Predicting User Purchase in E-commerce by Comprehensive Feature Engineering and Decision Boundary Focused Under-Sampling

[RecSys Challenge 2015]

Chanyoung Park, Donghyun Kim, Jinoh Oh and Hwanjo Yu

POSTECH

The Task

Given a sequence of click events performed by some user during a typical session in an e-commerce website,

1. Predict which session will end up with a purchase
2. Predict items that are going to be bought in the session

Data

- Click dataset, test dataset
  - SessionID
  - TimeStamp
  - ItemID
  - Category
- Purchase dataset
  - SessionID
  - TimeStamp
  - ItemID
  - Price
  - Quantity

Data	Number of entries
Click dataset (yoochoose_clicks.dat)	33,003,944
Purchase dataset (yoochoose_buys.dat)	1,150,753
Test dataset (yoochoose_test.dat)	8,251,791

Challenges

- Given data itself lack sufficient information
  - Existence of missing values
  - No user related information (User demographic information)
  - Not enough item related information
  - Hard to build accurate model

- Massive volume of dataset
  - 33 million clicks, 1 million purchases
  - Increases model training time and memory usage

- Highly imbalanced class distribution
  - Non-purchased clicks : Purchased clicks = 25 : 1
  - Model may be biased towards the majority class → poor accuracy

Our Approach

- Comprehensive Feature Engineering (CFE)
  - To make up for insufficiency of information
- Decision Boundary Focused Under-Sampling (DBFUS)
  - To reduce model training time and memory usage
  - To cope with class imbalance problem

Problem Setting: Binary classification

- Recall that there are two tasks in *RecSys Challenge 2015*
- We integrated these two tasks and converted them into a simple binary classification problem

I. Label each click instance in click dataset using purchase dataset
   - Clicks that contain a purchased item are labeled as positive, otherwise negative

II. For each given click instance, we predicted whether or not the click will end up with purchase regardless of the sessionID

III. After the prediction process, we can tell that a session with any positively predicted click is a session involving purchase

Problem Setting: Binary classification

- Example of labeling click dataset

Click dataset

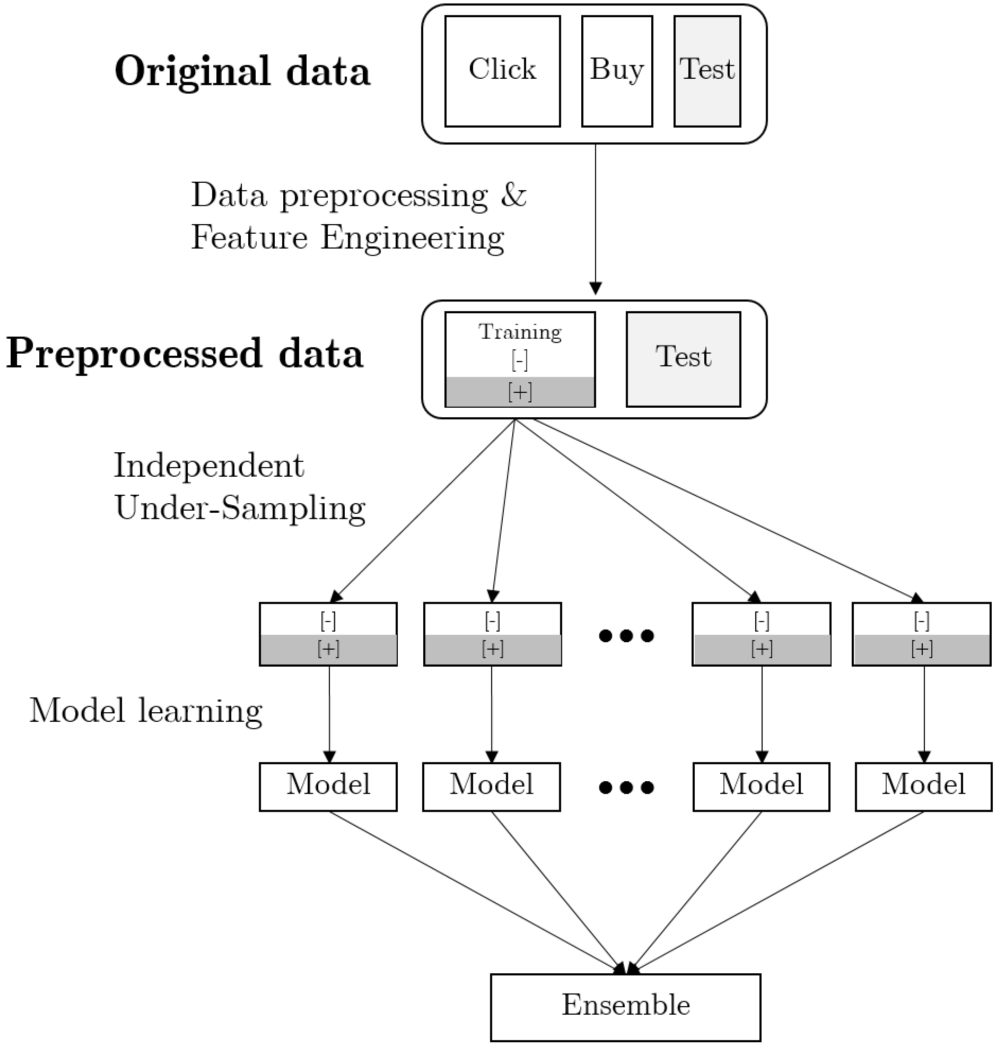
SessionID		ItemID		Label
1	…	100	…	**1**
1	…	100	…	**1**
1	…	200	…	**1**
1	…	100	…	0
2	…	300	…	**1**
2	…	400	…	0
3	…	100	…	0
3	…	500	…	**1**

