Social Search
Web 2.0

• Web 1.0: the classical notion of the Web which consists of non-interactive HTML documents

• Web 2.0 sites allow people with similar interests to interact with each other in various ways
  – Social media sites such as Digg, Twitter, Flickr, YouTube, Del.icio.us, and CiteULike
  – Social networking sites: MySpace, Facebook, LinkedIn

• Classroom discussion: What is the most significant feature of Web 2.0 web sites and applications?
  – Website holders only provide a framework, (a large number of) users are the real creators of the content
Social Search

• Search within a social environment
  – User interaction data can be searched and can help search

• A community of users actively participate in the search process
  – Defining individual user profiles and interests
  – Interacting with other users
  – Modifying the representations of the objects being searched
Selected Topics (1)

- **User tags**
  - A form of manual indexing
  - Searching for items using tags
  - Tag suggestion
  - Tag cluster visualization

- **Searching within communities**
  - Online community detection and tracking
  - Community-based question answering
Selected Topics (2)

• Filtering and recommendation systems
  – A fixed query representing a long-term information need, such as “tropical storms”
  – A system should alert the user when a new story matches the user’s interest

• Peer-to-peer (P2P) search
  – Query a set of “nodes” (e.g., individuals, organizations, or search engines) for an information need

• Metasearch
  – Running queries against a number of search engines and collecting and merging the results
  – Obtain better coverage and accuracy than one single search engine
Manual / Automatic Indexing

• Manual indexing: labeling objects manually into categories
  – Used extensively in libraries
  – Useful for social media objects

• Automatic indexing: using computers to assign identifiers (terms, phrases and features) to documents
  – Exhaustive – every word is indexed
  – Consistent – no bias in labeling
Manual Indexing of Social Media

• In libraries, experts conduct manual indexing using a controlled vocabulary (a fixed ontology) – more or less standardized

• Social media tagging is done by end users with little-to-no quality control
  – No fixed vocabulary

• Folksonomies: user generated ontologies or taxonomies – dynamic, community influenced ontology
Semantic Web

- Goal: to semantically tag Web content so that it becomes possible to find, organize and share information on the Web more easily
  - A standardized, fixed ontology of metadata tags must be developed and used consistently
- Users are generally more open to tagging data with a relatively unrestricted set of tags that are meaningful to them and reflect the specific context of the application
  - A fast growing number of social media sites based on flexible, user-driven folksonomies
  - Only a small number of semantic web sites based on rigid, predefined ontologies
Categories of Tags

- Content-based tags – describing the content of an item, e.g., “car”, “woman”, “sky”
- Context-based tags – describing the context of an item, e.g., “New York City”
- Attribute tags – describing implicit attributes of an item, e.g., “black and white”, “homepage”
- Subjective tags – describing an item subjectively, e.g., “pretty”, “amazing”, “awesome”
- Organizational tags – helping to organize items, e.g., “to-do-list”, “my pictures”
Role of Tags

• User tagging in Web 2.0 sites is not to organize all information into tidy categories
  – Subjective tags and folksonomies
• User tagging adds value to the huge amount of data that is already there
• In Flickr, more than 85% of the photos on its service have human-added tags
Searching Tags

• Task: searching tagged items
  – A textual or non-textual item may be labeled by multiple tags collaboratively – a textual dimension
  – A few critical difference from traditional text search – challenges in tag search
• The vocabulary mismatch problem
  – Tags are very sparse representations of very complex items
  – Query “fish bowl”
  – “fish AND bowl” may have a high precision but a low recall
  – “fish OR bowl” may have a high recall by a low precision
  – Boolean search cannot find a picture of fish bowl labeled by tag “aquariums”
Using Pseudo Relevance Feedbacks

- Query “tropical fish”
- Using the snippets of highly ranked results to derive the distribution of related terms
- Using the related terms to expand the query to improve both precision and recall

Age of Aquariums - Tropical Fish
Huge educational aquarium site for tropical fish hobbyists, promoting responsible fish keeping internationally since 1997.

The Krib (Aquaria and Tropical Fish)
This site contains information about tropical fish aquariums, including archived usenet postings and e-mail discussions, along with new ...

