Automatic Advertising Campaign Development

Matina Thomaidou, Kyriakos Liakopoulos, Michalis Vazirgiannis

Athens University of Economics and Business

April, 2011
Outline

1. Online Advertising
2. Keyword and Ad Creative Generation
3. Campaign Creation and Budget Optimization
4. Experiments
5. Conclusions and Future Work
Online Advertising

- **Online advertising** is a form of promotion that uses the Internet and World Wide Web for the expressed purpose of delivering marketing messages to attract customers.

- Benefits:
  - More targeted than traditional means - Better ROI
  - Immediate publishing of information
  - Good conversion tracking
  - Purchase offline but in most cases research online first (ROPO)
Online Advertising

- Textual ads - Two main channels for distributing such ads:
  1. Sponsored search (or paid search advertising) places ads on the result pages of a Web search engine, where ads are selected to be relevant to the search query
  2. Content match (or contextual advertising) places ads on third-party Web pages
- All major Web search engines (Google, Microsoft, Yahoo!) support sponsored ads and act simultaneously as a Web search engine and an ad engine
- Pricing Models: Pay-per-click (PPC), Pay per action (PPA), Pay-per-impression (PPI)
Terminology

- **Keyword**: A word or phrase that matches a web-users search query and at the same time describes the content advertised
- **Ad-Creative**: The text that a web-user reads on an advertisement
- **Impression**: The appearance of an advertisement in a SERP after a web-users query
- **Click**: The action of a web-user clicking on an advertisement
- **Conversion**: Action (e.g. purchase, registration) after arriving to a website
- **Campaign**: Set of components and preferences for the advertising purpose
- **Ad Group**: Set of related ads, keywords, and placements within a campaign
Google AdWords

- Google 2010 revenues: $30 billion, 97% come from ads
- Adwords is Google’s flagship advertising product

- Offers pay-per-click (PPC) advertising for text, banner, and rich-media ads
The problem

**Products**
- Advertiser has products to be advertised
- Products have landing pages

**Keywords, Ad Text**
- Keywords needed for bidding, based on intuition
- Ad-text needed for showing ad

**AdWords Management**
- Campaigns need to be created and configured
- Keyword bidding needs to be optimized regularly
Our approach

Keywords and Ad Creatives

- Extract features of products from landing pages
- Find keywords that best describe products
- Generate Ad text based on patterns
- Suggest more keywords related to a given keyword

Campaign Creation and Optimization

- Use keywords and AdTexts to create campaign
- Monitor and test keyword performance keeping statistics
- Select best keywords to optimize profit or traffic
- Automatically repeat optimization in regular time intervals
Automatic Advertising Campaign Development

Figure: Automatic Campaign Creation and Optimization System - Parts
Keyword Extraction Module

Table: Tag Weights

<table>
<thead>
<tr>
<th>Element</th>
<th>Assigned Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;title&gt;</td>
<td>50</td>
</tr>
<tr>
<td>meta keywords</td>
<td>40</td>
</tr>
<tr>
<td>meta description</td>
<td>40</td>
</tr>
<tr>
<td>anchor text</td>
<td>30</td>
</tr>
<tr>
<td>&lt;h1&gt;</td>
<td>30</td>
</tr>
<tr>
<td>&lt;b&gt;</td>
<td>10</td>
</tr>
<tr>
<td>other</td>
<td>1</td>
</tr>
</tbody>
</table>

- Java using JDK SE 6 on the Eclipse IDE
- Jericho HTML Parser: java library allowing analysis and manipulation of parts of an HTML document. Eclipse Public License (EPL) and GNU Lesser General Public License (LGPL).
- Stopwords:
  - KEA data file for english stopwords
  - Lucene GreekAnalyzer.java for greek stopwords
Keyword Suggestion Module

For each given seed keyword (extracted from previous step)

- keyword is entered as a query into a search engine API (Google JSON/Atom Custom Search API) example: "car rental"
- API returns a set of short text snippets relevant to the query
- The top 30 results are downloaded and loaded in Apache Lucene Library

**Figure: Snippet**
Suggestions

Data

- Response data in Atom format
- Retrieve feed/entry/summary/text() string type property indicating the snippet of the search result and feed/entry/title/text() string type property indicating the title of the search result
- Each set of title and snippet results that were retrieved after a seed query represents a document for Lucene
- Find new unique distinct words
- Co-occurrence find phrases
- Lucene scoring: Similarity class

