Collaborative Filtering

Jay Urbain, Ph.D.
Electrical Engineering and Computer Science Department
Milwaukee School of Engineering

What is Machine Learning?

- Subfield of Artificial Intelligence (AI) that allow computers to learn.

- *What does it mean to learn?*

What does it mean to learn?

- Improve our performance by interacting with our environment.
  - Build model of the world we interact with to improve future performance.

- *How do we interact with our environment to learn?*

How do we interact with our environment to learn a model?

- Supervised learning
  - Provide set of examples with answers
- Unsupervised learning
  - Group items based on similar attributes
- Reinforcement learning
  - Make choices, reinforce good decisions with rewards, update model, continue

Many important real life examples

- Biotech
- Financial fraud detection
- Machine vision
- Product marketing
- National security
- Web search
- Language learning

Collective Intelligence

- We often learn by interacting with each other.
- Create a model of our belief of how the world behaves and evolves – prediction.
- *Confabulation* - human brain is constantly deciding among multiple competing predications

- *Can we do this with machines?*

Collective Intelligence

- Can we do this with intelligent machines? ***Sure***.
  - Google
  - Amazon
  - Netflix
  - Pandora
  - Futures markets
- All of these companies combine the behavior, preferences, or ideas of a *group* of people to create novel insights.

Collective Intelligence

- Collecting answers from a large group of people lets you draw statistical conclusions about the group that no individual member would have known by themselves.

- Building new conclusions from independent contributors is what *collective intelligence* is all about.

- *So how do we go about doing that?*
  - *Collaborative Filtering…*

Collaborative Filtering

- What is the low-tech way to get recommends for products, movies , interesting web sites, or entertaining things to do?

Collaborative Filtering

- What is the low-tech way to get recommends for products, movies , interesting web sites, or entertaining things to do?
- Collective Intelligence - *ask your friends!*
- Do some of your friends have better taste than others?
  - These friends are more likely to *influence* our decisions.

Collaborative Filtering

Lets say we want to use the combined "*wisdom*" of an extended group of friends to pick movies.

1) Collect preferences
   - Use the Web
   - Build a database of individual ratings on movies

username	title	rating	rating_norm
bellmangreent	Raiders of the lost ark	3	0.5
bellmangreent	2012	3	0.5
bellmangreent	Aeon Flux	4	0.75
bellmangreent	Avatar	5	1
bellmangreent	Batman Begins	5	1
bellmangreent	Bourne Identity	3	0.5
bellmangreent	Children of Men	5	1
bellmangreent	Die Hard	4	0.75
bellmangreent	Die Hard with a Ven...	4	0.75
bellmangreent	Die Harder	4	0.75
bellmangreent	District 9	5	1
bellmangreent	Donnie Darko	3	0.5
bellmangreent	Harry Potter and th...	3	0.5

Collaborative Filtering

2) Finding similar users

Need method for correlating choices:

