Parallelizing LR using MapReduce

And It's commercial applications

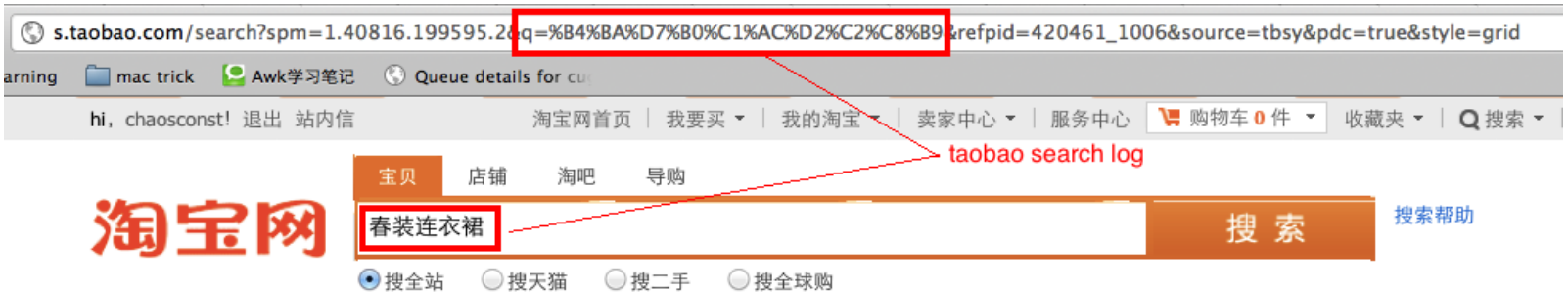
Outline

- Introduction to Map-Reduce programming : Top Query
- Basic Matrices Operation using Map-Reduce
- Solve LR: Gradient Descent, Newton's Method, and beyond
- Taobao Item CTR Prediction

Introduction to Map-Reduce programming

Finding Shopping Interests in online market

Find shopping interests in online market



PV from	Data size (after compress)	Row count
Search	Over 1.15T/day	1,600,000,000
Taobao Union	Over 1.18T/day	44,000,000,000

