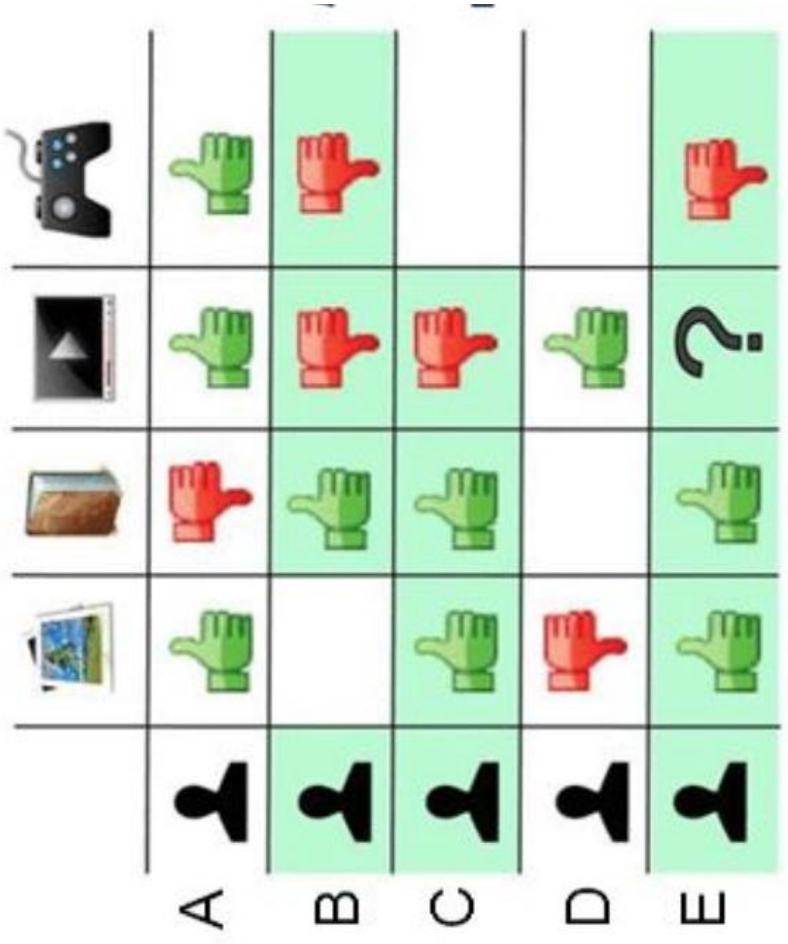
CISC 372
Network/Graph I



- Changed:

- Mini-exams (35% total): We have three tests and will only count ***top-2*** of those (17.5% each).

- Assignments (15%): Three assignments (#1, #2-3, and #4) and only count the ***top-1*** **of assignments**.

- Final project (50%): For the project presentation (10%), please send me the slides and notes of what are you going to say for each slide. Maximum 10 slides including the cover page.

- Part III: Network
  - Basic characteristics of network
  - Recommendation
  - Supervised/Self-supervised learning on information network
  - PageRank

A common problem we face when searching for information is how to find the good information in among the bad information.

Examples:

* finding a good restaurant
* choosing a good book
* deciding which movie to see
* deciding whether or not to take a course
* deciding who to marry
* deciding which grad school to go to
* …

Mechanisms to do this are often based on the opinions of others:

* asking friends
* editors
* newspaper columns
* people who liked this also liked …

Some of these are global authorities, `gatekeepers'

others are local authorities.

Because tastes, motivations, constraints differ, global recommendations are not always useful.

That's why we elect local members of parliament; read different newspapers; listen to different political talk shows.

Can we use the collective wisdom of individuals to make recommendations to others (like them) about wise/useful/fun choices?

Example: Amazon

Customers who bought Red Dwarf: Series 5/6 also bought:

Red Dwarf: Series 3/4

Red Dwarf Series 1 & 2

Doctor Who: The Green Death

Explore similar items

Basic idea:

For a given individual

find similar individuals

look at what these similar individuals

recommend

recommend this to the new individual

What does similar mean?

What does recommend mean?

