KDD'10 Tutorial: Recommender Problems for Web Applications

Deepak Agarwal and Bee-Chung Chen
Yahoo! Research

Agenda

- Focus:
  - Recommender problems for **dynamic**, **time-sensitive** applications
    - Content Optimization
- Introduction (20 min, Deepak)
  - Content optimization, match-making, example applications
- Offline components (40 min, Deepak)
  - Collaborative filtering (CF), methods for cold-start
- Online components + initialization (70 min, Bee-Chung)
  - Time-series, online/incremental methods, explore/exploit (bandit)
- Evaluation methods (15 min, Deepak)
- Challenges (5 min, Deepak)

Content Optimization

- Goal
  - Effectively and “pro-actively” learn from user interactions with content that are displayed to maximize our objectives.
- A new scientific discipline at the interface of
  - Large scale Machine Learning & Statistics
    - Offline Models
    - Online Models
    - Collaborative Filtering
    - Explore/Exploit
  - Multi-Objective Optimization in the presence of Uncertainty
    - Click-rates (CTR), Engagement,….
  - User Understanding
    - Profile construction
  - Content Understanding
    - Topics, “aboutness”, entities, follow-up of something, breaking news,…

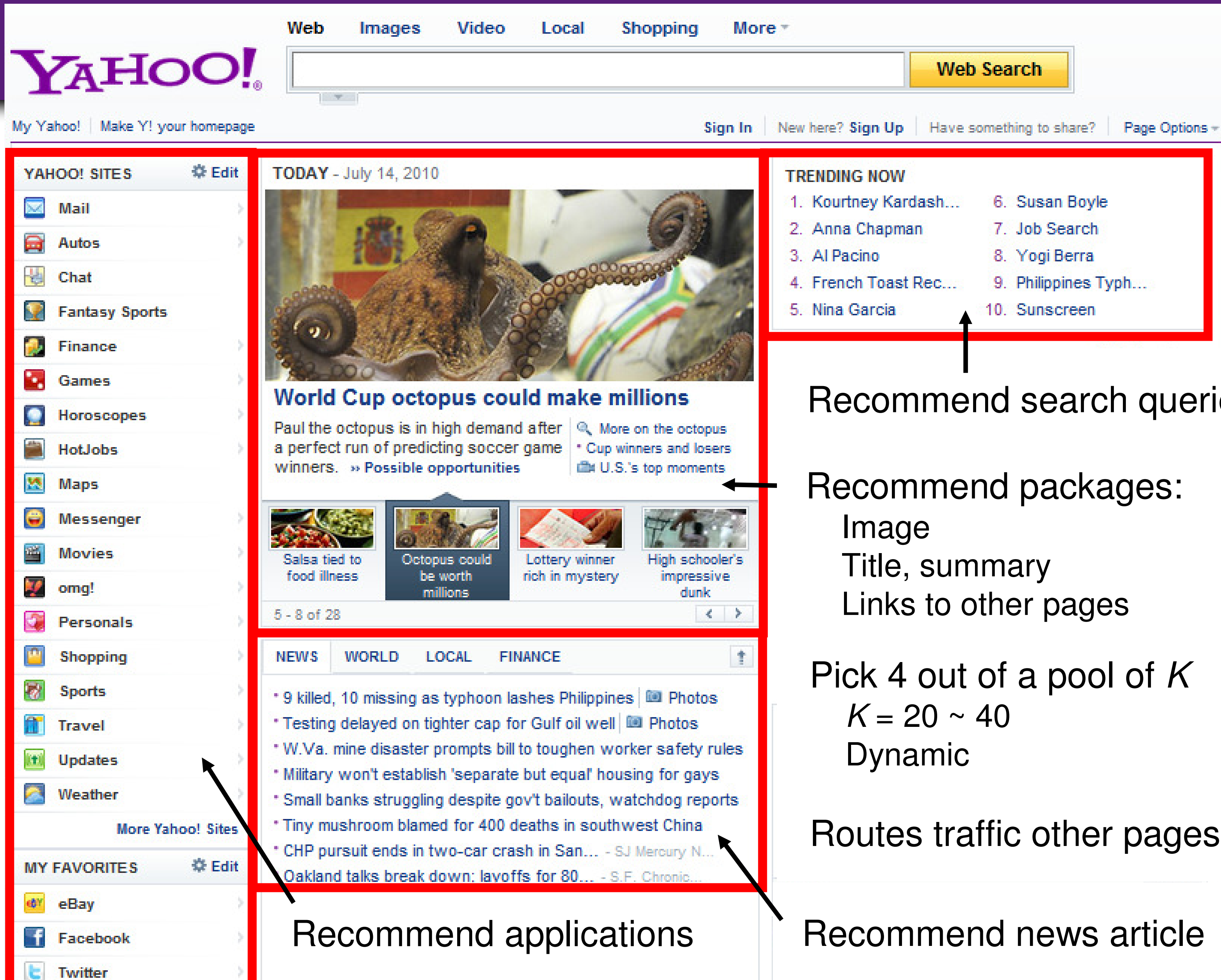
Content Optimization: High level flowchart

