Large-Scale Click-stream and transaction log mining in practice

Uwe Mayer, Nish Parikh, Gyanit Singh
October 6-9, 2013.
BIG DATA SCIENCE

Best Practices
Key Ideas

• Big Data Sets
• Big Data Properties
• Challenges in working with big data
• Practical Solutions
• Leveraging Hadoop
• Case Studies
Types of Data Used in this Tutorial

- Click-stream logs
  - PetaByte Scale
- Transactional Data
  - TeraByte Scale
  - More than ½ B items for sale
BEST PRACTICES USED IN PRESENTED CASE STUDIES

- Data Cleaning
  - Taking care of bad data
  - Importance of domain knowledge
- Data Sampling
  - Reservoir sampling
- De-duplication
- Normalization
- Handling Idiosyncrasies of long-tail data
- Understanding Tractability of Algorithms
- Efficiency at scale
- Bucketing data in the right way
- Bias Removal
  - System bias
  - Platform bias
  - User bias
- Handling curse of dimensionality
More Data is Good

Number of Unique Queries having Query Suggestion Recommendations (Millions)

- 1.5 weeks 100% training data
- 13 weeks 25% training data
- 4 weeks 100% training data
But it needs to be used carefully

Figure 3: Describes the effect of the parameter $\alpha$ on nDCG and MRR. The peak is for the value of 12 months. The metrics obtained by using temporal decay are normalized against those obtained without using temporal decay. The early dip shows that as we increase lookback into history we get better coverage and precision up to a certain point. We observe that using large corpuses of historical data leads to better coverage for intent inference. Also, the best prediction accuracy is achieved, when the past one year data is considered to be highly and equally relevant ($\alpha = 12$ months) and the data beyond the one year period is weighted by decaying exponentially ($\beta = 25$). $\alpha$ and $\beta$ are chosen through heuristical parameter sweeping techniques maximizing the prediction metrics over queries in $Q_{low}$. 
QUERY SUGGESTIONS

At Scale over Hadoop
Something different

yahoo
altavista
lycos
excite
hotbot

Searches related to google

google translate  google desktop
google jobs  google maps
google docs  google images
google search  google voice

RELATED SEARCHES
Warren Buffett Personal Holdings
Warren Buffett Interview
Warren Buffett Family
Warren Buffett Email
Warren Buffett Taxes
Warren Buffett Stocks
Oprah Winfrey
Berkshire Hathaway

"ansel adams"

Related Searches: ansel adams framed prints, ansel adams poster, ansel adams prints.
Query Suggestions at eBay

- Enable users to broaden or narrow searches.
- Lead users to related products or brands.
- Optimize the buying experience.
Query Suggestion Algorithms

• Various algorithms in literature
  – Agglomerative clustering
  – Query Similarity Measures (Linguistic, Latent)
  – Query Flow Graphs
• Our approach primarily based on user trails.
Challenges

• Large-scale data
  – 100M+ users.
  – 30TB+ click-stream logs.
  – 1B+ user sessions.
  – Several billion searches.

• Noisy Data
  – Robots
  – API Calls
  – Crawlers, spiders
  – Tools and scripts
  – User Bias

Query Suggestions for the query ‘calculator’. 
Challenges

- Long Tail
- Dynamic Inventory

Suggestions are more useful for tail queries.

Figure 2: (a) Long tail distribution of eBay search queries (b) Relation between query frequency and search result-set size (c) Related search click-through for different result-set size.
HADOOP TO THE RESCUE
Hadoop Cluster at eBay (One of several)

• Nodes
  – Cent OS 4 64 Bit
  – Intel Dual Hex Core Xeon 2.4 GHz
  – 72 GB RAM
  – 2 * 12 (24TB) HDD
  – SSD for OS

• Network
  – TOR 1Gbps
  – Core Switches uplink 40 Gbps

• Cluster
  – 532n – 1008n
  – 4000+ cores – 24000 vCPUs
  – 5 – 18 PB
<table>
<thead>
<tr>
<th>Layer</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBay Infrastructure &amp; Data Source Layer</td>
<td>eBay Data (Logs, Tables)</td>
</tr>
<tr>
<td>Mobius Layer</td>
<td>Low level Dataset access API</td>
</tr>
<tr>
<td></td>
<td>Hadoop Cluster</td>
</tr>
<tr>
<td>Application Layer</td>
<td>Mobius Studio (Eclipse plugin)</td>
</tr>
<tr>
<td></td>
<td>Generic Java Dataset API</td>
</tr>
<tr>
<td></td>
<td>Query Language</td>
</tr>
<tr>
<td></td>
<td>Click Stream Visualizer</td>
</tr>
<tr>
<td></td>
<td>Metrics Dashboard</td>
</tr>
<tr>
<td></td>
<td>Research Projects</td>
</tr>
</tbody>
</table>

Data Cleaning

• Data is cleaned during the processing phase.