Similar:

Take a geometric view;

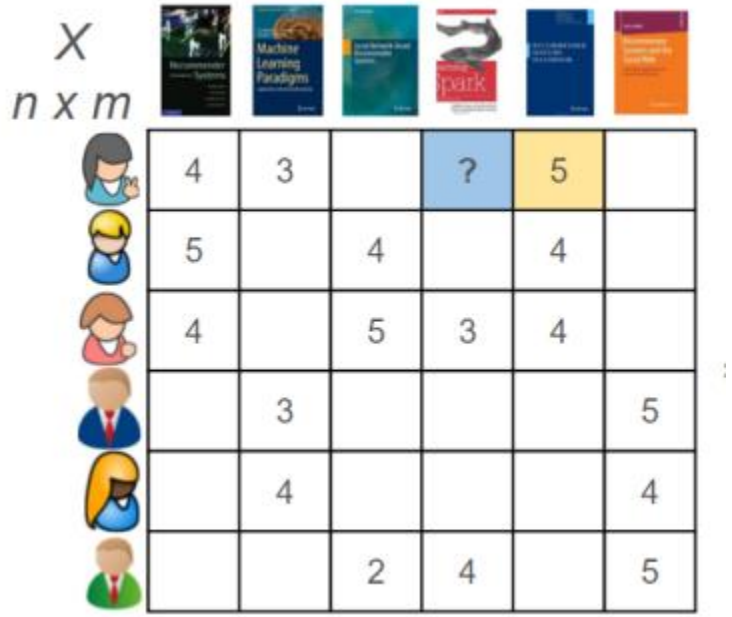
Similar individuals are those who are close in space to the new individual

Recommend:

Objects about which old users have (+ve) opinions and the new user doesn’t

Example: Movie recommender

Collect opinions/ratings for movies from a lot of people.



Ask each new person to rate several movies

Find neighbours who have rated the same movies similarly (so they probably have the same taste as the new user)

Look for movies that these neighbours liked but the new user hasn’t rated.

Recommend these.

Example: Amazon

Instead of explicit ratings, use the fact that someone has purchased a book as an implicit positive vote for it.

(Obvious problems:

* it was bought for someone else
* they thought they would like it but found that they hated it

)

It turns out that expressing preferences implicitly is important for making these systems work.

People are only happy to express an opinion explicitly when there's a big payoff for them in getting good opinions.

But this creates a chicken and egg problem – it's not worth working hard to provide a good opinion when the system hasn't learned enough to repay you with good recommendations.

When a new user selects a book/CD/DVD for their shopping cart, look for others who bought the same item,

and then for items that they bought as well

Recommend these items in the next displayed page.

Implicitly we are assuming that the space of possible items divides into regions that represent different sets of tastes or preferences.

But we're not assuming that there are clusters in this space – there might be a continuum, with tastes blending into each other.

It's not a global model.

The size of the problem:

Amazon sells about 372 million items,

to about 300 million people.

So this matrix is very big,

and very sparse

(most people don’t buy everything)

Techniques: how could we implement this idea?

Find nearest neighbours.

BUT

* we know that distances are not well-behaved in such a high-dimensional space

* the complexity of finding nearest neighbours is exponential in the number of dimensions, even using clever data structures, e.g. k-d trees

(big secret!!)

* we need to find nearest neighbours for every item purchased (lots of the computation is done on the critical path)

so this simple idea isn't very practical.

Second idea: Map all of the points to the unit sphere (a kind of normalization).

The sphere is still embedded in a high-dimensional space, and similarity is just about as hard to compute (angular difference → cosine → dot product).

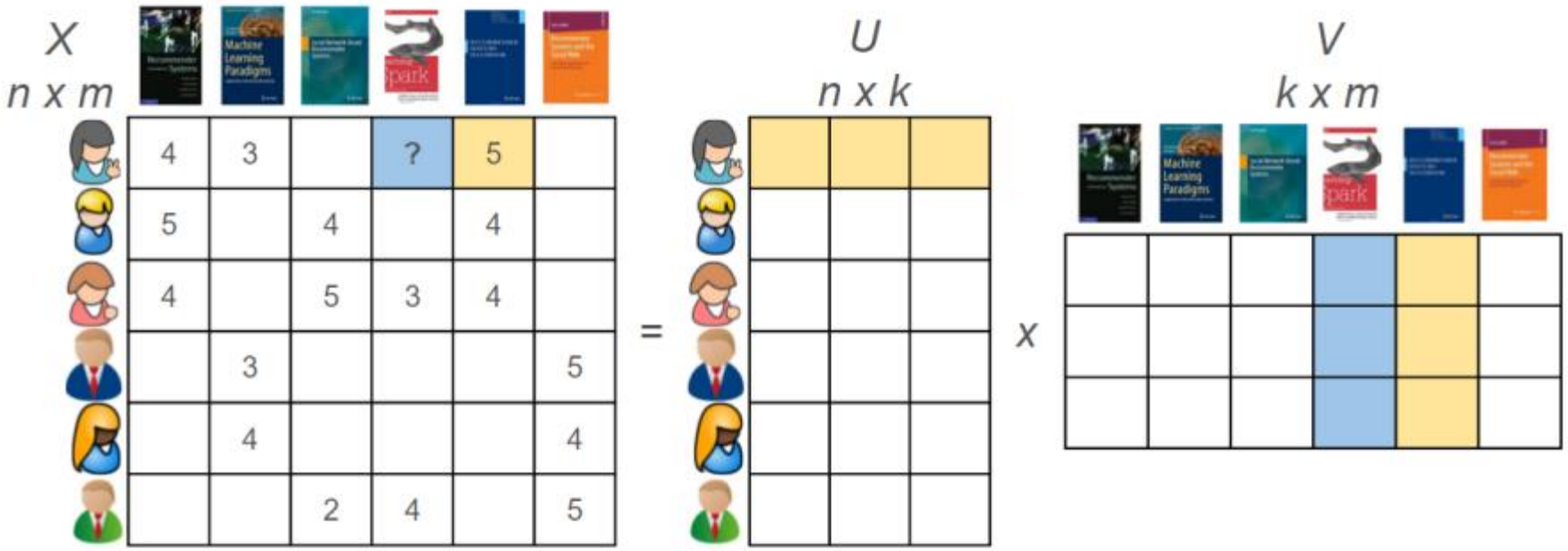
There's some evidence that this helps.

