Web-Scale Recommendation Engines

CS130 Fall 2012

Andrew Look
Senior Software Engineer
Shopzilla
About Shopzilla

● Comparison Shopping Engine
● Lots of traffic
  ○ > 200 million products
  ○ > 100 million impressions / day
  ○ > 8000 searches / second
  ○ > 30M unique visitors / month
● Operate web properties in US, Europe
  ○ Shopzilla.com
  ○ Bizrate.com
  ○ Beso.com
Beso.com: Personalization

- Focused on personalization for users
- Users can pick favorite:
  - Brands
  - Stores
  - Products
What's missing?

- Recommendations
- Relevancy
- ...but we have data!
Recommender: System Overview
Extracting User Preferences

- Shopzilla Engineers built this component
- Important in tuning recommender algorithm
Extracting User Preferences

--which brands did each user favorite?
SELECT user_id, brand_id
FROM favorites

--which brands did they click on?
SELECT C.user_id, P.brand_id
FROM clicks C, products P
WHERE C.product_id = P.id
Recommendations: Overview

- I like a brand. Which other brands do I like?
- Use latest available user data to answer this
Recommender: What will you learn?

- Leverage **Machine Learning** at scale
- Use **Hadoop** on **BigData**
- How to evaluate solutions to data problems

Ex. [Recommendations With Hadoop Streaming](#)
Recommender: Why is it cool?

- Run recommender on cluster of 332 CPU's
- Patterns emerge from behavioral data
  - Nike ↔ Adidas, Reebok
  - Prada ↔ Gucci, Louis Vuitton
  - Levi ↔ Wrangler, Lee'
- Data from 100's of millions of users
- Creative ways to represent user preferences
Web Service: Overview

- Load recs into NoSQL cache
- Build web service for recs
- Build client for Beso.com
Web Service: What will you learn?

● Build a high-performance web service
  ○ 100's of requests per second
● Learn enterprise Java best practices
● Understand Service-Oriented Architecture
Web Service: Why is it cool?

● Scaling
  ○ how many servers do we need?
  ○ how fast does cache need to be?
  ○ how much storage do we need?

● NoSQL Technology selection
  ○ read/write performance
  ○ reliability
  ○ availability
  ○ consistency

● See evolving data affect web layer
Evaluation Tool: Overview

- Build UI to compare algorithms side-by-side
Evaluation Tool: What will you learn?

- Build enterprise Java webapp from scratch
- Use frontend technologies (CSS/JS/AJAX)
- Consuming web services using REST client
Evaluation Tool: Why is it cool?

- Like CS144, with modern Java webapp tools
- Visualizing data problems
- Inform recommendation tuning process
Logistics

- Pre-configured VM to get you started
- Knowledgeable project mentors
  - available for questions / help daily
- Agile Tracking system
- Github / Continuous Integration build
Schedule

- Week 1: bootcamp
- Week 2: technology selection
- Week 3: interface definition
- Week 4: mock implementation
- Week 5: component buildout
- Week 6: component integration
- Week 7: hardening, performance
- Week 8: deployment
Big Picture

- Technology selection & platform rollout
- Mirrors process used by Shopzilla's Senior Architects and Engineers
- Program designed to teach enterprise engineering techniques
- Equips you with skills for your first job
Questions?

alook [at] shopzilla [dot] com
@andrewlook