About Dataset

Context

The dataset consists of three files: a file with behaviour data (events.csv), a file with item properties (item_properties.csv) and a file, which describes category tree (category_tree.csv). The data has been collected from a real-world ecommerce website. It is raw data, i.e. without any content transformations, however, all values are hashed due to confidential issues. The purpose of publishing is to motivate researches in the field of recommender systems with implicit feedback.

Content

The behaviour data, i.e. events like clicks, add to carts, transactions, represent interactions that were collected over a period of 4.5 months. A visitor can make three types of events, namely “view”, “addtocart” or “transaction”. In total there are 2 756 101 events including 2 664 312 views, 69 332 add to carts and 22 457 transactions produced by 1 407 580 unique visitors. For about 90% of events corresponding properties can be found in the “item_properties.csv” file.

For example:

- “1439694000000,1,view,100,” means visitorId = 1, clicked the item with id = 100 at 1439694000000 (Unix timestamp)
- “1439694000000,2,transaction,1000,234” means visitorId = 2 purchased the item with id = 1000 in transaction with id = 234 at 1439694000000 (Unix timestamp)

The file with item properties (item_properties.csv) includes 20 275 902 rows, i.e. different properties, describing 417 053 unique items. File is divided into 2 files due to file size limitations. Since the property of an item can vary in time (e.g., price changes over time), every row in the file has corresponding timestamp. In other words, the file consists of concatenated snapshots for every week in the file with the behaviour data. However, if a property of an item is constant over the observed period, only a single snapshot value will be present in the file.

For example, we have three properties for single item and 4 weekly snapshots, like below:

```
timestamp,itemid,property,value
1439694000000,1,100,1000
1439695000000,1,100,1000
1439696000000,1,100,1000
1439697000000,1,100,1000
1439694000000,1,200,1000
1439695000000,1,200,1100
1439696000000,1,200,1200
1439697000000,1,200,1300
1439694000000,1,300,1000
1439695000000,1,300,1000
1439696000000,1,300,1100
1439697000000,1,300,1100
```

After snapshot merge it would looks like:

```
1439694000000,1,100,1000
1439694000000,1,200,1000
1439695000000,1,200,1100
1439696000000,1,200,1200
1439697000000,1,200,1300
1439694000000,1,300,1000
1439696000000,1,300,1100
```

Because property=100 is constant over time, property=200 has different values for all snapshots, property=300 has been changed once.

Item properties file contain timestamp column because all of them are time dependent, since properties may change over time, e.g. price, category, etc. Initially, this file consisted of snapshots for every week in the events file and contained over 200 millions rows. We have merged consecutive constant property values, so it's changed from snapshot form to change log form. Thus, constant values would appear only once in the file. This action has significantly reduced the number of rows in 10 times.

All values in the “item_properties.csv” file excluding "categoryid" and "available" properties were hashed. Value of the "categoryid" property contains item category identifier. Value of the "available" property contains availability of the item, i.e. 1 means the item was available, otherwise 0. All numerical values were marked with "n" char at the beginning, and have 3 digits precision after decimal point, e.g., "5" will become "n5.000", "-3.67584" will become "n-3.675". All words in text values were normalized (stemming procedure: https://en.wikipedia.org/wiki/Stemming) and hashed, numbers were processed as above, e.g. text "Hello world 2017!" will become "24214 44214 n2017.000"

The category tree file has 1669 rows. Every row in the file specifies a child categoryId and the corresponding parent.
For example:

- Line “100,200” means that categoryid=1 has parent with categoryid=200
- Line “300,” means that categoryid hasn’t parent in the tree

Tasks

Task 1

When a customer comes to an e-commerce site, he looks for a product with particular properties: price range, vendor, product type and etc. These properties are implicit, so it's hard to determine them through clicks log.

Try to create an algorithm which predicts properties of items in "addtocart" event by using data from "view" events for any visitor in the published log.

Task 2

Description:

Process of analyzing ecommerce data include very important part of data cleaning. Researchers noticed that in some cases browsing data include up to 40% of abnormal traffic.

Firstly, abnormal users add a lot of noise into data and make recommendation system less effective. In order to increase efficiency of recommendation system, abnormal users should be removed from the raw data.

Secondly, abnormal users add bias to results of split tests, so this type of users should be removed also from split test data.

Goals:

- The main goal is to find abnormal users of e-shop.

Subgoals:

- Generate features
- Build a model
- Create a metric that helps to evaluate quality of the model