Third idea: Use SVD to map the enormous space into a lower-dimensional space, and find nearest neighbours there.

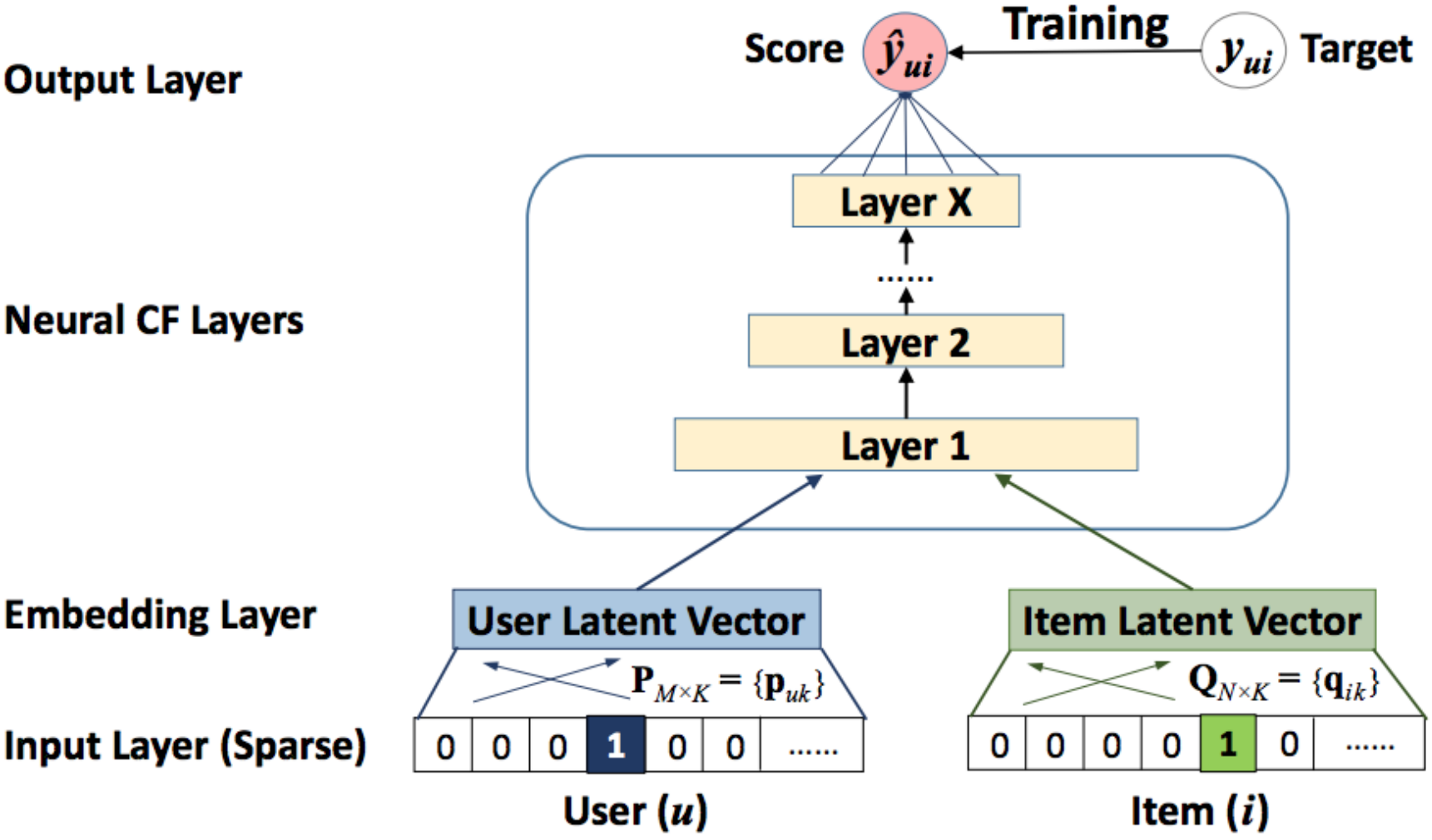
This requires a large precomputation, but is much faster for each individual item.