- Euclidian Distance
- Pearson Correlation

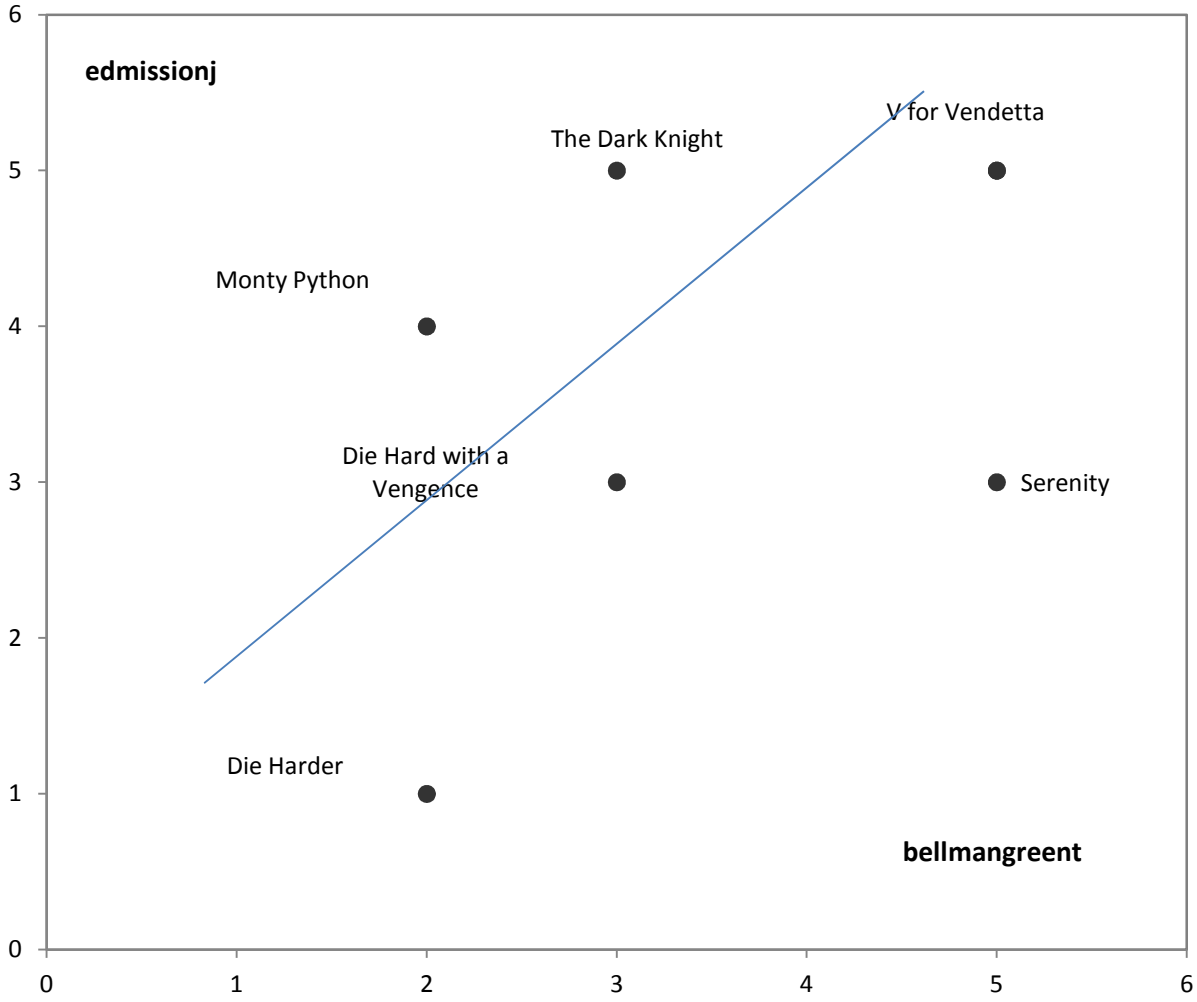
Compare Users

Select User 1: bellmangreent

Select User 2: edmissonj

User 1	User 2	Title	Rating 1	Rating 2
bellmangreent	edmissonj	Die Hard	4.0	4.0
bellmangreent	edmissonj	V for Vendetta	4.0	4.0
bellmangreent	edmissonj	Die Hard with a Vengence	4.0	4.0
bellmangreent	edmissonj	The Dark Knight	5.0	5.0
bellmangreent	edmissonj	Monty Python: The Holy Gr	4.0	4.0
bellmangreent	edmissonj	Die Harder	4.0	4.0
bellmangreent	edmissonj	Batman Begins	5.0	5.0
bellmangreent	edmissonj	Serenity	5.0	5.0
bellmangreent	edmissonj	Sherlock Holmes	4.0	4.0
bellmangreent	edmissonj	Iron Man	5.0	5.0
bellmangreent	edmissonj	Star Trek	5.0	4.0
bellmangreent	edmissonj	Matrix	5.0	4.0
bellmangreent	edmissonj	Avatar	5.0	4.0
bellmangreent	edmissonj	The Expendables	4.0	5.0
bellmangreent	edmissonj	District 9	5.0	4.0
bellmangreent	edmissonj	Bourne Identity	3.0	4.0
bellmangreent	edmissonj	The Hulk	3.0	4.0

