

INTRO to DATA SCIENCE

LECTURE 17: RECOMMENDATIONS

0. PRESENTATIONS DATA EXPLORATION FOR FINAL PROJECT

I. DIMENSIONALITY REDUCTION

II. SINGULAR VALUE DECOMPOSITION (SVD)

III. PRINCIPAL COMPONENT ANALYSIS (PCA)

IV. NOTEBOOK EXAMPLES & EXERCISES

I. RECOMMENDATION SYSTEMS

II. CONTENT-BASED FILTERING

III. COLLABORATIVE FILTERING

IV. MATRIX FACTORIZATION (ILLUSTRATIVE EXAMPLE)

V. THE NETFLIX PRIZE

- **EXPLAIN THE USE OF RECOMMENDATION SYSTEMS**
- **DISCUSS SEVERAL FAMILIAR EXAMPLES**
- **DESCRIBE THE UNDERLYING CONCEPTS, INCLUDING COLLABORATIVE & CONTENT-BASED FILTERING**
- **IMPLEMENT A RECOMMENDATION SYSTEM IN PYTHON**

I. RECOMMENDATION SYSTEMS

*The purpose of a **recommendation system** is to predict which new items are relevant for a user.*

*The purpose of a **recommendation system** is to predict which new items are relevant for a user.*

This predicted user-item rating is produced by analyzing other user-item ratings, and sometimes item characteristics, too, to provide personalized recommendations to users.

Let's look at a few examples of recommendations and try to guess how the recommendations were made

Recommendations for You in Books



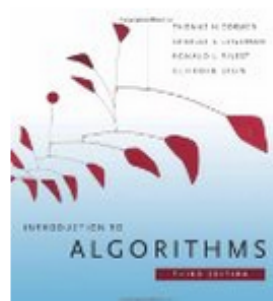
Cracking the Coding Interview: 150...

➤ Gayle Laakmann McDowell
Paperback

★★★★★ (166)

~~\$39.95~~ **\$23.22**

Why recommended?



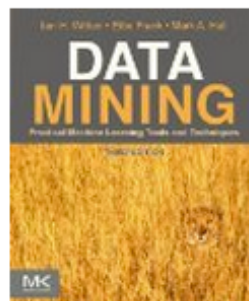
Introduction to Algorithms
Thomas H. Cormen, Charles E...

Hardcover

★★★★☆ (85)

~~\$92.00~~ **\$80.00**

Why recommended?



Data Mining: Practical Machine...

➤ Ian H. Witten, Eibe Frank, Mark A. Hall
Paperback

★★★★☆ (27)

~~\$69.95~~ **\$42.09**

Why recommended?



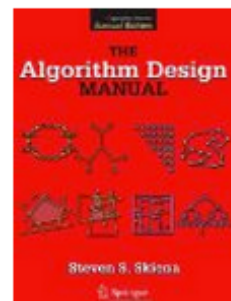
Elements of Programming Interviews...

➤ Amit Prakash, Adnan Aziz, Tsung-Hsien Lee
Paperback

★★★★☆ (25)

~~\$29.99~~ **\$26.18**

Why recommended?



The Algorithm Design Manual

➤ Steve Skiena
Paperback

★★★★☆ (47)

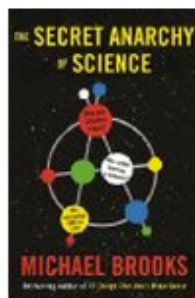
~~\$89.95~~ **\$71.84**

Why recommended?

Inspired by Your Wish List

You wished for

Customers who viewed this also viewed

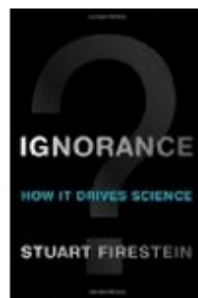


The Secret Anarchy of Science

► Michael Brooks

Paperback

★★★★☆ (6)



Ignorance: How It Drives Science

► Stuart Firestein

Hardcover

★★★★☆ (31)

~~\$21.95~~ **\$13.02**



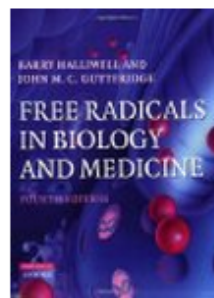
13 Things that Don't Make Sense: The...

► Michael Brooks

Paperback

★★★★☆ (65)

~~\$15.95~~ **\$12.49**



Free Radicals in Biology and Medicine

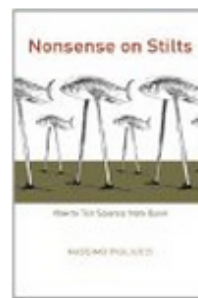
Barry Halliwell, John

Gutteridge

Paperback

★★★★★ (6)

~~\$90.00~~ **\$75.78**



Nonsense on Stilts: How to Tell...

