Scalable Audience Targeted Models for Brand Advertising on Social Networks

Kunpeng Zhang
Department of Information and Decision Sciences
University of Illinois at Chicago
Chicago, IL USA
kzhang6@uic.edu

Aris Ouksel
Department of Information and Decision Sciences
University of Illinois at Chicago
Chicago, IL USA
aris@uic.edu

Shaokun Fan
Department of Computer Information and Decision Management
West Texas A&M University
Canyon, Texas USA
fsk1234@gmail.com

Hengchang Liu
School of Computer Science and Technology
University of Science and Technology of China
Suzhou, Jiangsu China
hcliu@ustc.edu.cn

ABSTRACT

People are using social media to generate, share, and communicate information with each other. Finding actionable insights from such big data has attracted a lot of research attentions on, for example, finding targeted user groups based on their historical on-line activities. However, existing machine learning algorithms fail to keep up with the increasing large data volume. In this paper, we develop a scalable regression-based algorithm called distributed iterative shrinkage-thresholding algorithm (DISTA) that can identify potential users. Our experiments conducted on Facebook data containing billions of users and associated activities show that DISTA with feature selection not only enables online audience-targeted approach for precise marketing but also performs efficiently on parallel computers.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Data mining*; G.1.6 [**NUMERICAL ANALYSIS**]: Optimization—*Unconstrained optimization*

Keywords

Social brand; feature selection; DISTA; advertising

1. INTRODUCTION

Most social platforms, such as Facebook, Twitter, Youtube, and Amazon.com, have mechanisms allowing users to generate, share, and communicate with each other for their interested topics. For example, users can give rating scores and leave their reviews on products they purchased. People can also "like" or make comments on social brands (e.g. celebrities, institutes, organizations, companies, and products). Analyzing these user-generated contents to find actionable insights can help users make informed decisions, which has attracted a lot of attention in research. Research in social media data analysis falls into two categories. The first one is from the text-mining perspective: text sentiment analysis for decision making [4]; the second one is from the social network perspective: study of static and dynamic properties of networks [5].

Recently, the trend to social content-driven advertising is becoming increasingly evident in business management. Finding targeted audience for precise on-line advertising based on user historical behaviors is one of the most important marketing tasks. BIA/Kelsey's study estimates that the social advertising revenues in the U.S. will grow over 3 billion dollars by 2017 [1]. Machine-learning methods have been widely used, for example, for building a predictive model based on users' profile, historical activities, and social networking information. Many psychological and sociological models were also proposed to build user sociality from user access log data so that they can be used to guide marketing managers to find their targeted audience. In this work, we focus on user preference prediction on social brands.

However, there are some challenges given the big size of the training samples and the large number of training features. First, existing feature selection algorithms is infeasible and inefficient, which motivates us to find a scalable solution. Secondly, implementing distributed algorithms to efficiently and accurately learn predictive models is also not straightforward. To address the first challenge, we implement a MapReduce-based Apriori algorithm to find a given brand the group of correlative brands that share the most user activities. The identified brands will be used as the selected features in the model learning. To solve the second problem, we implement a distributed regression-based algorithm called iterative shrinkage-thresholding algorithm

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
Copyright 2014 ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645763.

(DISTA), a stochastic optimization algorithm that can handle a large amount of training instances. The experiments show that our DISTA can get up to 16% increase of accuracy by incorporating our feature selection strategy comparing to other baselines.

2. PROBLEM STATEMENT

Our problem is a typical classification in machine learning domain. The training features are social brands (b_1, b_2, ..., b_n) and the value of each feature is the number of historical activities a user had on the corresponding brands (e.g. the number of likes, the number of comments, or both). The target brand (b_t) is labeled in a binary form: 1 if a user is interested in this brand, 0 otherwise. Before mathematically formulating this problem, we define the terms of social brands and activity matrix used in this paper.

A social brand is an entity in the social network that allows other users to leave comments on its page. Examples are companies, organizations, individuals, or consumer products. The activity matrix is represented as the following.

$$A = \begin{array}{c c} & \begin{array}{ccccc} b_1 & b_2 & \dots & b_n & b_t \end{array} \\ \begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_m \end{array} & \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} & 1 \\ x_{21} & x_{22} & \dots & x_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} & 0 \end{pmatrix} \end{array}$$

where u_i is the i^{th} user; b_j is the j^{th} brand; The entry x_{ij} is the number of activities made by i^{th} user on brand j. $x_{ij} = like_{ij} + comment_{ij}$, where $like_{ij}$ is the number of likes user i gave to all posts initiated by brand j and $comment_{ij}$ is the number of comments made by user i on brand j.