• User Bias Removal
  – Filter information from robots, API calls, spiders and crawlers.
  – De-duplicate signals from the same user.

• Platform Bias Removal
  – Treat signals from different platforms like mobile phones, game consoles, computers differently.

• System Bias Analysis
  – Treat searches typed in by users differently from searches issued through user clicks on features.
Recommendation Computation – Phase 1

Input: User Click-stream data

**Mapper**
- Data Cleaning.
- Query Pair and Behavioral Frequency extraction.
- Query normalization.

**Reducer**
- User de-duplication.
- Computation of behavioral features.

**Key:** user, originating query
**Value:** Recommendation query and behavioral frequencies.

**Output:** Query pair and behavioral features per user
**Recommendation Computation – Phase 2**

Input: Query pairs, behavioral features per user

**Mapper**
- Identity Mapper

**Reducer**
- Compute textual features for query pair

**Key:** query, recommendation

- Query pairs with non-trivial textual similarity tend to have non-zero behavioral frequencies.
- Textual similarities computed only for 200M query pairs instead of several trillion.

**Output:** Query pair, behavioral features, textual features
## Results

<table>
<thead>
<tr>
<th>Evaluation Metrics</th>
<th>Baseline</th>
<th>Experiment I</th>
<th>Experiment II</th>
</tr>
</thead>
<tbody>
<tr>
<td># of unique queries with suggestions</td>
<td>4.8M</td>
<td>9.8M</td>
<td>22.6M</td>
</tr>
<tr>
<td>Impression Rate</td>
<td>1</td>
<td>1.12</td>
<td>1.22</td>
</tr>
<tr>
<td>Click-through Rate</td>
<td>1</td>
<td>1.41</td>
<td>1.13</td>
</tr>
</tbody>
</table>

**Live Site Experiments**

- **CTR Increase** attributable to better weighting of behavioral trail data.
- **CTR Increase** due to better data cleaning algorithm.
Remarks

- Log Mining algorithms are parallelizable.
- Easy to scale such algorithms using Hadoop.
- Hadoop empowers us to look at data-sets spanning larger time-frames.
- Hadoop enables us to iterate faster and hence run more user-facing experiments.
TIME SERIES MINING

Mining Large Scale Temporal Dynamics over Hadoop
Why study temporal dynamics?

- Stock Markets
- Bio-Medical Signals
- Traffic, Weather and Network Systems
- Web Search & Ranking
- Recommender Systems
- eCommerce…
Challenges

• Large Scale data
  – 100M+ users
  – Petabytes of click-stream logs
  – Billions of user sessions
  – Billions of unique queries

• Noisy Data
  – Robots
  – API Calls
  – Crawlers, Spiders
  – Tools, Scripts
  – Data Biases

• Data spread across long time frames
  – Differences in collection methodologies

• Complexity of certain algorithms
Mobius – Generic JAVA Dataset API

- Java-based, high-level data processing framework built on top of Apache Hadoop.
- Tuple oriented.
- Supports job chaining.
- Supports high level operators such as join (inner or outer) or grouping.
- Supports filtering.
- Used internally at eBay for various data science applications.
- [https://github.com/gysingh/openmobius](https://github.com/gysingh/openmobius)
Hadoop – Handling External Code

• Pre-compiled Java code can easily be used with Apache Hadoop
• User code needs to be assembled into one or more jar files
• Jars can be copied to the task nodes on the Hadoop cluster with the -libjar option (takes a comma-separated list of local jar names)
• The Hadoop software will add the contents from the Jar file(s) to the classpath on the task nodes
Mobius – Grouping

```java
@override
public int run(String[] args) throws Exception {
    Dataset order = TSVDatasetBuilder.newInstance(this, "orders",
    new String[] {"O_Id", "OrderNo", "P_Id"})
    .addInputPath(new Path("$INPUT_PATH"))
    .build();

    Dataset grouping_result = this
    .group(order).by("P_ID")
    .save(this,
          new Path("$OUTPUT_PATH"),
          new Column(order, "P_ID"),
          new Counts(new Column(order, "O_Id")))
    )
    return 0;
}
```
Mining Temporal Data

• When it’s in your mind, it’s in the Query Logs!
  – Queries as a proxy for demand
Mining Temporal Data

• Data Preparation
  – Robot Filtering
  – Session Log Analysis

• Data Cleaning
  – Normalization
  – De-duplication

Christmas trend – raw data

Christmas trend – prepared data
Mining Temporal Data – What’s Buzzing?

