Finding what People Like

(Shopping and Searching)

The Goal

- When customers visit an e-commerce site (for example Amazon) we want to recommend items to them.
- We have to guess what they like.
- We have to guess what items match their preferences.
- We use the term **filtering** for selecting items to recommend.

Two Strategies for Recommending Items to Customers

- **Content-based Filtering**
- **Collaborative Filtering**

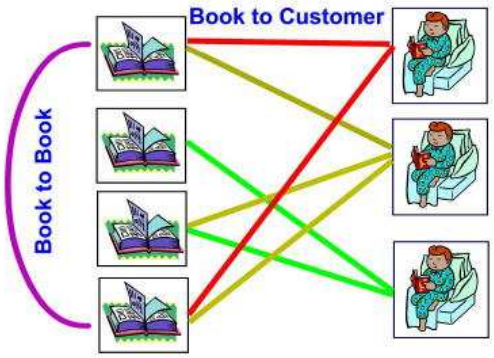
Content-based Filtering

- It relies on information about the content of various items, books, movies, music, etc.
- Content-based filtering requires human effort to add labels to the computer record of each item.
- More reliable but also far more expensive than collaborative filtering.

Collaborative Filtering

- It relies on the use of several kinds of information that are available from earlier customer actions.
- Dominant type of filtering in the web.

LOOKING FOR BOOKS ON AMAZON THROUGH COLLABORATIVE FILTERING



The Relationship between Two Books Part I

- Let *SALES(A*) and *SALES(B)* be the number of total sales of each of books *A* and *B*.
- Let *SALES_COM* be the number of people that bought both books. In order to express it as a percentage we need to divide it by a number combining the individual sales.

The Relationship between Two Books Part II

- Reportedly, Amazon uses the geometric mean of the individual sales, the square root of the product *SALES(A*) times *SALES(B). Then*
- *REL(A,B)* = *SALES_COM* / ***SQRT[***(*SALES(A*)**SALES(B)***]*** **{1}**
- Using the geometric mean rather than the average provides a better balance when one of the books has much bigger sales than the other.

The Relationship between Two Books Part III

- Suppose book A has sold 10,000 copies and book B only 60 but each person who bought book B has also bought book A, so the common sales are 60.
- The geometric mean of 10,000 and 60 is the square root of 600,000 or about 775, so that the relationship number will be 60/775 or about 0.077.
- The average sales number is 10060/2 or 5030 and that yields 0.012 for the relationship, a much smaller number.

A Generalization

- For each item (book, DVD, etc) construct a vector $\boldsymbol{U_A}$ that has as many components as users and each component equals the rating given by that user. If no rating is given the entry is 0. Then the relationship between two items is expressed as the cosine angle between the two vectors:
- $COS(A,B) = \langle U_A , U_B \rangle / ||U_A|| * ||U_B||$ **{2}**

$$\cos(A,B) = \frac{\sum U_A^i U_B^i}{\sqrt{\sum (U_A^i)^2}\sqrt{\sum (U_B^i)^2}}$$

U_A^i is the rating given to item A by the i^{th} user

If U_A^i can only be 0 or 1

$\sum U_A^i U_B^i$ is the common sales

$\sum (U_A^i)^2$ is the total sales of A.

Equ. {1} is a special case of Equ. {2}

- If the components are either 0 (user has not bought the item) or 1 (user has bought the item), then the sum equals the number of terms where both components are 1, in other words the number of users who bought both items. $||U_A||$ is the norm of the vector U_A and that equals the square root of the sum of the squares of each term. Again when the terms are either 0 or 1 the norm is the square root of the total sales for item A.

Problems with the Cosine

- We have only a gross image of the relationship between the items. We pay no attention to user categories (demographics).
- Suppose the only people who bought both items A and B are teenagers.
- If an older person expressed interest in A, does it make sense to recommend B?

Example of some not-so-hot recommendations

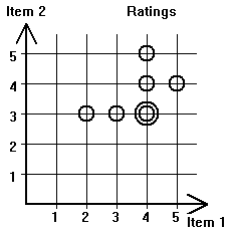
Only one of the recommendations is really related to the first item.

