Lecture 13: Collaborative Filtering, Missing & Relational Data

STATS 202: Data Mining and Analysis

Linh Tran

tranlm@stanford.edu

Department of Statistics
Stanford University

August 9, 2021

Announcements

- Homework 4 is due a week from Friday.
- Homework 3 is being graded.
- Final project submissions due in 2 weeks.
  - Write-up due the following Friday (last day of class).
- Office hours listed on course syllabus page.
- Panel of graduate research next Monday.

Outline

- Collaborative Filtering
- Missing data
- Relational data

The Netflix Prize

Goal: Predict user ratings (1 to 5 stars) for unwatched films

- 100M ratings of movies
- 18k movies and 48k users
- On average 5600 ratings / movie
- On average 208 ratings / user
- Data collected over several years
- Ratings are integers from 1 to 5

The Netflix Prize

Participant challenge: Reduce RMSE on new data by 10%

- Current was 0.951, so reduce to 0.856.
- New data may not have the same distributions as older data (Netflix is growing, more users and movies, fewer movies rated per user and per movie).

A baseline model

$$r_{ui} = \mu + b_u + b_i \tag{1}$$

Where

- μ is the item rating.
- b_i is an adjustment for that item.
- b_u is an adjustment for that user.

Models how “critical” a user is and how good a movie is, on average.

Collaborative Filtering

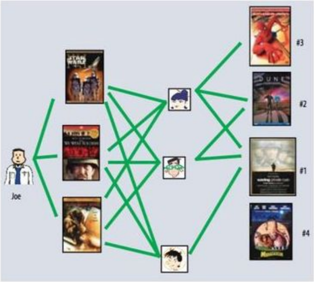
Produces recommendations of items based on patterns of ratings or usage (e.g. purchases) without the need for exogenous information about the item or user.

- Relates two fundamentally different entities: items and users

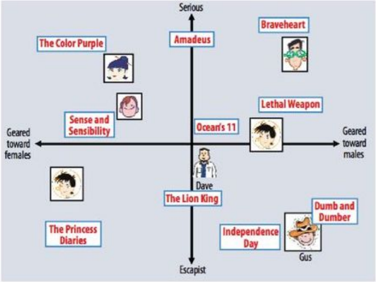
n.b. Doesn't require other predictors to make predictions

Two main techniques:

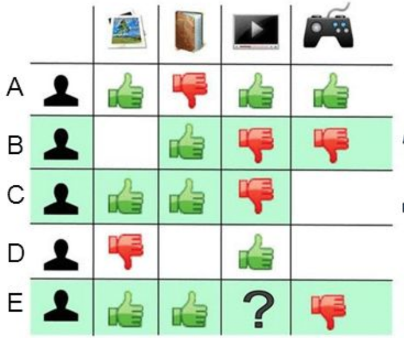
1. Neighborhood Methods



2. Latent Factor Methods



Neighborhood methods



Focus on relationships between items (or users), modeling the preference of a user to an item based on ratings of similar items by that user.

Neighborhood methods

Two items are more similar if a user rated them similarly.

Pearson correlation

$$\rho_{ij} = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - b_i)(r_{uj} - b_j)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - b_i)^2} \sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{uj} - b_j)^2}} \tag{2}$$