► Massimo Pigliucci

Paperback

★★★★☆ (35)

~~\$20.00~~ **\$11.94**

TV Shows

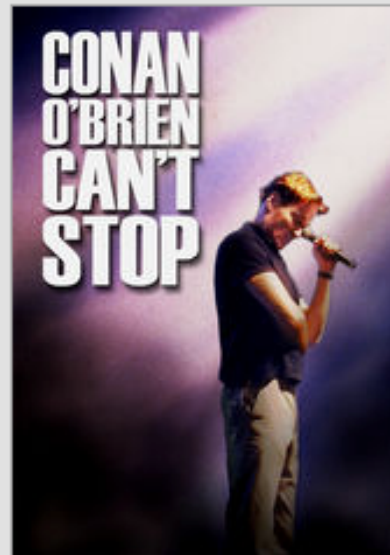
Your **taste preferences**
created this row.

TV Shows.

As well as your interest in...



Because you watched 30 Rock






Recommended for you because you watched

[Sugar Minott - Oh Mr Dc \(Studio One\)](#)



Mikey Dread - Roots and Culture

 by klaxonklaxon · 1,164,133 views

Lyrics:

Now here comes a special request
To each and everyone




Recommended for you because you watched

[Thelonious Monk Quartet - Monk In Denmark](#)



Bill Evans Portrait in Jazz (Full Album)

 by hansgy1 · 854,086 views

Bill Evans Portrait in Jazz 1960

1. Come Rain or Come Shine - 3.19 (0:00)
2. Autumn Leaves - 5.23 (3:24)




Recommended for you because you watched

[Bob Marley One Drop](#)



Bob Marley - She's gone

 by Dionysios29 · 1,058,704 views

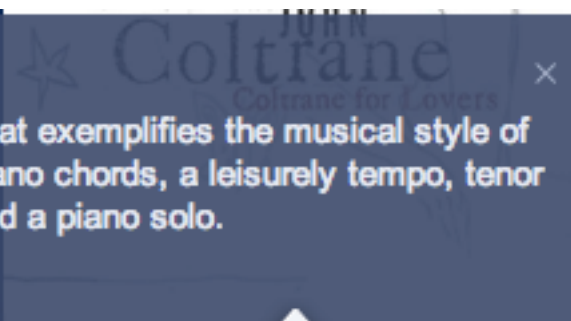
This is one of the eleven songs of album Kaya that Bob Marley and The Wailers creative in 1978.

Lyrics:

John Coltrane Radio

To start things off, we'll play a song that exemplifies the musical style of John Coltrane which features block piano chords, a leisurely tempo, tenor sax head, a melodic tenor sax solo and a piano solo.

That's not what I wanted, [delete this station](#)



MOST E-MAILED

RECOMMENDED FOR YOU

1. **How Big Data Is Playing Recruiter for Specialized Workers**
2. SLIPSTREAM
When Your Data Wanders to Places You've Never Been
3. MOTHERLODE
The Play Date Gun Debate
4. **For Indonesian Atheists, a Community of Support Amid Constant Fear**
5. **Justice Breyer Has Shoulder Surgery**
6. BILL KELLER
Erasing History



Recommendations

Knewton figures out what each student knows, then recommends the exact activities she should focus on next to meet learning goals.

So how do these examples actually work?

RECOMMENDATION SYSTEMS

There are two general approaches to “recsys” design:

RECOMMENDATION SYSTEMS

There are two general approaches to “recsys” design:

*In **content-based filtering**, items are mapped into a feature space, and recommendations depend on item characteristics.*

RECOMMENDATION SYSTEMS

There are two general approaches to “recsys” design:

*In **content-based filtering**, items are mapped into a feature space, and recommendations depend on item characteristics.*

*In contrast, the only data under consideration in **collaborative filtering** are user-item ratings, and recommendations depend on user preferences.*

II. CONTENT-BASED FILTERING

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

There are two approaches

- 1. Map users and items to same feature space, compute distance between a user and item*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

There are two approaches

- 1. Map users and items to same feature space, compute distance between a user and item*
- 2. Create features from user-item pairs and use a supervised algorithm to predict rating*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

There are two approaches

- 1. Map users and items to same feature space, compute distance between a user and item**
- 2. Create features from user-item pairs and use a supervised algorithm to predict rating*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

Item vectors *measure the degree to which the item is described by each feature*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

Item vectors *measure the degree to which the item is described by each feature, and* **user vectors** *measure a user's preferences for each feature.*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

Item vectors *measure the degree to which the item is described by each feature, and* **user vectors** *measure a user's preferences for each feature.*

Ratings are generated by taking **dot products** *of user & item vectors.*

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

Item vectors *measure the degree to which the item is described by each feature, and* **user vectors** *measure a user's preferences for each feature.*

NOTE

The idea is that users like items that are **similar** to other items they've consumed.

