

Data Valorization: Recommender System

Lionel Fillatre

fillatre@unice.fr


Outline

- Introduction
- Collaborative Filtering
- Memory-Based: Baseline Algorithm
- Matrix Factorization
- Practical Issues
- Conclusion



1 Introduction

Example: Similar Pages



Google Similar Pages

offered by google.com

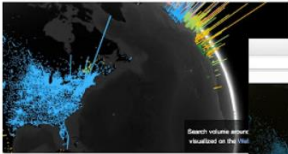
★★★★★ (919) | [Search Tools](#) | 162,542 users

AVAILABLE ON CHROME

OVERVIEW

REVIEWS

RELATED

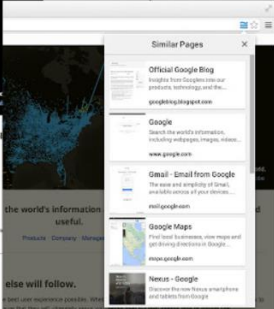


the world's information and make it universally accessible.

Products · Company · Management

else will follow.

Best user experience possible. Whether we're designing a new Internet browser or a new search engine, we want to make sure that they will ultimately serve you.



Similar Pages


- Official Google Blog
Insights from Google's internal products, technology and the...
[googleblog.blogspot.com](#)
- Google
Search the world's information, including webpages, images, videos...
[www.google.com](#)
- Gmail - Email from Google
The easy way to manage all your email, and get things done in Google...
[mail.google.com](#)
- Google Maps
Find your location, view maps and get driving directions in Google...
[www.google.com](#)
- News - Google
Browse the new News smartphone and tablet app today.

By Google

Discover webpages similar to the page you're currently browsing.

Discover webpages similar to the page you're currently browsing. Enjoying the page you're looking at and interested in other similar pages? Trying to find more pages about a topic you're researching, but having a hard time coming up with the right query on Google? Google Similar Pages can help!


Now you can quickly preview and explore other pages that are similar to the one you are browsing -- on the fly.

 [Report Abuse](#)

Additional Information

Version: 0.6.6.2
Updated: April 7, 2014
Size: 231KIB
Languages: [See all 40](#)

Example: on-line shopping



ISBN-10: 978-1617090543
ISBN-13: 978-1617090543
Why is ISBN important?

Trade in your item
Get a \$11.46
Gift Card

Trade In

Learn More

ADD TO LIST

Share

☐ Buy used \$26.31

☒ Buy new **\$36.00**

In Stock.
Ships from and sold by Amazon.com. Gift-wrap available.

Lowest Price \$48.00 Save \$12.00 (25%)
40 new from **\$23.00**
FREE Shipping
Want it Tuesday, April 26? Order within **12 hrs 28 mins** and choose **Two-Day Shipping** at checkout.
[Details](#)

Turn on 1-Click ordering

Ship to:
Select a shipping address.




More Buying Choices
40 new from **\$23.00** 21 used from **\$25.25**
[See All Buying Options](#)

amazon student **FREE TWO-DAY SHIPPING**
FOR COLLEGE STUDENTS [Learn more](#)

Summary
Big Data teaches you to build big data systems using an architecture that takes advantage of clustered hardware along with new tools designed specifically to capture and analyze web-scale data. It describes a scalable, easy-to-understand approach to big data systems that can be built and run by a small team. [Read more](#)

Check out the most buzzworthy new releases for Spring

Frequently Bought Together

















 +  + 
Total price **\$103.45**
[Add all three to cart](#)
[Add all three to list](#)

☒ This item: Big Data: Principles and best practices of scalable real-time data systems by Nathan Marz; Paperback; **\$36.00**

☒ Hadoop: The Definitive Guide by Tom White; Paperback; **\$22.00**

☒ Learning Spark: Lightning-Fast Big Data Analysis by Holden Kauri; Paperback; **\$24.00**