A simple way to count

Iphone
Ipad
连衣裙
龟苓膏
连衣裙
Iphone
Ipad
充值卡
Iphone
Ipad
连衣裙
充值卡

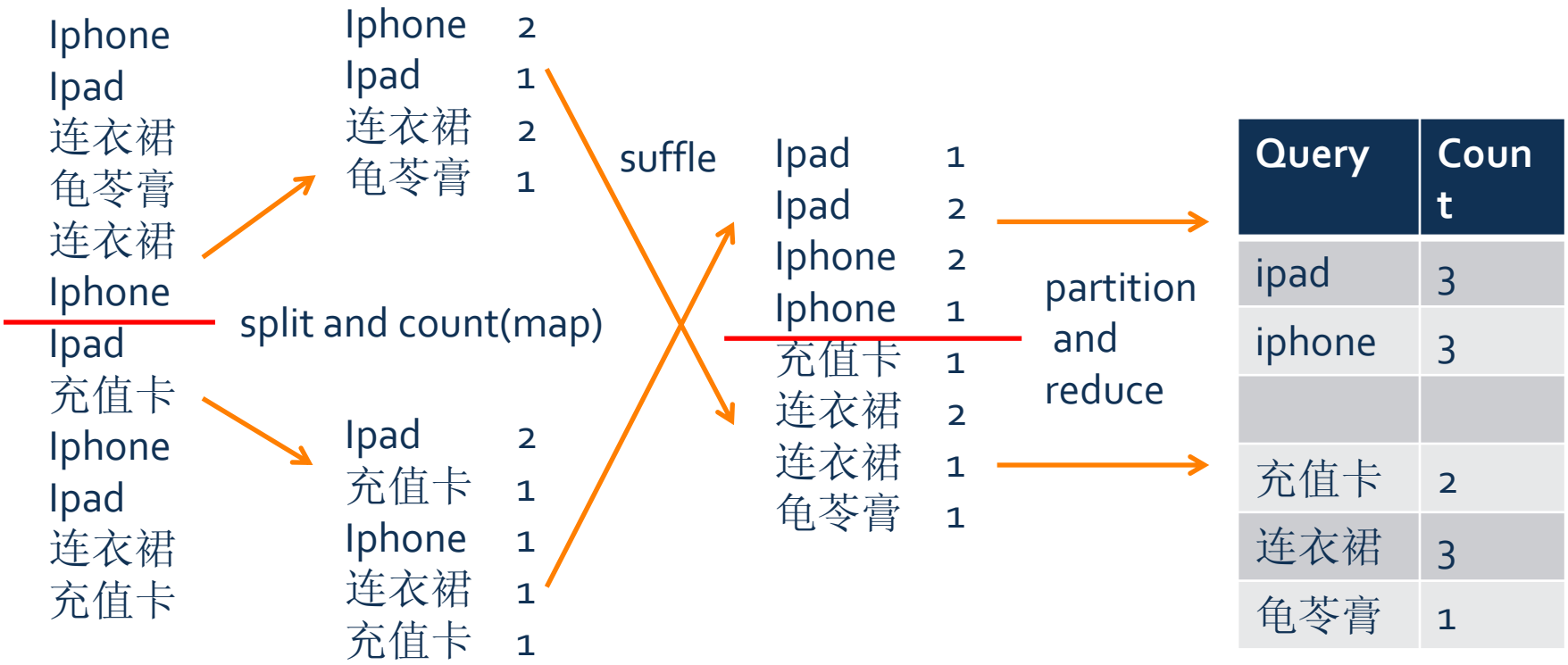
sort →

Ipad
Ipad
Ipad
Iphone
Iphone
Iphone
充值卡
充值卡
连衣裙
连衣裙
连衣裙
龟苓膏

sum →

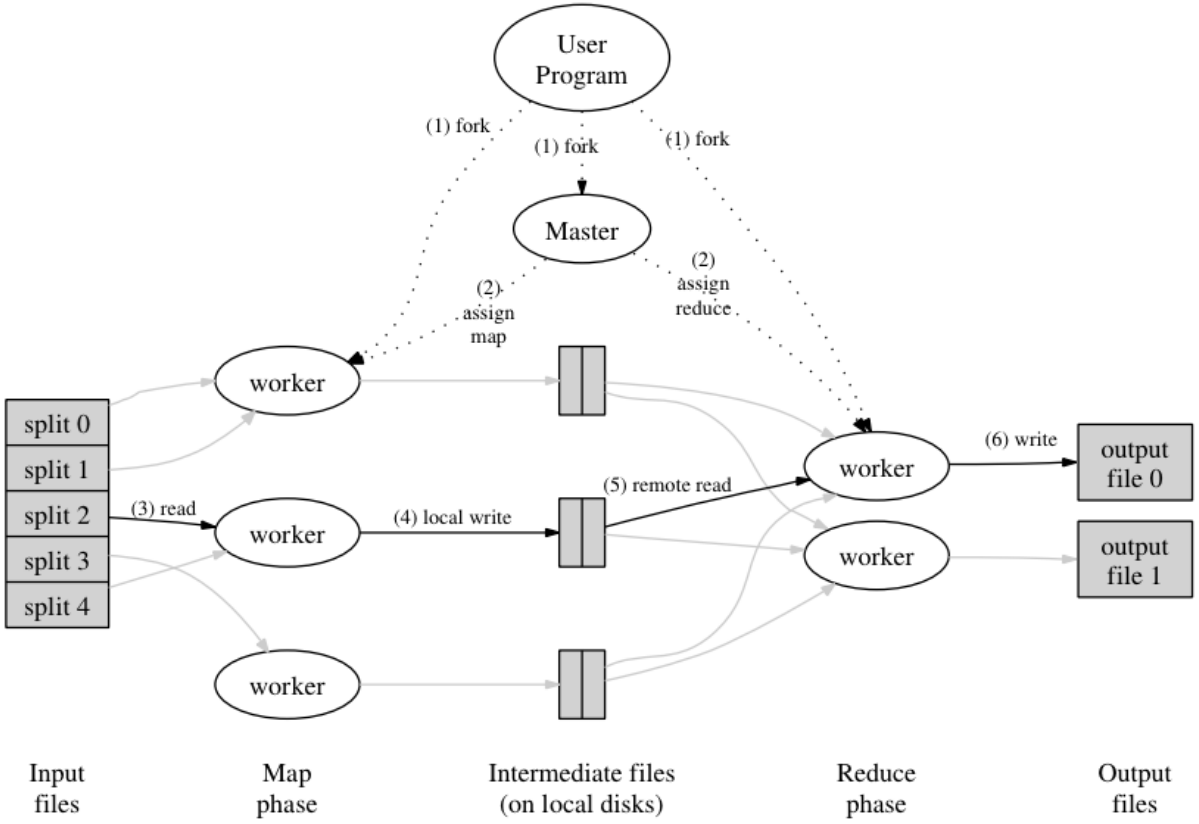
Query	Count
ipad	3
iphone	3
充值卡	2
连衣裙	3
龟苓膏	1