*Ratings are generated by taking **dot products** of user & item vectors.*

Movies

Finding Nemo

Mission Impossible

Jiro Dreams of Sushi

Movies

Finding Nemo

Mission Impossible

Jiro Dreams of Sushi

blockbuster
family
famous actors

Movies

	blockbuster	family	famous actors
<i>Finding Nemo</i>	5	5	2
<i>Mission Impossible</i>	3	-5	5
<i>Jiro Dreams of Sushi</i>	-4	-5	-5

Movies

	blockbuster	family	famous actors
<i>Finding Nemo</i>	5	5	2
<i>Mission Impossible</i>	3	-5	5
<i>Jiro Dreams of Sushi</i>	-4	-5	-5

Users

<i>Jason</i>	-3	2	-2
--------------	----	---	----

Movies	blockbuster	family	famous actors
<i>Finding Nemo</i>	5	5	2
<i>Mission Impossible</i>	3	-5	5
<i>Jiro Dreams of Sushi</i>	-4	-5	-5
Users			
<i>Jason</i>	-3	2	-2

*Jason doesn't like blockbusters
or movies with famous actors,
but is fine with family movies*

Movies	blockbuster family famous actors			Predicted ratings
<i>Finding Nemo</i>	5	5	2	$-3 \times 5 + 2 \times 5 - 2 \times 2 = -9$
<i>Mission Impossible</i>	3	-5	5	$-3 \times 3 + 2 \times -5 - 2 \times 5 = -20$
<i>Jiro Dreams of Sushi</i>	-4	-5	-5	$-3 \times -4 + 2 \times -5 - 2 \times -5 = 12$
Users				
<i>Jason</i>	-3	2	-2	

Moviesblockbuster
family
famous actors*Finding Nemo*

5 5 2

Mission Impossible

3 -5 5

Jiro Dreams of Sushi

-4 -5 -5

Predicted ratings

$$-3 \times 5 + 2 \times 5 - 2 \times 2 = -9$$

$$-3 \times 3 + 2 \times -5 - 2 \times 5 = -20$$

$$-3 \times -4 + 2 \times -5 - 2 \times -5 = 12$$

Users*Jason*

-3 2 -2

Movies

Finding Nemo

Mission Impossible

Jiro Dreams of Sushi

blockbuster
family
famous actors

5 5 2

3 -5 5

-4 -5 -5

Predicted ratings

$$-3 \times 5 + 2 \times 5 - 2 \times 2 = -9$$

$$-3 \times 3 + 2 \times -5 - 2 \times 5 = -20$$

$$-3 \times -4 + 2 \times -5 - 2 \times 5 = -12$$

NOTE

In practice, these predictions would be proportional to **deviations** from some global average rating (hence the negative values).

Users

Jason

-3 2 -2

Content-based filtering *begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.*

There are two approaches

- 1. Map users and items to same feature space, compute distance between a user and item*
- 2. Create features from user-item pairs and use a supervised algorithm to predict rating**

Movies	blockbuster	family	famous actors
<i>Finding Nemo</i>	5	5	2
<i>Mission Impossible</i>	3	-5	5
<i>Jiro Dreams of Sushi</i>	-4	-5	-5
<i>The Godfather</i>	5	-5	5
<i>Inequality For All</i>	-5	-1	-5

Movies	blockbuster	family	famous actors	Jason's rating
<i>Finding Nemo</i>	5	5	2	2.5
<i>Mission Impossible</i>	3	-5	5	2.0
<i>Jiro Dreams of Sushi</i>	-4	-5	-5	4.5
<i>The Godfather</i>	5	-5	5	1.0
<i>Inequality For All</i>	-5	-1	-5	

Movies	blockbuster	family	famous actors	Jason's rating
<i>Finding Nemo</i>	5	5	2	2.5
<i>Mission Impossible</i>	3	-5	5	2.0
<i>Jiro Dreams of Sushi</i>	-4	-5	-5	4.5
<i>The Godfather</i>	5	-5	5	1.0
<i>Inequality For All</i>	-5	-1	-5	3.5 predicted

Movies	blockbuster	family	famous actors	Jason's rating	
<i>Finding Nemo</i>	5	5	2	2.5	-
<i>Mission Impossible</i>	3	-5	5	2.0	-
<i>Jiro Dreams of Sushi</i>	-4	-5	-5	4.5	+
<i>The Godfather</i>	5	-5	5	1.0	-
<i>Inequality For All</i>	-5	-1	-5		

Movies	blockbuster	family	famous actors	Jason's rating	
<i>Finding Nemo</i>	5	5	2	2.5	-
<i>Mission Impossible</i>	3	-5	5	2.0	-
<i>Jiro Dreams of Sushi</i>	-4	-5	-5	4.5	+
<i>The Godfather</i>	5	-5	5	1.0	-
<i>Inequality For All</i>	-5	-1	-5		+ predicted

Movies	blockbuster	family	famous actors	Jason's rating	
<i>Finding Nemo</i>	5	5	2	2.5	-
<i>Mission Impossible</i>	3	-5	5	2.0	-
<i>Jiro Dreams of Sushi</i>	-4	-5	-5	4.5	+
<i>The Godfather</i>	5	-5	5	1.0	-
<i>Inequality For All</i>	-5	-1	-5	+ predicted	

NOTE

This is not practical when you have only a few ratings from the user. We don't use other users' ratings for inference.