Customers Who Bought This Item Also Bought

-  **Hadoop: The Definitive Guide** by Tom White
-  **Advanced Analytics with Hadoop** by James D. Warren
-  **Learning Spark** by Holden Kauri
-  **Hadoop Application Development** by James D. Warren
-  **Building Microservices** by Sam Newman
-  **Storm Applied: Strategies for Big Data** by James D. Warren
-  **NoSQL Cookbook** by Bruno Lamas
-  **Data Science from Scratch** by John D. Blumenthal
-  **Data Science for Business** by John D. Blumenthal
-  **Learning Apache Cassandra** by James D. Warren
-  **Amazon Web Services in Action** by James D. Warren
-  **Real-Time Analytics** by James D. Warren
-  **Big Data Analytics with Spark** by James D. Warren
-  **Programming Scala** by Martin Odersky
-  **Big Data Using Hadoop** by James D. Warren
-  **Java 8 in Action** by Ross W. Burton

User Ratings

- Many systems ask users to *rate* items – e.g. on a scale of 1 to 10. These ratings then enable the system to give more precise/accurate recommendations, and use a variety of sophisticated learning/prediction algorithms.
- Example: Here are user ratings for some items (“?” means unrated).

	A	B	C	D	E	F	G	H
You:	7	2	1	8	9	9	?	?
User1	1	8	8	2	?	2	8	7
User2	6	3	3	7	6	5	3	1
User3	7	2	1	7	7	?	3	1

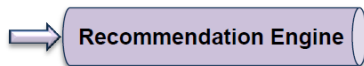
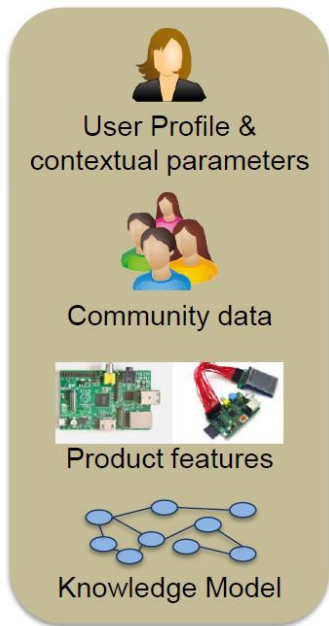
- How might a system predict your rating for items G and H?

Example: Netflix Prize

- Task
 - Given customer ratings on some movies
 - Predict customer ratings on other movies
- If John rates
 - “Mission Impossible” a 5
 - “Over the Hedge” a 3, and
 - “Back to the Future” a 4,
 - How would he rate “Harry Porter”, ... ?
- Performance
 - Error rate (accuracy)
- Grand Prize (2009)
 - \$1M
 - 10% improvement

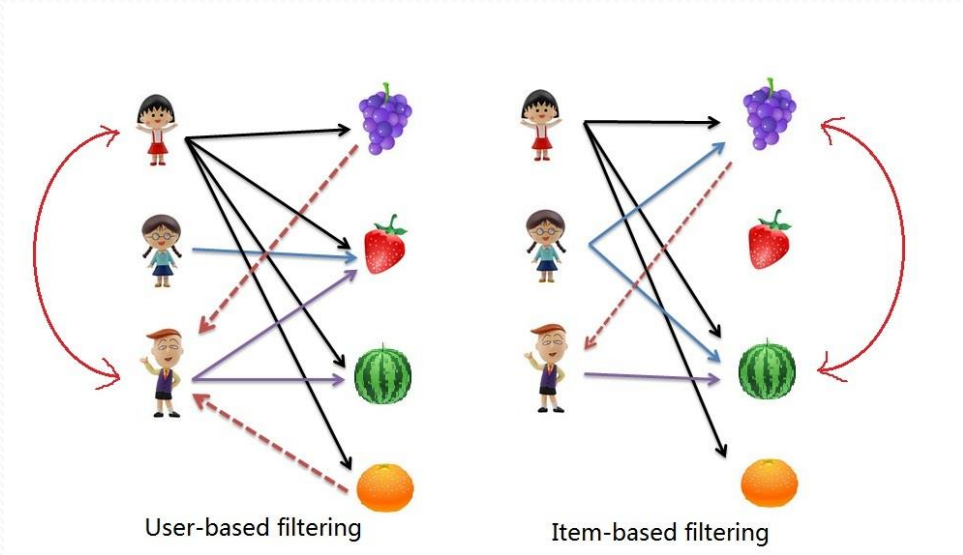


Types of Recommender Systems



1. Personalized recommendations
2. Collaborative: « Tell me what's popular among my peers »
3. Content-based: « Show me more of the same what I've liked »
4. Knowledge-based: « Tell me what fits based on my needs »
5. Hybrid: combinations of various inputs and/or composition of different mechanisms

User-based versus Item-based



User Profiles

- For user-based recommendation, sites need to have some kind of user profile.
- Similarity with other users is based on distance measurements based on the profile.
- What do you think could be in a user profile?

