Automatic characteristics generation for search keywords in paid search advertising

Pengxuan Liu Alan Montgomery

September 2007

Abstract

Paid search advertising has been a major form of online advertising in recent years. In this form of advertising, an advertiser submits a list of keywords to major search engines. When one of the keywords matches the query keyword that a search engines user submits, the ad of this advertiser will have a chance to be shown on the search result page. If the user is interested and clicks on the ad, the advertiser will be billed of each clickthrough with a predetermined cost-per-click fee by the search engine, regardless whether the user purchases anything after entering the advertiser's website. The advertiser will try to make a profit by hoping a higher probability that a clickthrough can end with a sale. So in an ad campaign the fundamental question the advertiser wants to ask is: what are the good keywords that can attract more clickthrough traffic form search engines, and more importantly have higher sale conversion rate given these clickthrough traffic? Several marketing literatures have addressed this issue but their keywords selection and evaluation methods need human interactions. Our goal is to try to develop a statistical learning method that can automate the keyword evaluation processes. This paper is our pilot study and we want to know whether such statistical learning method can really the same or even better job than human As a comparison, we compared our result with another study with the same data but using mainly manual evaluation process. The result shows our method has better prediction accuracy.

1. Background Introduction

Paid search advertising has emerged as a predominant form of Internet advertising in recent years, along with the advance of the search engine industry. Largely propelled by Google and followed by all other major search engine, paid search advertising industry has generated $8.2 billion ad revenues in 2005 (Satagopan et al. 2005). Before paid search advertising, form of online advertising is basically copied from that of offline media, especially printing industries. It is called Internet banner display or web banner display. Advertisers put the banner they create on high traffic websites and are charged with each banner display or impression. Web banner display still remains as a main advertising method because it does have some advantages that the advertisers like. It can let advertisers strategically exposes their website or company name right where they want it. It can help establish company name or brand in a long run. And more importantly, it gives advertisers the ability to measure results. By tracking banner advertisement performance, advertisers can review and determine which ad placements direct the most customers to their website or promote their brand image better.

The disadvantage of web banner display is also very obvious. It's untargeted and thus inevitably will increases the cost of the campaign for achieving the same level of overall impression. It is also annoying to online viewers who are not or at least at that time potential customers. The clickthrough rate of online banner display has been declining every year. Paid search advertising appears as an effective alternative form of advertising along with the growth of search engine industry. As a matter of fact, it has been the

major source of revenue for Google, the largest search engine. Contrary to the traditional online banner display, paid search advertising targets text-based ads to user search queries, making the ad campaign more efficient. In this form of advertising, an advertiser submits a list of keywords to search engines. When one of the keywords matches the query keyword that a search engines user submits, the ad of this advertiser will have a chance to be shown on the search result page. If the user is interested and clicks on the ad, the advertiser will be billed of each clickthrough with a predetermined cost-per-click fee by the search engine, regardless whether the user purchases anything after entering the advertiser's website. Just ad display won't cost advertiser anything and the advertiser pays only for actual traffic. This is why sponsored listings are referred to as "pay-per-click" (PPC) or "cost-per-click" (CPC) advertising. The advertiser will try to make a profit by hoping a higher probability that a clickthrough can end with a sale, or a "conversion". However there are no guarantees that visitors are buying anything.

In Google's paid search operation AdWords, for example, an advertiser may have a list of keywords such as "jacket", "leather jackets", and "men's leather jackets". Notice that the keywords we mentioned are actually keyword phrases and we only refer them as keyword through out our paper for convenience. The advertiser has to bid a CPC amount for each keyword indicating how much he wants to spend on each clickthrough generated by this keyword (can range from several cents up to a few hundred dollars). When a consumer searches Google using keyword "leather jacket", "leather jackets", or probably something similar such as slightly misspelled keywords, Google thinks this is a match for the keyword "leather jackets". For a generic and popular keyword like "leather

jackets", there are maybe thousands of advertisers bid on it. Google will choose several text ads and show them on the right of the search result page, depending on how much each advertiser wants to spend totally on his ad campaign, the CPC amount he bids for "leather jackets", and some other factors. Generally the more an advertiser wants to spend, the higher chance and the better location his ad will be shown. Google will also take into account the advertiser's clickthrough history, keywords "matching quality", and other factors, favoring those advertisers who generate more expected revenue for Google. The actual algorithm for matching and calculating the probability and location of banner display remain undisclosed by search engine mainly to prevent tempering by the third parties. Yahoo! use pure price auctions to determine the positions of the paid search ads before February 2007. This means the highest bidder is always given the best position; all others follow in order of bids. Now Yahoo! has abandoned their previous algorithm. They followed Google and started to take clickthrough rate into account too.