Parallelizing count



Query	Count
ipad	3
iphone	3
充值卡	2
连衣裙	3
龟苓膏	1

Map Reduce Execution Overview



Practice: top query count

Mapper: sort | uniq –c

Reducer: awk –F '\t' '{
    if ($2!=key) {
        if (key!="") print key"\t"sum
        key=$2
        sum=0
    }
    sum+=$1
}END{
    if (sum>0) print key"\t"sum
}'

Practice: top query count

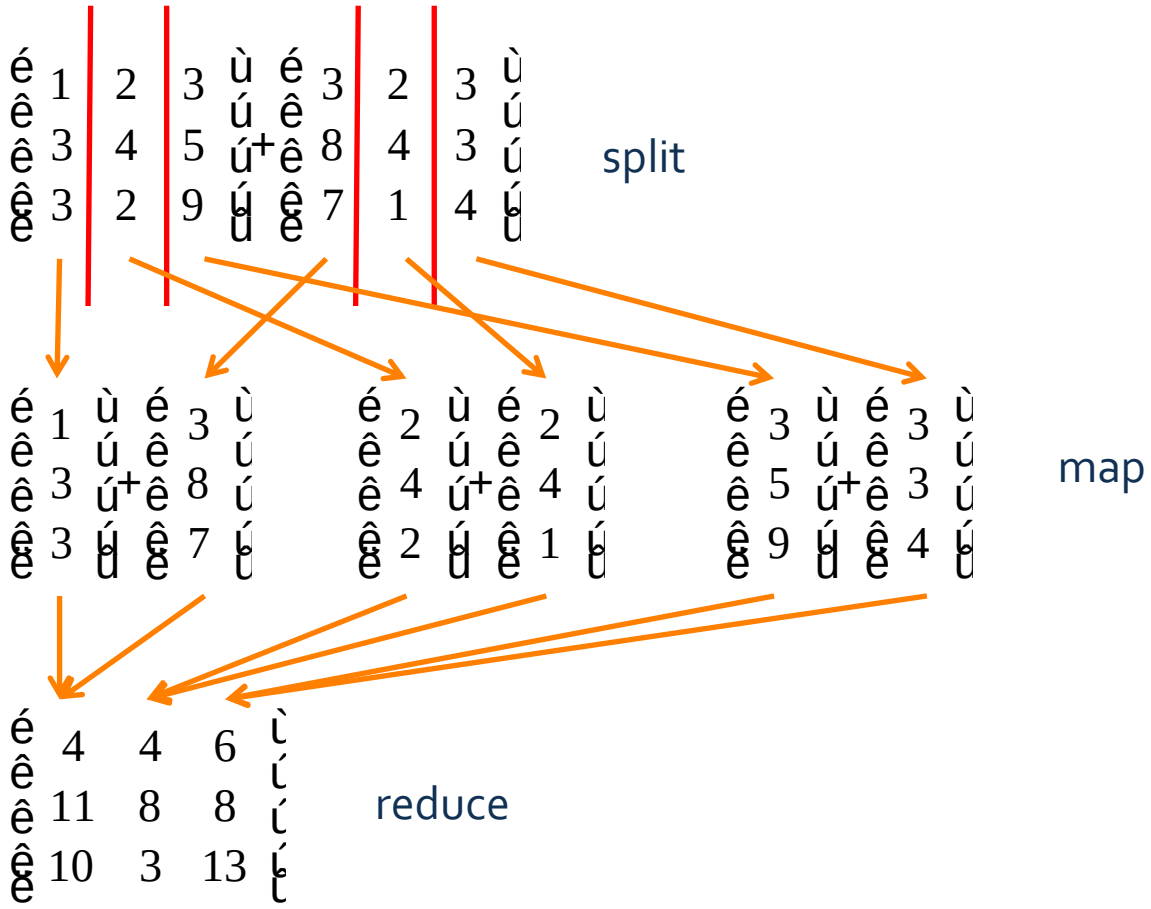
```
#!/bin/bash

/home/a/libexec64/hadoop/current/bin/hadoop

--config ~/conf.mutian/

jar /home/a/libexec64/hadoop/current/contrib/streaming/hadoop-0.19.2-streaming.jar

-D mapred.job.name=“get top query"

-input “/somedir/to/pv/log"

-output /somedir/for/output

-mapper mapper.sh -file ./mapper.sh

-inputformat SequenceFileAsTextInputFormat

-reducer reducer.sh –file ./reducer.sh
```