Keeping Tropical Fish and Goldfish in Aquariums, Fish Bowls, and ...
Keeping Tropical Fish and Goldfish in Aquariums, Fish Bowls, and Ponds at AquariumFish.net.
Quality Boosting in Tagging

- Tags may be very noisy and may contain spam
- Providing incentives to users to enter quality tags
  - Allowing users to report spam or inappropriate tags
  - Upgrading users if they provide some high quality tags
Missing Tags

- Many items may not be tagged – those items are invisible to any tag-based search engines
- Automatically infer tags for those items to improve recall
- If the items are textual such as books, news articles and research papers, we can choose $k$ terms with the highest weight as the inferred tags

\[
weight(w) = \log(tf(w, d) + 1) \log \frac{N}{df(w)}
\]

- $N$ is the total number items in the collection
Inferring Tags by Classification

• For each tag, train a classifier
  – Given an item, a classifier predicts whether the associated tag should be applied to the item
  – All the labeled items can be treated as the training data set

• For an unlabeled item, apply classifiers of all tags and choose the top-k tags with the highest classification confidence as the inferred tags
Redundant Tags

- A picture of children may have the tags “child”, “children”, “kid”, “kids”, “boy”, “boys”, …
  - They are relevant, but redundant
  - Drawback of the methods treating each tag independently
- The novelty problem – choose a set of tags that are both relevant and non-redundant
- Maximal Marginal Relevance (MMR) – given an item o and the current set T of tags for the item, choose the item of the largest MMR value as the next tag
  \[
  \text{MMR}(t, T) = \lambda \text{Sim}(t, o) - (1 - \lambda) \max_{t' \in T} \text{TagSim}(t, t')
  \]
  - Function Sim measures the similarity between a tag t and item o
  - Function TagSim measures the similarity between two tags
  - Tuning \( \lambda \) for balance between relevance (\( \lambda = 1 \)) and novelty (\( \lambda = 0 \))
Browsing Using Tags

• A user can browse through the collection of items by following a chain of tags –
  – When a user is viewing an item, all the item’s tag can be displayed
  – The user may then click on one of the tags and be shown a list of items that also have the tag

• A focused approach
Tag Clouds

- View the most popular tags

![Web 2.0 Tag Cloud](http://upload.wikimedia.org/wikipedia/commons/a/a7/Web_2.0_Map.svg)
Types of Tag Clouds

- Type I (displaying all tags for an item): size represents the number of times that tag has been applied to a single item
- Type II (displaying all tags in a collection of items): size represents the number of items to which a tag has been applied, as a presentation of each tag's popularity
- Type III (displaying item categories of various cardinality): tags are used as a categorization method for content items – tags are represented in a cloud where larger tags represent the quantity of content items in that category
Online Community

- By analyzing tags that users submit/search for, it is possible to find groups of users with related interests
  - Hockey fans may tag pictures of favorite players and web pages
- Online community: a group of entities that interact in an online environment and share common goals, traits or interests
  - Entities may be a mix of users, organizations, web pages and any other meaningful items
  - Example: Web community – a set of web pages that are all on the same topic
- Membership in online communities is often implicit
- One entity may be a member of multiple online communities
Communities in a Graph

• Each entity is a node
• Interactions or relationships between entities are modeled by edges – can be directed or undirected
• Communities in a graph
  – The set of entities must be similar to each other according to some similarity measure
  – The set of entities should interact with each other more than they do with other entities
Finding Communities in a Graph

- Identify candidate entities – a subset of entities that may possibly be members of the community
  - To find a community of hockey, we find all users that are interested in hockey, or search using “hockey” and find all pages match the query
- Use HITS to find the core of the community
  - For each candidate, HITS computes an authority score and a hub score
  - Rank entities in authority score – the entities with high authority scores are leaders and form the core of the community
Clustering Approach

• Represent each node as a vector of \(|V|\) components
  – If \(v_i \rightarrow v_j\) then \(v_i[j] = 1\) otherwise 0
• Using Euclidean distance to measure similarity
• A clustering algorithm can be applied – a cluster may be a community
What Is Clustering?

