Recommender Systems

Don van den Bergh

StatsLearn

Examples

- Internet: observe watched/ rated items, recommend new items (i.e. movies, books, anything in a webshop)

Examples

- Internet: observe watched/ rated items, recommend new items (i.e. movies, books, anything in a webshop)
- Medical: observe symptoms, recommend treatment

Examples

- Internet: observe watched/ rated items, recommend new items (i.e. movies, books, anything in a webshop)
- Medical: observe symptoms, recommend treatment
- Life choices: observe school skills/ preferences, recommend study/ profession

Types of Recommender Systems:

- Content Based: Recommend Items similar to those the user previously liked

Types of Recommender Systems:

- Content Based: Recommend Items similar to those the user previously liked
- Collaborative Filtering: Recommend a user what users with similar tastes liked in the past

Types of Recommender Systems:

- Content Based: Recommend Items similar to those the user previously liked
- Collaborative Filtering: Recommend a user what users with similar tastes liked in the past
- Demographic: Recommend based on user demographics

Types of Recommender Systems:

- Content Based: Recommend Items similar to those the user previously liked
- Collaborative Filtering: Recommend a user what users with similar tastes liked in the past
- Demographic: Recommend based on user demographics
- Knowledge: Recommend items based on specific knowledge about how item features meet users needs

Types of Recommender Systems:

- Content Based: Recommend Items similar to those the user previously liked
- Collaborative Filtering: Recommend a user what users with similar tastes liked in the past
- Demographic: Recommend based on user demographics
- Knowledge: Recommend items based on specific knowledge about how item features meet users needs
- Community: "Tell me who your friends are, and I will tell you who you are"

Types of Recommender Systems:

- Content Based: Recommend Items similar to those the user previously liked
- Collaborative Filtering: Recommend a user what users with similar tastes liked in the past
- Demographic: Recommend based on user demographics
- Knowledge: Recommend items based on specific knowledge about how item features meet users needs
- Community: "Tell me who your friends are, and I will tell you who you are"
- Hybrid: A mixture of the above

The Plan

General Approach

- Training set, Test set, Crossvalidation

The Plan

General Approach

- Training set, Test set, Crossvalidation
- Use a regression/ classification algorithm

The Plan

General Approach

- Training set, Test set, Crossvalidation
- Use a regression/ classification algorithm

Recommender System specific issues:

The Plan

General Approach

- Training set, Test set, Crossvalidation
- Use a regression/ classification algorithm

Recommender System specific issues:

- Cold start

The Plan

General Approach

- Training set, Test set, Crossvalidation
- Use a regression/ classification algorithm

Recommender System specific issues:

- Cold start
- Novelty, Adaptivity, Risk, Diversity of suggestions

The Plan

General Approach

- Training set, Test set, Crossvalidation
- Use a regression/ classification algorithm

Recommender System specific issues:

- Cold start
- Novelty, Adaptivity, Risk, Diversity of suggestions
- Sparsity/ curse of dimensionality

The Plan

General Approach

- Training set, Test set, Crossvalidation
- Use a regression/ classification algorithm

Recommender System specific issues:

- Cold start
- Novelty, Adaptivity, Risk, Diversity of suggestions
- Sparsity/ curse of dimensionality
- Statistical performance meausures do not capture all relevant aspects

General Idea

Given a person u and an item i:

$$R(u, i) = \text{Real Utility}$$

General Idea

Given a person u and an item i:

$$R(u,i) = \text{Real Utility}$$
$$\hat{R}(u,i_1), \quad \ldots, \quad \hat{R}(u,i_N)$$

General Idea

Given a person u and an item i:

$$R(u, i) = \text{Real Utility}$$
$$\hat{R}(u, i_1), \quad \ldots, \quad \hat{R}(u, i_N)$$

Recommend K items with highest utility ($\hat{R}$)