Finding similar users – Euclidian



Collaborative Filtering

2) Finding similar users – Euclidian Distance
   1) Take the differences in each axis
   2) Square them & add them together
   3) Then take the square root of the sum

$$\frac{1}{1+\sum_i \sqrt{(r_{1,i}-r_{2,i})^2}}$$

select 1/(1+sqrt(sum(pow(r1.rating_norm-r2.rating_norm,2)))) as sc

Bellmangreent	**edmissonj**	0.535898384862245
bellmangreent	**bellmangreent**	1.00

Collaborative Filtering

2) Finding similar users –

- Euclidian is just an absolute distance measure
- Correlation is a measure of how well two sets of data fit on a straight line.
  - Corrects for "*grade inflation*"
  - Remove consistent differences

Collaborative Filtering

2) Finding similar users –

Pearson product-moment correlation coefficient

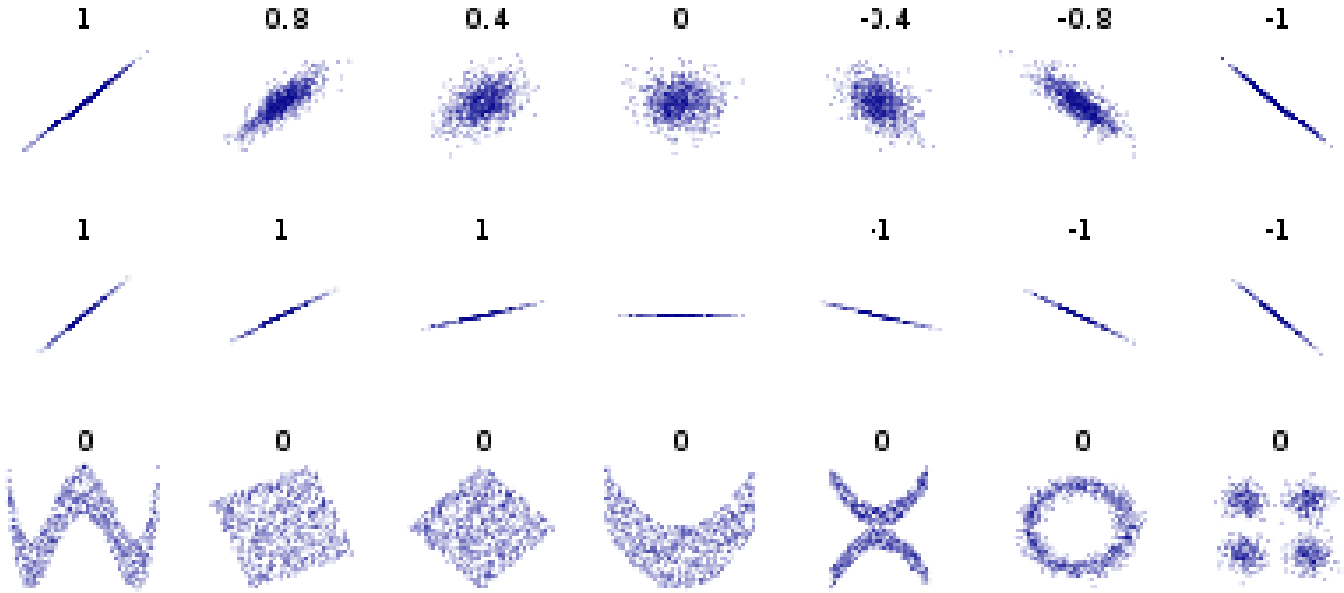
- Measure of the **correlation** (linear dependence) between two variables *X* and *Y*, giving a value between +1 and −1 inclusive.
- Widely used in the sciences as a measure of the strength of linear dependence between two variables.