Advertisers need to carefully consider their keyword selection for their ad campaigns. A better chosen keyword can potentially improve the "conversion rate" by attracting more targeted consumers. This is because the advantage of paid search advertising over traditional online banner display is that with keywords, advertisers can distinguish their target consumer group from the huge pool of mass consumers and effectively attract target consumers to their website. Poorly chosen keywords have no such ability of selection. Even worse, the advertisers are billed for each clickthrough, regardless of whether this click ends with a sale conversion or not. The traffic that poorly chosen keywords bring in has the same cost as high quality traffic for the advertisers but low

conversion rate, making them negative profit or "money burning" keywords. For example, it may not be a good idea for Victoria Secret, a retailer specializing in women's intimate apparel, to bid on keyword “men’s leather jacket”. Beside the low clickthrough rate, if a person enter Victoria Secret website through this keyword, mostly likely that he would find himself entering the wrong place and simply leaves.

2. Research Questions

In paid search advertising, an advertiser faces four different decisions (or levers): (1) which keywords to select, (2) how much to bid for each keyword, (3) how to design the text ad and (4) how to design the landing page (Rutz and Bucklin, 2007). The most fundamental question an ad campaign manager often asks, which is also the focus of this paper, is what is a good way to assess the quality of a keyword that makes the choice of the keyword simple? Theoretically, for a keyword to make profit, the inequation has to hold:

Profit of each conversion * conversion rate >= CPC

The profit of each conversion is the same for all keywords. Thus conditioned on a click with a given a CPC, conversion rate is the measure of the performance of each keyword. So our goal is to model the conversion rate against the characteristics of each keyword.

Marketing practitioners choose keywords usually in an ad hoc fashion. They often choosing keywords based their experiences and common knowledge, plus some

proprietary data of previous ad campaigns. There are some general guidelines of keyword selections, but none of them is hard rules. All paid search programs also provide basic keywords selection tools, which can give you suggestions of what other keywords are related to the keyword you choose. But they still can't provide prediction of how well a keyword will perform, i.e. what the convention rate will be. It is usually easy to assess the effect of a web banner display. But evaluating the profit performance of each keyword is not very simple. In web banner display form of advertising, there only a handful templates of banners, and advertisers can usually collect tons of display and sale conversion data for each banners. But in a big ad campaign, an advertiser usually has hundreds or thousands of keywords. In the keyword pool, some very good keywords can generate lots of clickthrough and the advertiser can probably easily assess the conversion rate. But most of the keywords can generate only a few or clickthrough or none during entire ad campaign. Without a proper model, we can't make any statistically meaningful assessment based on just a few clickthrough data of a keyword.

For those keywords that don't generate any clickthrough, since the advertiser is charged only for each clickthrough, not for each banner display, they don't cause any monetary cost to the advertiser (although they don't have any contribution to the profit either). But zero is not negative. The advertiser doesn't know that whether these keywords can generate any clickthrough and sales in the future and thus has no apparent reason to reject these keywords being in the keyword list. For keywords that only generate a few clickthrough, the advertiser can't easily reject them either. It's true that in most cases these clickthrough doesn't generate any sale conversion. But the keywords that have low

clickthrough rate are usually less competitive and thus have lower CPCs. A low conversion rate is still possible for these keywords to make profit. In real practice, most of these low-clickthrough-rate keywords are still profit-making keywords. So theoretically speaking before an advertiser knows for sure that a keyword can cause negative profit by generating only clickthrough and very few sale conversions, the advertiser should better choose this keyword because this is a better bet. In fact, there is a possibility that the power set of entire vocabulary of a language can be our keyword consideration set. In this case, most of the search keywords generate no clickthrough. And the majority of the rest that has clickthrough will have very low conversion rate. But if an advertiser can make high enough profit for a sale conversion, a very low conversation rate can still make profit.

In practice search engines usually don't allow advertisers to choose keywords that don't have a single clickthrough for a certain period of time. For example, Google give a Quality Score for each keyword based on clickthrough rate (CTR) and the relevance of ad text, keyword, and landing page, etc. Google will mark a keyword "inactive for search" and stop showing ads for this keyword on search results if it doesn't have a high enough Quality Score. So we can avoid the situation where we need to consider the entire vocabulary. And we can assume our keyword choice set is a compact set.