Potential contents of user profiles

- Demographic data: age, gender, salary, profession, country of residence, country of origin, religion ...
- Site behaviour: purchase history at the site; viewing history, perhaps including time spent on certain pages/items; clickstream sequence

Specificities

- Complexity grows linearly with the number of customers and items
- The sparsity of recommendations on the data set
 - Even active customers may have purchased well under 1% of the products



2 Collaborative Filtering

Basic Strategies

- Predict and Recommend
- **Predict** the opinion: how likely that the user will have on this item
- **Recommend** the “best” items based on
 - the user’s previous likings, and
 - the opinions of like-minded users whose ratings are similar
- **Assumption:** users with similar taste in past will have similar taste in future

Why “collaborative”?

- Basically, someone else (in fact many someones) have gone to the effort of viewing/filtering things, and chosen the best few. You get a recommendation of the best few, without having to spend the effort.
- Main CF Techniques
 - Clustering based
 - Memory based
 - Nearest neighbors (user, item)
 - Model based
 - Matrix factorization/Latent factors

Clustering Techniques

- Work by identifying **groups** of consumers who appear to have **similar** preferences
- Performance can be good with smaller size of group
- May **hurt accuracy** while dividing the population into clusters

Example: clustering

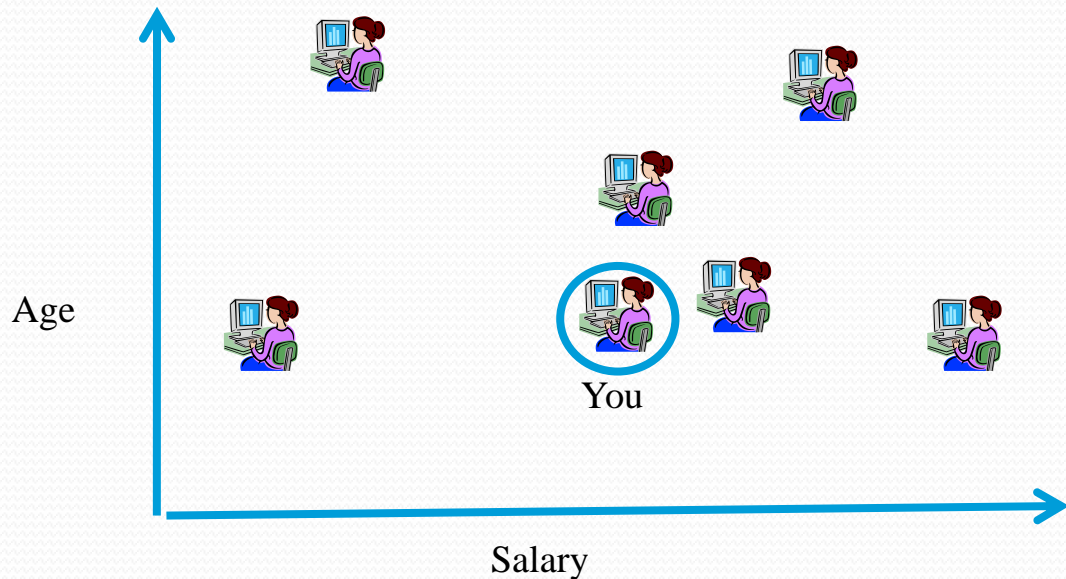
	Book1	Book2	Book3	Book4	Book5	Book6
Customer A	X			X		
Customer B		X	X		X	
Customer C		X	X			
Customer D		X				X
Customer E	X				X	

- B, C & D form the first cluster vs. A & E form another cluster.
- « Typical » preferences for first cluster are:
 - Book 2, very high
 - Book 3, high
 - Books 5 & 6, may be recommended
 - Books 1 & 4, not recommended