One notable example of content-based filtering is Pandora, which maps songs into a feature space using features (or “genes”) designed by the Music Genome Project.

Using song vectors that depend on these features, Pandora can create a station with music having similar properties to a song the user selects.

About The Music Genome Project®

We believe that each individual has a unique relationship with music – no one else has tastes exactly like yours. So delivering a great radio experience to each and every listener requires an incredibly broad and deep understanding of music. That's why Pandora is based on the Music Genome Project, the most sophisticated taxonomy of musical information ever collected. It represents over ten years of analysis by our trained team of musicologists, and spans everything from this past Tuesday's new releases all the way back to the Renaissance and Classical music.

Each song in the Music Genome Project is analyzed using up to 450 distinct musical characteristics by a trained music analyst. These attributes capture not only the musical identity of a song, but also the many significant qualities that are relevant to understanding the musical preferences of listeners. The typical music analyst working on the Music Genome Project has a four-year degree in music theory, composition or performance, has passed through a selective screening process and has completed intensive training in the Music Genome's rigorous and precise methodology. To qualify for the work, analysts must have a firm grounding in music theory, including familiarity with a wide range of styles and sounds.

Content-based filtering has some difficulties

Content-based filtering has some difficulties

- ▶ *need to map each item into a feature space (usually by hand!)*
- ▶ *recommendations are limited in scope*
(items must be similar to each other)
- ▶ *hard to create cross-content recommendations*
(e.g., books/music films...this would require comparing elements from different feature spaces!)

III. COLLABORATIVE FILTERING

Collaborative filtering *refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are only interested in the existing user-item ratings themselves.*

Collaborative filtering *refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are only interested in the existing user-item ratings themselves.*

In this case, our dataset is a ratings matrix whose columns correspond to items, and whose rows correspond to users.

Collaborative filtering *refers to a family of methods for predicting ratings where instead of thinking about users and items in the feature space, we are only interested in the existing user-item pairs themselves.*

NOTE

The idea here is that users get value from recommendations based on other users with similar tastes.

In this case, our dataset is a ratings matrix whose columns correspond to items, and whose rows correspond to users.

Collaborative filtering can be done in two different ways.

Collaborative filtering can be done in two different ways.

Item-based

Model-based

Collaborative filtering can be done in two different ways.

Item-based CF *uses ratings data to create an item-item similarity matrix.*

Collaborative filtering can be done in two different ways.

Item-based CF *uses ratings data to create an item-item similarity matrix.*

Recommendations are then made to a user for items most similar to those that the user has already rated highly.

Collaborative filtering can be done in two different ways.

Item-based CF *uses ratings data to create an item-item similarity matrix.*

Recommendations are then made to a user for items most similar to those that the user has already rated highly.

*This is also called **memory-based CF**.*

Collaborative filtering can be done in two different ways.

Item-based CF *uses ratings data to create an item-item similarity matrix.*

Recommendations are then made to a user for items most similar to those that the user has already rated highly.

*This is also called **memory-based CF**.*

NOTE

This is equivalent to a clustering problem in the space of column vectors (items)!

Item-based collaborative filtering is a ***neighborhood method***.

Collaborative filtering can be done in two different ways.

Item-based CF *uses ratings data to create an item-item similarity matrix.*

Recommendations are then made to a user for items most similar to those that the user has already rated highly.

*This is also called **memory-based CF**.*

NOTE

This is equivalent to a cluster analysis of the vectors.

NOTE

User-based collaborative filtering is possible, too, but is less efficient, since there are typically more users than items.

Customers Who Bought This Item Also Bought

 Pitch Dark (NYRB Classics)

› Renata Adler

Paperback

\$11.54



How Literature Saved My Life

› David Shields

★★★★☆ (60)

Hardcover

\$18.08



Bleeding Edge

Thomas Pynchon

Hardcover

\$18.05



The Flamethrowers: A Novel

› Rachel Kushner

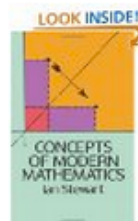
★★★★☆ (17)

Hardcover

\$15.79

amazon.com

Recommended for You



[Concepts of Modern Mathematics](#)

by Ian Stewart (February 1, 1995)

In Stock

List Price: ~~\$14.95~~

Price: **\$8.94**

87 used & new from **\$5.99**

Add to Cart

Add to Wish List

Because you purchased...



[Mathematics: Its Content, Methods and Meaning \(Dover Books on Mathematics\)](#)

by A. D. Aleksandrov (Author), et al.

Note that item-based collaborative filtering is different than content-based filtering:

Though we're making recommendations based on items, we are not embedding the items in a feature space.