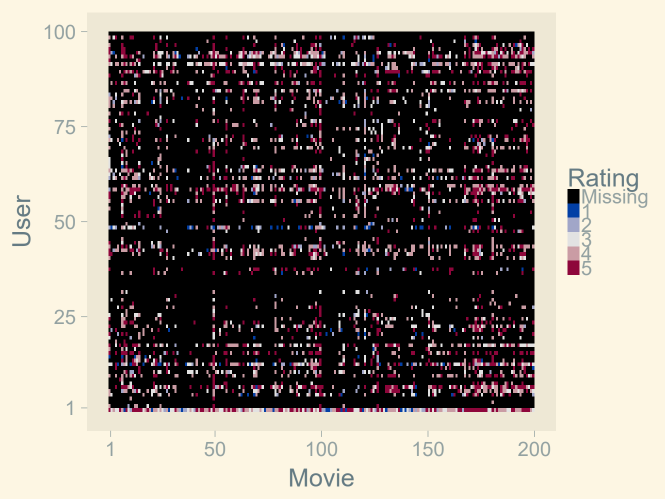
General Idea

Given a person u and an item i:

$$R(u,i) = \text{Real Utility}$$
$$\hat{R}(u,i_1), \quad \ldots, \quad \hat{R}(u,i_N)$$

Recommend K items with highest utility ($\hat{R}$)
Incorporating arbitrary aspects of the suggestions (i.e. variety) can be difficult

A snapshot of the Movielense data



Content Based

	u_1	u_2	u_3	u_4	u_5
m_1	5	4	2	3	3
m_2	3	?	3	3	3
m_3	4	3	2	?	2
m_4	?	2	?	2	?

Content Based

	u_1	u_2	u_3	u_4	u_5	x_1	x_2	x_3
m_1	5	4	2	3	3	0.94	0.98	0.12
m_2	3	?	3	3	3	0.48	0.56	0.90
m_3	4	3	2	?	2	0.14	0.99	0.95
m_4	?	2	?	2	?	0.08	0.51	0.39

Content Based

						Comedy	Romance	Action
	u_1	u_2	u_3	u_4	u_5	x_1	x_2	x_3
m_1	5	4	2	3	3	0.94	0.98	0.12
m_2	3	?	3	3	3	0.48	0.56	0.90
m_3	4	3	2	?	2	0.14	0.99	0.95
m_4	?	2	?	2	?	0.08	0.51	0.39

Content Based

- given movie features: $\mathbf{x}_m = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$
- find user preferences: $\theta_u = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$

$$R(u, m) = \theta_u^T \mathbf{x}_m$$

Content Based

- given movie features: $\mathbf{x}_m = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$
- find user preferences: $\theta_u = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$

$$R(u, m) = \theta_u^T \mathbf{x}_m$$

Estimate $\hat{\theta}$ from given ratings R.

$$J(\theta) = \frac{1}{2} \sum_{u,m \in OR} (\theta_u^T \mathbf{x}_m - R(u, m))^2 + \frac{\lambda}{2} \sum_{j=1}^{U} \theta_j^T \theta_j$$

Where OR are all observed ratings

Content Based

Make new predictions $\hat{R}$ according to:

$$\hat{R}(u, m) = \hat{\theta}_u^T \mathbf{x}_m$$

Content Based

Make new predictions $\hat{R}$ according to:

$$\hat{R}(u, m) = \hat{\theta}_u^T \mathbf{x}_m$$

Problem: obtaining features for many movies

Collaborative Filtering

- given user preferences: $\theta_u = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$
- find movie features: $\mathbf{x}_m = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$

$$R(u, m) = \theta_u^T \mathbf{x}_m$$

Collaborative Filtering

- given user preferences: $\theta_u = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$
- find movie features: $\mathbf{x}_m = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$

$$R(u, m) = \theta_u^T \mathbf{x}_m$$

Estimate $\hat{\mathbf{x}}$ from given preferences and ratings R.

$$J(\theta) = \frac{1}{2} \sum_{u,m \in OR} (\theta_u^T \mathbf{x}_m - R(u, m))^2 + \frac{\lambda}{2} \sum_{i=1}^{M} \mathbf{x}_i^T \mathbf{x}_i$$

Collaborative Filtering

Make new predictions $\hat{R}$ according to:

$$\hat{R}(u, m) = \theta_u^T \hat{\mathbf{x}}$$

Collaborative Filtering

Make new predictions $\hat{R}$ according to:

$$\hat{R}(u, m) = \theta_u^T \hat{\mathbf{x}}$$

Problem: obtaining preferences for many users

Low Rank Matrix Factorization

Why not both?

$$J(\mathbf{x}, \theta) = \frac{1}{2} \sum_{u,m \in OR} (\theta_u^T \mathbf{x}_m - R(u,m))^2 + \frac{\lambda}{2} \sum_{i=1}^{M} \mathbf{x}_i^T \mathbf{x}_i + \frac{\lambda}{2} \sum_{j=1}^{U} \theta_j^T \theta_j$$

Circumventing Problems

New user u^*:

Circumventing Problems

New user u^*:

- $\hat{R}(u^*, m) = \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is 0 for all movies!