Note connections to search engines – people-item matrix.

BUT it's like searching on a single word – you don't tend to get great results.


X
n x m
4 3 ? 5
5 4 4
4 5 3 4
3 5
4 4
2 4 5
=
U
n x k
x
V
k x m

- Neural Matrix Factorization



Fourth idea: compute the item-item similarities

Item relationships change much less frequently than people's preferences, so the computation can be done periodically.

Look at each item a new individual likes. Then choose other items that are ranked similarly by other people and recommend them.

Can the system be gamed to improve the ranking of some items?

Obviously basing rankings on purchases makes this difficult, but it's a big problem for systems that collect opinions.

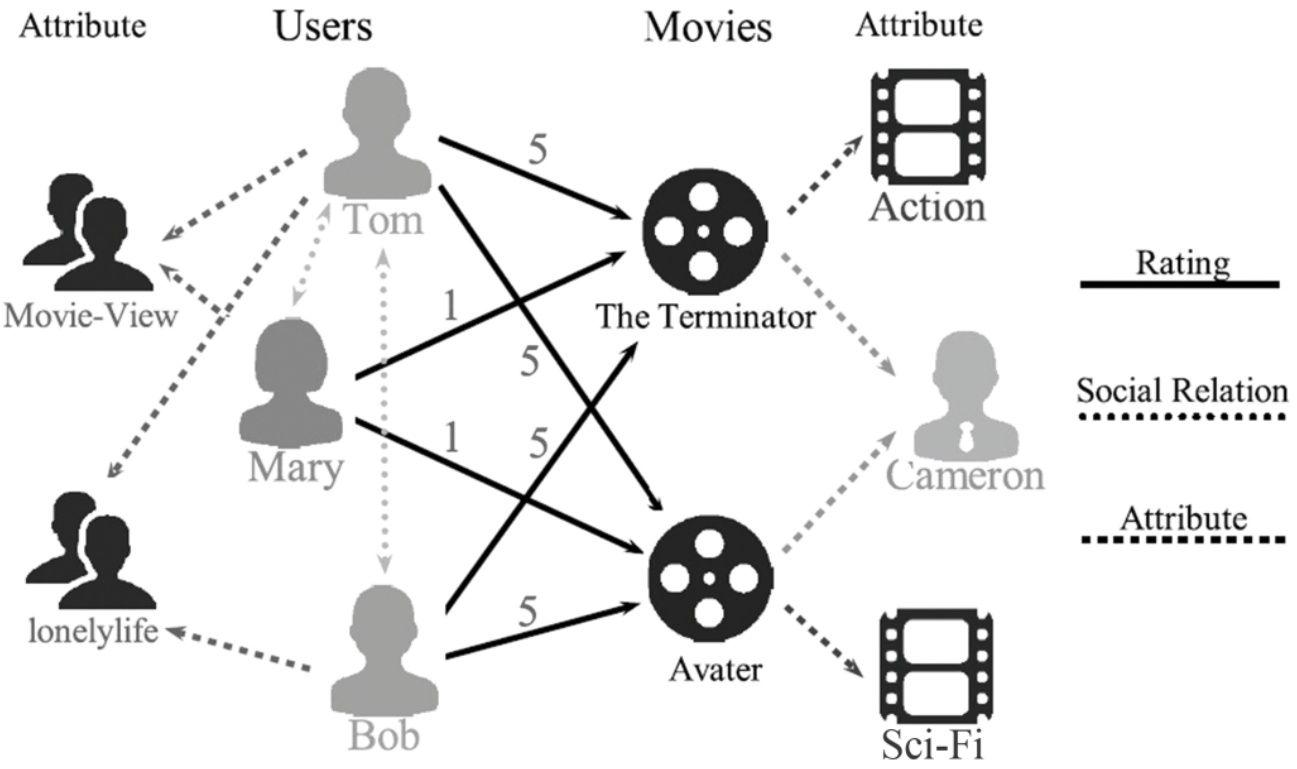
The problem is symmetric so the quality of those providing the rankings can also be calculated – reputation systems.

Is privacy a concern?

Do your opinions reveal too much about you?

Should you let organizations have access to your opinions, even when you benefit from it?

HIN – Review Network



Applications:

* recommendations for movies, restaurants, books, DVDs, etc.

* quality assessments, e.g. Slashdot

* resource discovery for complex systems

* publish-subscribe systems, RSS

* services discovery for mobile individuals