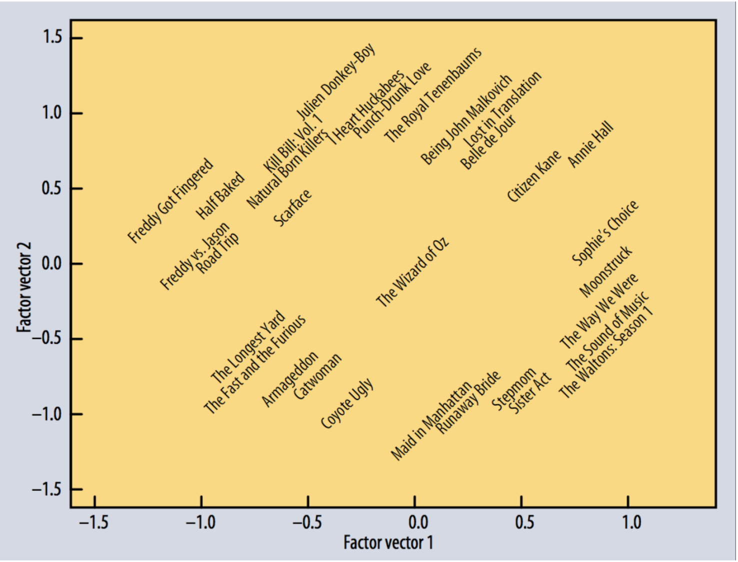
Cosine similarity

$$\cos_{ij} = \frac{\sum_{u \in \mathcal{U}_{ij}} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in \mathcal{U}_i} r_{ui}^2} \sqrt{\sum_{u \in \mathcal{U}_j} r_{uj}^2}} \tag{3}$$

- Items are clustered based on similarity.
- Alternatively, can build a KNN based predictive model.

Latent factor models

- Transform items and users to the same latent factor space.
- Explains ratings by characterizing products and users on factors inferred from user feedback.
- The new space might identify factors relating to "comedy", "romance", or a particular actor, etc.
  - Typically brings about a qualitatitve aspect of describing factors.
- The model provides weights for each user and item in this space.



Julien Donkey-Boy
I Heart Huckabees
Punch-Drunk Love
The Royal Tenenbaums
Being John Malkovich
Lost in Translation
Belle de Jour
Kill Bill: Vol. 1
Natural Born Killers
Freddy Got Fingered
Half Baked
Scarface
Citizen Kane
Annie Hall
Freddy vs. Jason
Road Trip
The Wizard of Oz
Sophie's Choice
Moonstruck
The Way We Were
The Sound of Music
The Waltons: Season 1
The Longest Yard
The Fast and the Furious
Armageddon
Catwoman
Coyote Ugly
Maid in Manhattan
Runaway Bride
Stepmom
Sister Act
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Factor vector 2
−1.5
−1.0
−0.5
0.0
0.5
1.0
Factor vector 1

Latent factor models

Map items and users into a latent factor space of dimensionality, f,

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^\top p_u \tag{4}$$

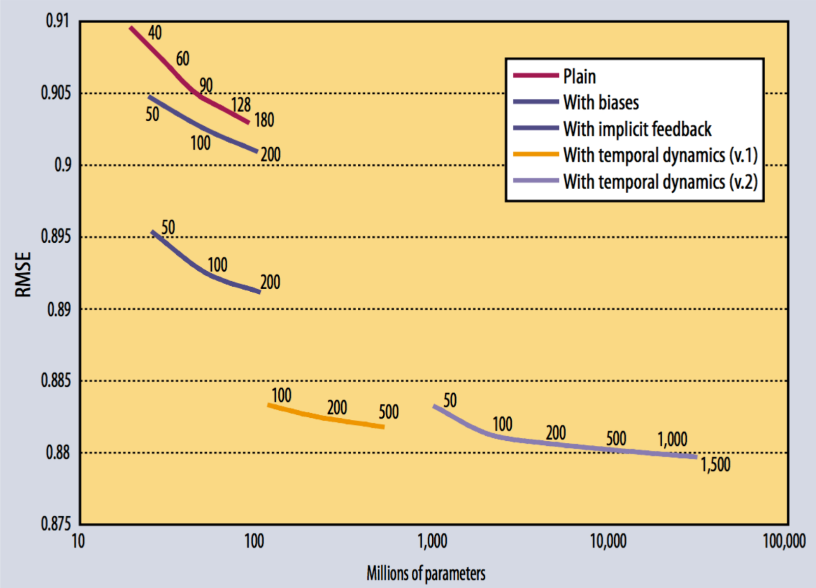
Estimate parameters with least squares + regularization

$$\min_{b^*, q^*, p^*} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2) \tag{5}$$

where $\hat{r}_{ui} = \mu + b_i + b_u$.

- Estimated with gradient descent.
- λ is a regularization parameter to bias parameters towards 0.

Latent factor models



Note: Can also include info about whether a result was rated *at all*.

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^\top \left(p_u + |R(u)|^{-1/2} \sum_{j \in R(u)} y_j \right) \qquad (6)$$

- Each item is now associated with a factor vector y, which is used to modify our user features based on the items they've rated.