Let us try to do better

- Keep closer track of individual user ratings.
- Regression Lines!

Regression Line - 1

Table 2.2.1 User Ratings for Two Items		
User	Item 1	Item 2
John	4	5
Richard	4	4
Tiffany	2	3
Lucy	4	3
Michael	5	4
Basil	4	3
Betty	3	3

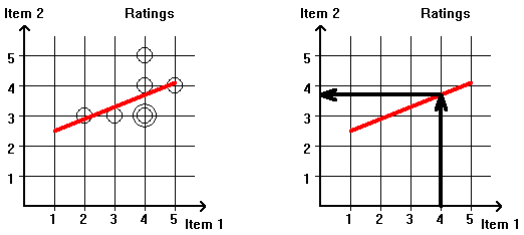


The graph on the right is another way of looking at the data of the table on the left.

Regression Line - 2

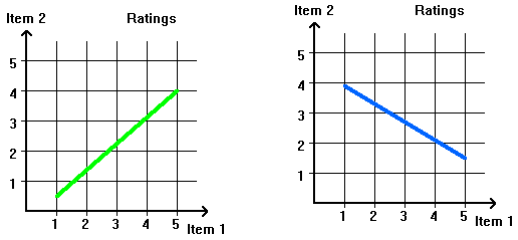
- We can fit a line (or other curve) to the points representing user choices for the two items.
- We use similar methods as those we discussed when we were fitting lines in dark pixels trying to enable computers to read.

Regression Line - 3



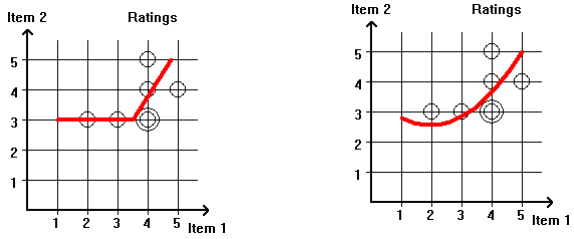
Regression Line is shown in red. Left: The construction of the line. Right: How the line can be used to predict the preference of a user.

Regression Line - 4



The slope is positive if users tend to give similar ratings to both items (Left). It is negative when opinions for the two items are in opposition (Right).

Regression Curves

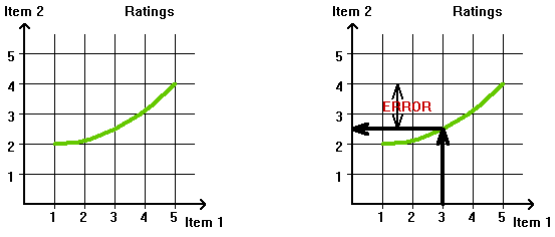


We may express relations between user ratings for two items by more complex curves than a straight line. Left: Two line segments (a *polyline)*. Right: A curve (parabola).

Machine Learning - 1

- A fancy term for finding regression lines or curves.
- In higher dimensions (say ratings of N items) we look for (hyper)planes or (hyper)surfaces.
- Split the data into two sets: use one (design set) to find a regression curve (or surface); use the other (test set) to estimate the errors of the responses.

Machine Learning - 2



Left: Regression Curve found from the design set. Right: Computing the error from the test set. The regression curve predicts a rating of 2.5 for item 2, if item 1 was rated 3. If someone rated the items as 3 and 4 we have a 1.5 error.

Machine Learning - 3

- In general we have N classes of objects.
  - For a reading machine we have N about 70 (letters of the alphabet in upper and lower case, numbers, symbols such as “?” or “!”)
- Each objects is mapped into a ***feature vector***.
  - Each feature is a property of the object. For a letter it might be number of holes, aspect ratio of a bounding box, etc, etc.
  - In a typical reading machine we may have over 100 features.

Machine Learning - 4

- If $\underline{v}$ is a feature vector we try to construct N discriminant functions F(k, $\underline{v}$) where k=1, 2, … N.
- In order to classify a letter we compute all N functions for the feature vector.
- We assign the class corresponding to the function that takes the highest value.