• Automatic Buzz Detection
Air conditioner searches become popular as summer approaches.

Why are searches related to monopoly pieces popular every October?

Mining Temporal Data – Does History Repeat Itself?

• Seasonality and Trend Prediction
Similar patterns for queries related to **Hanukkah**
Preparing Data – Getting Queries from User Sessions

Typical eBay flow

Search ➔ View ➔ Purchase

- **Search**: specify a query, with optional constraints
- **View**: click on an item shown on search results page
- **Purchase**: buy a fixed-price item or place winning bid on an auction item

Consider only queries typed in by humans. Ignore page views from robots or views from paid advertisements, campaigns or natural search links.
Cleaning Data

• Apply default robot detection and removal algorithm
  – Based on IP, number of actions per day, agent information.
• Find the right flows from the sessions.
  – Filter out noisy search events.
  – Remove anomalies due to outlier users.
  – Limit the impact a single user can have on aggregated data (de-duplication).
Finding the right flow in the session

Session 1

Search → Exit

May not consider flows without any interesting activity like clicks

Session 2

Ads/paid search → View → Purchase

May not consider searches coming from advertisements

Session 3

Search → View → Purchase

These kind of sessions are considered and information is aggregated.
Data Preparation - Map Reduce Flow

Preprocessing stage

- Read raw events
- Group events into sessions.
- Group sessions by GUID
- Apply bot filtering algorithm

Collecting stage

- Find the right flow.
- Emit query as key.
- Emit de-duplicated query volume as value
- Calculate sum per key

Save the result so it can be reused by other apps.

Query Volume output daily as `dailyQueryData`
Time Series Generation

**Input:** `dailyQueryData` for multi-year time-frames

- **Mapper**
  - Data Cleaning.
  - Query normalization.

- **Reducer**
  - Time Series formation for all unique queries
  - Time Series indicating total daily activity volume

**Output:** Vectors of Query → Volume Time Time Series
Buzz Detection – 2 state automaton model

• Arrival of queries as a stream.
• “low rate” state ($q_0$) and a “high rate” state ($q_1$).
• $f_0(x) = \alpha_0 e^{-\alpha_0 x}$, $f_1(x) = \alpha_1 e^{-\alpha_1 x}$, where $\alpha_1 > \alpha_0$.
• The automaton changes state with probability $p \in (0, 1)$ between query arrivals.
• Let $Q = (q_1, q_2 \ldots q_n)$ be a state sequence. Each state sequence $Q$ induces a density function $f_Q$ over sequences of gaps, which has the form

$$f_Q(x_1, x_2 \ldots x_n) = \prod_{t=1}^{n} f_{i_t}(x_t)$$

N. Parikh, N. Sundaresan. KDD 2008.
Scalable and Near Real-time Burst Detection from eCommerce Queries.
Buzz Detection – Modeling Queries as a Stream

Frequency of Query

Gaps between arrival times for queries
Buzz Detection – 2 state automaton model

- If number of state transitions in sequence Q are denoted as $b$
- Prior probability of Q is given as
  \[
  \left( \prod_{i_t \neq i_{t+1}} p \right) \left( \prod_{i_t = i_{t+1}} (1-p) \right) = p^b (1-p)^{n-b} = \left( \frac{p}{1-p} \right)^b (1-p)^n
  \]
- Using Bayes theorem, the cost equation is
  \[
  C(Q | X) = b \ln \left( \frac{1-p}{p} \right) + \sum_{t=1}^{n} \ln f_{i_t}(x_t)
  \]
- Sequence that minimizes the cost would depend on
  - Ease of jumps between 2 states.
  - How well the sequence conforms to the rate of query arrivals.
- Configurable Parameters for model are $\alpha_0$, $\alpha_1$ and cost $p$.
  - $\alpha_0$, $\alpha_1$ are calculated from data in the MR job.
  - Heuristically determined value of $p = 0.38$ is used.
Query Volume Time Series – 2 State Representation

harley davidson

April, May 2012

Normalized Query Volume  Bi-State Representation

cinco de mayo

April, May 2012

Normalized Query Volume  Bi-State Representation
### Time Series Normalization and Buzz Detection

