pCTR Challenge

Kenny He and Chris Yu

Background

- Search advertising has been one of the major revenue sources of the Internet industry for years. A key technology behind search advertising is to predict the click-through rate (pCTR) of ads, which drives the pricing model.
- This year we will be looking at the dataset of image ads provided by Tencent Multimedia

GOAL:

- Given training dataset of online advertising system results, contestants must accurately predict the test data

Dataset

The training data is published as a text file, where each line is a training instance derived from session log messages. (23,906,738)

UserID	AdvID	AdID	CreativeID	Impression	Click

Evaluation DS (3,253,943)   Validation DS (3,236,631)

Additional Files:

titles.txt (34,163 lines)

AdID	Title

images.zip: folder of all the image ads provided (24,125 jpg files)

users.txt (23,439,495 lines, 491MB in size)

UserID	Gender	Age

Approach

- Classification of Titles
- Sort User Data by UID
- Split Training Data by Gender and Age
- Analyze Image Characteristics
- Build the model

First Step: Classification of Ad Titles

- Put the different keywords from titles.txt into categories (for us it was 7 categories)
    - Men’s wear, Women’s wear, Gaming, Beauty, Household, Education, etc.
- Scan through each title, and keep weight of each Category that the keywords are from
- The category with the highest keywords weight will be the label for that title

2nd Step: Sort Users Data by UID

- Read records from Users.txt, and build a binary search tree of the user information
- The nodes in the tree are stored in an array
- A hash function to generate a number to represent a combination of age and gender:

  GA(age, gender) = 31 | (age*2 + gender) | 130

  age <=15   15<age<65   age>=65

3rd Step: Split Training Data

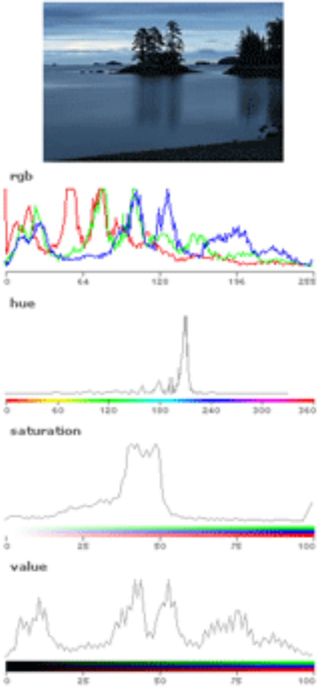
- We are sure that age and gender are the two most important factors related with the interest to certain ads
- Split the huge training dataset by the age and gender of the users into 100 groups
- Take advantage of the BST and hash function created in step 2 for higher performance

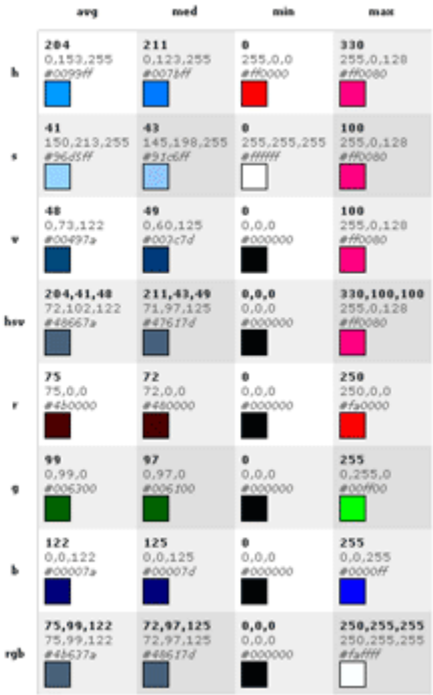
4th Step: Analyze Image Features

Using a Web-based API: http://mkweb.bcgsc.ca/color_summarizer/

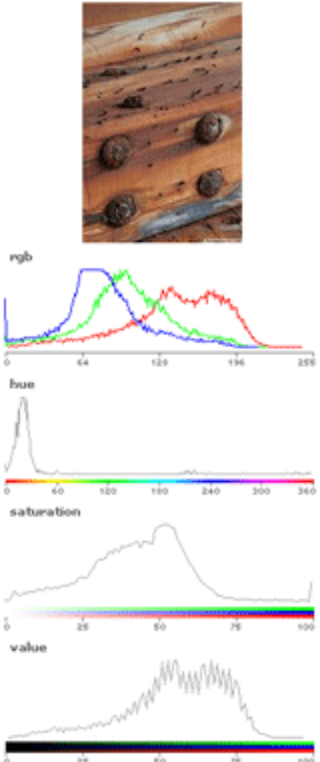
- The api is capable of analyzing image’s color palette and color statistics
- We will mine the api for image features to be added to the prediction model