3. Review of current research

Several research studies have been done to build model predict conversion. In Montgomery et al. (2004), the authors use navigational path information of visitors to predict purchase conversion. They show how path information can be categorized and modeled using a dynamic multinomial probit model of web browsing. But their research is not in paid search advertising setting and no keyword is used. In research where keyword is the main predictor variables, several different methods are used to create the characteristics. For example, in Rutz and Bucklin (2007), they enhance their data by introducing semantic keyword characteristics. The keywords used have certain common characteristics that are specific to the lodging industry. They "decompose" each of the 301 keywords along five set of location characteristics. Then they can pool the keywords with the same or similar characteristics together with the Bayesian hieratical model. Charles et al (2007) use the same dataset we used in this paper and attach 17 characteristics to all the keywords, such as whether the keyword includes brand name, or whether the keyword is sports related, etc. Then they use logistic regression model to model the conversion rates against the characteristics of these keywords.

In all these researches mentioned, categorization or characteristics assignment of these keywords are still being done manually. Charles et al. spent a lot of time on generating these categories that they think might have influences on conversion rate. This is apparently a not very efficient approach. Besides, manual categorization has to rely on the logical meaning and connections among these keywords, such as location etc. However being logically meaningful for these characteristics is by no means necessary. Sometime they can even be minor effects in the model. We shouldn't limit ourselves to

meaningful characteristics. This is the where machine learning (ML) and natural language processing (NLP) research will shine.

Our research goal is to have a model that could be able to automatically generate these categories and characteristics, and do an as-least-as-good if not better job than what we can achieve manually. In this pilot study paper, we used the same data that Charles et al. used in their research so that we can compare the result of our model with their result. And as a pilot study, the first thing we would like to know is if it's possible for statistical method to beat what human can do. If so, we can refine our method in our follow-up research studies.

4. Data

4.1 Sale conversion data

The data we use in our study come from two sources. First, we have a collection of clickstream data of several online stores. This is the data Charles et al. used in their research. These data are obtained from a major consulting firm for search engine marketing. This firm help manage clients' search marketing campaigns.

The clickstream data have totally more than 1.6 million records. Each record represents a clickthrough session. The variables includes the actual keyword that a consumer submitted to search engine, the keyword that search engine chose to match against the

actual keyword consumer submitted, matching rules (e.g. broad, exact), search engine used (Google, Yahoo! and Microsoft), indicator of whether the click converted to a sale, sales quantity amount when there is a sale, order id, and data and time of the clickthrough. Notice that since the data are from advertisers and not from the search engines, each record is an actual clickthrough to advertisers (and thus they have to pay for it). Those keywords that once generated ad display but no clickthrough are not in our dataset. These data must be obtained from search engine companies. Since we have assumed that any keyword that doesn't generate a single clickthrough during an ad campaign should be excluded from the keyword selection, those keywords are not relevant to our research problem.

The total 1.6 million clickthrough are generated by totally 15668 distinct keywords. But in these 15668 keywords, majority of them only generate very a few clickthrough and contribute very little to the total number of clickthrough. This is a very common scene in paid search engine advertising campaign. In our data, only 1933 (12%) keywords generate 31 or more clickthrough. However in total these 1933 keywords generate about 1.5 million clickthrough (94%). In sampling theory, 30 is the common threshold size for a sample to have any meaningful statistical results so we use only these 1933 keywords to estimate our model and exclude the rest.

4.2 Augmented data from Google

In general, the keywords are usually very short. Looking at the semantic links between only the keywords themselves usually won't give enough information. In previous researches, the authors all enhance their keyword data in one way or another. What we did was for each of 15668 distinct keywords, we submit the search query with it to Google search. As many already know, in Google search result page, each search result item has a title, and an excerpt of cached text that has about 30 or fewer words. Usually both title and the cached text contain the search query keywords. We collect the text of both title and cached text of the first 100 search result items and put all together. We call this collection of text a "document". Each document is corresponding to a keyword.

In NLP research, there are many researches that model the distribution of the distance between any words among the entire vocabulary. For example, in Ritter, A., et al. 2006, the authors used commonly Kullback Leibler divergence to measure the distance between a pair of words. Then based on this measure, they present a method of distributional clustering of text. In our research we only assume that the distance between two words follow a certain distribution. There is always a tendency that several words are more likely to appear close to each other than the other words in a random paragraph. Those words that are more likely to appear close to a keyword will be more likely to appear in the title or cached text in each search result item for this keyword. And thus they are more likely to appear in the document we constructed for each keyword. For the purpose of our paper, we don't need to know exactly or model this distribution. Nor do we need a measure for the distance between a pair of words.