Missing data is everywhere

Common situations with missing data:

- Survey data (non-response).
- Longitudinal studies and clinical trials (dropout).
- Recommendation systems.
- Data integration.

- **Missing Completely at Random** (MCAR): No relationship exists between the missingness of the data and any values, observed or missing.
  - *Example*. We run a taste study for 20 different drinks. Each subject was asked to rate only 4 drinks chosen at random.

Mechanisms for missing data

- **Missing Completely at Random** (MCAR): No relationship exists between the missingness of the data and any values, observed or missing.
  - *Example*. We run a taste study for 20 different drinks. Each subject was asked to rate only 4 drinks chosen at random.
- **Missing at Random** (MAR): The pattern of missingness depends on other *observed predictors*.
  - *Example*. In a survey, poor subjects were less likely to answer a question about drug use than wealthy subjects.

Mechanisms for missing data

- **Missing Completely at Random** (MCAR): No relationship exists between the missingness of the data and any values, observed or missing.
  - *Example*. We run a taste study for 20 different drinks. Each subject was asked to rate only 4 drinks chosen at random.
- **Missing at Random** (MAR): The pattern of missingness depends on other *observed predictors*.
  - *Example*. In a survey, poor subjects were less likely to answer a question about drug use than wealthy subjects.
- **Missing Not at Random** (MNAR): The pattern of missingness depends on the missing values (or unobserved predictors).
  - *Example*. High earners less likely to report their income.

Mechanisms for missing data

Formalizing our missing data structure, let us define:

$$
\begin{aligned}
O &= (X_1, \dots, X_p, Y) && (7)\\
(O, R) &= \text{Complete data} && (8)\\
R_j &= \text{Indicator that } j^{th} \text{ element of } O \text{ is missing} && (9)\\
O^{obs} &= \text{Observed part of } O && (10)\\
O^{mis} &= \text{Missing part of } O && (11)
\end{aligned}
$$

Mechanisms for missing data

Formalizing our missing data structure, let us define:

$$O = (X_1, \dots, X_p, Y) \tag{7}$$

$$(O, R) = \text{Complete data} \tag{8}$$

$$R_j = \text{Indicator that } j^{th} \text{ element of } O \text{ is missing} \tag{9}$$

$$O^{obs} = \text{Observed part of } O \tag{10}$$

$$O^{mis} = \text{Missing part of } O \tag{11}$$

Then for:

- Missing Completely at Random (MCAR)

$$\mathbb{P}(R|O) = \mathbb{P}(R) \tag{12}$$

Mechanisms for missing data

Formalizing our missing data structure, let us define:

$$O = (X_1, \ldots, X_p, Y) \tag{7}$$

$$(O, R) = \text{Complete data} \tag{8}$$

$$R_j = \text{Indicator that } j^{th} \text{ element of } O \text{ is missing} \tag{9}$$

$$O^{obs} = \text{Observed part of } O \tag{10}$$

$$O^{mis} = \text{Missing part of } O \tag{11}$$

Then for:

- Missing Completely at Random (MCAR)

$$\mathbb{P}(R|O) = \mathbb{P}(R) \tag{12}$$

- Missing at Random (MAR)

$$\mathbb{P}(R|O) = \mathbb{P}(R|O^{obs}) \,\forall\, O^{mis} \tag{13}$$

Mechanisms for missing data

Formalizing our missing data structure, let us define:

$$O = (X_1, \ldots, X_p, Y) \quad (7)$$
$$(O, R) = \text{Complete data} \quad (8)$$
$$R_j = \text{Indicator that } j^{th} \text{ element of } O \text{ is missing} \quad (9)$$
$$O^{obs} = \text{Observed part of } O \quad (10)$$
$$O^{mis} = \text{Missing part of } O \quad (11)$$

Then for:

- Missing Completely at Random (MCAR)

$$\mathbb{P}(R|O) = \mathbb{P}(R) \quad (12)$$

- Missing at Random (MAR)

$$\mathbb{P}(R|O) = \mathbb{P}(R|O^{obs}) \, \forall \, O^{mis} \quad (13)$$

- Missing Not at Random (MNAR)

$$\mathbb{P}(R|O) \text{ depends on } O^{mis} \quad (14)$$

Dealing with missing data

The *missing-data mechanism* is ignorable when (Rubin 1976):

1. The missing data are MAR (or MCAR).
2. The parameters of O and R are distinct (i.e. the joint parameter space (ψ, ξ) can be factorized).
   - (ψ, ξ) are parameters for our our distributions of O and R.
   - If not distinct, ignoring missing-data mechanism is still valid, but not fully efficient.