**Table:**

<table>
<thead>
<tr>
<th>Time Frame</th>
<th>Event</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov-Jan</td>
<td>Winter</td>
<td>Snowboarding pants, cashmere gloves, fleece lined jeans. Mens ugly Christmas sweater</td>
</tr>
<tr>
<td>Nov</td>
<td>Thanksgiving</td>
<td>Thanksgiving decorations, thanksgiving tablecloth, thanksgiving dress, vintage thanksgiving</td>
</tr>
<tr>
<td>Dec-Jan</td>
<td>Orange Bowl</td>
<td>Orange bowl,</td>
</tr>
<tr>
<td>Jan-Feb</td>
<td>Superbowl</td>
<td>Super bowl tickets, superbowl tickets, ahmad bradshaw, ny giants jersey</td>
</tr>
<tr>
<td>Jun</td>
<td>Father’s Day</td>
<td>Father’s day, father’s day gifts</td>
</tr>
<tr>
<td>Dec</td>
<td>Kwanzaa</td>
<td>Kwanzaa</td>
</tr>
</tbody>
</table>

### Category - click on a category to view
- Antiques
- Art
- Baby
- Books
- Business & Industrial
- Cameras & Photo
- Cell Phones & PDAs
- Clothing, Shoes & Accessories
- Coins & Paper Money
- Collectibles
- Computers & Networking
- Crafts
- Dolls & Bears
- DVDs & Movies
- Electronics
- Entertainment Memorabilia
- Gift Certificates

### Hot Keywords

<table>
<thead>
<tr>
<th>Hot Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>ballerinas</td>
</tr>
<tr>
<td>nike air yeezy 2</td>
</tr>
<tr>
<td>air yeezy 2</td>
</tr>
<tr>
<td>red white blue</td>
</tr>
<tr>
<td>air jordan &amp; olympic</td>
</tr>
<tr>
<td>the dark knight rises</td>
</tr>
<tr>
<td>coach sale</td>
</tr>
<tr>
<td>nike yeezy 2</td>
</tr>
</tbody>
</table>
Binary data structure generation from MR job

- Created new FileOutputFormat
- Write time series data to two files
  - Binary File with fixed sized records indicating time series volume
  - Text file mapping each unique query string to binary file and offset
- Index created by reducers directly loaded by custom servers written in C++.
- Used for an internal Query Trends Application
Query Trends

Popular Queries:
- vintage
- anthropologie
- prada

Trending Queries:
- monopoly pieces
- monopoly metal pieces
- monopoly pieces wooden
- monopoly pieces wood
- pokem
- monopoly monopoly pieces
- monopoly monopoly pieces
- monopoly monopoly monopoly
- monopoly monopoly monopoly monopoly

Chart by amCharts.com

Current query trend period: 2006-03-23 - 2011-09-12
Zoom: 1D 1M 3M 6M 9M 1Y 2Y 3Y YTD MAX
World of Warcraft fans queue to buy Cataclysm expansion

Thousands of people around the world queued into the night to get hold of the latest expansion for World of Warcraft (WoW).

The expansion, called Cataclysm, is the first for two years and makes big changes to the game.

The expansion re-makes the world in which WoW is set and rips up the geography of many familiar places.
Trends – Comparing Queries

Popular Queries:
- missoni target
- star wars
- ralph lauren
- burberry
- target missoni
- nike
- ipad
- iphone 4
- gucci

Trending Queries:
- missoni target
- star wars
- target missoni
- samsung galaxy s ii
- nike mag
- halloween costumes
- missoni c

chart by amCharts.com

Current query trend period: 2006-03-25 - 2011-09-14

Zoom: 10D 1M 3M 6M 9M 1Y 2Y 3Y YTD MAX
Temporal Similarity

- 1+ Billion Queries
- Naïve Algorithm – Quadratic Complexity
- Pearson’s Correlation

\[
\frac{1}{d} \sum_i \left( \frac{X_{p,i} - \mu(X_p)}{\sigma(X_p)} \right) \left( \frac{X_{q,i} - \mu(X_q)}{\sigma(X_q)} \right)
\]

- Candidate Set Reduction
  - Correlations useful only for event-based or seasonal queries
  - Correlations useful in applications only for head and torso queries
  - These filters reduce candidate space from B+ to a few M.

\[
\tilde{X}_{p,i} = \frac{1}{\sqrt{d}} \frac{X_{p,i} - \mu(X_p)}{\sigma(X_p)}
\]
Exact Correlations amongst candidates – All pairs similarity on Reduced Set