Purchase dataset

SessionID		ItemID	
1	…	100	…
1	…	200	…
2	…	300	…
3	…	500	…

POSTECH

System Overview



Comprehensive Feature Engineering (CFE)

Decision Boundary Focused Under-Sampling (DBFUS)

Training with Gradient Boosting Classifier

Comprehensive Feature Engineering (CFE)

Preliminaries
- Class imbalance Problem

- # of Negative class instances >> # of Positive class instances

  (Non-purchased clicks) (Purchased clicks)

- Classifiers are biased towards the majority class resulting in poor accuracy in correctly classifying the positive class instances
- Different methods
  - Over-sampling
  - Under-sampling
  - Hybrid

Preliminaries
- Gradient Boosting Classifier (GBT)

- Predictive modeling algorithm for classification & regression
- Decision tree is typically used as a base learner
- Boosting
  - Multiple weak learners are combined to improve overall performance
- Provides *feature importance* information

Comprehensive Feature Engineering (CFE)

- *Feature Engineering* implies …
  - Imputation of missing values
  - Extraction of informative features from the original data (Engineered feature)
- Feature selection
- Verification of the quality of CFE using Principal Component Analysis (PCA)

Comprehensive Feature Engineering (CFE)
- Imputation of missing values

- ***Category*** **(0: missing, 1~12: valid, >12: brand)**
  - Missing category information of an item can be induced by looking at the data of other months
  - Category = “0” (Missing)
    - Converted into “1 ~ 12” if possible, otherwise remains as “0”
  - Category > “12” (Brand)
    - Converted into “1 ~ 12” if possible, otherwise converted into “13”

Comprehensive Feature Engineering (CFE)
- Imputation of missing values

- ***Price / Quantity*** **(0: missing)**
  - Price / quantity = "0" for *item A* in April → look if other logs of April contain information about *item A*
    - Fill in the missing entries with the mean value of the *item A* in April
  - No other logs of April contain information about *item A*
    - Fill in the missing entries with the mean value of the *item A* in the entire data
  - No price / quantity information about *item A* at all
    - Fill in the missing entries with the mean value of all items

Comprehensive Feature Engineering (CFE)
- Engineered Features

No.	Feature Name	Type	Description
1	Day	Categorical	31 days of a month are divided into 4 bins according to the number of clicks forming a binary vector of length 4
2	Weekday	Categorical	7 weekdays of a week form binary vector of length 7
3	Hour	Categorical	24 hours of a day are divided into 5 bins according to the number of clicks forming binary vector of length 5
4	Category	Categorical	a binary vector of length 14 is formed after the imputation step
5	Price / Quantity	Numerical	price and quantity of items purchased
6	Category S	Boolean	whether an item is in sale or not
7	Last Session	Boolean	whether an instance is the last click in the session or not
8	One category in a session	Boolean	whether the user browsed only one category in a session or not
9	Category ratio vector	Numerical Vector	If there are 3 clicks occurred in a session and each one of them clicked on a different category, then (0.33, 0.33, 0.33)
10	Weekend	Boolean	whether it is weekend or not