To obtain the k^{th} user's preference on a specified target brand b_t, we calculate P_{kt}.

$$P_{kt} = A_k * \alpha = \alpha_1 x_{k1} + \alpha_2 x_{k2} + \cdots + \alpha_i x_{ki} + \cdots + \alpha_n x_{kn}$$

where A_k is the k^{th} row of activity matrix A. $P_{kt} \in [0, 1]$ is the output value for the target brand b_t, representing the preference on brand t of the k^{th} user; $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]^T$; All these x_{ki} are given for a testing user and all α_i obtained through the training process, which involves solving the following convex optimization problem.

$$min_\alpha f(\alpha) + \lambda\|\alpha\|_1 = min_\alpha \|A\alpha - b_t\|_2^2 + \lambda\|\alpha\|_1 \qquad (\star)$$

where α is a vector of n dimensions; b_t is a vector of the targeted brand t of n dimensions; λ is a constant and $\|\alpha\|_1$ is the l_1-norm of the parameter vector. $\|\alpha\|_1 = \sum |\alpha_1| + |\alpha_2| + \ldots + |\alpha_n|$.

In this work, we used Facebook Graph API to download social brand data from Facebook. The data covers many different categories, including sports, movies, politics, fast-food, and many others. The first issue we need to handle is feature selection, because not all brands in the feature set are related to the targeted brand. We start with selecting top related brands to reduce the size of the activity matrix A. The method we use here is association rule mining to find patterns like "$b_i \Rightarrow b_t$" with high confidence scores. In the next section, we will discuss a MapReduce-based technique to find top k other brands $(b_1, b2, \ldots, b_k)$ in terms of confidence score with the pattern of "$b_i \Rightarrow b_t$". Then the size of the new activity matrix A' is significantly reduced from $m * n$ to $u * k$ ($k << n$ and u is the number of users having activities on at least one of top k brands, $u << m$). The next step is the binary classification problem to identify potential users.

3. METHODOLOGY

In this section, we describe our distributed iterative shrinkage-thresholding algorithm. In addition, to address the large data volume challenge in feature selection, we use MapReduce-based Apriori to select top associated brands.

3.1 Feature Selection

As most of the features/brands are not closely related to the targeted brand, removing irrelevant features during the learning can not only reduce the size of training data, but also help mitigate bias. In this work, we use a distributed Apriori algorithm to select top brands (features) based on the confidence score of associated rules like "$b_i \Rightarrow b_t$". Before diving into the details, we first describe the data we collect from the Facebook.

Data Preparation For the public social brands, users can like or make comments on campaigns posted by brand administrators. In this work, we assume that a user is interested in a brand if he/she makes positive comments on it or likes campaigns on that brand. OpinionFinder [6] is used to identify sentiments. We consider likes and comments as user activities, which can be represented as a 3-tuple: [$user_{id}$, $brand_{id}$, $\#\_of\_activities$]. We then combine all activities across all brands for each user. After this process, each user is described with the format of <$user_{id}$ DEL $b_1|w_1, b_2|w_2, \cdots, b_i|w_i, \cdots$>, where b_i is the i^{th} brand and w_i is the corresponding number of activities, DEL could be any delimiter.

Confidence score: The goal here is to find the frequent pattern "$b_i \Rightarrow b_t$" based on a large amount of user historical activities across brands. Two-itemset (I_x, I_y) Apriori ("$I_x \Rightarrow I_y$") indicates their correlation. Here, I_x could be any brand $b_i \in \{n$ features: $b_1, b_2, \ldots, b_n\}$ except the target brand, I_y is the target brand b_t. We choose top k brands based on the confidence score of the pattern "$b_i \Rightarrow b_t$". The confidence is calculated using the following equation.

$$Conf(b_i \Rightarrow b_t) = \frac{Support(b_i, b_t)}{Support(b_i)}$$

Where $Support(X)$ is the occurrence frequency of X. In our case, it is the number of users who have activities on both brands b_i and b_t for $Support(b_i, b_t)$, on brand b_i only for $Support(b_i)$. The key sketchlon of the MapReduce-based algorithm of calculating confidence score (CSC) is shown in Algorithm 1.