Number of Mappers = M
Number of Iterations of the job = K
In every iteration, every mapper $m$ pre-loads $\frac{N}{K}$ queries in RAM
Every mapper is then streamed $\frac{N}{M}$ queries
Every mapper in a single iteration computes $\frac{N}{M} \times \frac{N}{K}$ correlations
In a single iteration all mappers together compute $\frac{N}{M} \times \frac{N}{K} \times M = \frac{N \times N}{K}$ correlations
In $K$ iterations $N \times N$ correlations are computed
As correlation is commutative $\tau_{p,q} = \tau_{q,p}$ we compute $\tau_{p,q}$ only when $p < q$ as an optimization
To save disk space only useful correlations ($\tau > T$) are stored
Parameters we use; $M = 5000, R = 100$ or $0, K = 100, T = 0.7$
Exact correlations for all pairs in reduced set can be computed and used in practice
Applications of Temporal Correlations – Query Suggestions

Welcome! Sign in or register.

CATEGORIES  ELECTRONICS  FASHION  MOTORS  TICKETS  DEALS  CLASSIFIEDS

hanukkah

Related Searches: menorah, hanukkah menorah, hanukkah decorations, hanukkah brass, hanukkah fisher price, judaica, d

7,312 results found for hanukkah

Save search | Tell us what you think

Categories
Collectibles (2,527)
Religion & Spirituality (2,044)
Decorative Collectibles (115)

View as:
Remarks

• Log Mining and Time Series mining algorithms are parallelizable.
• Easy to scale such algorithms using Hadoop.
• Hadoop empowers us to look at data-sets spanning years and years.
• Hadoop enables us to iterate faster and hence run more user-facing experiments.
SHIPPING RECOMMENDATIONS
Outline

• Introduction to selling on eBay
• Shipping suggestion opportunity
• Data to the rescue
• Shipping suggestions: Base approach
• Inhomogeneous category problem
• Improved data mining to the rescue
• Shipping suggestions: Current approach
Listing an item for sale on eBay

• Specify listing title
• Accept / override suggested listing category
• Upload one or more pictures
• Specify item condition (eg, New, Used)
• Type in item description
• Set start price or fixed price, and listing duration
• Specify shipping (service, cost, who pays: buyer / seller)
• Specify accepted payment methods
Shipping on eBay

• eBay would like to help sellers choose a shipping method
• Many different and unique items are offered on eBay
• Weight and dimensions are usually unknown
• Asking sellers to type in weight and dimensions creates friction
• Would like an automatic approach
Data to the rescue

• Sellers on eBay often buy their postage labels through eBay’s label printing platform
• Many different shipping services are offered through eBay label printing (from US Postal Service, FedEx)
• Shipping labels usually include weight and dimensions to determine pricing
• While items are often unique, all items are assigned to categories during listing
Data to the rescue (cont.)

• Approach: aggregate past shipping label data by category
• Run statistics on the weight and dimension data for each category
• Derive a usable data-driven estimate on weight and dimensions
• Choose a suitable service and carrier, and make a suggestion
Label data at eBay

• eBay has at any given time more than 350 million listings worldwide
• Many millions of shipping labels for the US are printed through eBay every year
• Thousands of categories
Processing of label data with Hadoop

• Use Mappers to extract desired fields (weight, dimensions)
• Use Mappers for filtering (eg, exclude USPS flatrate)
• Mapper output key = category, value = weight and dimensions
• Use Reducers to perform statistical evaluation
• Reducer output key = category, value = suggested weight and dimensions
• Pick a suitable carrier and service for each category
Opportunities for Improvement

• Many categories contain a wide variety of items
Improved Approach

• Differentiate items within a category into light and heavy
• Light vs. heavy:
  – “trumpet” category: mouthpiece vs. trumpet with case
  – “dinnerware” category: single plate vs. dinnerware set
  – “computer accessories” category: mouse vs. keyboard
• Besides the listing category use the listing title
• Different words are important for different categories
Improved Approach: What precisely is “heavy”?