• Group data into clusters
  – Similar to one another within the same cluster
  – Dissimilar to the objects in other clusters
  – Unsupervised learning: no predefined classes
Requirements of Clustering

• Scalability
• Ability to deal with various types of attributes
• Discovery of clusters with arbitrary shape
• Minimal requirements for domain knowledge to determine input parameters
• Can deal with noise and outliers
• Insensitive to the order of input records
• Can handle high dimensionality
• Incorporation of user-specified constraints
• Interpretability and usability
Partitioning Algorithms: Basics

- Partition n objects into k clusters
  - Optimize the chosen partitioning criterion
- Global optimal: examine all possible partitions
  - \((k^n-(k-1)^n-\ldots-1)\) possible partitions, too expensive!
- Heuristic methods: k-means and k-medoids
  - K-means: a cluster is represented by the center
  - K-medoids or PAM (partition around medoids): each cluster is represented by one of the objects in the cluster
K-means

- Arbitrarily choose k objects as the initial cluster centers
- Until no change, do
  - (Re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster
  - Update the cluster means, i.e., calculate the mean value of the objects for each cluster
K-Means: Example

K=2
Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

reassign

reassign

Update the cluster means
Hierarchical Clustering

- Group data objects into a tree of clusters

Diagram:

- Agglomerative (AGNES)
- Divisive (DIANA)

Steps:

- Step 0
- Step 1
- Step 2
- Step 3
- Step 4
AGNES (Agglomerative Nesting)

- Initially, each object is a cluster
- Step-by-step cluster merging, until all objects form a cluster
  - Single-link approach
  - Each cluster is represented by all of the objects in the cluster
  - The similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters
Dendrogram

- Show how to merge clusters hierarchically
- Decompose data objects into a multi-level nested partitioning (a tree of clusters)
- A clustering of the data objects: cutting the dendrogram at the desired level
  – Each connected component forms a cluster
Evaluation of Community Finding

• Evaluating the effectiveness of community finding algorithms is very difficult
  – Hard to determine whether or not an entity should be part of a given community
  – Even judgments made by human being may disagree with each other due to the vague definition of a community

• How to use communities found?
  – When a user issues a query, results can be personalized according to the communities that the user belongs to
  – Enhanced browsing, identifying experts, web site recommendation, …
Community-based Question Answering

• An information need cannot be answered using conventional search engines if no document in the collection satisfies the information need
  – Example: what is the potential interactions between a medicine and an herbal tea

• Documents answering an information need can be created by asking experts in a community
  – A user can ask a large group of other users, some of them may be pharmacists or herbal experts

• Community-based question answering – a person submits a question to a community consisting of both experts and non-experts in a wide range of topics, each of whom can opt to answer the question
  – Example: Yahoo!Answers
Pros and Cons

• Pros
  – May get answers to complex or obscure information needs
  – May see multiple, possibly different opinions about a topic
  – May interact with other users who may share common interests, problems and goals

• Cons
  – Possibly no answer at all
  – Waiting for a few days for an answer
  – Incorrect, misleading, offensive or spam answers

• “garbage in, garbage out”
  – May users may ask low quality questions
  – Many answers are of low quality
Examples of Questions

- **Straightforward question**
  - What part of Mexico gets the most tropical storms?
  - How do you pronounce the French words, 'coeur' and 'miel'?
  - GED test?
  - Why do I have to pay this fine?
  - What is Schrödinger's cat?
  - What's this song?
  - Hi...can u ppl tell me sumthing abt death dreams??
  - What are the engagement and wedding traditions in Egypt?
  - Fun things to do in LA?
  - What lessons from the Tao Te Ching do you apply to your everyday life?
  - Foci of a hyperbola?
  - What should I do today?
  - Why was iTunes deleted from my computer?
  - Heather Locklear?
  - Do people in the Australian Defense Force (RAAF) pay less tax than civilians?
  - What's a psp xmb?
  - If C(-3, y) and D(1, 7) lie upon a line whose slope is 2, find the value of y.?
  - Why does love make us so irrational?
  - Am I in love?
  - What are some technologies that are revolutionizing business?

- **Ambiguous question**
  - What are the engagement and wedding traditions in Egypt?

- **A question very hard to answer**
  - Am I in love?
Searching Questions-and-Answers

- Community-based question answering services provide users with the ability to search the archive of previously-asked questions and the corresponding answers.
- If a user finds that a similar question has been asked and answered well in the past, the user does not need to ask the question and wait for responses.
  - Other users are not overwhelmed by similar questions again and again.
- Search engines may augment traditional search results with hits from the question and answer database.
  - Example: a query “Schrödinger’s cat” can be matched with the answers to question “What is Schrödinger’s cat?”
Matching for New Queries

• Given a new query, which can be either a new question or a conventional Web search query, how can we find answers to the query?
  – A new query can be matched with archived questions alone, archived answers alone, and the combination of archived questions and answers

• It is better to match queries against archived questions rather than answers
  – In general, it is easier to find related questions (which are likely to have related answers) than to match queries directly to answers
Matching Queries

• Traditional IR methods are ineffective in matching queries in community-based question answering systems – there are many (very) different ways to ask the same question
  – “who is the leader of Canada?”
  – “who is the current head of the Canada government?”
  – “who is the prime minister in Ottawa?”