Note that item-based collaborative filtering is different than content-based filtering:

Though we're making recommendations based on items, we are not embedding the items in a feature space.

Neighborhood methods such as item-based CF are popular and easy to understand, but they don't scale well.

Model-based *collaborative filtering* abandons the neighborhood approach and applies other techniques to the ratings matrix.

Model-based *collaborative filtering abandons the neighborhood approach and applies other techniques to the ratings matrix.*

The most popular model-based CF techniques use matrix decomposition techniques to find deeper structure in the ratings data.

Model-based *collaborative filtering* abandons the neighborhood approach and applies other techniques to the ratings matrix.

The most popular model-based CF techniques use matrix decomposition techniques to find deeper structure in the ratings data.

*For example, we could decompose the ratings matrix via SVD to reduce the dimensionality and extract **latent variables**.*

Once we identify the latent variables in the ratings matrix, we can express both users and items in terms of these latent variables.

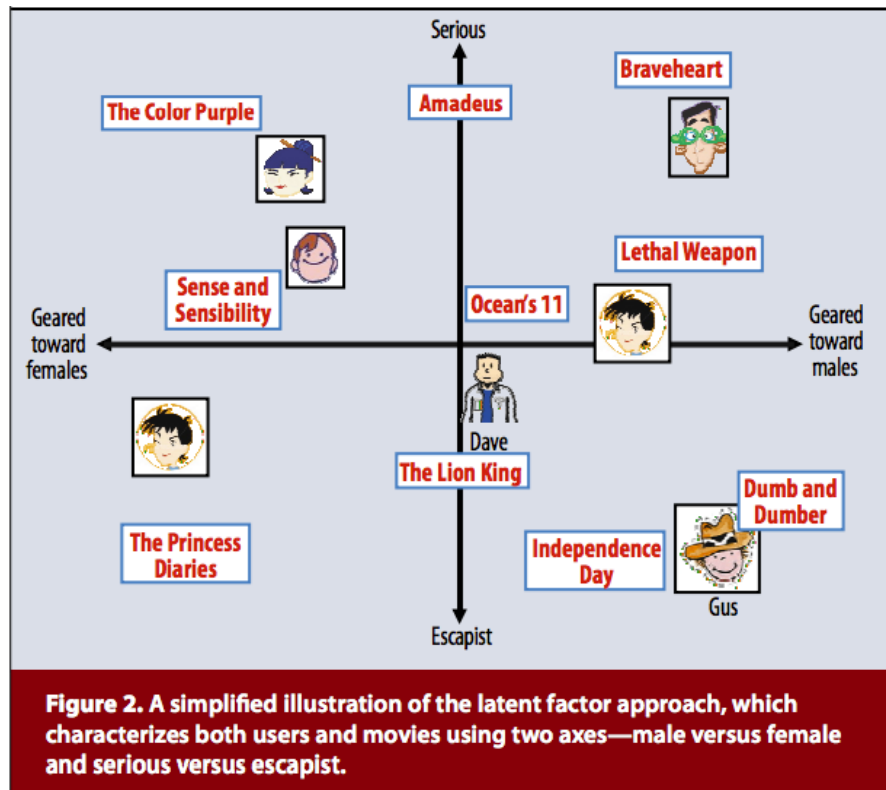
Once we identify the latent variables in the ratings matrix, we can express both users and items in terms of these latent variables.

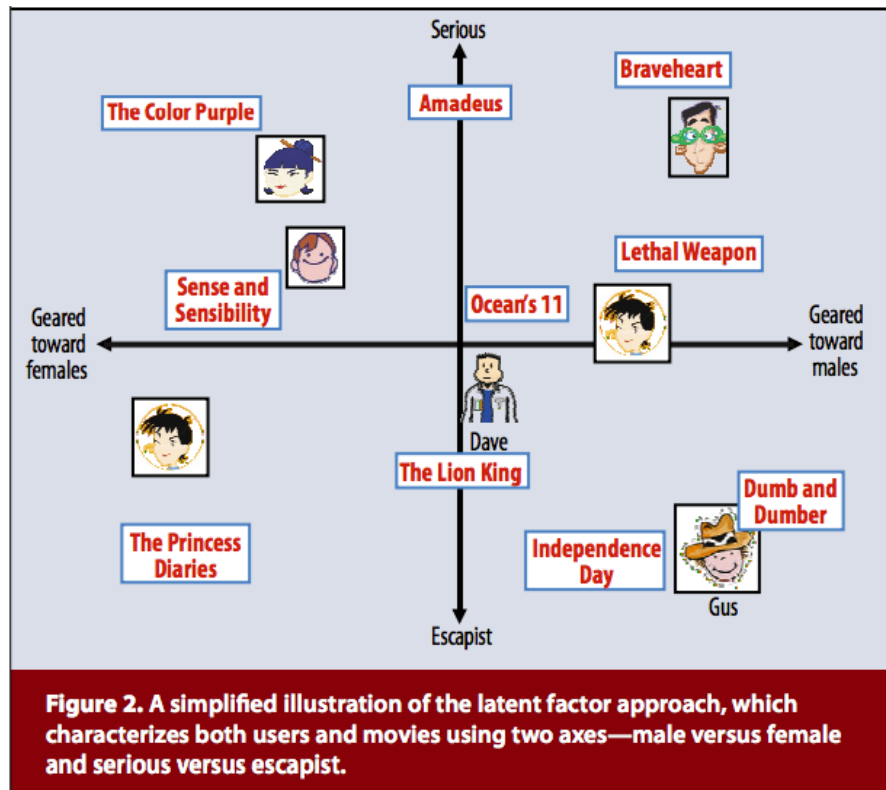
As before, values in the item vectors represent the degree to which an item exhibits a given feature, and values in the user vectors represent user preferences for a given feature.