Our likelihood:

$$L_{full}(\psi, \xi | O^{obs}, R) = L(\psi | O^{obs}) \cdot L(\xi | O^{obs}, R) \tag{15}$$

In blue: The likelihood for our observed data.

In red: The likelihood for our missing data mechanism.

Dealing with missing data

Practically, what we'll sometimes do:

- Some tree-based methods can deal with missing data naturally.

Dealing with missing data

Practically, what we'll sometimes do:

- Some tree-based methods can deal with missing data naturally.
- **Single imputation**: We replace each missing value with a single number. We can replace with e.g.
  1. The mean or median of the column.
  2. A random sample from the non-missing values in the column.
  3. A regression estimate from other predictors, X_{-j}.

Dealing with missing data

Practically, what we'll sometimes do:

- Some tree-based methods can deal with missing data naturally.
- **Single imputation**: We replace each missing value with a single number. We can replace with e.g.
  1. The mean or median of the column.
  2. A random sample from the non-missing values in the column.
  3. A regression estimate from other predictors, X_{-j}.
  - Methods 1 and 2 can give biased coefficients if the data is not missing completely at random.
  - Method 3 does not have bias if the missing variable is predicted well by X_{-j}.
  - Method 3 yields standard errors that are artificially small.

Dealing with missing data

- **Multiple imputation**: We replace each missing value in X_j with a regression estimate from the other predictors X_{-j}, plus some noise. This is repeated several times.
  - If the regression fit of X_j onto X_{-j} is good, the standard errors from this method can be unbiased.

Missing data in more than one variable

Problem: What if we have missing data in almost every column $X_1, X_2, \ldots , X_p$?

Missing data in more than one variable

Problem: What if we have missing data in almost every column $X_1, X_2, \ldots, X_p$?

- **Iterative multiple imputation**: Start with a simple imputation. Then, iterate the following:

  1. Multiple imputation of X_1 from X_{-1}.
  2. Multiple imputation of X_2 from X_{-2}.

     ...
  3. Multiple imputation of X_p from X_{-p}.

Missing data in more than one variable

Problem: What if we have missing data in almost every column $X_1, X_2, \ldots, X_p$?

- **Iterative multiple imputation**: Start with a simple imputation. Then, iterate the following:
  1. Multiple imputation of X_1 from X_{-1}.
  2. Multiple imputation of X_2 from X_{-2}.
     $\ldots$
  3. Multiple imputation of X_p from X_{-p}.
- **Model based imputation**: Fit the missing values to a joint statistical model for all the predictors.

Missing data in more than one variable

Problem: What if we have missing data in almost every column $X_1, X_2, \ldots, X_p$?

- **Iterative multiple imputation**: Start with a simple imputation. Then, iterate the following:
  1. Multiple imputation of X_1 from X_{-1}.
  2. Multiple imputation of X_2 from X_{-2}.
     ...
  3. Multiple imputation of X_p from X_{-p}.
- **Model based imputation**: Fit the missing values to a joint statistical model for all the predictors.
  - *Rarely worth the trouble.*

Missing data in the outcome

Question: What if the outcome is missing?

- If MCAR, then can just drop the observations.
- If MAR, then (if measured) you can model it.
  - *Example*. Survival analysis - If a person is censored due to poorer health, you can model the probability of censoring based on poorer health & use it in propensity score models.
- If MNAR, then not many options. Two options include:
  - *Selection models*: simultaneously model Y and the probability that it's missing.
  - *Pattern mixture*: perform multiple imputations under a variety of assumptions about the missing data mechanism.

Formalizing the missing data mechanism.

