Scalable Audience Targeted Models for Brand Advertising on Social Networks

Kunpeng Zhang, Aris Ouksel, Shaokun Fan, Hengchang Liu

Department of Information and Decision Sciences,

University of Illinois at Chicago, Chicago, IL 60607, USA

UIC UNIVERSITY OF ILLINOIS AT CHICAGO

The ACM Conference Series on Recommender Systems

Introduction

Recently, the trend to social content-driven advertising is becoming increasingly evident in business management. Finding targeted audience for precise online advertising based on user historical behaviors is one of the most important marketing tasks.

There are some challenges: (1) Existing feature selection algorithms is infeasible and inefficient, which motivates us to find a scalable solution; (2) Implementing distributed algorithms to efficiently and accurately learn predictive models is also not straightforward.

In this work, we implement a MapReduce based feature selection algorithm to find for a given brand the group of correlative brands that share the most user activities. We also implement a distributed stochastic optimization algorithm called iterative shrinkage thresholding algorithm (DISTA) that can handle a large amount of training instances. Our experiment results on Facebook data show that our DISTA can get up to 16% increase of accuracy by incorporating our feature selection strategy comparing to other baselines.

Problem Definition

Our problem is a typical classification in machine learning. 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. The target brand (b_t) is labeled in a binary form: 1 if a user is interested in this brand, 0 otherwise. Then we will have an activity matrix A.

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

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

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

Where A_k is the k^{th} row in the matrix A. P_{kt} is in [0,1]. It represents the preference on brand t of the k^{th} user; $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$

Then we try to solve the following convex optimization problem:

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

Where $\|\alpha\|_1 = \sum(|\alpha_1 + \alpha_2 + \ldots + \alpha_n|)$

Data Collection

We use Facebook Graph API to download the available activities. 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 this work. 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.

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

Feature Selection

The goal here is to find the frequent pattern "" based on a large amount of user historical activities across brands. Two-itemset (I_x, I_y) association rule ("$I_x \rightarrow I_y$") indicates their correlation. Here I_x could be any brand 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 "$b_i \rightarrow b_t$".

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

Where *Support(X)* is the occurrence frequency of *X*. The MapReduce-based algorithm of calculating confidence score (CSC) is shown in Algorithm 1.

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

DISTA: Distributed Iterative Shrinkage Thresholding Algorithm

We want to find a model to have the following two properties: (1) less sensitive to outliers; (2) can promote sparse solutions because most of the features are irrelevant to the class/label, even suing top k features after feature selection. Consider the unconstrained minimization problem of a continuously differentiable function:

$$f(\alpha): R^n \rightarrow R : \min\{f(\alpha), \alpha \in R^n\}$$

One of the simplest methods for solving this is the gradient descent algorithm. We also found the independence when we calculate α_i. Therefore,

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

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 $\blacksquare$}
    $\alpha_i^+ = T_{\lambda t^k}\{\alpha_i^{k-1} - t^k \nabla f(\alpha_i^{k-1})\}$;
  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^+;

α^k is Separable

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:

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

Experimental Results

Accuracy comparison with baseline algorithms: Naïve Bayes, SVM, and Logistic Regression for both with row normalization and without shown in the Table 2. All are average accuracy on 10 target brands.

		Classification Accuracy			
Row Normalization	Model	Without Feature Selection		With Feature Selection	
		Size (10,000)	Size (20,000)	Size (10,000)	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%**

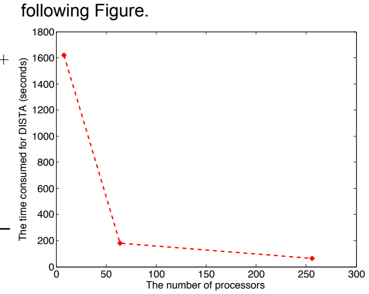
Feature Selection Results

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

Time Complexity

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



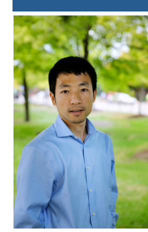
Conclusion and Future Work

- Build a user predictive model based on user historical activities on social media platforms by mplementing a distributed feature selection algorithm to reduce training dataset and a distributed iterative shrinkage thresholding model to find user's preferences.
- Needs to incorporating semantic understanding of user-generated content

References

- Amir Beck and Marc Teboulle: "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM J. Img. Sci., pages 183–202, 2009.
- Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen: "A singular value thresholding algorithm for matrix completion," SIAM J. on Optimization, 20(4):1956–1982, 2010.

Contact



Email: kzhang6@uic.edu

Phone: (312)-996-0819

Website: http://kzhang6.people.uic.edu