boldsymbol{\beta}$$

which is often called Bayesian Personalized Ranking.

Recommendation

- Given a set of candidate docs $\{d_1, d_2, \cdots, d_N\}$, we can easily obtain the utility on viewing each d_i for user u
- $\{U(\boldsymbol{x}_{u,d_1}^{\mathrm{T}}\boldsymbol{\beta}), U(\boldsymbol{x}_{u,d_2}^{\mathrm{T}}\boldsymbol{\beta}), \cdots, U(\boldsymbol{x}_{u,d_N}^{\mathrm{T}}\boldsymbol{\beta})\}$
- Sorting these utility scores, we obtain the ranking for recommendation



yifanhu.net › PUB ▾ PDF
Collaborative Filtering for Implicit Feedback Datasets - Yifan Hu
Types of **implicit feedback** include purchase history, browsing history, search patterns, or even mouse movements. For example, a user that pur-. Page 2. chased ...
by Y Hu - Cited by 2311 - Related articles

www.hindawi.com › journals › mpe ▾
Exploiting Explicit and Implicit Feedback for Personalized ...
The problem of the previous researches on personalized ranking is that they focused on either explicit feedback data or **implicit feedback** data rather than ...
by G Li - 2016 - Cited by 34 - Related articles
Abstract · Introduction · Related Work · Conclusion

terpconnect.umd.edu › ~oard › pdf › aaai98 ▾ PDF
Implicit Feedback for Recommender Systems - TerpConnect
Implicit feedback techniques seek to avoid this bottleneck by inferring something similar to the ratings that a user would assign from observations that are available ...
by DW Oard - Cited by 459 - Related articles

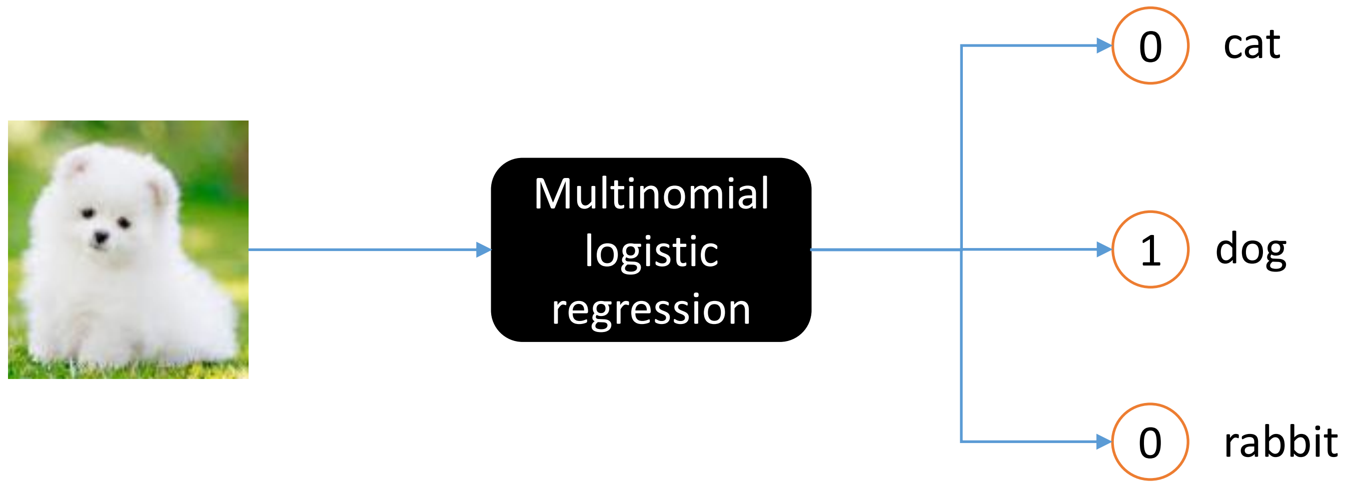
www.semanticscholar.org › paper › Implicit-Feedback-for...
Implicit Feedback for Recommender Systems | Semantic Scholar
Can **implicit feedback** substitute for explicit ratings in recommender systems? If so, we could avoid the difficulties associated with gathering explicit ratings from ...