- Flow
  - Understand content (Offline)
  - Serve content to optimize our objectives (Online)
  - quickly learn from feedback obtained using ML/Statistics (Offline + Online)
  - Constantly enhance our content inventory to improve future performance (Offline)
  - Constantly enhance our user understanding to improve future performance (Offline + Online)

  - Iterate

Some examples

- Simple version
  - I have an important module on my page, content inventory is obtained from a third party source which is further refined through editorial oversight. Can I algorithmically recommend content on this module? I want to drive up total CTR on this module
- More advanced
  - I got X% lift in CTR. But I have additional information on other downstream utilities (e.g. dwell time). Can I increase downstream utility without losing too many clicks?
- Highly advanced
  - There are multiple modules running on my website. How do I take a holistic approach and perform a simultaneous optimization?



Recommend search queries

Recommend packages:
- Image
- Title, summary
- Links to other pages

Pick 4 out of a pool of K
- $K = 20 \sim 40$
- Dynamic

Routes traffic other pages

Recommend applications

Recommend news article

Problems in this example

- Optimize CTR on different modules together in a holistic way
  - Today Module, Trending Now, Personal Assistant, News, Ads
  - Treat them as independent?

- For a given module
  - Optimize some combination of CTR, downstream engagement and perhaps revenue.

Single module CTR optimization problem

- Display “best” articles for each user visit
- Best - Maximize User Satisfaction, Engagement
  - BUT Hard to obtain quick feedback to measure these

- Approximation
  - Maximize utility based on immediate feedback (click rate) subject to constraints (relevance, freshness, diversity)
- Inventory of articles?
  - Created by human editors
  - Small pool (30-50 articles) but refreshes periodically

Recommendation: A Match-making Problem

- Recommendation problems
- Search: Web, Vertical
- Online advertising
- …

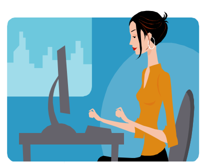
Item Inventory
Articles, web page, ads, …

Use an automated algorithm to select item(s) to show

Get feedback (click, time spent,..)
Refine the models

Repeat (large number of times)
Measure metric(s) of interest
(Total clicks, Total revenue,…)

Opportunity
Users, queries, pages, …



Important Factors affecting solution in Match-making Problems

- **Items:** Articles, web pages, ads, modules, queries, users, updates, etc.
- **Opportunities:** Users, query keywords, pages, etc.
- **Metric** (e.g., editorial score, CTR, revenue, engagement)
  - Currently, most applications are single-objective
  - May be multi-objective optimization (maximize *X* subject to *Y*, *Z*,..)
- **Properties of the item pool**
  - Size (e.g., all web pages vs. 40 stories)
  - Quality of the pool (e.g., anything vs. editorially selected)
  - Lifetime (e.g., mostly old items vs. mostly new items)