Then naturally, we use the Bag-of-Words (BOW) model which is very popular for analyzing documents (e.g., Ueda and Saito, 2003). BOW model ignores the order of word occurrence in a document. Each document is represented by a distribution over a fixed vocabulary. Formally suppose the n[th] document d^n can be represented by a word-frequency vector $x^n = (x_1^n, ..., x_V^n)$. Where x_i^n denotes the frequency of word w_i occurrence in d^n among the vocabulary $\upsilon = (w_1, ..., w_V)$. V is the total number of words in the vocabulary. A document can be a webpage, a very long paragraph, a multi-million words novel, or simply just a search phrase composed of only one or two words. In our case, a document is the text collection of first 100 search result items. After we collected totally 15668 documents, we pool all the text of these documents together (in a "bag"), find out all the distinct words that ever appear, and generate the vocabulary υ for our problem. The total number of word in our vocabulary is 333,033, and thus the dimension of each word-frequency vector is also 333,033. Then we built a word-frequency vector from each document like the way the BOW model specified and finally we got 15668 huge vectors of which each has a dimension 333,033. Each component of the vectors is a frequency count.

Before constructing our vocabulary and the big document matrix, we actually did another processing called stemming on all 15668 documents. Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form (en.wikipedia.org/wiki/stemming). For example, a stemmer for English should identify "cats" as based on the root "cat", and "specializes", "specialized", "specialise", "specialises", "specialised" as based on "specialize". After stemming, all the derived

words are replaced by their root forms and we merged them by adding the frequency count of derived words to the frequency count of their roots. There isn't a standard or authoritative stemmer. Instead, many versions of stemmers are available for English. A widely used stemming algorithm was created by Martin Porter called Porter stemmer. But what we used was a stemmer built in Miner module version 4.3 in SAS system. We choose this stemmer just for the conveniences because we process all of our data in SAS. So after stemming, in the vocabulary and all documents we constructed, there is no derived word. All words are roots.

Now we've had all the data we need before carrying out a logistic regression. First we have the clickstream data that contain search keywords used and sale conversion indicator. Secondly, we have 15668 document vectors and each has a 333,033 frequency count of each word in a same vocabulary. When carrying out a regression, 333,033 independent variables are too much and not necessary. Only top 100 most frequent words overall in the vocabulary are selected to represent each keyword. As a result, each document vector was shortened to a dimension of 100. We pick out 1933 document vectors whose keywords have generated more than 30 clickthrough and exclude the rest. We also exclude one keyword because it has generated unusually large amount of clickthrough than any other keywords. We finally settled with 1932 keyword document vectors and 892,230 clickthrough (55%) data corresponding to the 1932 keywords. Then we match merge the clickthrough data with 1932 keyword document vectors using search keywords as index, and generate an 892,230 by 101 matrix. The first variable is sale conversion indicator. The other 100 are the frequency components. We then did a

partition of the data and use a sample random of 1000 keywords (434,487 observations) as training data for model estimation. The rest of 931 keywords (457,743 observations) are used as hold out data to test out model.

5. Model and Estimation

The model we used is classic logistic regression model. This is also what Charles et al. used in there research. Formally, the model is:

$$Log(\frac{P_i}{1-P_i}) = \beta X^i$$

Where

P_i is the probability of sale conversion, i.e. the conversion rate.

X^i is the combined vector of 1 and i^{th} document vector of 1 by 100

i = 1, 2, … , 600,000

The result of the estimation is shown in Table 1. We can see that our model does have very good prediction power. More than half of the parameters are statistically significant.

Then we use hold out sample data to test how well the model performed. We use $\hat{\beta}$ to calculate the conversion rate for each clickthrough. In hold sample, the average conversion rate is 4.26%. So we use 4.26% as cutoff for predicting conversion. Any keyword with higher than 4.26% predicted probability is predicted as converting. Table 2 shows the actual vs. predicted values on our holdout sample. Table 3 shows the result

of Charles et al. Clearly, our overall accuracy is a little better than theirs. Besides, both our true positive and true negative prediction achieves about 75% accuracy while their result of true positive is quit poor. All these comparisons show that our statistical model does work and can do at least as good as human.