Outline

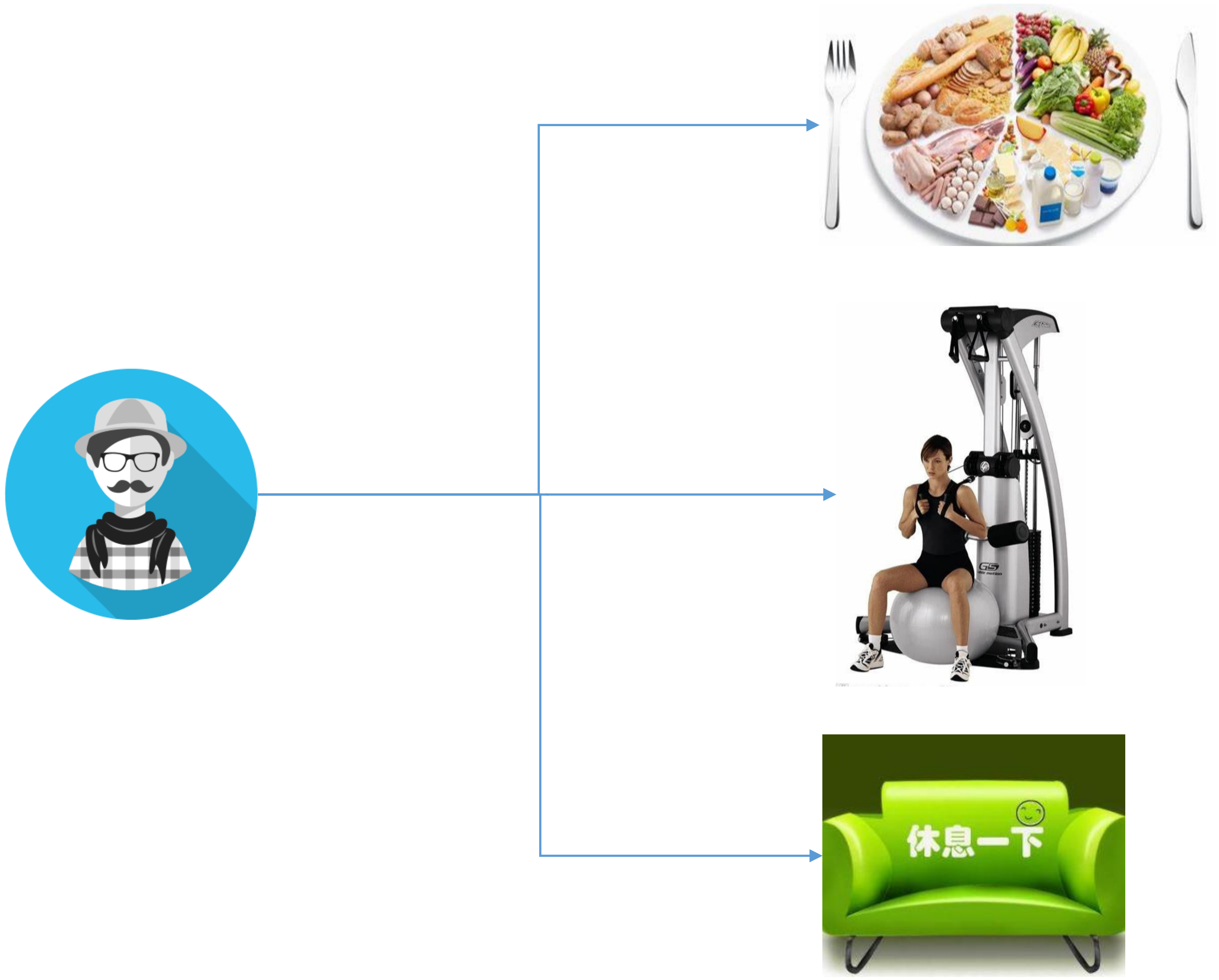
- Continuous and Categorical Data
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression

Multinomial logistic regression (softmax regression)

- Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems



Recommendation for multiple choices



The probability of a choice over candidates

- $\mathbf{V} = \{v_1, \cdots, v_N\}$: candidate set
- $\mathbf{Y}_u = \{y_{u,1}, \cdots, y_{u,N}\}$: $y_{u,i}$ denotes user u select item i
- $\boldsymbol{x}_u$**:** A feature vector of user u
- $\boldsymbol{x}_i$**:** A feature vector of item i
- $\boldsymbol{\beta}$**:** Weights on features to estimate
- $softmax(x_i|\boldsymbol{X}) = \frac{e^{x_i}}{\sum_{x_j \in \boldsymbol{X}} e^{x_j}}$
- The probability user u select item i

$$\mathrm{P}\left(y_{u,i} = 1 | \boldsymbol{x}_u, \mathbf{V}, \boldsymbol{\beta}\right) = \frac{e^{\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}}}{\sum_{v_j \in \mathbf{V}} e^{\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}}}$$

where $\boldsymbol{x}_{u,i} = [\boldsymbol{x}_u, \boldsymbol{x}_i]$

Question: What's the form of softmax for two classes?

