Dealing with Structured and Unstructured Data at Facebook

Lars Backstrom
Agenda

1. Data at Facebook
2. Data Analyses
3. Application: People You May Know
   3a. Problem Description
   3b. Ranking System
   3c. Performance Summary
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data at Facebook</td>
</tr>
<tr>
<td>2</td>
<td>Data Analyses</td>
</tr>
<tr>
<td>3</td>
<td>Application: People You May Know</td>
</tr>
<tr>
<td>3a</td>
<td>Problem Description</td>
</tr>
<tr>
<td>3b</td>
<td>Ranking System</td>
</tr>
<tr>
<td>3c</td>
<td>Performance Summary</td>
</tr>
</tbody>
</table>
500 million 30-day active users
Over **3 billion** photos uploaded each month
~ 1.2TB/day March 2008
~ 12-18 TB/day March 2009
~ 30-36 TB/day Sept-Oct 2009
80-90 TB/day Sept 2010
Profiles (ca. 2008)

Holly Ann Calloway

About Me
Call me Miss Calloway, Room 2A. I'm a new kindergarten teacher who's just a minute of her job!
I enjoy trying new things, especially if they involve being outdoors. And Zoe!
I also like music, dancing, a great meal with wine and friends, and of course, snowboarding!

A few of my new year resolutions:
- climb Kilimanjaro
- learn a new language
- run a marathon

Check back with me to see how I'm doing with those. And tell me about yours!

Personal Information

About Me:
- Call me Miss Calloway, Room 2A.
- A new kindergarten teacher who's just a minute of her job!
- Enjoy trying new things, especially if they involve being outdoors.
- Zoe is important.
- Likes music, dancing, great meals with wine, friends, and snowboarding.

Activities:
- Climb Kilimanjaro
- Learn a new language
- Run a marathon

Interests:

Favorite Music:

Favorite TV Shows:

Favorite Movies:
- Juno, Love Actually, Before Sunrise, Breakfast at Tiffany's, Fight Club, Notting Hill, Varsity Blues, Remember the Titans, Titanic

Favorite Books:
- Lucky Jim, Anything by David Sedaris, The Kite Runner, Snowfall at Willow Lake, The Audacity of Hope

Favorite Quotes:
- A bird doesn't sing because it has an answer, it sings because it has a song.
  - Maya Angelou
- Never take life too seriously. Nobody gets out alive anyway.
- Dates are for having fun, and people should use them to get to know each other. Even boys have something to say if you listen long enough.
  - One of my students
- Tell your wife that she looks pretty, even if she looks like attract.
  - Another student
- Smile, it enhances your face value.

Unstructured free-text interests
Profiles (2011)

• General trend towards more structure, less text
• Everything (almost) on new profile is linked entity
Structured and Unstructured Data

- **Structured Data**
  - Typically relates one object to another in some way
    - Friendship edges relate two people
    - Photo tags related a person to a photo
    - Liking a page relates a person to the page
  - Can be thought of as one large heterogeneous graph

- **Unstructured Data**
  - Mostly text and photos
  - Hard to deal with
  - Goal is to annotate as much as possible
The Friendship graph

500M users each connect to an average of 130 other users = ~ 60 Billion Edges
Users connect to pages by *liking* them

Average user connected to 80 pages

\[(80*500M = 40B)\]
Each photo is connected to its owner, and all the people tagged in it.
Places

- Places are pages + location
- Check-ins link people to:
  - Place
  - Time

![Map of places with connections and photos](image)
Open Graph Pages

Allows users to socially interact with many webpages, create more structured data
More than just nodes and edges

- Most relationships are annotated in some way
  - Structured
    - Direction (friendships, hierarchies)
    - Creation time
  - Unstructured
    - Free-text (e.g. status with check-in)

- What sorts of things can we do with all this data?
Challenges

• Even with all this structure, much ambiguity remains
  • Pages on multiple sites refer to the same entity
    • E.g. IMDB, Rotten Tomatoes, FB Page
  • Multiple places for the same entity
    • Athens airport
  • Multiple meanings for the same phrase
Challenges

• Merging multiple nodes representing the same entity results in
  • More complete social context across the web
  • Instant cross site personalization
  • Metadata available comes from union of all nodes
• But, maintain high precision and don’t overmerge!
Why Bother?