Factors affecting Solution continued

- **Properties of the opportunities**
  - Pull: Specified by explicit, user-driven query (e.g., keywords, a form)
  - Push: Specified by implicit context (e.g., a page, a user, a session)
  - Size (e.g., user base); continuity (e.g., session vs. single event)
- **Properties of the feedback on the matches made**
  - Types and semantics of feedback (e.g., click, vote)
  - Latency (e.g., available in 5 minutes vs. 1 day)
  - Volume (e.g., 100K per day vs. 300M per day)
- **Constraints specifying legitimate matches** (e.g., business rules)
- **Available Metadata** (e.g., link graph, various user/item attributes)

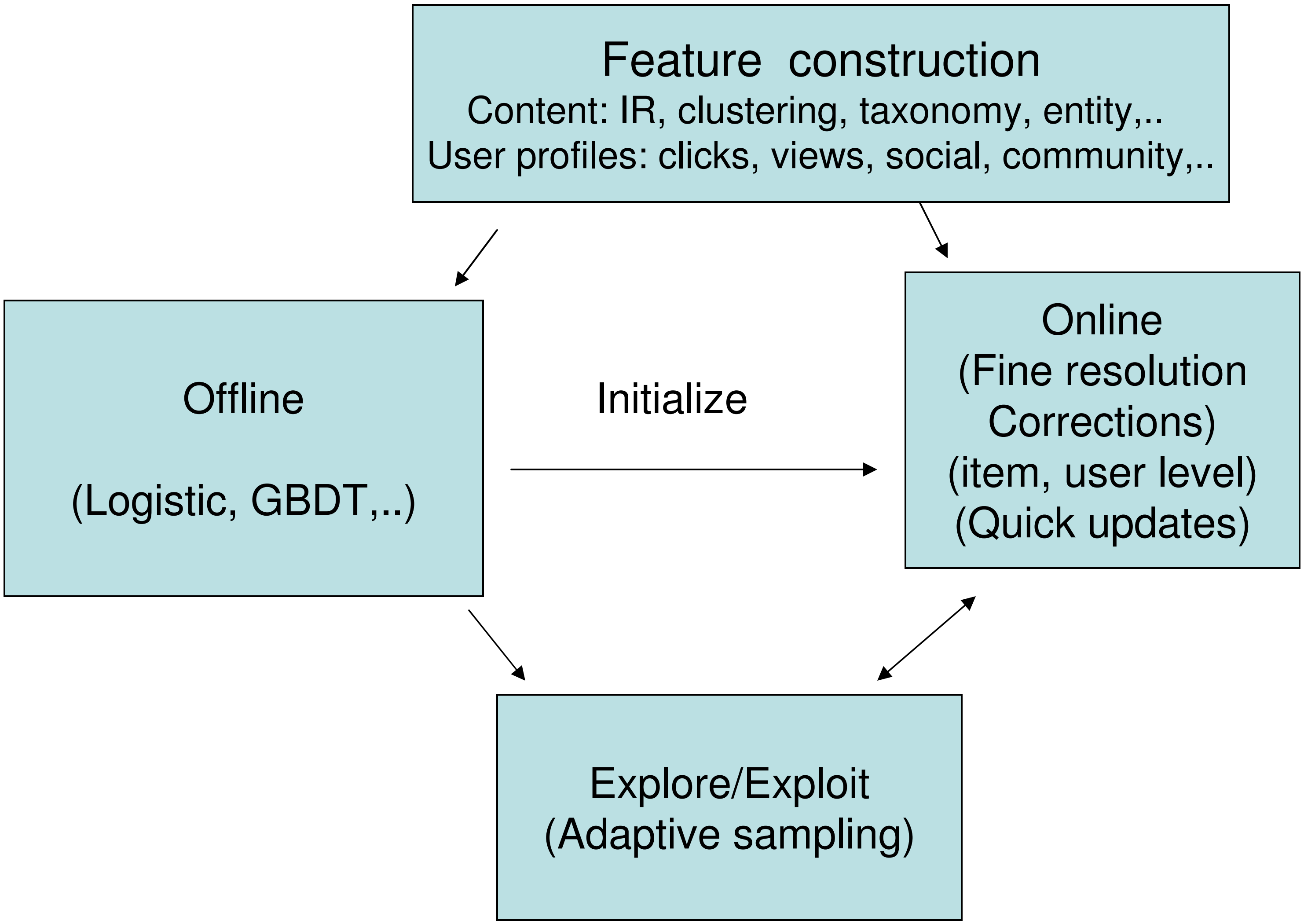
Recommendation vs. Other Match-Making Problems