Circumventing Problems

New user u^*:

- $\hat{R}(u^*, m) = \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is 0 for all movies!
- $\hat{R}(u^*, m) = \mu_m + \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is not 0 for all movies!

Circumventing Problems

New user u^*:

- $\hat{R}(u^*, m) = \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is 0 for all movies!
- $\hat{R}(u^*, m) = \mu_m + \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is not 0 for all movies!

Related movies:

- related movies: $||\mathbf{x}_i - \mathbf{x}_j||$ (or some other distance)

Circumventing Problems

New user u^*:

- $\hat{R}(u^*, m) = \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is 0 for all movies!
- $\hat{R}(u^*, m) = \mu_m + \hat{\theta}_{u^*}^T \hat{\mathbf{x}}_m$ is not 0 for all movies!

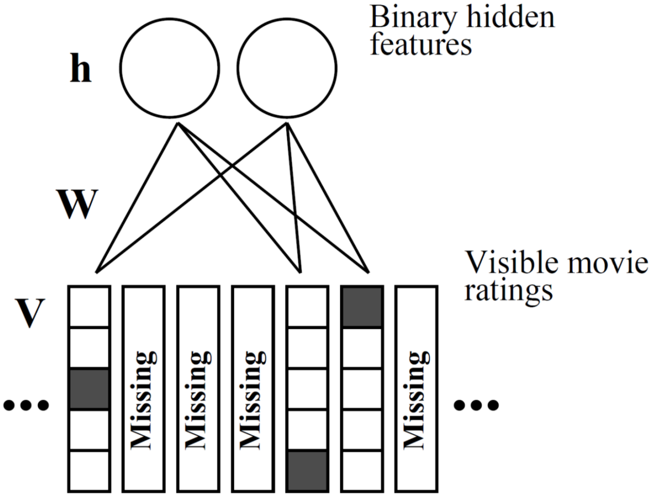
Related movies:

- related movies: $||\mathbf{x}_i - \mathbf{x}_j||$ (or some other distance)

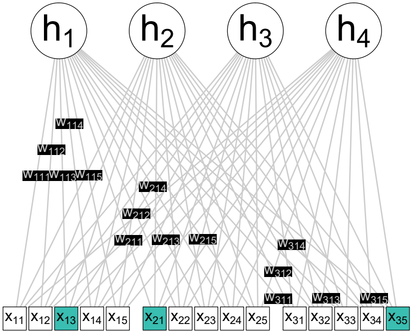
New problems:

- what do the features mean?
- Number of features is a hyperparameter

Graphical Representation



A user is a vector of movie ratings. Each movie is a "visible unit", composed of 5 dummy variables for each category. $h, x \in \{0, 1\}$, $w \in \mathcal{R}$



In math

rating k out of K for movie i and hidden feature j

$$P(x_{ik} = 1|\mathbf{h}) = \frac{\exp\left(b_{ik} + \sum_{j=1}^{F} h_j W_{ijk}\right)}{\sum_{l=1}^{K} \exp(b_{il} + \sum_{j=1}^{F} h_j W_{ijl})}$$

In math

rating k out of K for movie i and hidden feature j

$$P(x_{ik} = 1|\mathbf{h}) = \frac{\exp\left(b_{ik} + \sum_{j=1}^{F} h_j W_{ijk}\right)}{\sum_{l=1}^{K} \exp(b_{il} + \sum_{j=1}^{F} h_j W_{ijl})}$$

$$p(h_j = 1|\mathbf{X}) = \sigma(b_j + \sum_{i=1}^{M} \sum_{k=1}^{K} x_{ik} W_{ijk})$$

In math

rating k out of K for movie i and hidden feature j

$$P(x_{ik} = 1|\mathbf{h}) = \frac{\exp\left(b_{ik} + \sum_{j=1}^{F} h_j W_{ijk}\right)}{\sum_{l=1}^{K} \exp(b_{il} + \sum_{j=1}^{F} h_j W_{ijl})}$$

$$p(h_j = 1|\mathbf{X}) = \sigma(b_j + \sum_{i=1}^{M} \sum_{k=1}^{K} x_{ik} W_{ijk})$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Implementation