• When social context is provided, social activity goes up
  • 2.5x increase in social activity
  • 4x increase in the number of likes

• Only 0.3% of the like buttons had faces previously
  • Social context increases engagement
  • Social advertisement results in significant increase in both CTR and purchase intent
Data Takeaways

• Facebook has lots of structured data.
  • Not just the friendship graph!
• We want more structured data.
  • Extraction/Mining
  • Higher Quality (Unambiguous)
• Next, we want to do stuff with structured data.
  • Improved ranking
  • New services
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data at Facebook</td>
</tr>
<tr>
<td>2</td>
<td>Data Analyses</td>
</tr>
<tr>
<td>3</td>
<td>Application: People You May Know</td>
</tr>
<tr>
<td>3a</td>
<td>Problem Description</td>
</tr>
<tr>
<td>3b</td>
<td>Ranking System</td>
</tr>
<tr>
<td>3c</td>
<td>Performance Summary</td>
</tr>
</tbody>
</table>
What are users talking about?

- Simple analysis of unstructured text
- Use automated techniques to extract interesting unigrams, bigrams
- Can easily find major events
- Swine Flu Scare
  - Started May ‘09
  - Actual cases in fall ‘09
What are users talking about?

- Simple analysis of unstructured text
- Use automated techniques to extract interesting unigrams, bigrams
- Can easily find major events
- Weekly trends
  - Most flu on Wednesday
  - Much less on weekends!
FML?

- FML – F*#% My Life
  - Big internet meme
  - Took off in early 2009
  - Mondays and Tuesdays suck
Enriched Status Updates

- We have some structure with each status update
  - Not just words, but time posted, user posting
  - Combining user information (country, language) gives us a richer picture
- Japanese Earthquake
  - Spiked first in Japan
- Took two hours for English speakers to take note

Languages on different scales
Unstructured Updates + Structured Data

- Extract specific words from user generated content
- Slice by age, region
- Can now find interesting trends
- Keyword association

Number of messages containing "Laid off"

Distribution of words related to "Vodka"
Vodka Map

Age

Percent Male
Page Like Graph (O’Connor)

- Use page like information to explore:
  - How concepts are organized
  - What space of interests does a user like?

- Useful for:
  - Page recommendations
  - Analytics
  - Ad targeting
Finding Related Pages

# Mutual Fans = |x ∩ y|

- Ira Glass
- Barack Obama: 64k of 94k
- The Office: 41k of 194k
- The Onion: 11k of 1.1M
- The Moth: 9179 of 34k
- Car Talk: 8448 of 116k
- Dexter: 8575 of 3.5M
- TED: 7058 of 519k
- This American Life: 64k of 220k
- Wait Wait Don't Tell Me: 64k of 249k
- PBS: 12k of 354k
- The Colbert Report: 5771 of 1.1M
- The New York Times: 9725 of 1.9M
- It's Always Sunny In Philadelphia: 6655 of 2.2M
- Michelle Obama: 5355 of 1.8M
- Flight of the Conchords: 8974 of 1.1M
- Lost: 8898 of 8.4M
- PostSecret: 7861 of 823k
- Apple Students: 6100 of 1.4M
- Weeds: 6325 of 1.8M
- House: 6351 of 12.3M
- Lady Gaga: 5309 of 15.6M
- Family Guy: 6025 of 15.2M
- The Wire: 4431 of 621k
- Human Rights Campaign: 5625 of 475k
- Anthony Bourdain: No Reservations
**Finding Related Pages**

# Mutual Fans = |x ∩ y|

Normalize for background popularity

\[ p(x | y) \]
“Israel” Page-Page Graph

- Relatedness between pages induces a page-page graph
“Israel” Page-Page Graph

- Relatedness between pages induces a page-page graph
Ranking

• Many users have too many content producing friends to reasonably show it all
  • FB must filter and rank all this content to generate Newsfeed

• Example: Ranking a place check-in
  
  Lars Backstrom was at Athens International Airport.
  Yesterday at 12:37am via iPhone • Like • Comment • Tag Friends

• Try to predict interestingness of checkin story
  • Depends on: relationship of me to actor, me to place, actor to place, details of place, etc.
Ranking

- Understand my relationship to actor based on various edge annotations (time created, number of interactions, etc.)
- How interesting is place historically?
  - Have previous check-ins to Athens airport elicited interest
- How far am I from place?
  - Today, not very interesting if a friend checks into a restaurant in Palo Alto
- Treat as classification problem to predict clicks, likes comments.
Feedback vs. Distance

![Graph showing feedback vs. distance on a logarithmic scale.](image)
Checkin Heatmaps

- Joining check-ins with other data and interesting patterns emerge
San Francisco Heatmap
San Francisco Heatmap (Male)

Gold's Gym

Lone Star Saloon
San Francisco Heatmap (Female)

Forever 21
Nordstrom
San Francisco Heatmap (Republican)