	Recommendation	Search	Advertising
Main Metric	User engagement	Relevance to the query	Revenue
Items	Anything (except for ads)	Anything (except for ads)	Ads
Opportunities	Push (implicit) The system guesses users info needs	Pull (explicit) Users specify their info needs	Push
Examples	Recommend articles, friends, feeds to users Recommend related items given an item	Web search Vertical search	Sponsored search Content match Behavior targeting Display advertising (non-guaranteed)

More on Recommendation vs. Search

- Recommendation
  - User intent: See something “interesting” (browse mode, implicit)
    - The system tries to guess what a user likes on an entity/topic page
  - No query reformulation (unless we suggest related topics/entities)
  - False +ve more costly than false –ve
    - Showing a bad article is a worse than missing a good one
- Search
  - User intent: Explicit, users express what they want
  - Users can reformulate queries
  - False –ve more costly (but depends on the query)
    - Users want to get the results they are looking for

Modeling: Key components



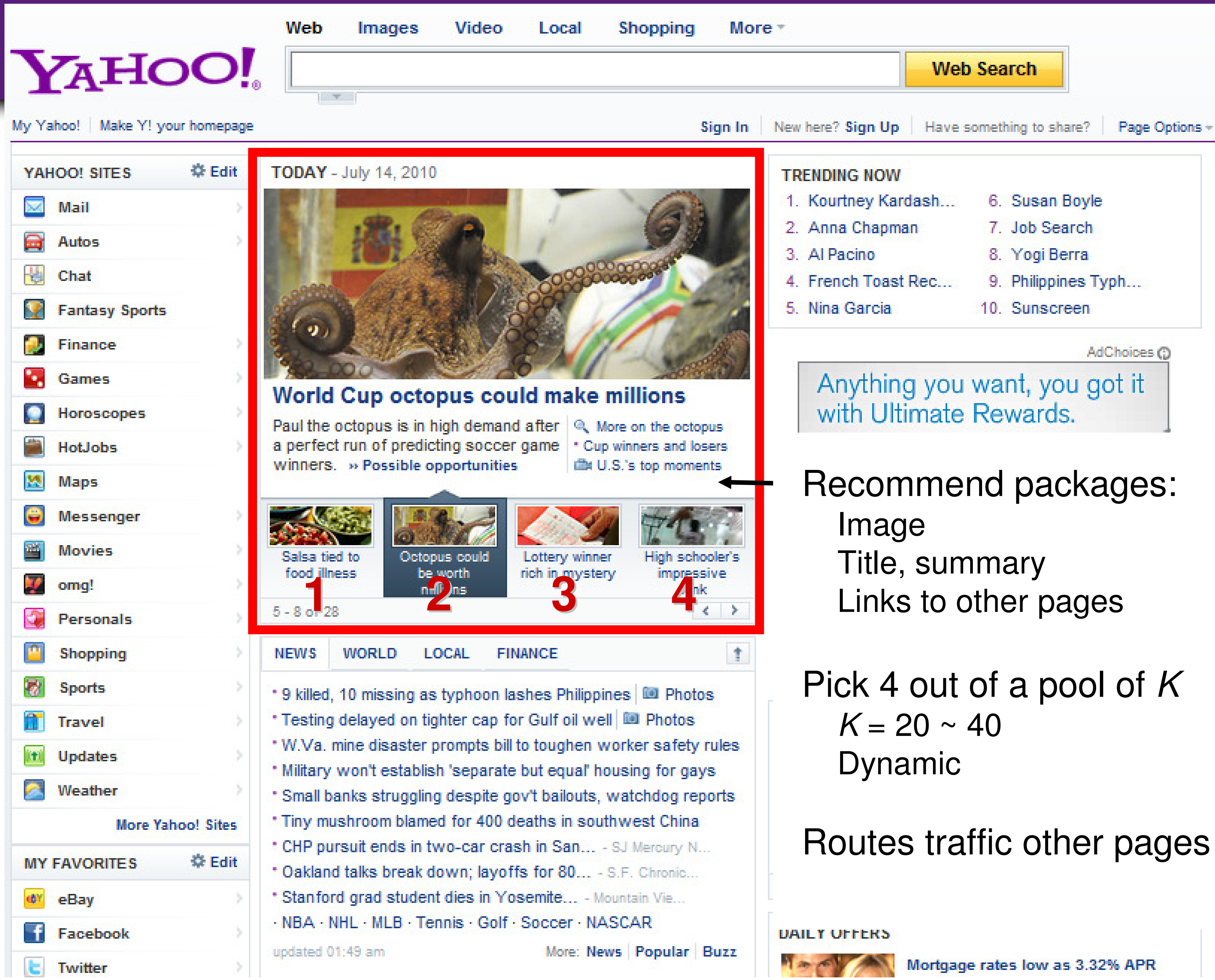
Modeling Problems that has received attention