No.	Feature Name	Type	Description
11	SNC	Numerical	number of clicks in a session
12	INW	Numerical	number of clicks of an item among the whole training data
13	INC	Numerical	number of clicks of an item in a session
14	IBW	Numerical	number of purchases of an item among the complete training data
15	DUR	Numerical	duration of a session in seconds
16	S1	Numerical	INC / SNC. Higher value implies higher probability of ending with purchase
17	S2	Numerical	IBW / INW. Higher value implies higher probability of ending with purchase
18	IMC	Numerical	number of clicks of an item in a month
19	IMB	Numerical	number of purchases of an item in a month
20	IR1	Numerical	ratio of an item in a session
21	IR2	Numerical	ratio of an item clicks in a session
22	CR1	Numerical	ratio of a category in a session
23	CR2	Numerical	ratio of a category clicks in a session

Comprehensive Feature Engineering (CFE)
- Feature Selection

- Gradient Boosting Classifier provides *feature importance* information
- Calculated feature importance scores for numerical features
  - Categorical features give useful information as a whole

Feature	Import.	Feature	Import.	Feature	Import.
P	0.036	IBW	**0.005**	IMB	0.037
Q	0.042	DUR	0.146	IR1	0.034
SNC	0.027	S1	0.025	IR2	0.027
INW	0.034	S2	0.062	CR1	**0.007**
INC	0.05	IMC	0.04	CR2	0.031

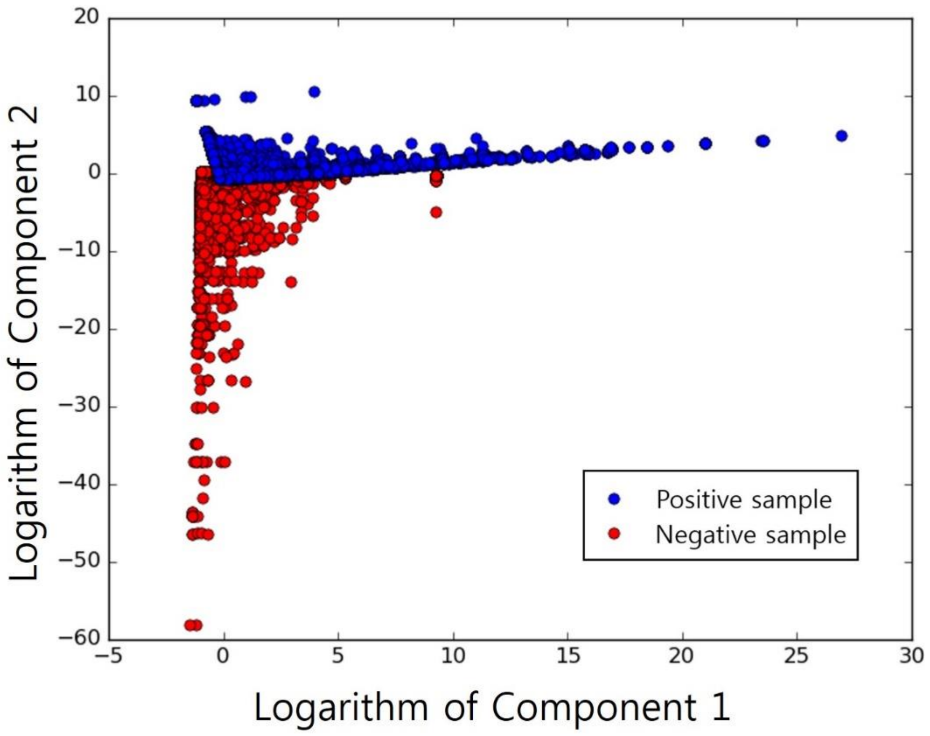
Feature importances of numerical features

Comprehensive Feature Engineering (CFE)
- Quality verification of engineered features

- To verify the quality of engineered features
  I. Perform Principal Component Analysis (PCA) on the data represented by engineered features
  II. Take first two principal components to visualize the data

Comprehensive Feature Engineering (CFE)
- Quality verification of engineered features

- The instances are *<u>surprisingly well divided</u>* according to the first two principal components
  - Our feature engineering process made success in extracting valuables features that represent the data!