3.2 DISTA: Distributed Iterative Shrinkage-Thresholding Algorithm

Given large amounts of user historical activities, a very intuitive way to solve the problem mentioned in ($\star$) is building a regression model. We intend to develop our model to have the following two properties: (1) less sensitive to outliers, and (2) can promote sparse solutions because most of the features are irrelevant to the class/label, even using top k features after feature selection. Consider the unconstrained minimization problem of a continuously differentiable function $f(\alpha)$: $R^n \rightarrow R$: $min\{f(\alpha), \alpha \in R^n\}$ ($\triangle$). One of the simplest methods for solving ($\triangle$) is the gradient descent algorithm which generates a sequence of α^k via

Algorithm 1 CSC. al: an activity list for a user

map function:
for all $b_i \in al$ **do**
  if $b_t \in al$ **then**
    output <(b_i, b_t), **1**>;
  end if
  output <b_i, **1**>;
end for

reduce function:
for all keys: (b_i, b_t) and b_i **do**
  sum all values $\rightarrow S_{it}$ or S_i;
end for

for all $b_i \Rightarrow b_t$ sequentially **do**
  $Conf(b_i \Rightarrow b_t) = S_{it}/S_i$;
end for

$\alpha^k = \alpha^{k-1} - t^k \nabla f(\alpha^{k-1})$ ($\diamondsuit$), where $\alpha^0 \in R^n$, $t^k > 0$ is a suitable step size. It is very well known [3] that the gradient iteration in ($\diamondsuit$) can be viewed as a proximal regularization of the linearized function f at α^{k-1}, and written equivalently as $argmin_\alpha\{f(\alpha^{k-1}) + \nabla f(\alpha^{k-1})^T(\alpha - \alpha^{k-1}) + \frac{1}{2t^k}\|\alpha - \alpha^{k-1}\|_2^2\}$. Adopting this same basic gradient idea to the non-smooth l_1 regularized problem: $min\{f(\alpha)+\lambda\|\alpha\|_1 : \alpha \in R^n\}$. It leads to the iterative scheme: $\alpha^k = argmin_\alpha\{f(\alpha^{k-1})+\nabla f(\alpha^{k-1})^T(\alpha - \alpha^{k-1}) + \frac{1}{2t^k}\|\alpha - \alpha^{k-1}\|_2^2 + \lambda\|\alpha\|_1\}$. α^k can be solved as: $\alpha^k = T_{\lambda t^k}\{\alpha^{k-1} - t^k\nabla f(\alpha^{k-1})\}$, where $T_x(\cdot) : R^n \rightarrow R^n$ is the shrinkage soft threshold; $T_x(y) = (|y| - x)^+ sign(y)$, where $(Y)^+ = max\{0, Y\}$ and $sign$ is the sign function. Therefore, $\alpha^k = (|\alpha^{k-1} - t^k\nabla f(\alpha^{k-1})| - \lambda t^k) sign(\alpha^{k-1} - t^k\nabla f(\alpha^{k-1}))$

THEOREM 1. α^k ***is separable to calculate. Since the*** l_1 ***norm is separable, the computation of*** α^k ***reduces to solving a one-dimensional minimization problem for each of its components.***

Proof: α^k is equivalent to $argmin_\alpha\{\frac{1}{2t^k}\|\alpha - \alpha^{k-1} + t^k\nabla f(\alpha^{k-1})\|_2^2 + \lambda\|\alpha\|_1\}$ after ignoring constant terms, because:

$$\begin{aligned}\alpha^k &= argmin_\alpha\{\tfrac{1}{2t^k}(\|\alpha - \alpha^{k-1}\|_2^2 + 2t^k\nabla f(\alpha^{k-1})^T(\alpha - \alpha^{k-1}) \\ &\quad +(t^k)^2\|\nabla f(\alpha^{k-1})\|_2^2) + \lambda\|\alpha\|_1\} \\ &= argmin_\alpha\{\tfrac{1}{2t^k}(\|a\|_2^2 - 2a^Tb + \|b\|_2^2) + \lambda\|\alpha\|_1\} \\ &= argmin_\alpha\{\tfrac{1}{2t^k}\|\alpha - \alpha^{k-1} + t^k\nabla f(\alpha^{k-1})\|_2^2 + \lambda\|\alpha\|_1\} \\ &= argmin_\alpha\{\tfrac{1}{2t^k}\|\alpha - c\|_2^2 + \lambda\|\alpha\|_1\} \\ &= argmin_\alpha\{\tfrac{1}{2t^k}\textstyle\sum_{i=1}^n(\alpha_i - c_i)^2 + \lambda|\alpha_i|\}\end{aligned}$$