Once we identify the latent variables in the ratings matrix, we can express both users and items in terms of these latent variables.

As before, values in the item vectors represent the degree to which an item exhibits a given feature, and values in the user vectors represent user preferences for a given feature.

Ratings are constructed by taking dot products of user & item vectors in the latent feature space.



**NOTE**

The dimensions in the latent feature space are *inferred*, and not pre-defined. It is not always clear what these dimensions represent.

This approach is domain independent, and requires no explicit user or item profiles to be created.

This approach is domain independent, and requires no explicit user or item profiles to be created.

It combines predictive accuracy, scalability, and enough flexibility for practical modeling (we'll see what this means in a moment).

This approach is domain independent, and requires no explicit user or item profiles to be created.

It combines predictive accuracy, scalability, and enough flexibility for practical modeling (we'll see what this means in a moment).

Since the conclusion of the Netflix prize, these latent factor methods for collaborative filtering have been regarded as the state-of-the-art.

But they do have some drawbacks:

- ▶ *lots of high-dimensional ratings data needed*
- ▶ *data is typically very sparse*
(in the Netflix prize dataset, ~99% of possible ratings were missing)
- ▶ *susceptible to fraud (e.g. shilling attacks)*
- ▶ **cold start problem:** *need lots of data on new user or item before recommendations can be made*

*The cold start problem arises because we've been relying only on ratings data, or on **explicit feedback** from users.*

*The cold start problem arises because we've been relying only on ratings data, or on **explicit feedback** from users.*

Until a user rates several items, we don't know anything about her preferences!

*The cold start problem arises because we've been relying only on ratings data, or on **explicit feedback** from users.*

Until a user rates several items, we don't know anything about her preferences!

*We can get around this by enhancing our recommendations using **implicit feedback**, which may include things like item browsing behavior, search patterns, purchase history, etc.*

While explicit feedback (ratings, likes, purchases) leads to high quality ratings, the data is sparse and cold starts are problematic.

While explicit feedback (ratings, likes, purchases) leads to high quality ratings, the data is sparse and cold starts are problematic.

Meanwhile implicit feedback (browsing behavior, etc) leads to less accurate ratings, but the data is much more dense (and less invasive to collect).

While explicit feedback (ratings, likes, purchases) leads to high quality ratings, the data is sparse and cold starts are problematic.

Meanwhile implicit feedback (browsing behavior, etc) leads to less accurate ratings, but the data is much more dense (and less invasive to collect).

Implicit feedback can help to infer user preferences when explicit feedback is not available, therefore easing the cold start problem.

Hybrid filtering methods *provide another way to get around the cold start problem by combining filtering methods (e.g., by using content-based info to “boost” a collaborative model).*

Hybrid filtering methods *provide another way to get around the cold start problem by combining filtering methods (e.g., by using content-based info to “boost” a collaborative model).*

This content-based info can be item-based as above, or even user-based (e.g., demographic info).

Hybrid filtering methods *provide another way to get around the cold start problem by combining filtering methods (e.g., by using content-based info to “boost” a collaborative model).*

This content-based info can be item-based as above, or even user-based (e.g., demographic info).

Hybrid methods can also make the data sparsity issue easier to deal with, by broadening the set of features under consideration.

INTRO TO DATA SCIENCE

OVERVIEW

Content-based filtering

Mapping items and users into feature space

Collaborative filtering

Using user-item rating matrix only

User-item similarity measured by dot-product

Content-based filtering

Mapping items and users into feature space

Collaborative filtering

Using user-item rating matrix only

Content-based filtering

Mapping items and users into feature space

User-item similarity measured by dot-product

*Create features from user-item pairs and
use a supervised algorithm to predict rating*

Collaborative filtering

Using user-item rating matrix only

Content-based filtering

Mapping items and users into feature space

Collaborative filtering

Using user-item rating matrix only

User-item similarity measured by dot-product

*Create features from user-item pairs and
use a supervised algorithm to predict rating*