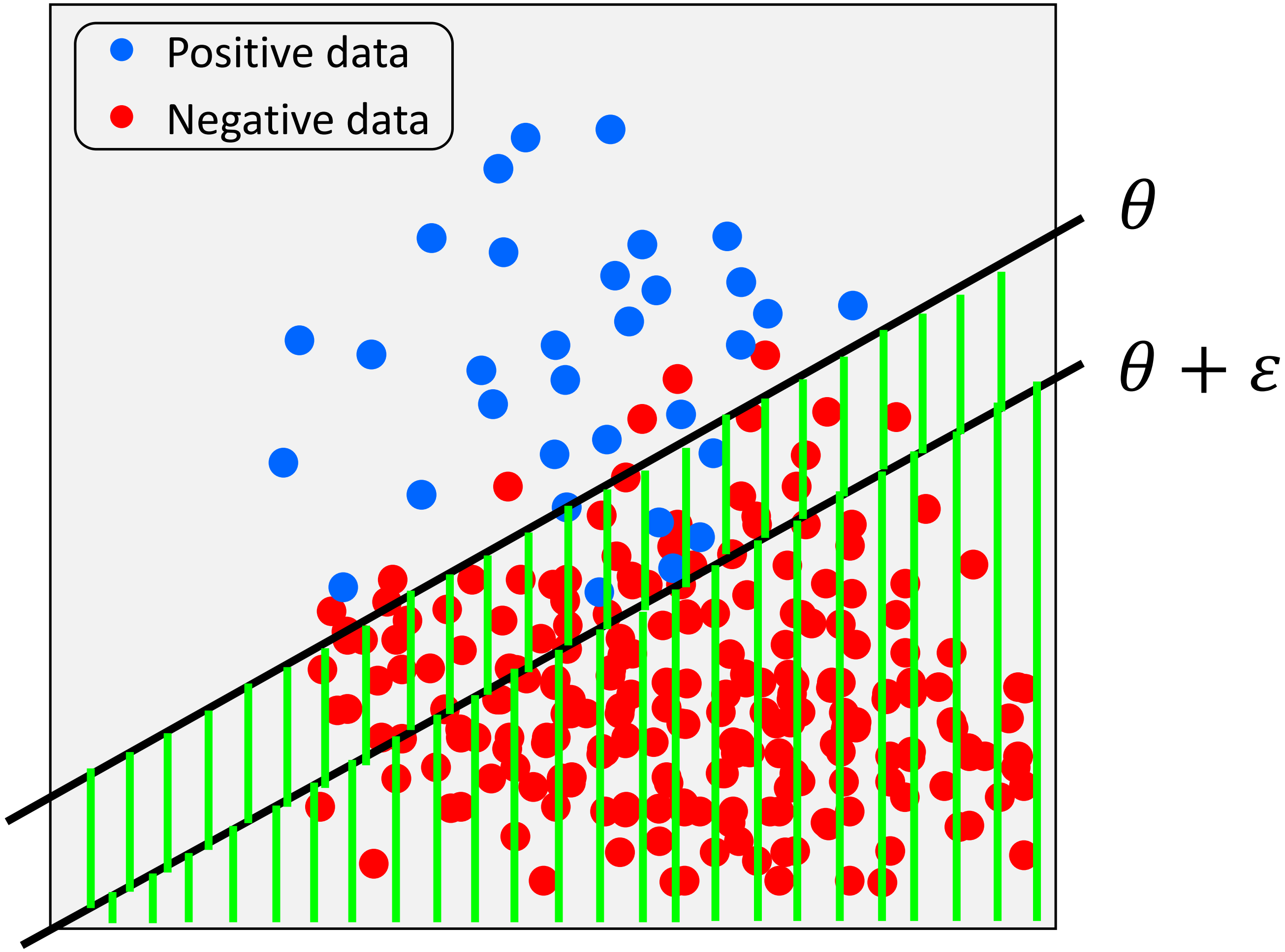


Decision Boundary Focused Under-Sampling (DBFUS)

Decision Boundary Focused Under-Sampling (DBFUS)

- Perform **under-sampling** on the data of majority class while keeping all the data in the minority class
  - To reduce model training time and memory usage
  - To alleviate class imbalance problem
    - Non-purchased clicks : Purchased clicks = 25 : 1
- Consider the distance to the decision boundary such that more data is sampled near the decision boundary

Decision Boundary Focused Under-Sampling (DBFUS)