5. Conclusion and future research

Our research have been trying to develop an automated method to build characteristics from search keywords. To the best of our knowledge, all the previous researches create the feature of the search keywords using manual methods. Our research is the first one that has used NLP models and done a fully unattended creation of characteristics. Our very preliminary result shows that the NLP model can do at least as good as human. This is already inspiring result to us because NLP model can free people from tedious mind work in the whole keyword performance evaluation process.

Of course, our method is still very rudimentary. First, the document of each keyword is largely depended on the search result of Google, which is constantly changing. This makes our result less robust. Secondly, the search result of Google is not a random sample from a large amount of text corpus, which is the sample we should use. Google search results are the text that Google thinks will match the query keyword best. This makes our result likely to be a little different from the result derived from true random sample of text corpus. Thirdly, Google's matching algorithm is undisclosed and

proprietary. This means that there exists a “black box” in our unattended process that we don’t know what happens. We certainly need to develop our own open sourced matching models. Fourthly, there are still a lot can be improved in our data processing and model selections. The estimation model is certainly not limited to only logistic model.

References

Chen, Y., & He, C., Paid Placement: Advertising and Search on the Internet, working paper

Griffiths, T. L., & Ghahramani, Z. (2005). Infinite latent feature models and the Indian buffet process (Technical Report 2005-01). Gatsby Computational Neuroscience Unit, University College London.

Satagopan, S., D. Card, N. Soevak, and G. Stein. 2005 "US Paid Search Forecast, 2005 to 2010." Online Search. 3 1-14. Published by Jupitermedia Corporation.

Ueda, N., & Saito, K. (2003). Parametric mixture models for multi-labeled text. In Advances in Neural Information Processing Systems 15. Cambridge: MIT Press.

Montgomery, A., Li, S., Srinivasan, K. & Liechty, J.C. (2004). Modeling online browsing and path analysis using Clickstream data, Marketing Science, Vol. 23 (Fall 2004), 579-595

Ritter, A., Hearne, J., & Nelson, P., Distributional Word Clustering in Parallel, Proceedings of the ISCA 19th International Conference on Parallel and Distributed Computing Systems 2006, 267-272

Rutz, O., & Bucklin, R., A Model of Individual Keyword Performance in Paid Search Advertising, working paper

Table 1. Parameter estimation

Parameter	Estimate	Standard Error	Chi-Square	P Value
Intercept	-3.6396	0.0495	5416.646	<.0001
COL1	0.0166	0.00347	22.9227	<.0001
COL2	-0.031	0.00628	24.3501	<.0001
COL3	0.00617	0.00423	2.1275	0.1447
COL4	-0.0298	0.00551	29.2988	<.0001
COL5	-0.0228	0.00424	28.8816	<.0001
COL6	-0.00724	0.00221	10.7137	0.0011
COL7	0.0981	0.0555	3.1233	0.0772
COL8	-0.0804	0.0893	0.8117	0.3676
COL9	0.000075	0.000664	0.0128	0.91
COL10	0.00393	0.000406	93.9565	<.0001
COL11	0.0119	0.00319	14.0255	0.0002
COL12	-0.00087	0.000523	2.7946	0.0946
COL13	0.00103	0.000704	2.1434	0.1432
COL14	-0.00338	0.00108	9.7752	0.0018
COL15	0.00225	0.00131	2.9518	0.0858
COL16	-0.00961	0.00309	9.6469	0.0019
COL17	-0.00804	0.00385	4.3499	0.037
COL18	-0.0193	0.00305	40.213	<.0001
COL19	0.00264	0.000779	11.4914	0.0007
COL20	-0.00051	0.00107	0.2294	0.632
COL21	0.00424	0.00212	3.9904	0.0458
COL22	-0.00121	0.00237	0.262	0.6087
COL23	0.00168	0.000533	9.9292	0.0016
COL24	-0.00054	0.00045	1.4467	0.2291
COL25	0.000863	0.000732	1.3868	0.2389
COL26	0.0108	0.000304	1257.444	<.0001
COL27	-0.00305	0.00522	0.3415	0.5589
COL28	0.00293	0.000848	11.9259	0.0006
COL29	-0.00036	0.000665	0.2978	0.5853
COL30	-0.00052	0.00064	0.6596	0.4167
COL31	0.00418	0.0035	1.4253	0.2325
COL32	0.00402	0.000439	84.1122	<.0001
COL33	-0.0077	0.00404	3.6234	0.057
COL34	-0.0303	0.00442	46.9827	<.0001
COL35	-0.0491	0.0384	1.6377	0.2006
COL36	-0.0018	0.00162	1.2482	0.2639
COL37	-0.00045	0.000498	0.8101	0.3681
COL38	-0.0378	0.0431	0.7679	0.3809
COL39	-0.0515	0.0038	183.3239	<.0001
COL40	0.00514	0.00141	13.2217	0.0003
COL41	-0.0353	0.0061	33.4108	<.0001