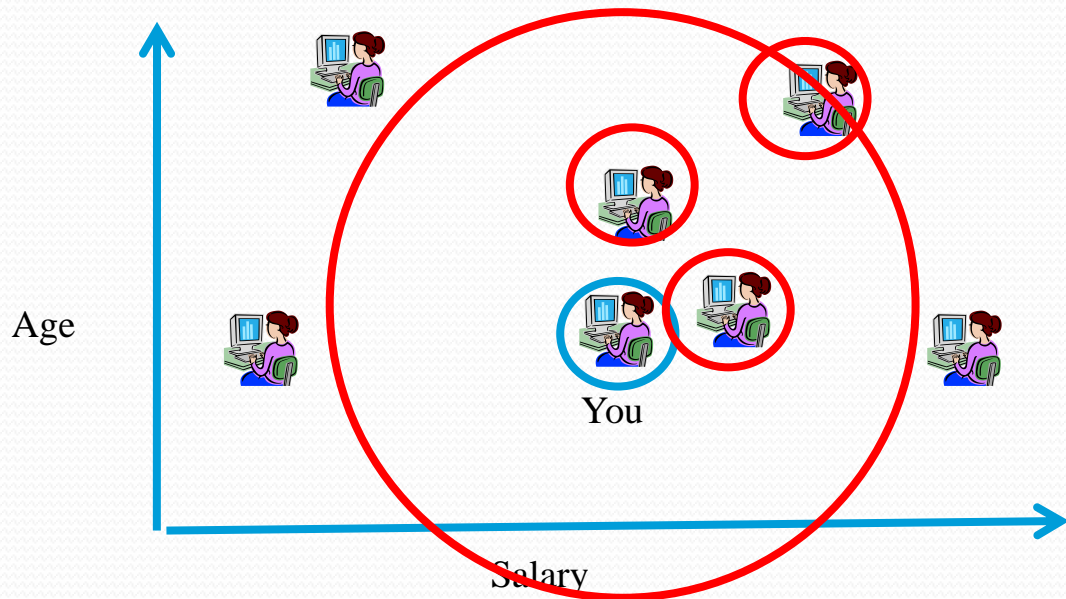
3 Memory-Based: Baseline Algorithm

K-Nearest Neighbour based Recommendation



(Think in terms of many dimensions, not just these two)

K-Nearest Neighbour based Recommendation



Your neighbours: recommend things that they have viewed/purchased

Item-to-Item Collaborative Filtering

- No more matching the user to similar customers
- Build a **similar-items table** by finding that customers tend to purchase together
- Amazon.com used this method
- Scales independently of the catalog size or the total number of customers
- Acceptable performance by creating the expensive similar-item table offline

Memory-Based Algorithms

- $v_{b,j}$ = vote of user b on item j
- I_b = set of items for which user b has voted
- Mean vote for user b is $\bar{v}_b = \frac{1}{|I_b|} \sum_{j \in I_b} v_{b,j}$
- Predicted vote for “active user” a is weighted sum

	$v_{b,j}$	item j							
		A	B	C	D	E	F	G	H
User1		7	2	1	8	9	9	?	?
User2		1	8	8	2	?	2	8	7
User3		6	3	3	7	6	5	3	1
User4		7	2	1	7	7	?	3	1

$$p_{a,j} = \bar{v}_a + \gamma \sum_{b=1}^n \underbrace{w(a,b)}_{\text{weights of } n \text{ similar users who have voted for item } j} (v_{b,j} - \bar{v}_b)$$

normalizer

$$\gamma = 1 / \sum_{b=1}^n |w(a,b)|$$

weights of n similar users
who have voted for item j

Memory-Based Algorithms

- K-nearest neighbor:

$$w(a, b) = \begin{cases} 1 & \text{if } b \in \text{neighbors}(a) \\ 0 & \text{else} \end{cases}$$

- Pearson correlation coefficient:

$$w(a, b) = \frac{\sum_{j \in I_a \cap I_b} (v_{a,j} - \bar{v}_a) (v_{b,j} - \bar{v}_b)}{\sqrt{\sum_{j \in I_a \cap I_b} (v_{a,j} - \bar{v}_a)^2 \sum_{j \in I_a \cap I_b} (v_{b,j} - \bar{v}_b)^2}}$$