• Common words are not informative
  – Who, is, the, Canada
  – “who is the finance minister of Canada?” and “who is the fastest man in Canada?”

• Generalizing “leader” to include other concepts such as “prime minister”, “head” helps
Using Translation-Based Models

• Many cross-language retrieval methods are based on translation probabilities $P(s|t)$ where $s$ and $t$ are two terms in two different languages
  – $P(s|t)$ is the probability that $s$ is used in place of $t$

• The same idea can be used in query matching in the same language
Independent Term Translation

• Given a query $Q$, answers are ranked according to

$$P(Q | A) = \prod_{w \in Q} \sum_{t \in V} P(w | t)P(t | A)$$

  – $A$ is an answer
  – $V$ is the vocabulary
  – $P(w|t)$ is the translation probability
  – $P(t|A)$ is the smoothed probability of generating $t$ given answer $A$ (i.e., the answer language model)

• Every term is translated independently – no guarantee that the answer is related to the query
Weighted Translation

- \( P(Q | A) = \prod_{w \in Q} \frac{(1 - \beta)tf(w, A) + \beta \sum_{t \in V} P(w | t)tf(t, A) + \mu \frac{c_w}{|C|}}{|A| + \mu} \)

- \( \beta \) is a parameter between 0 and 1 that controls the influence of the translation probabilities
- \( \mu \) is the Dirichlet smoothing parameter

- When \( \beta = 0 \), it is the query likelihood model
  - no influence from the translation model
- When \( \beta \) approaches 1, approaching a pure independent term translation model
Translation Probability Estimation

• How to estimate translation probability $P(t|w)$?
  – Use question-answer pairs $\{(Q_1, A_1), \ldots, (Q_n, A_n)\}$

• Computing the translation probability between every pair of words is expensive
  – For each query term, we can consider only a small number of translations

• A possible extension: for a question $Q$, find answers in a large document collection
Collaborative Search

- A group of users, with a common goal, search together in a collaborative setting.
- When a group of students work together on a survey of the latest advances in web search:
  - The traditional divide-and-conquer approach assigns a subtopic to each student.
  - A collaborative search allows the students to search the Web and other resources together so that every student can contribute and understand every subtopic.
- When a group of people plan for a Christmas party, a collaborative search system helps to coordinate information gathering tasks such as finding recipes, choosing decorations, selecting music, deciding on invitation, etc.
Two Types of Collaborative Search

• Co-located collaborative search: all of the search participants are in the same location and using the same computer

• Remote collaborative search: the search participants are physically located in different locations
CoSearch – A Co-Located System

• Driver – the person leading the search task
  – A primary display, keyboard and mouse
• Observers – additional participants
  – A mouse or a Bluetooth-enabled mobile phone
• Search results are displayed on the primary display and the mobile phones
• Participants vote for the next pages to be navigated and the next queries to be searched
Example
SearchTogether – A Remote System

- Each participant has her/his own computer
- Participants may not be online at the same time
- Queries asked by a participant is logged and shared with other participants
- Participants can add ratings and comments to pages that are viewed during the search process
  - The rating information is aggregated and shared with other participants
- A participant can recommend a page to others
- New participants can be added at any time and be brought up to speed by browsing the query history, page ratings, comments and recommendations
Example

(a) integrating messaging  (b) query awareness  (c) current results  (d) recommendation queue  (e-g) search buttons  (h) page specific meta data  (i) toolbar  (j) browser
Examples

Search Results from george using Windows Live Search

Diabetic-Lifestyle: Recipes and Practical Information for Managing...
Diabetic-Lifestyle Cooking Tips features useful ways to cook with more flavor, using less fat ... Diabetes and Cooking with Alcohol: Herbs - The Natural, delicious way to replace salt.
http://www.diabetic-lifestyle.com/cookingtips.htm