Scoma's
Bubba Gump's
San Francisco Heatmap (Democrat)

Glas Kat

SF street food fest
## Agenda

<table>
<thead>
<tr>
<th></th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data at Facebook</td>
</tr>
<tr>
<td>2</td>
<td>Data Analyses</td>
</tr>
<tr>
<td>3</td>
<td>Application: People You May Know</td>
</tr>
<tr>
<td>3a</td>
<td>Problem Description</td>
</tr>
<tr>
<td>3b</td>
<td>Ranking System</td>
</tr>
<tr>
<td>3c</td>
<td>Performance Summary</td>
</tr>
</tbody>
</table>
People you may know

- Top 1-2 suggestions shown on homepage of Facebook
  - See all link leads to more suggestions
- Many more friend adds from home than ‘see all’ page.
- ‘Xing’ a user removes that person from list permanently
  - Pulls in next suggestion
- Accounts for a significant chunk of all friending on Facebook
Helping people find friends on FB

- Recommendation has proven itself in many contexts
  - Amazon, Netflix, etc. all have sophisticated systems

- Like them, we can increase value to users by making good suggestions
  - People with more friends use the site more, get more out of it

- Unlike those systems (collaborative filtering) our’s must take social context into account
How to make suggestions

- Most friendships go to friends-of-friends
  - Previous work shows over 5x more friendships to FoFs (2-hops) than 3+ hop users (Lescovec et. al ‘08)
  - 92% of new friendships on FB
- From a practical point of view, doing more than FoF is impossible
  - Average user has over 130 friends
    - $130 \times 130 = 17K$ FoFs
    - $130^3 = 2.2M$ FoFoFs
  - Power users have up to 5K friends
How to make suggestions

- Most friendships go to friends-of-friends
  - Previous work shows over 5x more friendships to FoFs (2-hops) than 3+ hop users (Lescovec et. al ‘08)
  - 92% of new friendships on FB
- From a practical point of view, doing more than FoF is impossible
  - Average user has over 130 friends
    - $130 \times 130 = 17\text{K FoFs}$
    - $130^3 = 2.2\text{M FoFoFs}$
  - Power users have up to 5K friends
How to make suggestions

- Most friendships go to friends-of-friends
  - Previous work shows over 5x more friendships to FoFs (2-hops) than 3+ hop users (Lescovec et. al ‘08)
  - 92% of new friendships on FB
- From a practical point of view, doing more than FoF is impossible
  - Average user has over 130 friends
    - $130 \times 130 = 17K$ FoFs
    - $130^3 = 2.2M$ FoFoFs
  - Power users have up to 5K friends

![Friendship by Number of Hops](image)
Suggesting Friends of Friends

- Problem Statement:
  - Given a source user, find the best FoFs to suggest

- Challenges:
  - A typical user has tens of thousands of FoFs (about 40K on average, 99th percentile 800K!)
    - What features will help us pick from these
    - How can we combine network and demographic features
Friends in Common

- Number of friends in common is a good start
  - Two people are 12x more likely to become friends with 10 mutual friends than 1
- Other social network features are also helpful
  - For example, if your good friend just made a new friend, that is a good suggestion
- We can combine network properties:
  - $\delta_{u,v}$ gives the time since edge creation

$$v(fof) = \sum_{f_i} \frac{(\delta_{u,f_i} \cdot \delta_{f_i,fof})^{-0.3}}{\sqrt{\text{friends}_{f_i}}}$$
System Overview

- System examines all FoFs
  - Generates list of top 100 candidates

- Scores are stored and used along with cheaply available data to predict real-time CTRs
  - Candidates are re-ranked and displayed on each impression

- Results are fed back into system for retraining
  - Real-time model depends on input scores, is trained online to keep CTR-predictions accurate

```
Lars

FoF Discovery and Feature Generation

Lars, Greg: Mutual Friends = 10, Age(Lars) = 27, ...

Bagged Decision Trees

Score(Lars, Greg) = 0.045
Score(Lars, Shelly) = 0.021

Real-Time CTR Prediction

CTR(L, G) = 0.012

Impressions(Lars, Greg) = 3
Impressions(Lars, Shelly) = 2
```

Outcome
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data at Facebook</td>
</tr>
<tr>
<td>2</td>
<td>Data Analyses</td>
</tr>
<tr>
<td>3</td>
<td>Application: People You May Know</td>
</tr>
<tr>
<td>3a</td>
<td>Problem Description</td>
</tr>
<tr>
<td>3b</td>
<td>Ranking System</td>
</tr>
<tr>
<td>3c</td>
<td>Performance Summary</td>
</tr>
</tbody>
</table>
Making Static Predictions