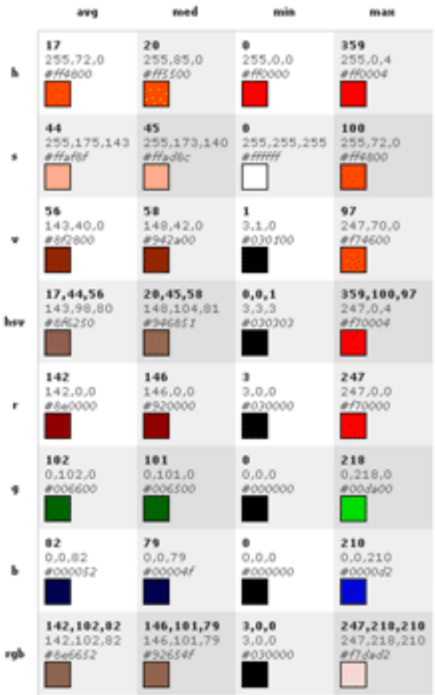
Examples





	avg	med	min	max
h	**204** 0,153,255 *#0099ff*	**211** 0,123,255 *#007bff*	**0** 255,0,0 *#ff0000*	**330** 255,0,128 *#ff0080*
s	**41** 150,213,255 *#96d5ff*	**43** 145,198,255 *#91c6ff*	**0** 255,255,255 *#ffffff*	**100** 255,0,128 *#ff0080*
v	**48** 0,73,122 *#00497a*	**49** 0,60,125 *#003c7d*	**0** 0,0,0 *#000000*	**100** 255,0,128 *#ff0080*
hsv	**204,41,48** 72,102,122 *#48667a*	**211,43,49** 71,97,125 *#47617d*	**0,0,0** 0,0,0 *#000000*	**330,100,100** 255,0,128 *#ff0080*
r	**75** 75,0,0 *#4b0000*	**72** 72,0,0 *#480000*	**0** 0,0,0 *#000000*	**250** 250,0,0 *#fa0000*
g	**99** 0,99,0 *#006300*	**97** 0,97,0 *#006100*	**0** 0,0,0 *#000000*	**255** 0,255,0 *#00ff00*
b	**122** 0,0,122 *#00007a*	**125** 0,0,125 *#00007d*	**0** 0,0,0 *#000000*	**255** 0,0,255 *#0000ff*
rgb	**75,99,122** 75,99,122 *#4b637a*	**72,97,125** 72,97,125 *#48617d*	**0,0,0** 0,0,0 *#000000*	**250,255,255** 250,255,255 *#faffff*





	avg	med	min	max
h	**17** 255,72,0 *#ff4800*	**20** 255,85,0 *#ff5500*	**0** 255,0,0 *#ff0000*	**359** 255,0,4 *#ff0004*
s	**44** 255,175,143 *#ffaf8f*	**45** 255,173,140 *#ffad8c*	**0** 255,255,255 *#ffffff*	**100** 255,72,0 *#ff4800*
v	**56** 143,40,0 *#8f2800*	**58** 148,42,0 *#942a00*	**1** 3,1,0 *#030100*	**97** 247,70,0 *#f74600*
hsv	**17,44,56** 143,98,80 *#8f6250*	**20,45,58** 148,104,81 *#946851*	**0,0,1** 3,3,3 *#030303*	**359,100,97** 247,0,4 *#f70004*
r	**142** 142,0,0 *#8e0000*	**146** 146,0,0 *#920000*	**3** 3,0,0 *#030000*	**247** 247,0,0 *#f70000*
g	**102** 0,102,0 *#006600*	**101** 0,101,0 *#006500*	**0** 0,0,0 *#000000*	**218** 0,218,0 *#00da00*
b	**82** 0,0,82 *#000052*	**79** 0,0,79 *#00004f*	**0** 0,0,0 *#000000*	**210** 0,0,210 *#0000d2*
rgb	**142,102,82** 142,102,82 *#8e6652*	**146,101,79** 146,101,79 *#92654f*	**3,0,0** 3,0,0 *#030000*	**247,218,210** 247,218,210 *#f7dad2*

5th Step: Build the model

- Building the model to predict pCTR for each ad for gender/age groups:
- Apply logistic regression for CTR predictions

			Image Features							
UserID	Age	Gender	F1	F2	F3	...	Fn	image size	title category	pCTR

Potential Trends

Gender-based ads vs Color trends (pink/red for women, black/blue for men)

Gaming-based ads vs saturation and contrast (high contrast, full colors)

CTR rate vs Text Area (what % of image area is text or non-text)

Difficulties

- Huge amount of user data and training data
- No tool or conventional method for data splitting
- Need to do title classification, which requires lots of work to manually identify keywords and categories
- Time consuming on image data analysis
- Our goal is to build a model to predict the pCTR for different age/gender groups, rather than a model for predicting the individual clicks -- avoid overfitting problem

Summary & Thanks!

- Classification / Data Split / Logistic Regression

- Q & A

- Contact us if you have good ideas about the image analysis