Item-based

*Item-item similarity measured by dot-product
Recommend items similar to highly-rated items*

Content-based filtering

Mapping items and users into feature space

User-item similarity measured by dot-product

*Create features from user-item pairs and
use a supervised algorithm to predict rating*

Collaborative filtering

Using user-item rating matrix only

Item-based

*Item-item similarity measured by dot-product
Recommend items similar to highly-rated items*

Model-based

Express items and users in latent feature space

Content-based filtering

Mapping items and users into feature space

User-item similarity measured by dot-product

*Create features from user-item pairs and
use a supervised algorithm to predict rating*

Collaborative filtering

Using user-item rating matrix only

Item-based

*Item-item similarity measured by dot-product
Recommend items similar to highly-rated items*

Model-based

Express items and users in latent feature space

NOTE

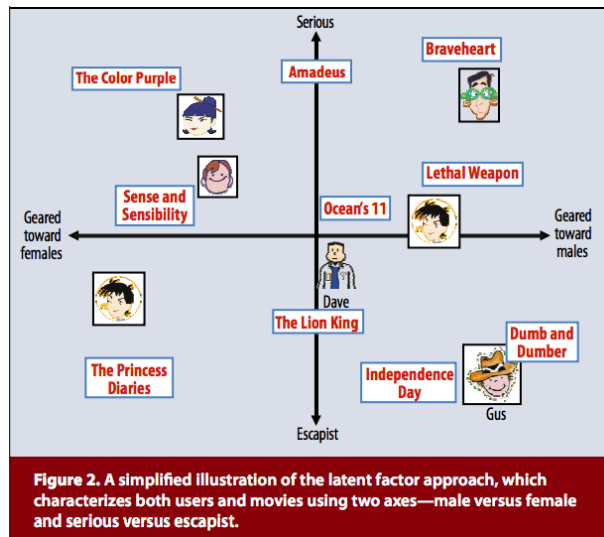
Please see the notebook about recommending beers for examples of implementations of a few of those.

IV. MATRIX FACTORIZATION

(SIT BACK & RELAX)

Matrix factorization decomposes the ratings matrix and maps users and items into a low-dimensional vector space spanned by a basis of latent factors.

Matrix factorization decomposes the ratings matrix and maps users and items into a low-dimensional vector space spanned by a basis of latent factors.



Matrix factorization decomposes the ratings matrix and maps users and items into a low-dimensional vector space spanned by a basis of latent factors.

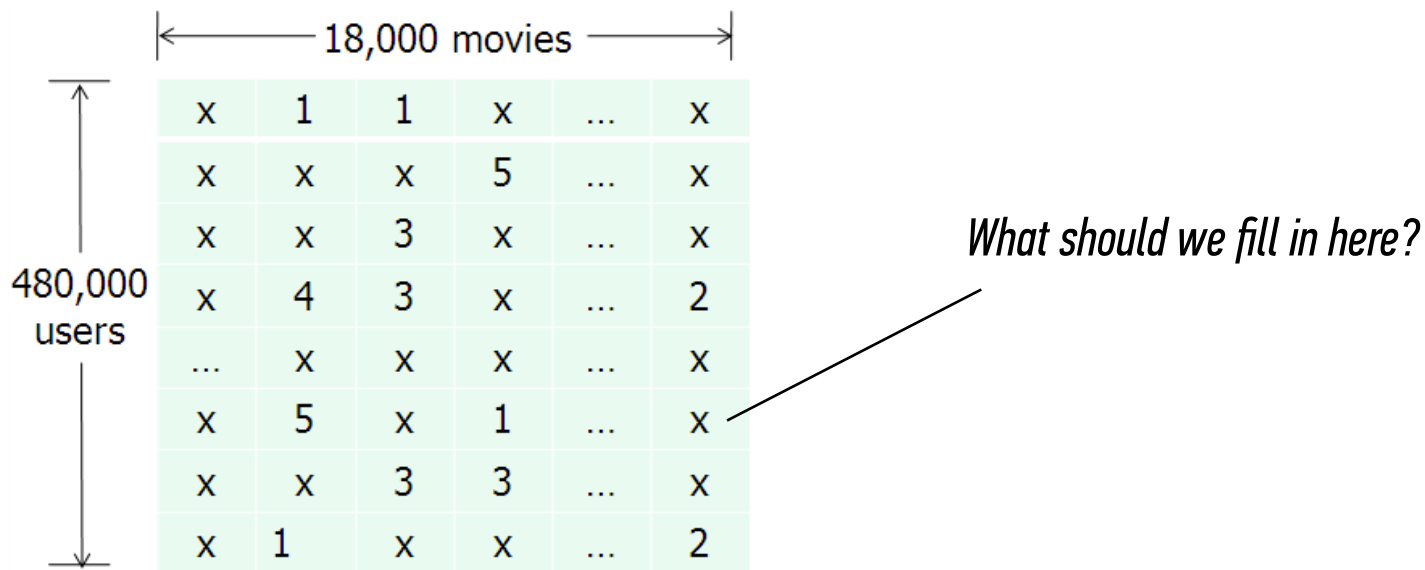
Predicted ratings are given by inner products in this space, so for user u and item i we can write:

$$\hat{r}_{ui} = q_i^T r_u$$

predicted rating — \hat{r}_{ui} — q_i^T — r_u — *user vector*
item vector

Factoring the ratings matrix via SVD leads to difficulty, since the matrix is typically sparse and therefore our information is incomplete.