Apache Lucene

- Cross-Platform Solution
- Available as Open Source software under the Apache License which lets you use Lucene in both commercial and Open Source programs
- 100% pure Java
- Implementations in other programming languages available that are index-compatible
Scoring

\[ w_{i,j} = tf \times idf = \frac{freq_{i,j}}{\text{Max}_{l} freq_{l,j}} \times \log \left( \frac{N}{n_i} \right) \] (1)

\[ w_{j\text{tag}} = \text{weight}_{\text{tag}} \times f_{j\text{tag}} \] (2)

\[ \text{special}_{-} \text{weight}_j = \sum w_{j\text{tag}} \] (3)

\[ \text{relevance}_{-} \text{score}_j = \frac{\text{special}_{-} \text{weight}_j}{\text{MAX}_{-} \text{WEIGHT}} \] (4)

\[ \text{boosted}_{-} \text{score}_j = \text{relevance}_{-} \text{score}_j \times k^{\text{noOfWords}} \] (5)
Ad Creative Module

- Extract all the text from the HTML document
- Summarize to keep the most important meaning for the description of our advertising page
- Classifier4J: Java library. Text summary facility
- Divide the sentence into the two description lines
- Cut exceeded words
Components Diagram

Figure: Components Diagram
Configuration

- Java using JDK SE 6 on the Eclipse IDE
- MySQL 5.5 Community Server
- Adwords API Java Library (adwords-api-8.2.0.jar)
- MySQL JDBC Connector (mysql-connector-java-5.1.10-bin.jar)
- Simple Adwords account (client)
- Google Analytics account (monitor conversions)
- My Client Center (MCC account) for Adwords API
Task Scheduler

- The system needs "memory"
- Tasks
  - When to collect statistics
  - When to optimize budget
  - When to test for new keywords
- Tasks are executed in date and time priority
- Tasks are stored in file
- System "sleeps" until the next task
Keyword Statistics

Uses database to get statistics

- Number of Impressions
- Number of Clicks
- Number of Conversions
- Click-through rate $= \text{Clicks} / \text{Impressions}$
- Conversion rate $= \text{Conversions} / \text{Clicks}$

Result: Find keyword options $(k,b)$ and their statistics
### Database Tables

**Table: Database tables and attributes - I**

<table>
<thead>
<tr>
<th><strong>AdGroup</strong></th>
<th><strong>Relevance</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>adgroupId-INTEGER</td>
<td>adgroupId-INTEGER</td>
</tr>
<tr>
<td>landingPageUrl-VARCHAR(2000)</td>
<td>keywordText-VARCHAR(60)</td>
</tr>
<tr>
<td>profitPerSale-DOUBLE</td>
<td>relevance-DOUBLE</td>
</tr>
<tr>
<td>Tested-BOOLEAN</td>
<td></td>
</tr>
</tbody>
</table>

**Table: Database tables and attributes - II**

<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Statistics</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>adgroupId-INTEGER</td>
<td>statisticsId-INTEGER</td>
</tr>
<tr>
<td>keywordId-INTEGER</td>
<td>adgroupId-INTEGER</td>
</tr>
<tr>
<td>text-VARCHAR(60)</td>
<td>keywordId-INTEGER</td>
</tr>
<tr>
<td>relevance-DOUBLE</td>
<td>maxCpc-DOUBLE</td>
</tr>
<tr>
<td>matchType-VARCHAR(7)</td>
<td>startDate-DATE</td>
</tr>
<tr>
<td></td>
<td>daysDuration-MEDIUMINT</td>
</tr>
<tr>
<td></td>
<td>impressions-INTEGER</td>
</tr>
<tr>
<td></td>
<td>conversions-INTEGER</td>
</tr>
<tr>
<td></td>
<td>averageCpc-DOUBLE</td>
</tr>
<tr>
<td></td>
<td>averageCpm-DOUBLE</td>
</tr>
<tr>
<td></td>
<td>averagePosition-DOUBLE</td>
</tr>
</tbody>
</table>

Goal: maximize profit or traffic.

$$\sum_{k \in K} w_k(k, b_k) \leq B$$

where $w_k$ is the cost of bidding price $b$ on keyword $k$ (weight) that produce:

$$\max \sum_{k \in K} v_k(k, b)$$

where $v_k$ is the expected profit of bidding price $b$ on keyword $k$ (value).
Knapsack and Genetic algorithm

- Multiple-choice Knapsack problem (NP-hard)
- Genetic algorithm solves it
- Finds an approximately optimum solution
- A possible solution is modeled as a chromosome

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>$k_2$</td>
<td>$k_3$</td>
<td>...</td>
<td>$k_N$</td>
</tr>
<tr>
<td>$0.60$</td>
<td>$0.00$</td>
<td>$0.45$</td>
<td>...</td>
<td>$0.50$</td>
</tr>
</tbody>
</table>

Fitness Function:

$$\sum_{k \in K} v_k(k, b)$$
Genetic Algorithm basics

- **Population** is a set of chromosomes
- The population evolves in each **generation**
- **Selection** is the process of finding the fittest chromosomes to become the parents of the next generation.