Machine Learning - 5

- We need to estimate a lot of parameters. With d features and N classes we must compute at least N*(d+1) parameters.
  - For a reading machine we need several thousands of parameters.
- We must compute these parameters from the design set. To avoid "over-fitting" we need at least twice as many samples as parameters.

Machine Learning - 6

- A computer needs several hundreds (if not thousands) of samples of letters in order to learn how to read.
- A human needs only one or two samples!
- HUMAN LEARNING IS QUITE DIFFERENT THAN MACHINE LEARNING.

The *Netflix* Challenge - 1

- When someone orders a movie, *Netflix* recommends other movies that this person might like. Users rate movies by assigning from one to five stars and the company uses these ratings to guess what a particular person might like.
- Netflix would like such guesses to be as accurate as possible.

The *Netflix* Challenge - 2

- In 2007 the company announced that it offered a million dollar prize to anyone that could improve upon its own system for predicting user tastes.
- The company made public data from the ratings of past users that could be used by those trying to meet the challenge.

The *Netflix* Challenge - 3

- The first round of the Netflix Challenge was won by a group from the AT&T Research Labs (they used to be called Bellcore) who combined several methods to improve upon the ratings. Then they combined their efforts with those of two other top teams creating a new team called *BellKor's Pragmatic Chaos* and that team won the final contest.

The *Netflix* Challenge - 4

- The following is from the *Netflix* web site. (Emphasis has been added.)
  "It is our great honor to announce the $1M Grand Prize winner of the Netflix Prize contest as team **BellKor's Pragmatic Chaos** for their verified submission on July 26, 2009 …, achieving the winning RMSE of 0.8567 on the test subset. This represents a 10.06% improvement over Cinematch's* score on the test subset at the start of the contest. We congratulate the team of ….. for their superb work advancing and **integrating many significant techniques** to achieve this result."

  * Cinematch is the in-house system of *Netflix*.

The *Netflix* Challenge - 5

- The improved results relied a lot on user demographics.
- The company was planning a second challenge but it changed plans in response to legal objections regarding the privacy of the company's customers whose movie ratings had been used in the design and test sets.

Google

- Google relies , in effect, on collaborative filtering.
- EXPAND

The Price of Collaborative Filtering

- Suppose you want to find an article about *headhunters*, groups of people who kill others and collect their heads.
- Unfortunately for you, another meaning of the word *headhunter* is that of a corporate recruiter.
- Many more people are interested in corporate recruiters than killers.

The Price of Collaborative Filtering (continued)

- Because many more people are interested in corporate recruiters than killers, if that if you type "headhunter" in *Google* the returns will be for the corporate type.

The Price of Collaborative Filtering (continued)

- You may try, for example, "headhunter amazon" because you have heard that headhunting was practiced amongst primitive people living in the Amazon rain forest.
- *Google* will return some items that are indeed close to what you are looking for but it will also return several irrelevant ones, for example about a music band called "Headhunters" whose albums are sold on *amazon.com*.
- "headhunter tribes" seems to be the best bet for finding what you are looking for.

Looking for a Dog

- Suppose we are looking for stories about a **dog named Lucy**. Typing "dog named Lucy" on *Google* is too restrictive because the word "named" may not appear in a story.
- Also using all three words does not guarantee that we will get only what we want. When I did that on Google one of the stories that was returned contained the sentence: *Lucy and I spent the weekend alone together. We have a dog named Kyler*.

Looking for a Dog (cont)

- Insisting on the **exact** phrase "dog named Lucy" does not produce any unwanted returns but it missed a lot of stories.
- It is better to try "dog Lucy". That did produce some unwanted returns (for example: "Ask **Lucy** - How to feed a senior **dog**") but the first 22 returns were all about a dog named Lucy!

Conclusion

- A computer cannot search for stories with a given meaning but only for stories with given words. It is human intervention that enables a search for meaning.
- On one end is the construction of the query by the user and at the other end is the feedback from previous users of the system.