For $i = 1, 2, \dots, n$, define:

- $T_i \sim P_0(T) : T_i \geq 0$ to be our survival time.
- $C_i \sim P_0(C) : C_i \geq 0$ to be our censoring time.

Then our observed survival time is simply $Y_i = \min(T_i, C_i)$.

Survival analysis

Formalizing the missing data mechanism.

For $i = 1, 2, \ldots, n$, define:

- $T_i \sim P_0(T) : T_i \geq 0$ to be our survival time.
- $C_i \sim P_0(C) : C_i \geq 0$ to be our censoring time.

Then our observed survival time is simply $Y_i = \min(T_i, C_i)$.

Ways not to deal with censored data:

- Discard the censored observations.
- Treat the censored observations as uncensored

Both introduce bias and (possibly severely) under estimate $P_0(T)$.

Some practical considerations

- It is important to visualize summaries or plots for the pattern of missingness.
- If the pattern of missingness is informative, include it as a dummy variable.
- If a variable has too many missing values, it is worth it to include it?
- If we are using a method that allows it, consider weighting variables according to the rate of missing data.
  Example. In nearest neighbors, scale each variable and multiply by $(100 - \%\mathit{missing})$.
- Some variables are restricted to be positive, or bounded above.
- Are there any variables that are non-linear functions of others?

Relational data

The observations have the form of a graph.

Examples.

- Links between websites.
- Relationships between accounts in social networks.
- Transmission networks for contagious diseases.
- Causal graphs (e.g. matches cause lung cancer).
- Relationships between named entities (e.g. Santa ¡lives-in¿ The North Pole)

Relational data

The observations have the form of a graph.

Examples.

- Links between websites.
- Relationships between accounts in social networks.
- Transmission networks for contagious diseases.
- Causal graphs (e.g. matches cause lung cancer).
- Relationships between named entities (e.g. Santa ¡lives-in¿ The North Pole)

The links can be *directed* or *undirected*.
There can be different types of link (friend, follower, followed).
We can observe the graph in time (social networks growing).
Each vertex can have additional features or metadata.

PageRank algorithm

- Invented by Sergei Brin and Larry Page of Google.
- Uses a graph of links between websites to rank websites by "importance".

PageRank algorithm

- Invented by Sergei Brin and Larry Page of Google.
- Uses a graph of links between websites to rank websites by "importance".
- **Motivation**:
  - Consider the problem of searching the web using the query "birth control".
  - There are millions of pages containing the term.
  - Analyzing the content of each website semantically to infer which one is more likely to satisfy the user is very expensive.
  - We need a way to rank websites, to filter out all those that are rarely visited. This information is given by links.

PageRank algorithm

Example:

Consider a hypothetical surfer who is jumping from website to website by clicking on random links.

Example:

Consider a hypothetical surfer who is jumping from website to website by clicking on random links. Intuitively, the websites that are visited more frequently can be considered more important in the network of links.

Example:

Consider a hypothetical surfer who is jumping from website to website by clicking on random links. Intuitively, the websites that are visited more frequently can be considered more important in the network of links.

Will the surfer visit every website eventually?

Example:

Consider a hypothetical surfer who is jumping from website to website by clicking on random links. Intuitively, the websites that are visited more frequently can be considered more important in the network of links.

Will the surfer visit every website eventually? No. It is possible to get stuck in a website with no outgoing links, or to be stuck in a loop between two websites, for example.

Example:

Consider a hypothetical surfer who is jumping from website to website by clicking on random links. Intuitively, the websites that are visited more frequently can be considered more important in the network of links.

Will the surfer visit every website eventually? No. It is possible to get stuck in a website with no outgoing links, or to be stuck in a loop between two websites, for example.

To avoid this problem, we modify the random walk, such that at every step, with probability $1 - q$, we pick a website at random, and with probability q we go through one of the links in the current website at random.

PageRank algorithm

- The surfer's random walk is a Markov chain on the set of websites.
- It is a fact that the frequency with which the surfer visits any website converges to some limit.
- The PageRank of a website is this limiting frequency.