Diabetes > Healthy Cooking Tips for Persons With Diabetes
Methodist Hospital System ... Healthy cooking and diabetes management: A healthy diet is critical to proper diabetes management by...
http://www.methodisthealth.com/cgi-bin/hrmdin/home/basic.do?channelId=107383214&programId=1073769640&contentId=1073791149&contentType=HEALTHTOPICCONTENTTYPE

Healthy Cooking Tips for Persons With Diabetes - New York Presbyterian...
New York Presbyterian. The University Hospital of Columbia and Cornell. One Hospital affiliated with two Ivy League Medical Schools. Columbia University College of Physicians & Surgeons and Weill...
http://www.nyp.org/health/diabetes_cooking.html

WebMD Diabetes Health Center - Information on Type 1 and Type 2...
Diabetes Health Center Diabetes affects an estimated 18.2 million ... Plus, find daily support and nutritional tips in our online ... Healthy Cooking & Special Needs: Elaine Magee, RD Need some recipes...
http://www.webmd.com/diseases_and_conditions/diabetes.htm

Diabetes Control for Life - Healthy Eating
Tips on Keeping a Journal. Cooking With Less Fat ... Free Weight Loss Plan Specifically for People With Diabetes.
http://diabetescontrolforlife.com/healthy_eating/

Current Results Summary Search Results from george using Wi...

Diabetes mellitus type 2 - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Type_2_diabetes
This site has good general purpose background info, a overly technical.

MedlinePlus Medical Encyclopedia Type 2 diabetes
This information is from the NIH, so it's very trustworthy.

Insulin and Diabetes Medications by Lifeclinic
Betty, you should ask your doctor if this brand of insulin covered by your insurance. Will do. Thanks!

Guide to Diabetes Medications
You should print out a copy of this medication guide.

Diabetes Control for Life - Healthy Eating
http://diabetescontrolforlife.com/healthy_eating/
The article on vitamins look interesting. I should ask about that.

ADA Virtual Grocery Store - Recipes: Cooking Tips
http://vgs.diabetes.org/recipe/cookingtips.jsp
Future of Collaborative Search

- Collaborative search systems provide a unique set of tools for effective collaboration
- Very few commercial collaborative search systems exist today
- Beginning to gain considerable attention in the research community
- Only a matter of time to become widely available
Document Filtering

• Given a user’s information need, identify (filter) the relevant new documents in a dynamic document collection and send them to the user
  – A push application alternative to the standard ad hoc search where a user has many different information needs on a relatively static document collection

• Filtering as a classification problem
  – User profiles as training data, for a new document, predict whether it is relevant or not
  – User profiles are often very small, e.g., a single query – classification methods do not work well due to insufficient training data
Applications of Document Filtering

• News alerts in news sites
  – A user subscribes to some topics
  – When breaking news arrives, related users are alerted by emails, SMS (text messages) or news feeds

• Two important components in document filtering
  – A user profile to represent a user’s long-term information need
  – A highly accurate and efficient decision mechanism to identify documents relevant to user profiles
Filtering Profile

• The representation of a user’s long-term information need
  – Often in greater detail than short queries to search engines

• Components
  – A Boolean or keyword query
  – Example documents that are known relevant or irrelevant
  – Social tags and related named entities
  – (hard filters) Relational constraints such as time/price constraints (e.g., published on weekdays, stock price less than $100)
Filtering Models

• Playing a role similar to retrieval models
  – Many filtering systems use retrieval models where profiles are treated as queries

• Static models – users’ profiles do not change over time

• Adaptive models – users’ profiles are changing over time
Static Filtering Models

- Derived from the standard retrieval models
  - Instead of returning a ranked list, when a new document arrives, a static filtering model decides whether it is relevant to each profile.
Using a Boolean Model

• A filtering profile consists of a Boolean query
• A new document is retrieved for a profile only if the document satisfies the Boolean query
• Simple and effective, especially when precision is important
• Many web-based filtering systems use a Boolean model
• A Boolean model may have a low recall
  – In some domains, users may prefer to have good coverage than very high precision results
Using Complex Models

- The vector space model, the probabilistic model, BM25, or language models can be extended for filtering
  - Profiles are specified as keyword queries or sets of documents

- Those complex models return a score instead of a decision of “retrieve” or “not retrieve”
  - Using a global threshold to make the retrieval decision may not be effective
Using Language Models

• Given a profile P as a single keyword query, multiple queries, a set of documents, or some combinations of the above, we estimate the profile language model as

\[ P(w \mid P) = \frac{1 - \lambda}{\sum_{i=1}^{k} \alpha_i} \sum_{i=1}^{k} \alpha_i \frac{tf(w, T_i)}{|T_i|} + \lambda \frac{C_w}{|C|} \]