where $a = \alpha - \alpha^{k-1}$, $b = t^k\nabla f(\alpha^{k-1})$, and $c = \alpha^{k-1} - t^k\nabla f(\alpha^{k-1})$. t^k is the step length. From this derivation, we could see that we can minimize each component of α separately. This also provides our opportunities of distributed computing. Therefore,

$\alpha_i^k = (|\alpha_i^{k-1} - t^k\nabla f(\alpha_i^{k-1})| - \lambda t^k) sign(\alpha_i^{k-1} - t^k\nabla f(\alpha_i^{k-1}))$ ■

There are still some key points that need to be addressed, including: (I) step length. Usually, we use $t^k = \frac{1}{L}$ as the step length where L is the lipschitz continuity. In this work, we set L to $\|A^TA\|_2$. (II) Stopping condition. We use the following criteria to stop the iterative learning process.

$$\frac{\|\alpha^{k+1} - \alpha^k\|_F^2}{\|\alpha^k\|_F^2} \leq \epsilon$$

where $\|X\|_F$ is called the Frobenius norm and $\|X\|_F = \sqrt{\sum_{i=1}^m\sum_{j=1}^n|x_{ij}|^2}$. (III) Convergence. Previous work has show ISTA algorithm behaves like: $f(\alpha^k) - f(\alpha^\star) \simeq \mathcal{O}(1/k)$ ($\alpha^\star$ is the optimal value of α), namely, shares a sublinear global rate of convergence. In [2], authors proved the converge in function values as $\mathcal{O}(1/k^2)$, where k is the iteration counter. (IV) Backtracking. There are a number of different accelerated backtracking schemes and these are made under different criteria for the same reason. We use one of the simpler schemes - line search backtracking. Algorithm 2 describes the learning process of DISTA.

Algorithm 2 DISTA: Distributed Iterative Shrikage-Thresholding Algorithm with Line Search Backtracking

choose β, such that $0 < \beta < 1$;
$t^0 = 1$;
repeat
  $t^k = t^{k-1}$;
  for all i such that $1 \leq i \leq n$ **do**
    {distributed computing of α_i as indicated in ■}
    $\alpha_i^+ = T_{\lambda t^k}\{\alpha_i^{k-1} - t^k\nabla f(\alpha_i^{k-1})\}$;
  end for
  while $(f(\alpha^+) > f(\alpha^{k-1}) + \nabla f(\alpha^{k-1})^T(\alpha^+ - \alpha^{k-1}) + \frac{1}{2t^k}\|\alpha^+ - \alpha^{k-1}\|_2^2)$ **do**
    {line search backtracking step}
    $t^k = \beta t^k$;
    for all i such that $1 \leq i \leq n$ **do**
      $\alpha_i^+ = T_{\lambda t^k}\{\alpha_i^{k-1} - t^k\nabla f(\alpha_i^{k-1})\}$;
    end for
  end while
until the stopping criteria meets
return α^+;

4. EXPERIMENTS AND RESULTS

In this section, we first describe the structure of data used in our experiments. As the social media data is generated by the public, there are many noisy factors. It is necessary to filter out spams to obtain a high-quality data for producing unbiased results. Then, we discuss the experimental results of feature selection and DISTA under different parameter settings and compare them with some baselines.

4.1 Experimental Data and Cleaning

On Facebook, the largest and most popular social network platform, many companies, organizations, and individuals build their own pages to communicate with social users (fans), which generates an extensive amount of networked and textual information. In this paper, we mainly consider social brands as our target objects. We use Facebook Graph API to download the available activities made on brand side such as posts and user side, such as comments on posts, likes on posts, and public profiles. We have designed some rules to filter out spam users and their activities in our previous work, such as users having an abnormal amount of brand accesses (e.g. >100). Table 1 describes the cleaned data used in our experiments. For labels in the training dataset, we consider users who make all positive comments on the target brand as positive samples and negative comments as negative samples.