Optimization objective

- The probability user u select item i

$$\mathrm{P}\left(y_{u,i} = 1 | \boldsymbol{x}_{u,i}, \mathbf{V}, \boldsymbol{\beta}\right) = \frac{e^{\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}}}{\sum_{v_j \in \mathbf{V}} e^{\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}}}$$

- A unified representation

$$\mathrm{P}\left(\boldsymbol{Y}_u | \boldsymbol{x}_{u,i}, \mathbf{V}, \boldsymbol{\beta}\right) = \prod_{v_j \in \mathbf{V}} \left(\frac{e^{\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}}}{Z}\right)^{y_{u,i}}, \qquad where\ Z = \sum_{v_j \in \mathbf{V}} e^{\boldsymbol{x}_{u,j}^{\mathrm{T}}\boldsymbol{\beta}}$$

- Negative log-likelihood

$$\mathrm{L}_{\boldsymbol{\beta}}\left(\boldsymbol{Y}_u | \boldsymbol{x}_{u,i}, \mathbf{V}\right) = -\log \prod_{v_j \in \mathbf{V}} \left(\frac{e^{\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}}}{Z}\right)^{y_{u,i}} = \boxed{-\sum_{v_j \in \mathbf{V}} y_{u,i} \log \frac{e^{\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}}}{Z}} = -\sum_{v_j \in \mathbf{V}} y_{u,i}\left(\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta} - \log Z\right)$$

which is often called multinomial cross-entropy

Recommendation

- Given a set of candidate item $\{x_1, x_2, \cdots, x_N\}$, we can easily obtain the probability to select an item x_i for user u, i.e. $\mathrm{P}\left(y_{u,i} = 1 | \boldsymbol{x}_u, \mathbf{V}, \boldsymbol{\beta}\right) = \frac{e^{\boldsymbol{x}_{u,i}^{\mathrm{T}}\boldsymbol{\beta}}}{Z}$

- Note that Z is the same for all candidates, and the exponential function is a monotonic increasing function.

- Therefore, we simply find the $\underset{i}{\operatorname{argmax}}\{\boldsymbol{x}_{u,1}^{\mathrm{T}}\boldsymbol{\beta}, \cdots, \boldsymbol{x}_{u,N}^{\mathrm{T}}\boldsymbol{\beta}\}$

$\boldsymbol{x}_{u,1}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{1}$

$\boldsymbol{x}_{u,2}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{2}$

$\boldsymbol{x}_{u,3}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{6}$

Summary

- Numeric, binary, categorical data representation
- Linear regression model for numeric data based recommendation
- Logistic regression model for binary data based recommendation
- Multinomial logistic regression model for categorical data based recommendation

Assignment I

TMDB 5000 Movie Dataset

This dataset was generated from The Movie Database API. This product uses the TMDb API but is not endorsed or certified by TMDb. Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows.

Task: Apply linear regression to predict movie ratings (vote_average) using movie attributes.

- Evaluation metrics: MAE, RMSE

Experimental dataset: https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata

- Training set: tmdb_5000_movies.csv (80%)
- Testing set: tmdb_5000_movies.csv (20%)

Assignment II

CTR In Advertisement

A company wants to know the CTR (Click Through Rate) in order to identify whether spending their money on digital advertising is worth or not.

Task: Apply logistic regression to predict whether an Ad. is clicked by each user.

- Evaluation metrics: Recall@1~20, MAP@5, MAP@10, nDCG@5, nDCG@10

Experimental dataset: https://www.kaggle.com/datasets/arashnic/ctr-in-advertisement

The dataset divided to train (463291, 15) and test (128858, 14). Features are clear and target is "is_click" , 0 (No) , 1(Yes).

- Training set: Ad_click_prediction_train (1).csv
- Testing set: Ad_Click_prediciton_test.csv