- Cosine distance (unobserved item receive a zero vote):

$$w(a, b) = \sum_{j \in I_a \cup I_b} \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{b,j}}{\sqrt{\sum_{k \in I_b} v_{b,k}^2}}$$



4 Matrix Factorization

Matrix of Ratings

n customers

d products

$$\begin{pmatrix} A \\ A_{ij} = \text{rating of } j\text{-th product} \\ \text{by the } i\text{-th customer} \end{pmatrix}$$

Find subsets of products that capture the behavior or the customers

Singular Value Decomposition

$$A = U \Sigma V^T = [u_1, u_2, \dots, u_r] \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_r^T \end{bmatrix}$$

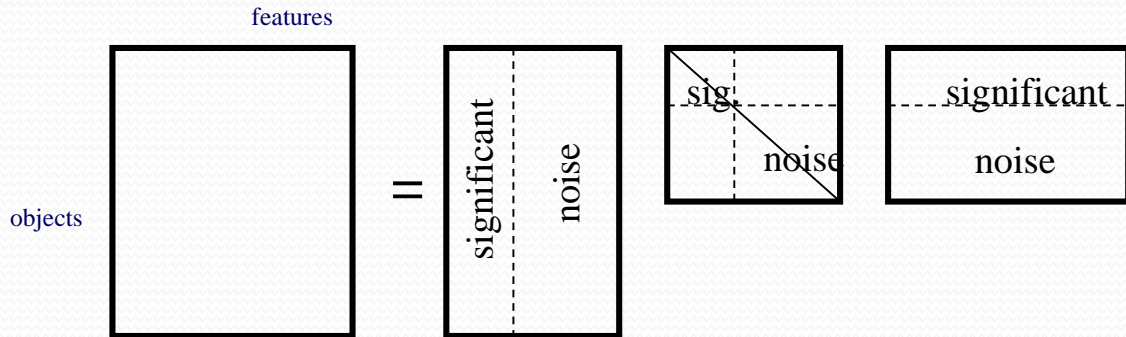
$[n \times m] = [n \times r][r \times r][r \times m]$
 r : rank of matrix A

- $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$: singular values of matrix A (also, the square roots of eigenvalues of AA^T and $A^T A$)
- u_1, u_2, \dots, u_r : left singular vectors of A (also eigenvectors of AA^T)
- v_1, v_2, \dots, v_r : right singular vectors of A (also, eigenvectors of $A^T A$)

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T$$

SVD and Rank- k approximation

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$








Application: Recommender systems

- Data: Users rating movies
 - Sparse and often noisy
- Assumption: there are r basic user profiles, and each user is a linear combination of these profiles
 - E.g., action, comedy, drama, romance
 - Each user is a weighted combination of these profiles
 - The “true” matrix has rank r
- What we observe is a noisy, and incomplete version \tilde{A} of this matrix A
 - The rank- k approximation \tilde{A}_k is provably close to A
- Algorithm: compute \tilde{A}_k and predict for user u and movie m , the value $\tilde{A}_k[m, u]$.

Example: Matrix of Ratings (2 factors)

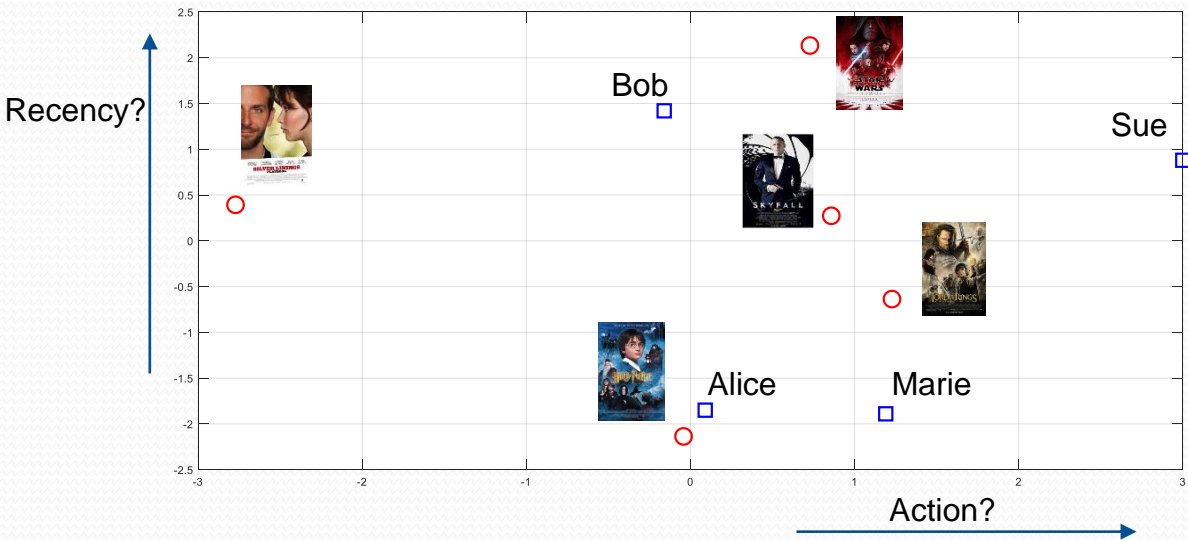
A_{ij}	M1	M2	M3	M4	M5
Alice	-4	-1	0	1	4
Bob	3	1	0	-1	-3
Mary	-3	-4	0	3	4
Sue	4	-8	3	3	-2