Top Query Result

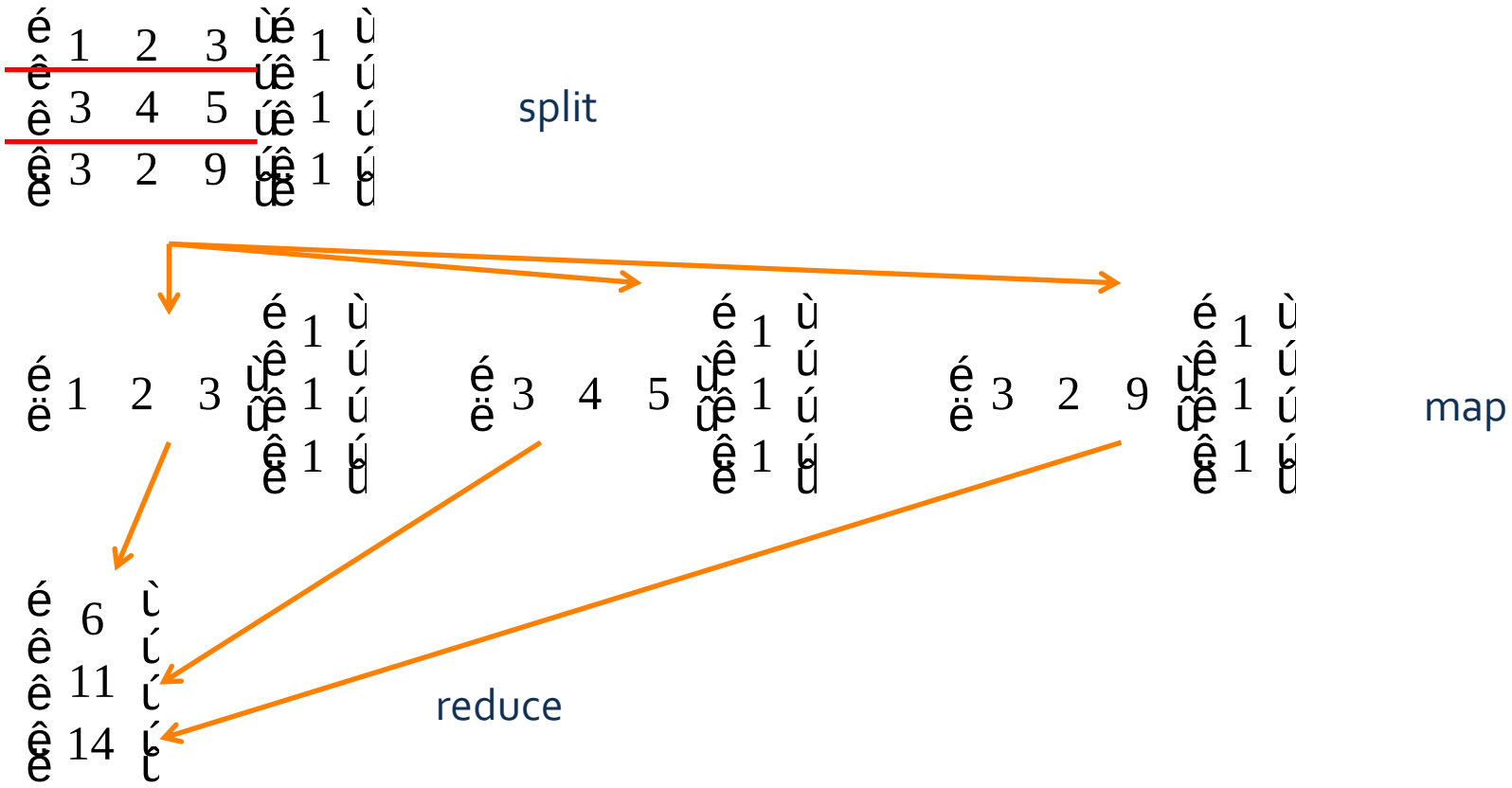
Basic Matrices Operation

Sum, Multiply and Inverse a Matrix using map-reduce

Basic Matrix Operation : Sum



Basic Matrix Operation : Multiply



~~Basic~~ Matrix Operation: Inverse

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 3 & 2 & 9 \end{bmatrix}^{-1}$$

How could you split this?

If a matrix A has the property that:

$$\lim_{n\to\infty} (\mathbf{I} - \mathbf{A})^n = 0$$

Then A is nonsingular and its inverse may be expressed by :

$$\mathbf{A}^{-1} = \sum_{n=0}^{\infty} (\mathbf{I} - \mathbf{A})^n.$$

Which can be calculated by basic matrix operation we have already discussed.

Example Code

```
#!/bin/bash

alpha=0.15

awk -F '\t' 'BEGIN{
        join=0;
}{
    if (key==$1)
    {
        if (split($2,inf,"^A")>1)
        {
            last_pr=inf[1];
            init_pr=inf[2];
            cons=inf[3];
            join=1;
        }
        else sum_strength+=$2;
    }
    else
    {
        if (join==1)
        {
            pr='$alpha'*init_pr+(1-'$alpha')*sum_strength;
            print key"^A"pr"^A"init_pr"^A"cons;
        }

        last_pr=0;
        init_pr=0;
        cons="";
        join=0;
        key=$1;
        sum_strength=0;

        if (split($2,inf,"^A")>1)
        {
            last_pr=inf[1];
            init_pr=inf[2];
            cons=inf[3];
            join=1;
        }
        else sum_strength+=$2;
    }
}END{
    if (join==1)
    {
        pr='$alpha'*init_pr+(1-'$alpha')*sum_strength;
        print key"^A"pr"^A"init_pr"^A"cons;
    }
}'
```