• Each category has its own separation into light and heavy
• Some categories are uniform and have no such separation
• Attempt to cluster items by weight in each category into precisely two clusters
• Split the category if both the light and the heavy clusters have sufficient items
Improved Approach: Bag of title words

- Each category has its own collection of title words indicating light and heavy items
- Preselect words important for each category
- Fit a statistical model on the title words that for each listing produces a probability that the item is heavy (or light)
Improved Approach with Hadoop

• Use Mappers to extract desired fields (weight, dimensions, title)
• Use Mappers for filtering (eg, exclude USPS flatrate)
• Mapper output key = category, value = weight, dimensions, and title
• Use Reducers to perform machine learning
  – Clustering to determine light / heavy cut-off
  – Title word selection
  – Title word model fitting
Sampling

• Categories have very different numbers of listings
  – Searching on 2013/09/23 on ebay.com yields:
  – 2,576,202 results for ”dvd”
  – 487 results for ”Climbing Holds”

• Above results are “active items”, if using historical data then some categories’ data will be too large to fit into a single reducer

• The reducer does not know ahead of time how large the category is (records are streamed by Hadoop)

• Use reservoir sampling in case leaf category is too large to fit into a single reducer (hundreds of thousands of records)
Modeling Details

• K-means for clustering of weights, K=2
• Discard clustering if almost all records are in larger cluster or too few records in smaller cluster
• For each category, fit a binary Maximum Entropy model (aka Logistic Regression) on item titles predicting light vs. heavy using standard public-domain Java software
• Perform cross-validation
Improved Approach with Hadoop (cont)

• Reducer also performs data-driven validation and testing of goodness of model fits
• Reducer output key = category, value = model words, model word parameters, and suggested weight / dimensions for light and heavy, model performance statistics
Final System

• Thousands of categories with title models to have suggestions for light and heavy items
• For thousands more rarely used categories have the baseline suggestions
• All transparent to the seller, no additional input required
• Sellers can override if they want
• Abandoning rate of listing flow at shipping stage is significantly improved
Example: Trumpet Mouthpiece

trumpet mouthpiece
Condition: Used  Auction starts at: $0.99

Select the shipping option most sellers use

<table>
<thead>
<tr>
<th>Service</th>
<th>Package</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>USPS First Class Package</td>
<td>Package (or thick envelope)</td>
<td>$2.58 paid by buyer</td>
</tr>
<tr>
<td>2 to 5 business days</td>
<td>6oz. 10.0in. x 7.0in. x 5.0in.</td>
<td></td>
</tr>
</tbody>
</table>

Create your own shipping option

Add international shipping

Previous  Next
Example: Trumpet with Case and extra Mouthpiece

trumpet with case and extra mouthpiece
Condition: Used  Auction starts at: $0.99

Select the shipping option most sellers use

<table>
<thead>
<tr>
<th>Service</th>
<th>Package</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>USPS Priority Mail</td>
<td>Package (or thick envelope)</td>
<td>$13.50-$48.15 paid by buyer varies by buyer's location</td>
</tr>
<tr>
<td>2 to 3 business days</td>
<td>11lbs. 0oz. 24.0in. x 14.0in. x 11.0in.</td>
<td>□ Offer free shipping to attract buyers</td>
</tr>
</tbody>
</table>

Create your own shipping option

□ Add international shipping
References

• Hasan et al. Query suggestion for E-commerce sites. WSDM 2011.
• Parikh et al. Inferring semantic query relations from collective user behavior. CIKM 2008.
• Sundaresan et al. Scalable Stream Processing and Map Reduce. Hadoop World 2009.
• Parikh et al. Scalable and near real-time burst detection from eCommerce queries. KDD 2008.
• Pantel et al. Web-Scale Distributional Similarity and Entity Set Expansion. EMNLP 2009.
• Gyanit Singh, Nish Parikh, Neel Sundaresan. Query Suggestion at Scale with Hadoop. Hadoop Summit 2011.
• Nish Parikh. Mining Large-scale Temporal Dynamics with Hadoop. Hadoop Summit 2012.
• Uwe Mayer. Parallel and Distributed Computing, Data Mining and Machine Learning. EBay Shipping Recommendations over Hadoop. Hadoop Innovation Summit 2013.
• Nish Parikh, Gyanit Singh. Large scale user-interaction log analysis. ACM Data Mining SIG Bay Area Summit 2010.
• Halevy et al. The Unreasonable effectiveness of data. IEEE Intelligent Systems, 2009.
• Banko and Brill. Scaling to very very large corpora for natural language disambiguation. ACL 2001.
• Pilaszy and Tikk. Recommending new movies: even a few ratings are more valuable than metadata. RecSys 2009.
Acknowledgments

• Neel Sundaresan
• Evan Chiu
• Mohammad Al Hasan
• Karin Mauge
• Jack Shen
• Rifat Joyee
• Zhou Yang
• Hui Hong
• Long Hoang
• Narayanan Seshadri
Questions