![Weighted Roulette Wheel Selection]

**Figure:** Weighted Roulette Wheel Selection

- The process of combining two chromosomes is called **crossover**.
- **Mutation** is used to escape local optimums
- With **elitism** the fittest chromosomes pass to the next generation directly.
Configuration

- JGAP Library (jgap.jar)
  - The Java Genetic Algorithms Package library is an open source framework for implementing and configuring genetic algorithms and genetic programming. It is published under the GNU Lesser Public License for non-commercial use.

- Michael Thomas Flanagan’s Java Scientific Library (flanagan.jar)
  - This library implements many mathematical areas, one of them is regression. It also provides methods for plotting the data. Permission to use for non-commercial purposes is granted.
We want to predict Impressions

Impressions affect Clicks and Conversions

Multivariate regression

\[ Y = \theta_0 + \theta_1 \times X_1 + \theta_2 \times X_2 + \theta_3 \times X_3. \]
Experiments

AdWords API Sandbox
- Testing and development environment that replicates the functionality of the AdWords API web services.
- Develop and test applications without spending API units (calls made in the sandbox are free)

Budget Optimization Testing - 4 optimization scenarios
- Budget Optimization for Profit with No Prediction (NoPredProfit)
- Budget Optimization for Traffic with No Prediction (NoPredTraffic)
- Budget Optimization for Profit With Prediction (PredProfit)
- Budget Optimization for Traffic With Prediction (PredTraffic)

During consecutive weeks of March 2011, we run an automatic campaign...
<table>
<thead>
<tr>
<th>Budget</th>
<th>Clicks</th>
<th>Cost</th>
<th>Profit</th>
<th>Counter</th>
<th>AverageBid</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoPredProfit:</td>
<td>60.451</td>
<td>49.944</td>
<td>219.51</td>
<td>24</td>
<td>1.495</td>
</tr>
<tr>
<td>NoPredTraffic:</td>
<td>61.5</td>
<td>49.938</td>
<td>206.22</td>
<td>23</td>
<td>1.436</td>
</tr>
<tr>
<td>PredProfit:</td>
<td>82.363</td>
<td>49.901</td>
<td>317.18</td>
<td>16</td>
<td>1.379</td>
</tr>
<tr>
<td>PredTraffic:</td>
<td>86.51</td>
<td>49.886</td>
<td>274.82</td>
<td>18</td>
<td>1.429</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoPredProfit:</td>
<td>197.83</td>
<td>199.88</td>
<td>621.32</td>
<td>56</td>
<td>1.552</td>
</tr>
<tr>
<td>NoPredTraffic:</td>
<td>200.84</td>
<td>199.91</td>
<td>582.22</td>
<td>54</td>
<td>1.506</td>
</tr>
<tr>
<td>PredProfit:</td>
<td>236.94</td>
<td>199.87</td>
<td>787.63</td>
<td>31</td>
<td>1.423</td>
</tr>
<tr>
<td>PredTraffic:</td>
<td>248.6</td>
<td>199.86</td>
<td>638.13</td>
<td>32</td>
<td>1.438</td>
</tr>
<tr>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoPredProfit:</td>
<td>333.55</td>
<td>389.6</td>
<td>798.9</td>
<td>97</td>
<td>1.618</td>
</tr>
<tr>
<td>NoPredTraffic:</td>
<td>343.99</td>
<td>405.17</td>
<td>795.29</td>
<td>107</td>
<td>1.631</td>
</tr>
<tr>
<td>PredProfit:</td>
<td>607.74</td>
<td>599.85</td>
<td>1645.6</td>
<td>70</td>
<td>1.569</td>
</tr>
<tr>
<td>PredTraffic:</td>
<td>622.69</td>
<td>599.82</td>
<td>1569.2</td>
<td>68</td>
<td>1.523</td>
</tr>
</tbody>
</table>
Contributions

- Fully implemented and functional system with a comprehensive experimental evaluation in real world data
- Showcase that the proposed architecture and methodology is plausible and feasible and provide proof of concept
- Good basis for a larger system that has more features and can provide more functionality useful for the field of web-search advertising
- Find and implement tangible solutions to the issues arising at the very core of a Campaign creation and Budget optimization software
Selected Bibliography

Keyword generation for search engine advertising using semantic similarity between terms. 
In Proceedings of the ninth international conference on Electronic commerce.

Keyword Generation for Search Engine Advertising. 
Sixth IEEE International Conference on Data Mining - Workshops (ICDMW06).

Ravi, S. et al., 2010.
Automatic generation of bid phrases for online advertising. 
Proceedings of the third ACM international conference on Web search and data mining.

Algorithm for the Stochastic Multiple-Choice Knapsack Problem and Application to Keywords Bidding. 

A genetic algorithm for the multidimensional knapsack Problem. 
J Heuristics 4.

Farhad Djannaty, Saber Doostdar, 2008.
A Hybrid Genetic Algorithm for the Multidimensional Knapsack Problem 
Thank you!

email: thomaidous@aueb.gr

webpage: http://www.db-net.aueb.gr/matina