reducer.awk 2,0-1 All

```
echo "";

for ((i=0;i<$max;i++))
do
    turnNum=$(echo $i+1|bc)
    echo "---------------------------------------";
    echo "Turn $turnNum/$max";
    echo "---------------------------------------";
    echo "rm old dir ...";
    if hadoop dfs -test -e /ppr/compute/20110815/turn$turnNum; then
        hadoop dfs -rmr /ppr/compute/20110815/turn$turnNum ;
    fi
    echo "running ...";
    hadoop jar $HADOOP_HOME/contrib/streaming/hadoop-streaming-0.20.203.0.jar \
            -D mapred.job.name="people rank iteration turn"$turnNum \
            -input /ppr/compute/20110815/turn$i/ \
            -output /ppr/compute/20110815/turn$turnNum \
            -mapper mapper.awk \
            -reducer reducer.awk \
            -file /home/chaosconst/peopleSearch/dev/pprMapRduce/awk/reducer.awk \
            -file /home/chaosconst/peopleSearch/dev/pprMapRduce/awk/mapper.awk

    echo "turn $i done.";
done;
```

./run_distribute.sh

```
#!/bin/bash
awk -F '^A' '{
    #print my self
    print $1"\t"$2"^A"$3"^A"$4;

    #print connections;
    split($4,array,"^B");
    for (x in array)
    {
        split(array[x],con,"^C");
        target=con[1];
        strength=con[2];
        print target"\t"strength*$2;
    }
}'
```

mapper.awk

Solve LR Problem

Gradient Descent, Newton's Method, and beyond

Gradient Descent for Linear Regression

Quesion:

$$\min L(\vec{\theta}) = \min \frac{1}{2} \|X\vec{\theta} - \vec{y}\|_2$$

where X is input feature matrix, $\vec{y}$ is target vector, $\vec{\theta}$ is the parameter vector we want to find.