4.2 Experimental Results

The input data used in our experiments is big. Using single machine to do feature selection, and regression model

Table 2: The comparison of classification accuracy using DISTA between with and without incorporating feature selection under different size of training sets with three baselines. All these results are average accuracy on 10 target brands.

Row Normalization	Model	Classification Accuracy: Without Feature Selection, Size (10,000)	Without Feature Selection, Size (20,000)	With Feature Selection, Size (10,000)	With Feature Selection, Size (20,000)
No	Naive Bayes	55.52%	57.30%	58.82%	55.44%
	SVM	61.31%	60.52%	63.04%	56.62%
	Logistic Regression	70.14%	70.10%	71.18%	79.58%
	DISTA	**72.07%**	**73.14%**	**77.58%**	**81.68%**
Yes	Naive Bayes	68.95%	71.04%	86.65%	86.24%
	SVM	77.53%	79.76%	**87.89%**	88.52%
	Logistic Regression	76.70%	79.50%	86.78%	88.07%
	DISTA	**80.32%**	**80.50%**	81.76%	**89.25%**

Table 1: Data descriptions after cleaning.

# of unique users	$97,699,832$
# of social brands	$7,580$
# of the triple (user, page, comments)	$102,517,478$
# of the triple (user, page, likes)	$192,442,757$
The number of total post likes	$5,275,921,875$

Table 3: Top 5 associated brands sorted by the confidence score of the rule: "$b_i \Rightarrow Nordstrom$"

Rank	Brand Name (b_i)	Confidence Score
1	NORDSTROM RACK	0.288
2	NEIMAN MARCUS	0.225
3	HAUTELOOK	0.185
4	SAKS FIFTH AVENUE	0.181
5	LORD & TAYLOR	0.169

building is infeasible. In fact, we could not finish the job within 10 hours using only single machine. Hence, we conduct our experiments on a Hadoop-based environment which has 10 machines. Each machine has 8 compute processors. We randomly select 10 different target brands in our experiments. Table 3 shows top 5 correlated brands to the target brand "Nordstrom" in terms of the confidence score. Table 2 compares the performance of using DISTA between with and without incorporating this feature selection strategy under different size of training sets with three other baselines. It shows that with our feature selection strategy can obtain up to 16% increase of accuracy and also always beat without incorporating feature selection.

To build the model, we used the training dataset of size $10,000$ positive instances and $10,000$ negative instances. We use 10-fold cross validation. For such training sets, it takes a long time to finish learning. But our DISTA learning algorithm significantly speeds it up, as shown in Figure 1.

5. CONCLUSION AND FUTURE WORK

In this work, we build a user predictive model based on their historical behaviors on social media for on-line advertising. We implemented a distributed Apriori feature selection for reducing the training dataset. In addition, we implemented a distributed iterative shrinkage thresholding model to predict user's preference. The experiments conducted on Facebook data has shown that all proposed techniques in this work are scalable and efficient for social audience-targeted advertising. Future work includes deeply understanding and incorporating semantics of user-generated contents; finding more accurate and fast predictive learning algorithms.

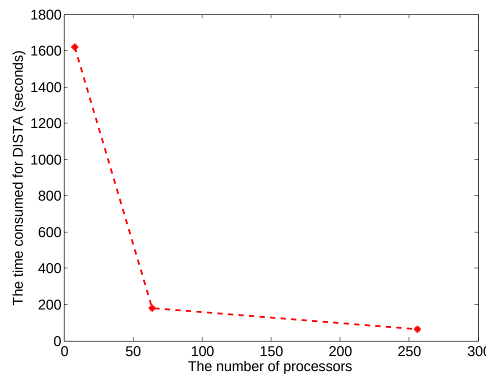

Figure 1: The time (seconds) consumed for DISTA on different number of processors.

6. REFERENCES

[1] http://www.biakelsey.com/index.asp.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, pages 183–202, 2009.

[3] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, 2010.

[4] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[5] Jie Tang, Tiancheng Lou, and Jon Kleinberg. Inferring social ties across heterogenous networks. In *WSDM*, pages 743–752, New York, USA, 2012.

[6] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35, 2005.