COL42	-0.0845	0.00842	100.4839	<.0001
COL43	0.00593	0.00512	1.3457	0.246
COL44	-0.00063	0.000441	2.0134	0.1559
COL45	-0.00057	0.000807	0.5008	0.4791
COL46	-0.00076	0.00102	0.5531	0.4571
COL47	0.00165	0.000369	20.0987	<.0001
COL48	0.00391	0.00132	8.753	0.0031
COL49	-0.00295	0.00146	4.0688	0.0437
COL50	-0.00056	0.00104	0.2898	0.5903
COL51	0.000056	0.000333	0.0285	0.866
COL52	0.0034	0.000557	37.2861	<.0001
COL53	-0.00003	0.000599	0.0023	0.9621
COL54	-0.00017	0.00042	0.1592	0.6899
COL55	0.00292	0.00102	8.2297	0.0041
COL56	-0.00146	0.00119	1.516	0.2182
COL57	0.00058	0.00259	0.0502	0.8227
COL58	0.000257	0.00132	0.0379	0.8455
COL59	-0.00113	0.00213	0.2835	0.5944
COL60	0.00128	0.000294	19.0047	<.0001
COL61	-0.00729	0.0454	0.0257	0.8725
COL62	0.0025	0.000461	29.3445	<.0001
COL63	-0.00744	0.00209	12.7406	0.0004
COL64	0.00309	0.000684	20.4034	<.0001
COL65	0.0377	0.00483	60.9222	<.0001
COL66	0.000381	0.00475	0.0064	0.9361
COL67	-0.0447	0.00483	85.499	<.0001
COL68	0.0399	0.00721	30.556	<.0001
COL69	-0.0387	0.00613	39.8325	<.0001
COL70	0.00433	0.00231	3.5019	0.0613
COL71	-0.00123	0.000593	4.2857	0.0384
COL72	-0.00162	0.000582	7.7154	0.0055
COL73	0.00203	0.000658	9.5556	0.002
COL74	-0.0125	0.00234	28.8115	<.0001
COL75	-0.00231	0.00289	0.6383	0.4243
COL76	0.00213	0.0013	2.6722	0.1021
COL77	-0.0004	0.000709	0.3257	0.5682
COL78	0.00825	0.00213	14.973	0.0001
COL79	-0.00756	0.00385	3.8527	0.0497
COL80	0.000629	0.000712	0.7804	0.377
COL81	-0.00262	0.00107	6.0074	0.0142
COL82	-0.00189	0.00128	2.1976	0.1382
COL83	-0.00371	0.00205	3.2896	0.0697
COL84	-0.00135	0.00117	1.3361	0.2477
COL85	-0.00952	0.00414	5.2915	0.0214
COL86	-0.00343	0.00116	8.7495	0.0031
COL87	-0.00034	0.000539	0.4053	0.5244
COL88	0.00606	0.000649	87.2547	<.0001
COL89	-0.00157	0.000466	11.3583	0.0008

COL90	0.00485	0.000376	167.021	<.0001
COL91	0.00412	0.00119	12.0094	0.0005
COL92	0.00382	0.000445	73.8742	<.0001
COL93	0.0383	0.00482	63.0947	<.0001
COL94	-0.0133	0.00332	16.0098	<.0001
COL95	-0.0222	0.0043	26.6822	<.0001
COL96	0.00151	0.000725	4.3105	0.0379
COL97	-0.00009	0.000399	0.0535	0.8171
COL98	0.000143	0.00152	0.0088	0.9254
COL99	0.000982	0.000557	3.1016	0.0782
COL100	0.000852	0.00108	0.6249	0.4292

Table 2. Actual vs. predicted values on holdout sample

		Predicted		Row percent
		0	1	
TRUE	0	325480	112761	25.73%
	1	4478	15024	77.04%
Column percent		1.36%	11.76%	
Overall Accuracy				**74.39%**

Table 3. Charles et al. Actual vs. predicted values on holdout sample

		Predicted		Row percent
		0	1	
TRUE	0	27430	539	1.93%
	1	10693	2298	17.69%
Column percent		28.05%	81.00%	
Overall Accuracy				**72.58%**