Slove:

$$\begin{aligned} \vec{\theta}_{next} &:= \vec{\theta} - \alpha \nabla L(\vec{\theta}) \\ &:= \vec{\theta} - \alpha X^T (X\vec{\theta} - \vec{y}) \end{aligned}$$

All the operation are basic matrix operation we have already discussed.

Gradient Descent for Logistic Regression

Quesion:

$$\max l(\vec{\theta}) = \max log(\Pi_{i=1}^{m} Pr(y^{(i)}|x^{(i)}, \theta))$$

where X is input feature matrix, $\vec{y}$ is target vector, $\vec{\theta}$ is the parameter vector we want to find.

Slove:

$$\begin{aligned} \vec{\theta}_{next} &:= \vec{\theta} + \alpha \nabla l(\vec{\theta}) \\ &:= \vec{\theta} - \alpha X^T (g(X\vec{\theta}) - \vec{y}) \end{aligned}$$

$$g(\vec{x}) = \begin{bmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_n) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-x_1}} \\ \frac{1}{1+e^{-x_2}} \\ \vdots \\ \frac{1}{1+e^{-x_n}} \end{bmatrix}$$

Newton-Raphson for Logistic Regression

Quesion is the same one,

Slove:

$$\begin{aligned} \vec{\theta}_{next} &:= \vec{\theta} - H^{-1}\nabla l(\vec{\theta}) \\ &:= \vec{\theta} - H^{-1}X^T(g(X\vec{\theta}) - \vec{y}) \end{aligned}$$

where,

$$H(i,j) := \frac{\partial l(\theta)}{\partial(\theta_i)\partial(\theta_j)}$$

Parallelizing Newton-Raphson

- Based on paper
  - "Map-Reduce for Machine Learning on Multicore" by Andrew Y. Ng Group
- Parallelizing compute H^{-1} and $\nabla l(\vec{\theta})$
- They said:

Logistic Regression (LR) For logistic regression [23], we choose the form of hypothesis as $h_\theta(x) = g(\theta^T x) = 1/(1 + \exp(-\theta^T x))$ Learning is done by □tting θ to the training data where the likelihood function can be optimized by using Newton-Raphson to update $\theta := \theta - H^{-1}\nabla_\theta \ell(\theta)$. $\nabla_\theta \ell(\theta)$ is the gradient, which can be computed in parallel by mappers summing up $\sum_{subgroup}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$ each NR step i. The computation of the hessian matrix can be also written in a summation form of $H(j,k) := H(j,k) + h_\theta(x^{(i)})(h_\theta(x^{(i)}) - 1)x_j^{(i)}x_k^{(i)}$ for the mappers. The reducer will then sum up the values for gradient and hessian to perform the update for θ.

Implementations

- LIBLINEAR
  - http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- WEKA
  - http://www.cs.waikato.ac.nz/ml/weka/
- Apache Mahout
  - https://cwiki.apache.org/confluence/display/MAHOUT/Logistic+Regression
- Glm in Rmpi
  - http://stat.ethz.ch/R-manual/R-patched/library/stats/html/glm.html
  - http://www.stats.uwo.ca/faculty/yu/Rmpi/

Beyond Newton-Raphson

- Discretization of Features: Millions of Dimensions and more
  - Find all the possible relation between input features and the target
- Logistic Regression with Regularization
  - Add norm of \theta into objective function
  - Anti-over-fitting
- Quasi-Newton Method
  - DFP
  - BFGS
- Batch Gradient Descent and Batch Newton Method
  - Train faster

Item CTR Prediction

How to judge goods in Taobao.com

AdWords in Taobao.com

Under the query :
clothing
Which item is better?

