* 1. DATA SET

Data sets used to make predictions in this assignment are reviews of businesses from Google Local and meta and reviews of Video Games from Amazon. The attributes we may make use of in each review are:

1. For reviews of businesses from Google Local

businessID: The hashed ID of each unique business (eg.B660702032)

userID: The hashed ID of each unique user (eg.U342536218)

categories: A list of categories this business belongs to (eg.[Donut Shop, Dessert Shop, Bakery])

ratings: The star rating of the user’s re-view (eg.4.0)

unixReviewTime: Time of the review in seconds since 1970 (eg.1383057010)

1. For detailed information of Video Games(meta) from Amazon data set:

asin: The hashed ID of each unique video game (eg.B00005O0I2)

categories: A list of categories this video game belongs to (eg.['Video Games', 'PC', 'Games'])

price: The unit price of each unique video game (eg.23.88)

salesRank: The rank of each unique game in all kinds of Video Games (eg. 22579)

1. For reviews of video games from Amazon data set

We built an adjacency matrix based on the (user, item) pairs extracted from reviews of video games from Amazon. If the user reviewed this item, then the corresponding value in this matrix is 1, else is 0.

* 1. MOTIVATION

Based on features extracted from detailed information of Video Games(meta), such as price, platform in categories and salesRank, those feature-based models – logistic regression and SVM—can be used to predict whether a particular video game is visited by a user or not. A part of the adjacency matrix extracted from Amazon data set can be used as labels to train the model and another part can be used to calculate the accuracy of prediction as test set.

Since those features included in meta may not be that related to the visit labels, performance of those models will not be very good. So, the collaborative filter based on the data of reviews of video games from Amazon data set which computing similarities between items can also be applied.

2

This task requires a binary output for each user-business pair, the model we have learn-ng related to it are logistic regression, SVM and the collaborative filter etc. Thus, there are several ways to solve it.

For convenience, we define I to represent all items in training set, S to be the data in train-ing set, and

公式

2.1 EVALUATION APPROACH

To evaluate performances of those models, we choose to compute the accuracy of prediction of each model. At first, extract visited (user, item) pairs from reviews of video games from Amazon data set. Then, by choosing those random generated (user, item) pairs which are not in visited pairs, we generate a negative labeled train set whose length is exactly equal to the positive labeled train set extracted directly from reviews of video games from Amazon data set. We combine these two train sets as a final train set. Test set generated by the same method.

2.2 BASELINE: MOST POPULAR

略

2.3 BASELINE: LOGISTIC REGRESSION

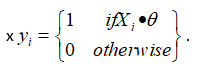
Logistic regression is a method to convert a real valued expression into a probability and then train a classifier for prediction. As illustrated in the above equation, when the score is greater than 0 then the prediction will be 1, and when the score is less than 0 then the prediction will be 0.

\[{{y}\_{i}}=\left\{ \begin{matrix}

1 & if{{X}\_{i}}\centerdot \bullet \theta \\

0 & otherwise \\

\end{matrix} \right\}\]



Sigmoid function is used to accomplish the conversion between the real-valued expression and the probability.

\[\sigma (t)={}^{1}/{}\_{1+{{e}^{-t}}}\]



should be maximized when label is positive and minimized when label is negative. Thus, by calculating the log-likelihood, computing the gradient and solving the problem with gradient ascent, we could achieve an appropriate model.

Log-Likelihood:  \[{{l}\_{\theta }}(y|X)=\sum\nolimits\_{i}{-\log (1+{{e}^{-{{X}\_{i}}\centerdot \theta }})}+\sum\nolimits\_{{{y}\_{i}}=0}{-{{X}\_{i}}\centerdot \theta }-\lambda ||\theta ||\_{2}^{2}\]

Derivative: \[\frac{\partial l}{\partial {{\theta }\_{k}}}=\sum\nolimits\_{i}{{{X}\_{ik}}(1-\sigma ({{X}\_{i}}\centerdot \theta )})+\sum\nolimits\_{{{y}\_{i}}=0}{-{{X}\_{ik}}}-2\lambda {{\theta }\_{k}}\]

Considering our dataset, features we selected for the logistic regression include the platforms, the price and the sales ranking of the video game.



\[Visited=\left\{ \begin{matrix}

1,if\left[ ifVGA,ifPC,ifLinux,ifMac,ifN\operatorname{int}edno,ifPlayStation,ifSonyPSP,ifWii,ifXbox,\Pr ice,salesRank \right]\cdot \theta >0 \\

0,otherwise \\

\end{matrix} \right\}\]

2.4 BASELINE: SVM

Since the logistic regressors don’t optimize the number of mistakes, and we want to minimize the number of misclassifications, the SVM will fix the issue:

\[\arg {{\min }\_{\theta }}\sum\nolimits\_{i}{\delta ({{y}\_{i}}({{X}\_{i}}\centerdot \theta -\alpha )}\le 0)\]

Compared to logistic regression, SVM is trying to find a perfect classifier and the soft-margin formulation is used to figure out the classifier that maximizes the distance to the nearest points.

\[{{\forall }\_{i}}{{y}\_{i}}({{X}\_{i}}\centerdot \theta -\alpha )\ge 1\]

In practical, features we selected for the SVM also include the platforms, the price and the sales ranking of the video game.



\[Visited=\left\{ \begin{matrix}

1,if\left[ ifVGA,ifPC,ifLinux,ifMac,ifN\operatorname{int}edno,ifPlayStation,ifSonyPSP,ifWii,ifXbox,\Pr ice,salesRank \right]\cdot \theta -\alpha >0 \\

-1,otherwise \\

\end{matrix} \right\}\]

2.5 BPR

略