- \( T_1, \ldots, T_k \) are the pieces of text that make up the profile
- \( \alpha_i \) is the weight associated with \( T_i \)

• Given an incoming document, we estimate the document language model as

\[ P(w \mid D) = \frac{1 - \lambda}{|D|} \frac{tf(w, D)}{|D|} + \lambda \frac{C_w}{|C|} \]

• Make decision based on the negative KL divergence – if

\[-KL(P \mid D) = \sum_{w \in P} P(w \mid P) \left( \log P(w \mid D) - \log P(w \mid P) \right)\]
Adaptive Filtering

• Allowing dynamic profiles, more robust than static filtering techniques
Updating Profiles Using Feedback

• If a profile is represented as vector in a vector space model, Rochio’s algorithm can be used to update the profiles when the user provides relevance feedback.

• For a profile $P$, given a set of irrelevant feedback documents $\text{Irrel}$ and a set of relevant feedback documents $\text{Rel}$, update $P$ to $P'$

$$P' = \alpha P + \beta \frac{1}{|\text{Rel}|} \sum_{D_i \in \text{Rel}} \hat{D}_i - \gamma \frac{1}{|\text{Irrel}|} \sum_{D_j \in \text{Irrel}} \hat{D}_j$$
Using Language Models

\[ P(w \mid P) = \frac{1}{|\text{Rel}|} \sum_{D_i \in \text{Rel}} \sum_{D \in \mathcal{C}} P(w \mid D)P(D_i \mid D) \approx \frac{1}{|\text{Rel}|} \sum_{D_i \in \text{Rel}} P(w \mid D_i) \]

- C is the set of documents in the collection
- Rel is the set of documents that have been judged relevant
- \( P(D_i \mid D) \) is the probability that document \( D_i \) is generated from the language model of document D
  - \( P(D_i \mid D) = 1 \) or very close to 1 when \( D_i = D \), and nearly 0 for most other documents
## Summary Filtering Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Profile representation</th>
<th>Profile updating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Boolean expression</td>
<td>N/A</td>
</tr>
<tr>
<td>Vector space</td>
<td>Vector</td>
<td>Rocchio</td>
</tr>
<tr>
<td>Language modeling</td>
<td>Probability distribution</td>
<td>Relevance modeling</td>
</tr>
</tbody>
</table>
Efficient Filtering

• Conceptually, each profile can be regarded as a (multi-term) query
  – Running a profile query on incoming documents to retrieve relevant documents for the user
• There can be thousands or even millions of profiles
• Solution
  – Treat each profile as a document – index profiles using the document indexes
  – Treat each document as a query – when a new document comes, run the document query on the profiles and find relevant profiles
Collaborative Filtering

• Some users may share interests – their profiles may be similar
  – If A and B judge consistently a large number of the same documents to be relevant and/or irrelevant, A and B have similar profiles – a document judged relevant by A is likely be relevant to B
  – In static and adaptive filtering, profiles are assumed to be independent of each other

• Collaborative filtering – filtering that considers the relationship between profiles and uses this information to improve how incoming items are matched to profiles
Collaborative/Static/Adaptive Filtering

- Collaborative filtering typically associates a single profile with each user
  - In static/adaptive filtering, a user may have multiple profiles

- Collaborative filtering provides a rating for every incoming item and every item in the database for which the current user has not explicitly provided a judgment
  - Static/adaptive filtering makes a binary decision (retrieve or not) for each incoming document
Business Advantages

• Collaborative filtering is heavily used in recommendation systems such as Amazon.com and NetFlix
  – Providing users with a list of recommended products in the hopes that the user will see something they may like but may not have know about, and thus make a purchase
• Users are likely to see relevant products that they do not know before
• Retailers can sell products more effectively
• Search engines companies can increase revenue
Problem Formalization

- A set of users $U$
- A set of items $I$
- $r_u(i)$ is user $u$’s rating of item $i$ in range $[1, M]$
- $\hat{r}_u(i)$ is our system’s prediction for user $u$’s rating for item $i$ in range $[1, M]$
- Input: $r_u(i)$
- Output: for every user/item pair that does not have an explicit rating
Rating Using User Clusters