- Use traditional machine learning
  - For a user $u$, consider all FoFs $w_1, ..., w_k$
  - For each pair $(u, w_i)$ generate a bunch of features
    - Mutual friends, time discounted mutual friends, new mutual friends, etc.
    - Also incorporate features of just $u$ and $w_i$
      - Age, gender, country, total friends, time on FB, etc.
  - We use bagged decision trees (the average of many decision trees)
    - Training data comes from past PYMK
    - Only train on ‘first impressions’ on /home.php

Feature Vector
Mutual Friends = 5
Gender$_u$ = M
Age$_{wi}$ = 23
...
Friend of Friend Features

- Two types of features
  - Weighted Mutual Friends
    - Actual MFs, Pending MFs, Time Weighted MFs, Strength Weighted MFs, Directed MFs
  - Demographic features
    - Age, country, Facebook age, gender, friend count, etc.
    - Because average person has 40K FoFs, these must be local, and hence are not sharded, but are duplicated on every machine.

- Most important features for prediction
  1. Time discounted mutual friends: \[ v(f_{of}) = \sum_{f_i} \frac{(\delta_{u,f_{i}} \cdot \delta_{f_{i},f_{of}})^{-0.3}}{\sqrt{\text{friends}_{f_i}}} \]
  2. Country and Facebook age of source user
  3. Number of friends
Showing the best suggestion every time

- To optimize the suggestions, we re-rank after every impression
  - Decision models can only be run once per 2 days
    - They output a score for each \((u, w_i)\) pair
  - Can’t do much too much computation for each impression, but can do a little
    - Simple features are available at each impression, for each suggestion
      - \(\text{score}(u, w_i)\), number of impressions for \((u, w_i)\), friend count\((u)\), friend count\((w_i)\)

Combine what is available with score to re-rank via Logistic Regression

<table>
<thead>
<tr>
<th>Suggestion</th>
<th>Impressions</th>
<th>CTR Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>David</td>
<td>1</td>
<td>0.012</td>
</tr>
<tr>
<td>Alice</td>
<td>3</td>
<td>0.011</td>
</tr>
<tr>
<td>Bob</td>
<td>2</td>
<td>0.010</td>
</tr>
<tr>
<td>Carol</td>
<td>2</td>
<td>0.009</td>
</tr>
</tbody>
</table>
Reranking with logistic regression

- Simple logistic regression model does well at predicting CTRs
  - Given features for a suggestion $F_1, F_2, F_3, \ldots$
  - Predicted CTR = logistic($C_0 + C_1 \times F_1 + C_2 \times F_2 + \ldots$)
- Improves quality on two fronts
  - Don’t repeat a suggestion over and over – show the best suggestion given history
  - If a user never interacts with PYMK, just give up
Reranking with logistic regression

- Most important features have to do with offline score and user’s PYMK history
  - What score did the decision trees give?
  - How many friends has the user added through PYMK in the last week
  - How many has she rejected?
  - How many suggestions did we make?
  - How many times have we shown her each suggestion?

- Simple to implement
  - Using user history data to personalize gives HUGE improvements!
Putting it all together

Lars

FoF Discovery and Feature Generation

Lars, Greg: Mutual Friends = 10, Age(Lars) = 27, ...

Bagged Decision Trees

Score(Lars, Greg) = 0.045
Score(Lars, Shelly) = 0.021

Cache-only memcache

Impressions(Lars, Greg) = 3
Impressions(Lars, Shelly) = 2

Real-Time CTR Prediction

CTR(L, G) = 0.012 ...

CTR > threshold?

Outcome

Outcome
Agenda

1. Data at Facebook
2. Data Analyses
3. Application: People You May Know
   3a. Problem Description
   3b. Ranking System
   3c. Performance Summary
Performance

- Two performance metrics
  - Friendships created
  - Click-through Rate
- Can always increase one at the cost of the other

- Initial launch of offline model and CTR prediction in early March
  - A few hiccups since then ☹️
  - Overall, increase in total adds up 60%
  - At the same time, CTR prediction has cut impressions have been cut by 1/3
  - Hence, CTR is up by 130%
Summary

- Direction at FB is to add more structure everywhere
  - Annotations on relationships, objects makes them richer and creates opportunities for new applications
  - Services (Search, News Feed, PYMK, etc.) can all use this structured data to improve quality

- Major hurdles still remain
  - Disambiguation, unification of entities across the web, inside Facebook
  - Maintaining authenticity, fighting spam
  - Scale (petabytes) introduces lots of systems challenges
Questions