- Weights $\mathbf{W}$ and biases $\mathbf{b}$ are fixed across all users

Implementation

- Weights $\mathbf{W}$ and biases $\mathbf{b}$ are fixed across all users
- Users vary in activation of the hidden layer $\mathbf{h}$

Implementation

- Weights $\mathbf{W}$ and biases $\mathbf{b}$ are fixed across all users
- Users vary in activation of the hidden layer $\mathbf{h}$
- An RBM for a given user only contains the movies that user has rated

Implementation

- Weights $\mathbf{W}$ and biases $\mathbf{b}$ are fixed across all users
- Users vary in activation of the hidden layer $\mathbf{h}$
- An RBM for a given user only contains the movies that user has rated
- Estimate parameters using an adaptation of SGD called Contrast Divergence

Contrast Divergence

SGD:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \Delta \mathbf{W}$$

with some learning rate η

Contrast Divergence

SGD:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \Delta \mathbf{W}$$

with some learning rate η

$$\Delta \mathbf{W} = \frac{\partial \log P(\mathbf{X} = \mathbf{x})}{\partial \mathbf{W}}$$

$$\Delta \mathbf{W} = \frac{\partial F}{\partial W}(\mathbf{x}) - \sum_{\mathbf{x}} p(\mathbf{x}) \frac{\partial F}{\partial W}(\mathbf{x})$$

Contrast Divergence

SGD:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \Delta \mathbf{W}$$

with some learning rate η

$$\Delta \mathbf{W} = \frac{\partial \log P(\mathbf{X} = \mathbf{x})}{\partial \mathbf{W}}$$

$$\Delta \mathbf{W} = \frac{\partial F}{\partial W}(\mathbf{x}) - \sum_{\mathbf{x}} p(\mathbf{x}) \frac{\partial F}{\partial W}(\mathbf{x})$$

Left positive term is analytically solvable, right term can be obtained using Gibbs sampling

Contrast Divergence

- We want to maximize the likelihood

Contrast Divergence

- We want to maximize the likelihood
- This cannot be done analytically

Contrast Divergence

- We want to maximize the likelihood
- This cannot be done analytically
- Thus we use SGD (iterative procedure #1)

Contrast Divergence

- We want to maximize the likelihood
- This cannot be done analytically
- Thus we use SGD (iterative procedure #1)
- However, the gradient cannot be evaluated analytically

Contrast Divergence

- We want to maximize the likelihood
- This cannot be done analytically
- Thus we use SGD (iterative procedure #1)
- However, the gradient cannot be evaluated analytically
- Therefore use Gibbs sampling (iterative procedure #2)

Predictions

Assume we know the weights matrix **W**, the intercepts/ biases **b**, and the hidden activations for all users **h** for p observed. Recall that **W** and **b** are fixed over users.

$$\hat{p}_j = p(h_j = 1|\mathbf{X}) = \sigma(b_j + \sum_{i=1}^{M}\sum_{k=1}^{K} x_{ik} W_{ijk})$$

Predictions

Assume we know the weights matrix **W**, the intercepts/ biases **b**, and the hidden activations for all users **h** for p observed. Recall that **W** and **b** are fixed over users.

$$\hat{p}_j = p(h_j = 1|\mathbf{X}) = \sigma(b_j + \sum_{i=1}^{M}\sum_{k=1}^{K} x_{ik} W_{ijk})$$

$$p(x_{qk} = 1|\hat{\mathbf{p}}) = \frac{\exp\left(b_{qk} + \sum_{j=1}^{F} \hat{p}_j W_{qjk}\right)}{\sum_{l=1}^{K} \exp\left(b_{qk} + \sum_{j=1}^{F} \hat{p}_j W_{qjk}\right)}$$

To Summarize

- RBMs and MFs were the most successful in the netflix competition

To Summarize

- RBMs and MFs were the most successful in the netflix competition
- RBMs do slightly better than MFs

To Summarize

- RBMs and MFs were the most successful in the netflix competition
- RBMs do slightly better than MFs
- Most importantly, the errors of RBMs are completely different than those of MFs

To Summarize

- RBMs and MFs were the most successful in the netflix competition
- RBMs do slightly better than MFs
- Most importantly, the errors of RBMs are completely different than those of MFs
- Best predictor is a combination of multiple RBM's and MFs (model averaging)