	Dim1	Dim2
Alice	0.09	-1,85
Bob	-0.16	1.42
Mary	1.19	-1.89
Sue	3.00	0.88

					
Dim1	0.73	-2.77	0.86	1.23	-0.04
Dim2	2.13	0.39	0.27	-0.64	-2.14

Prediction: $\hat{r}_{ij} = 0,09 \times 1,23 + (-1,85) \times (-0,64) = 1,2947 \approx 1$

Lower Dimensional Feature Space





5 Practical Issues

Practical Issues : Ratings

- Rating Scales
 - Scalar ratings
 - Numerical scales
 - 1 – 5, 1 – 7, etc.
 - Binary ratings
 - Agree/Disagree, Good/Bad, etc.
 - Unary ratings
 - Good, Purchase, etc.
 - Absence of rating indicates no information

Practical Issues : Cold Start

- New user
 - Rate some initial items
 - Non-personalized recommendations
 - Describe tastes
 - Demographic info
- New item
 - Non-CF : content analysis, metadata
 - Randomly selecting items “close” to the new item

Evaluation Metrics

- Accuracy

- Predict accuracy

- The ability of a CF system to predict a user's rating for an item
 - Mean absolute error (MAE)

$$\text{MAE} = \frac{\sum_{(a,j) \in W} |v_{a,j}^p - v_{a,j}|}{|W|}$$

- $v_{a,j}^p$ is the predicted value of $v_{a,j}$
 - W is the set of all predicted couples (user,item)
 - The MAE used the same scale as the data being measured.

- Rank accuracy

- Percentage of items in a recommendation list that the user would rate as useful

Evaluation Metrics

- Novelty
 - The ability of a CF system to recommend items that the user was not already aware of.
- Coverage
 - The percentage of the items known to the CF system for which the CF system can generate predictions.
- Serendipity
 - Users are given recommendations for items that they would not have seen given their existing channels of discovery.

Serendipity

- Unsought finding
- Unexpected, but useful result
- Do not recommend items the user already knows or would find anyway, try something more interesting
- Example
 - I like movies by Steven Spielberg, Peter Jackson, and James Cameron
 - Recommending another movie by Steven Spielberg not very useful
 - Recommending Quentin Tarantino = serendipity

Evaluation Metrics

- Learning Rate
 - How quickly the CF system becomes an effective predictor of taste as data begins to arrive.
- Confidence
 - Ability to evaluate the likely quality of its predictions.
- User Satisfaction
 - By surveying the users or measuring retention and use statistics

Additional Issues : Privacy & Trust

- User profiles
 - Personalized information
- Distributed architecture
 - Security of distributed systems
- Recommender system may break trust when malicious users give ratings that are not representative of their true preferences.



6 Conclusion

Conclusion

- Recommender systems have had broadly visible impact:
 - Google, TIVO, Amazon, personal radio stations, ...
- Critical tool for finding “consensus information” present in a large community (or large corpus of web pages, or large database of purchase records,)
- Relatively-well established, especially in certain narrow directions, on a few datasets
- Set of applications still being expanded