PageRank algorithm

Let P_{ij} be the probability of jumping from website i to website j, then

$$P_{ij} = (1-q)\frac{1}{n} + q\left[\frac{\#\text{of links from } i \text{ to } j}{\#\text{of links out of } i}\right] \tag{16}$$

PageRank algorithm

Let P_{ij} be the probability of jumping from website i to website j, then

$$P_{ij} = (1-q)\frac{1}{n} + q\left[\frac{\#\text{of links from } i \text{ to } j}{\#\text{of links out of } i}\right] \tag{16}$$

The limiting frequency of website j, π_j, must satisfy

$$\pi_j = \sum_{i=1}^{n} \pi_i P_{ij} \tag{17}$$

or in matrix notation $\pi = \pi P$.

Let P_{ij} be the probability of jumping from website i to website j, then

$$P_{ij} = (1-q)\frac{1}{n} + q\left[\frac{\#\text{of links from } i \text{ to } j}{\#\text{of links out of } i}\right] \tag{16}$$

The limiting frequency of website j, π_j, must satisfy

$$\pi_j = \sum_{i=1}^{n} \pi_i P_{ij} \tag{17}$$

or in matrix notation $\pi = \pi P$. That is, π is an eigenvector of the transition probability matrix P with eigenvalue 1.

Finding the limiting factor π

In principle, finding the limiting frequencies could require solving the eigendecomposition of a matrix P which is $n \times n$, and this has a complexity which grows as n^3.

Finding the limiting factor π

In principle, finding the limiting frequencies could require solving the eigendecomposition of a matrix P which is $n \times n$, and this has a complexity which grows as n^3.

However, it is possible to compute π by starting with the approximation $\pi^{(0)} = (1/n, ..., 1/n)$, and iterating:

$$\pi^{(t)} = \pi^{(t-1)} P \tag{18}$$

The number of iterations necessary for convergence is typically small.

Finding the limiting factor π

In principle, finding the limiting frequencies could require solving the eigendecomposition of a matrix P which is $n \times n$, and this has a complexity which grows as n^3.

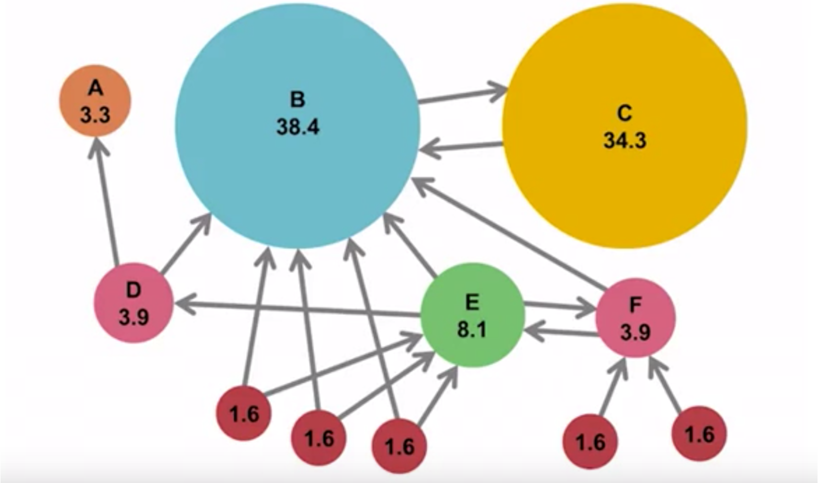
However, it is possible to compute π by starting with the approximation $\pi^{(0)} = (1/n, ..., 1/n)$, and iterating:

$$\pi^{(t)} = \pi^{(t-1)} P \tag{18}$$

The number of iterations necessary for convergence is typically small.

The matrix-vector multiplication in each iteration can be sped up using sparse matrix techniques.

PageRank example



How can PageRank be used in web search?

One idea:

1. Find all websites that contain all query terms.
2. Display them in order of their PageRank.

How can PageRank be used in web search?

One idea:

1. Find all websites that contain all query terms.
2. Display them in order of their PageRank.

A more likely approach:

1. Use PageRank to select the $10,000$ most important pages which contain the query terms.
2. Rank these $10,000$ pages by analyzing their content, integrating information about the user, etc.

References

[1] Su, Xiaoyuan (2009). A Survey of Collaborative Filtering Techniques.

[2] Rubin, Donald et al (2012). Missing data and imputation methods.