- Univariate response (e.g. click); single objective (e.g. maximize CTR)

- Our solution
  - Initialize online through offline models
  - Learn "corrections" to offline models at very granular levels (user, item) and learn rapidly in an online fashion
  - Online correction models have reduced dimension through clever representations of parameters and by exploiting the fallback mechanism to coarser models
  - The models are tightly coupled with Explore-exploit to ensure fast convergence to areas of high valued response

Example Application:
Today Module on Yahoo! Homepage

Currently in production powered by some methods discussed in this tutorial



Recommend packages:
- Image
- Title, summary
- Links to other pages

Pick 4 out of a pool of K
- $K = 20 \sim 40$
- Dynamic

Routes traffic other pages

Problem definition

- Display “best” articles for each user visit
- Best - Maximize User Satisfaction, Engagement
  - BUT Hard to obtain quick feedback to measure these
- Approximation
  - Maximize utility based on immediate feedback (click rate) subject to constraints (relevance, freshness, diversity)
- Inventory of articles?
  - Created by human editors
  - Small pool (30-50 articles) but refreshes periodically

Where are we today?

- Before this research
  - Articles created and selected for display by editors
- After this research
  - Article placement done through statistical models
- How successful ?

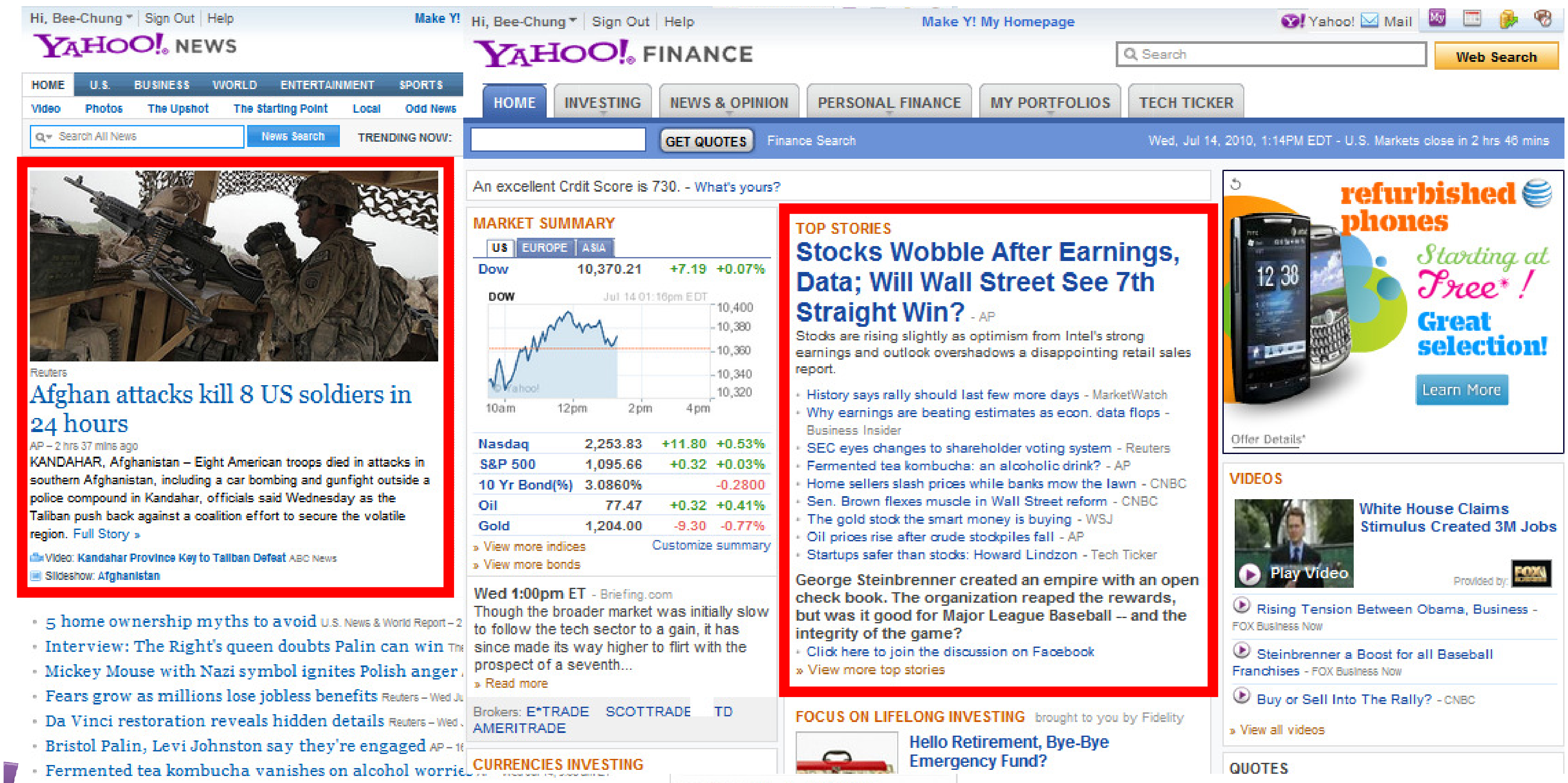
"Just look at our homepage, for example. Since we began pairing our content optimization technology with editorial expertise, we've seen click-through rates in the Today module more than double. ----- Carol Bartz, CEO Yahoo! Inc (Q4, 2009)

Main Goals

- Methods to select most popular articles
  – This was done by editors before
- Provide personalized article selection
  – Based on user covariates
  – Based on per user behavior
- Scalability: Methods to generalize in small traffic scenarios
  – Today module part of most Y! portals around the world
  – Also syndicated to sources like Y! Mail, Y! IM etc

Similar applications

- Goal: Use same methods for selecting most popular, personalization across different applications at Y!
- Good news! Methods generalize, already in use



Next few hours

	Most Popular Recommendation	Personalized Recommendation
Offline Models		Collaborative filtering (cold-start problem)
Online Models	Time-series models	Incremental CF, online regression
Intelligent Initialization	Prior estimation	Prior estimation, dimension reduction
Explore/Exploit	Multi-armed bandits	Bandits with covariates