	Row count
Taobao P4P Item	>10,000,000
Queries	>10,000,000

We need a Static Score for every (Query,Ad) pair

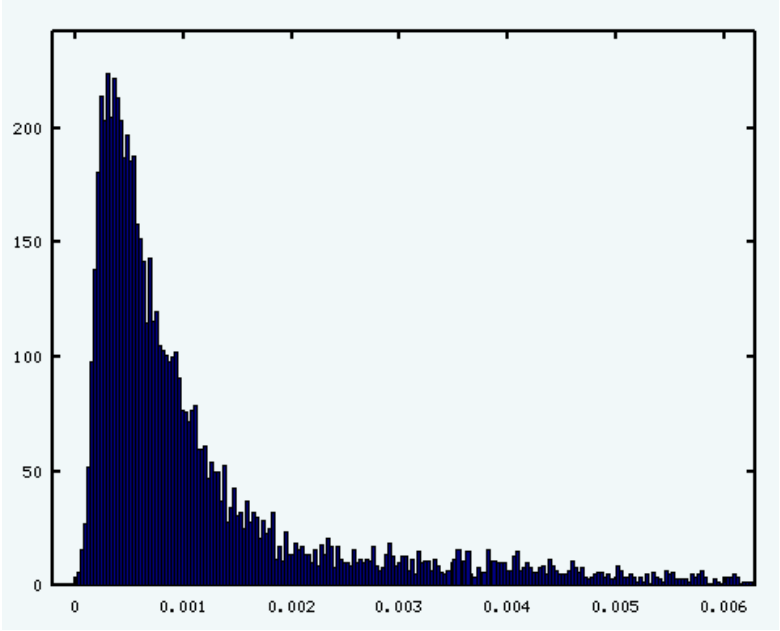
关键词管理（125） 推广内容管理（2） 无展现词管理(0) 昨天 过去7天 过去30天

关键词： 搜索 关键词查询工具

批量修改出价 删除 添加关键词

关键词(125)	状态	出价	展现量	点击量	点击率	花费	平均点击费用	质量得分	操作
默认出价		¥ 0.65	-	-	-	-	-	-	
类目出价	已启用	¥ 0.65 默认出价	591352	1625	0.27%	¥ 2183.44	¥ 1.34	4分	
定向推广出价		¥ 0.5	52	0	0%	¥ 0	¥ 0	-	
2010 新款	推广中	¥ 0.65 默认出价	171	1	0.58%	¥ 1.25	¥ 1.25	2分	
立领 短款 羽绒服	推广中	¥ 0.65 默认出价	61	0	0%	¥ 0	¥ 0	8分	
淑女 羽绒服 短款	推广中	¥ 0.65 默认出价	20	1	5%	¥ 0.69	¥ 0.69	8分	

Target: Click Through Rate



$$ctr = \frac{click}{impression}$$

Item Quality

ItemQuality = history CTR ?

$$ItemQuality = \frac{click}{impression}$$

What about new ads without enough impression?

How to Anti-Fraud?

$$ItemQuality = g(q^T x)$$

x is a feature vector reflect multi-dimension click-feedback history and The relevance between query and item.

Top Secret I: What are these features?

- Query Keywords Exist in Item Title?
- Item Category Match Query Category?
- How many times Item's Category has been clicked by user under certain Query.
- Every day Impression and Click since two weeks ago
- Impression and Click from the same query, same store, same category.

Top Secret II: What's the model?

- Before 2010.6
  - No model at all, ranking by price
- 2010.6
  - Linear Model, CTR +30%
- 2010.11
  - Logistic Smoothed CTR, CTR +10%
- 2011.12
  - Boosting Tree, Expanded CTR Feature, CTR +20%

Summary

- Large Scale Problem can be done by map-reduce procedure
- Key Point : Divide and Conqueror
- Matrix Inverse is not so easy
- Solve Optimize Problem by Iteration maybe a good choice
- Item CTR prediction can be done with Logistic Regression

References

- Itunes U video Lecture
  - http://deimos3.apple.com/WebObjects/Core.woa/Feed/itunes.stanford.edu-dz.4331558558.04331558560
- Machine Learning using MapReduce by Andrew Y. Ng Team
  - http://www.cs.stanford.edu/people/ang//papers/nips06-mapreducemulticore.pdf
- Google Paper about map-reduce
  - http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//archive/mapreduce-osdi04.pdf
- 拟牛顿法
  - http://wenku.baidu.com/view/024a1ad184254b35eefd3441.html
- LR model
  - http://cseweb.ucsd.edu/~elkan/250B/logreg.pdf
  - http://people.csail.mit.edu/jrennie/writing/lr.pdf
- PTLR: 云计算平台商处理大规模移动数据的置信域逻辑回归算法
  - http://www.klmp.pku.edu.cn/Paper/UsrFile/449.pdf

Thank you

by YuanXingyuan

@etao.com