Factoring the ratings matrix via SVD leads to difficulty, since the matrix is typically sparse and therefore our information is incomplete.



Factoring the ratings matrix via SVD leads to difficulty, since the matrix is typically sparse and therefore our information is incomplete.

Interpolating missing values is an expensive process and can lead to inaccurate predictions, so we need another way to perform this factorization.

One possibility is to learn the feature vectors using the observed ratings only. Since this dramatically reduces the size of the ratings matrix, we have to be careful to avoid overfitting.

One possibility is to learn the feature vectors using the observed ratings only. Since this dramatically reduces the size of the ratings matrix, we have to be careful to avoid overfitting.

We can learn these feature vectors by minimizing the loss function:

$$\min_{q,p} \sum_{(u,i) \in \kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

where κ denotes the set of known ratings, and λ is a hyperparameter.

One possibility is to learn the feature vectors using the observed ratings only. Since this dramatically reduces the size of the ratings have to be careful to avoid overfitting.

NOTE

The loss function has two unknowns (q, p) and is generally **not convex!**

This can be minimized using a method called **alternating least squares**.

We can learn these feature vectors by minimizing the loss f

$$\min_{q,p} \sum_{(u,i) \in \kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

where κ denotes the set of known ratings, and λ is a hyperparameter.

It turns out that much of the variation in observed ratings is due to user or item biases (e.g., some users are very critical, or some items are universally popular).

It turns out that much of the variation in observed ratings is due to user or item biases (e.g., some users are very critical, or some items are universally popular).

We can capture these biases in our model by generalizing \hat{r}_{ui} .

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T r_u$$

Here μ is a global average rating, b_i is the item bias, b_u is the user bias, and $q_i^T r_u$ is the user-item interaction.

With this generalization, our minimization problem becomes:

$$\min_{q,p,b} \sum_{(u,i) \in \kappa} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2)$$

With this generalization, our minimization problem becomes:

$$\min_{q,p,b} \sum_{(u,i) \in \kappa} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2)$$

Further modifications can be made to this model (incorporating implicit feedback, capturing temporal effects, attaching confidence scores to predictions), and you can look up the details in the references.

V. THE NETFLIX PRIZE

(KEEP RELAXIN')

The Netflix prize was a competition to see if anyone could make a 10% improvement to Netflix's recommendation system (accuracy measured by RMSE).

The Netflix prize was a competition to see if anyone could make a 10% improvement to Netflix's recommendation system (accuracy measured by RMSE).

The grand prize was \$1mm dollars, with annual \$50k progress prizes to the leader at the end of each year if the 10% threshold had not yet been met. Approx 50k teams participated from >180 countries.

The Netflix prize was a competition to see if anyone could make a 10% improvement to Netflix's recommendation system (accuracy measured by RMSE).

The grand prize was \$1mm dollars, with annual \$50k progress prizes to the leader at the end of each year if the 10% threshold had not yet been met. Approx 50k teams participated from >180 countries.

The ratings matrix contained >100mm numerical entries (1-5 stars) from ~500k users across ~17k movies. The data was split into train/quiz/test sets to prevent overfitting on the test data by answer submission (this was a clever idea!)

The competition began in 2006, and the grand prize was eventually awarded in 2009. The winning entry was a stacked ensemble of 100's of models (including neighborhood & matrix factorization models) that were blended using boosted decision trees.

The competition began in 2006, and the grand prize was eventually awarded in 2009. The winning entry was a stacked ensemble of 100's of models (including neighborhood & matrix factorization models) that were blended using boosted decision trees.

Ultimately, the competition ended in a photo finish. The winning strategy came down to last-minute team mergers & creative blending schemes to shave 3rd & 4th decimals off RMSE (concerns that would not be important in practice).

The competition began in 2006, and the grand prize was eventually awarded in 2009. The winning entry was a stacked ensemble of 100's of models (including neighborhood & matrix factorization models) that were blended using boosted decision trees.

Ultimately, the competition ended in a photo finish. The winning strategy came down to last-minute team mergers & creative blending schemes to shave 3rd & 4th decimals off RMSE (concerns that would not be important in practice).

The competition did much to spur interest and research advances in recsys technology, and the prize money was donated to charity.

Though they adopted some of the modeling techniques that emerged from the competition, Netflix never actually implemented the prizewinning solution.

Why do you think that's true?

INTRO TO DATA SCIENCE

DISCUSSION