- Apply a clustering algorithm to find profile groups
  - Each profile is represented as $r_u = [r_u(i_1), \ldots, r_u(i_{|U|})]$
- Challenge: not all users judge all items
- Using correlation measure as the distance

$$\sqrt{\frac{\sum_{i \in I_u \cap I_{u'}} (r_u(i) - \bar{r}_u)(r_{u'}(i) - \bar{r}_{u'})}{\sqrt{\sum_{i \in I_u \cap I_{u'}} (r_u(i) - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_{u'}} (r_{u'}(i) - \bar{r}_{u'})^2}}}$$
Prediction

• Any user within a cluster that has not judged some item can be assigned the average rating for the item among other users in the cluster

\[ \hat{r}_u(i) = \frac{1}{\left| \text{Cluster}(u) \right|} \sum_{u' \in \text{Cluster}(u)} r_{u'}(i) \]

− Cluster(u) is the cluster that u belongs to

• Alternatively, we can use expectation

\[ \hat{r}_u(i) = \sum_{x=1}^{M} x \cdot P(r_u(i) = x \mid C = \text{Cluster}(u)) = \sum_{x=1}^{M} x \cdot \frac{\left| \{u' \mid r_{u'}(i) = x\} \right|}{\left| \text{Cluster}(u) \right|} \]

− \( P(r_u(i)=x \mid C=\text{Cluster}(u)) \) is the probability that user u will rate the item with rating x given that they are in cluster Cluster(u)
A Problem in Clustering Approach

• Very sparse clusters – a profile may not fit into a cluster well
  – What prediction should be made to those outlier profiles?
• No straightforward solutions
  – Assigning global average is not a good solution
Rating Using Nearest Neighbors

- To predict ratings for user u, we find the k users that are closest to the user according to some distance measure, then use the ratings of the neighbors for prediction:

\[
\hat{r}_u(i) = \bar{r}_u + \frac{\sum_{u' \in N(u)} \text{dist}(u, u')(r_{u'}(i) - \bar{r}_{u'})}{\sum_{u' \in N(u)} \text{dist}(u, u')}
\]

- \text{dist}(u, u') is the distance between u and u';
- N(u) is the set of u’ s nearest neighbors

- Tend to be more robust to noise than the clustering approach
Collaborative Filtering Evaluation

- The difference between the actual and predicted ratings should be considered.
- Absolute error
  \[
  ABS = \frac{1}{|U||I|} \sum_{u \in U} \sum_{i \in I} |\hat{r}_u(i) - r_u(i)|
  \]
- Mean squared error
  \[
  MSE = \frac{1}{|U||I|} \sum_{u \in U} \sum_{i \in I} (\hat{r}_u(i) - r_u(i))^2
  \]
- Which one is better? It depends on the task and the data set.
Distributed Search

• Search using communities of nodes
  – Each node can store and search information, and can communicate with other nodes
• Metasearch engine – each node is a complete web search engine, and the results from a relatively small number of different search engines are combined to improve the effectiveness of the ranking
• Peer-to-peer (P2P) search – a large number of nodes, each with a relatively small amount of information and only limited knowledge about other nodes
Additional Functions in Distributed Search

- Resource representation: generating a description of the information resource (i.e., the documents) stored at a node
- Resource selection: selecting one or more resources to search based on their descriptions
- Result merging: merging the ranked result lists from the searches carried out at the nodes containing the selected information resources
- There may be one special node that provides the directory services of selection and merging, and every other node is responsible for providing its own representation
Metasearch Architecture

- A query is broadcast by the metasearch engine to all engines
Resource Selection

• Generally, in distributed search, for a given query, which nodes should be queried?
• Build a language model for each node using the documents stored at the node
• Rank nodes in query likelihood
• Use the top-k nodes or nodes passing some threshold
• Conduct local search in the selected nodes and normalize local results using the node ranking scores
• Merge normalized local search results
Summary

• Social search concepts
• Using user tags
  – Searching tags
  – Tag recommendation
  – Browsing using tags
• Online communities
  – Finding communities
  – Community-based question answering
  – Collaborative searching
• Filtering and recommendation
  – Document filtering, static/adaptive filtering
  – Collaborative filtering
• Distributed search
  – Metasearch and peer-to-peer search
To-Do List

• Reach Chapter 8
• Go to http://www.gwap.com/gwap/ and play the game ESP and some other games
• Try out SearchTogether at http://research.microsoft.com/searchtogether/