Collaborative Filtering

2) Finding similar users –

Pearson product-moment correlation coefficient

- Covariance of the two variables divided by the product of their standard deviations.
- Corresponds to the cosine of the angle θ between two regression lines.
- Correlation reflects the non-linearity and direction of a linear relationship , but not the slope of that relationship



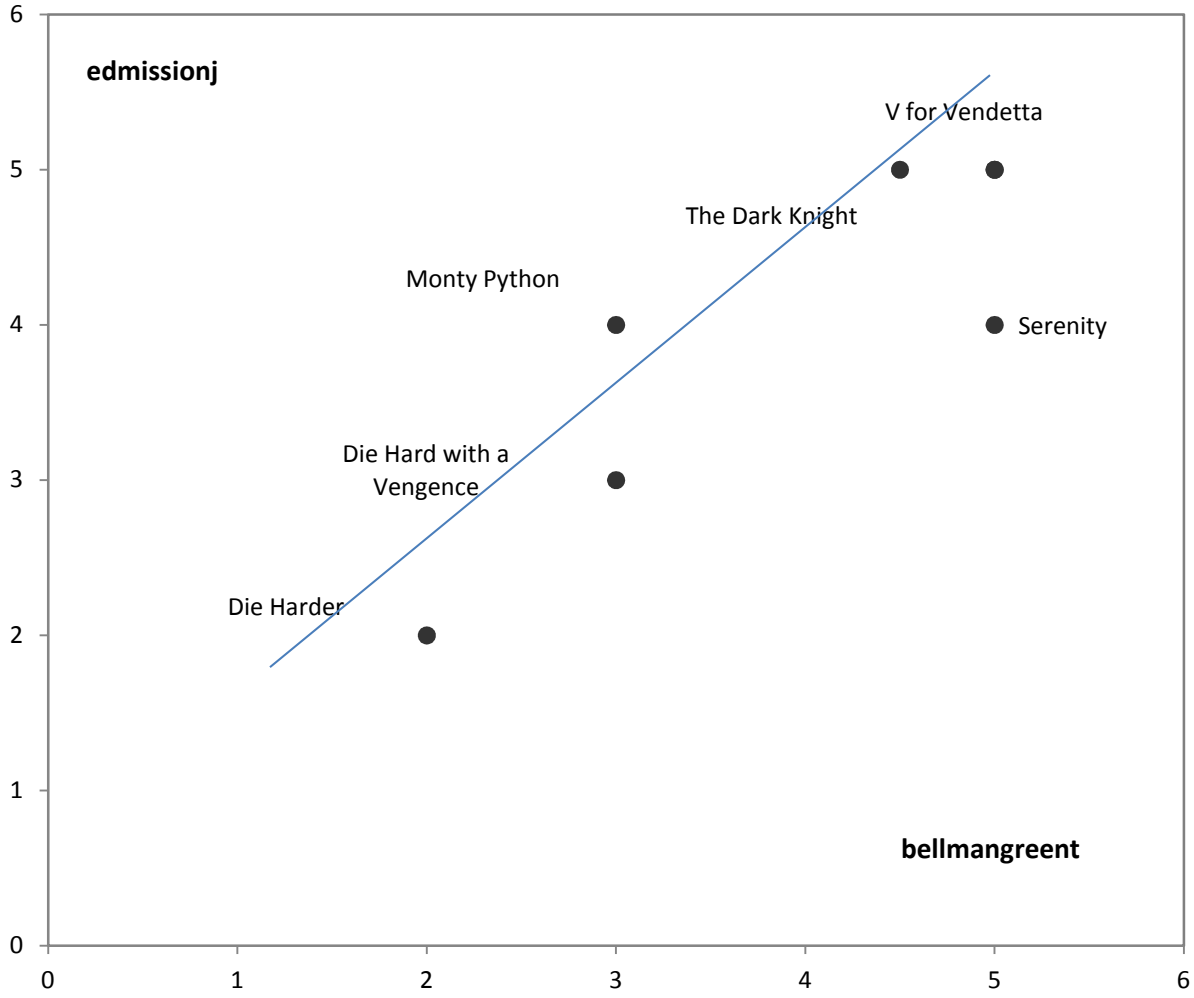
Collaborative Filtering

2) Finding similar users – Pearson Correlation Coefficient

$$\frac{\sum_i (r_{1,i} * r_{2,i}) - \left(\frac{\sum_i (r_{1,i}) * \sum_i (r_{2,i})}{n} \right)}{\sqrt{\left(\sum_i (r_{1,i})^2 - \left(\frac{\sum_i (r_{1,i})^2}{n} \right) \right) * \left(\sum_i (r_{2,i})^2 - \left(\frac{\sum_i (r_{2,i})^2}{n} \right) \right)}}$$

```
( sum(r1.rating_norm*r2.rating_norm) -
    (sum(r1.rating_norm)*sum(r2.rating_norm))/count(*) ) /
sqrt( ( sum(pow(r1.rating_norm,2))-
    (pow(sum(r1.rating_norm),2)/count(*)) ) * (
    sum(pow(r2.rating_norm,2))-(pow(sum(r2.rating_norm),2)/count(*))
    ) )
as sc
```

Finding similar users – Normalized (Pearson Correlation Coefficient)



Collaborative Filtering

3. Ranking the reviewers

- Now that we can compare any two people.
- Score everyone against a given person and find the closest matches.
- Allows us to find the reviewers whose taste are most similar to any individual.

Collaborative Filtering

3. Ranking the reviewers

create cross-product Pearson correlation coefficient similarity table

```
insert into sim
select u1.userid as userid1, u2.userid as userid2,
( sum(r1.rating_norm*r2.rating_norm) -