I. Calculate decision boundary θ using click dataset

II. Keep positive data and only sample from negative data

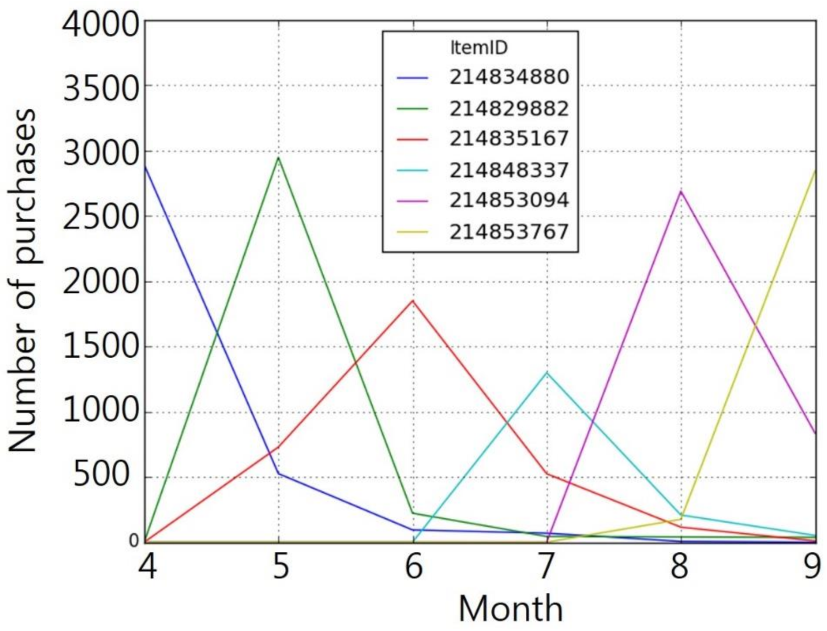
- A half from $\theta < instances < \theta + \varepsilon$
- The other half from $instances > \theta + \varepsilon$

Decision Boundary Focused Under-Sampling (DBFUS)

- *"Non-purchased clicks : Purchased clicks = 3 : 1"* shows the best performance
- Reduced imbalance ratio from 25:1 to 3:1
  - Reduced model training time and memory usage + alleviated class imbalance problem
  - However, may cause information loss
    - Solution: Independently perform DBFUS 25 times and train 25 different models
- *"Ensemble of ensembles"*
  - Gradient boosting classifier is used to train the model

Learning Strategy

- *Splitting Monthly*
  - Purchase patterns for each month are significantly different
  - Thus, we split the data monthly and constructed our model for each month
    - Improvement by more than 5,000 points on the leaderboard!



Summary of Results

- Implemented using *Python Scikit-Learn*
- Parameters for gradient boosting classifier

Parameters	Value
num estimators	5000
max leaf nodes	20
max depth	N/A
min samples split	1
learning rate	0.17
max features	number of whole features

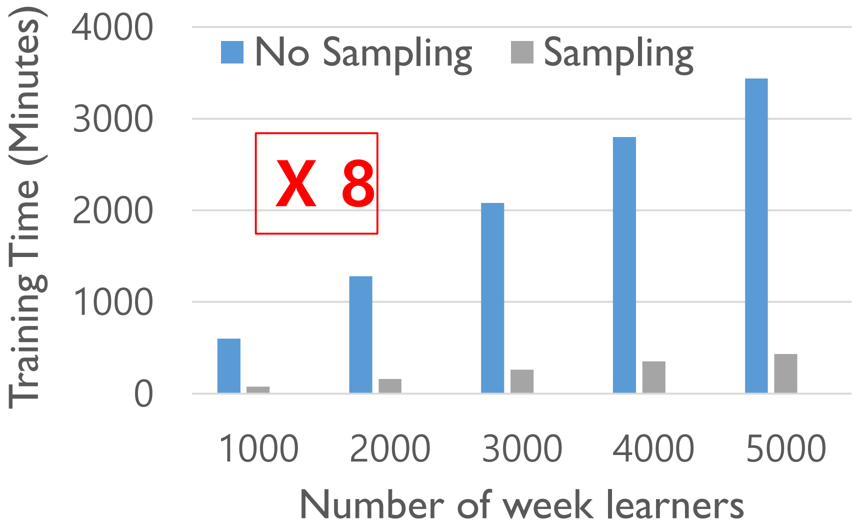
Summary of Results

- Final result

Model	Leaderboard score
Less features + No sampling + GBT	49587.8
CFE + DBFUS + Neural Net	49444.4
CFE + No sampling + GBT	52525.5
CFE + DBFUS + GBT	**54403.6**

Summary of Results

- Training time



- Memory usage

Conclusion

- **Challenges** for *RecSys Challenge 2015*
  - Insufficiency of information
  - Inefficiency in model training time and memory usage
  - Class imbalance problem

Solution

Comprehensive Feature Engineering (CFE)

Decision Boundary Focused Under-Sampling (DBFUS)

- We achieved 54,403.6 in the final leaderboard (10th/569 teams)