INTRODUCTION

• Motivation: Display recommended content that are more likely to get clicked when common variables are taken into consideration
• Problem Definition: Given some webpage and ad information, determine which ad is more likely to be clicked on the given webpage
• Input: Sparse information related to each webpage, the ads on the webpage
• Output: Which ad is clicked of a set of ads that user is reading
• Challenge: Not given string-based information about the webpage or ads
• Result: SVM resulted in the best accuracy, but achieving high accuracy is limited due to the limited amount of features and the anonymized nature of the data

PRE-PROCESS DATA CONT.

• Predicting: 6 ad id’s will be given under a display id, in which one ad is clicked
• Features:
  • Topic id: Topic of webpage
  • Category id: Category of webpage
  • Source id: source that webpage is from
  • Advertiser id: Advertiser of ad
  • Campaign id: Campaign of ad

LOGISTIC REGRESSION

• Logistic regression (LR): LR optimizer helps reduce the logistic loss through SGD. We extracted features based on the top 50 topics, categories, advertisers, and campaigns that are clicked

PCA

• Principle Component Analysis: PCA helps reduce high dimension data to lower dimensions, so one can visualize how hard it is to find a hypothesis to separate the data

SVM

• Support Vector Machine: SVM optimizer helps reduce the hinge loss through SGD. We use the same features as LR

NAÏVE BAYES

• Naïve Bayes: Naïve Bayes assumes each feature is conditionally independent. We found the conditional probability based on how many times advertisers and campaigns are clicked

MULTILAYER PERCEPTRON

• MLP: Hyperparameters are tuned to select the optimal number of hidden layers, the learning rate, and the optimization algorithm

PRE-PROCESS DATA

• Raw Data: Collect data from multiple csv files and map it to dictionaries
• Graphic Representation of Data: Display ID Raw Data: Ad ID

RESULT

• Comparison:
<table>
<thead>
<tr>
<th>Method</th>
<th>Rand</th>
<th>Naïve Bayes</th>
<th>LR</th>
<th>SVM</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>N/A</td>
<td>45%</td>
<td>51%</td>
<td>52%</td>
<td>57%</td>
</tr>
<tr>
<td>Testing</td>
<td>15%</td>
<td>45%</td>
<td>46%</td>
<td>48%</td>
<td>43%</td>
</tr>
</tbody>
</table>

DISCUSSION

• Problem was simplified by cutting down the data effectively with the risk of missing information relevant for training
• Even effectively trimming data still resulted in low accuracy because the data does not contain rich features
• All of the advertisement and webpage information have been transformed through some unknown mapping to numbers
• The best results from others on Kaggle yield accuracy about 60%, so an accuracy more than 60% is not expected

FUTURE

• Use of parallel computing to go through the complete dataset and to train the MLP model more efficiently
• Determine an optimal clustering for the users based on location
• Find a better numerical relationship between the ad campaign id’s and advertiser id’s rather than assuming the given mapping is valid

REFERENCE

• (2) Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.