      (sum(r1.rating_norm)*sum(r2.rating_norm))/count(*) ) /
sqrt( ( sum(pow(r1.rating_norm,2))-(pow(sum(r1.rating_norm),2)/count(*)) ) * (
      sum(pow(r2.rating_norm,2))-(pow(sum(r2.rating_norm),2)/count(*)) ) )
as sc
from ratings r1, ratings r2, users u1, users u2, movies m
where r1.userid=u1.userid
and r2.userid=u2.userid
and r1.itemid=r2.itemid
and r1.itemid=m.movieid
and u1.username<>u2.username
group by u1.username, u2.username
order by u1.username, sc desc;
```

Collaborative Filtering

3. Ranking the reviewers

```
# rank the reviewers
select x.username1, max(x.sc) as maxsc
from (
select u1.username as username1, u2.username as username2, s.sc
from sim s, users u1, users u2
where s.userid1=u1.userid
and s.userid2=u2.userid) x
group by x.username1
order by maxsc;
```

Collaborative Filtering

4. Recommending Items

- Using the correlation coefficient that ranks users with respect to each other.
- Use this score to rank products.

```
# Finally! Make recommendations for a given user!
select u1.username, m.title, sum( r.rating_norm*s.sc ) / sum(s.sc) rating
from ratings r, users u1, users u2, movies m, sim s
where r.userid=u2.userid
and r.itemid=m.movieid
and s.userid1=u1.userid
and s.userid2=u2.userid
and u1.username='breuerr'
and r.itemid not in (
select r.itemid
from users u, ratings r
where r.userid=u.userid
and u.username='breuerr'
)
group by m.title
order by rating desc;
```

References

- Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second Edition, 2009.
- Anand Rajaraman, Jeffrey D. Ullman. *Mining of Massive Datasets*. 2009.
- Toby Seagram*, Collective Intelligence*. 2007.