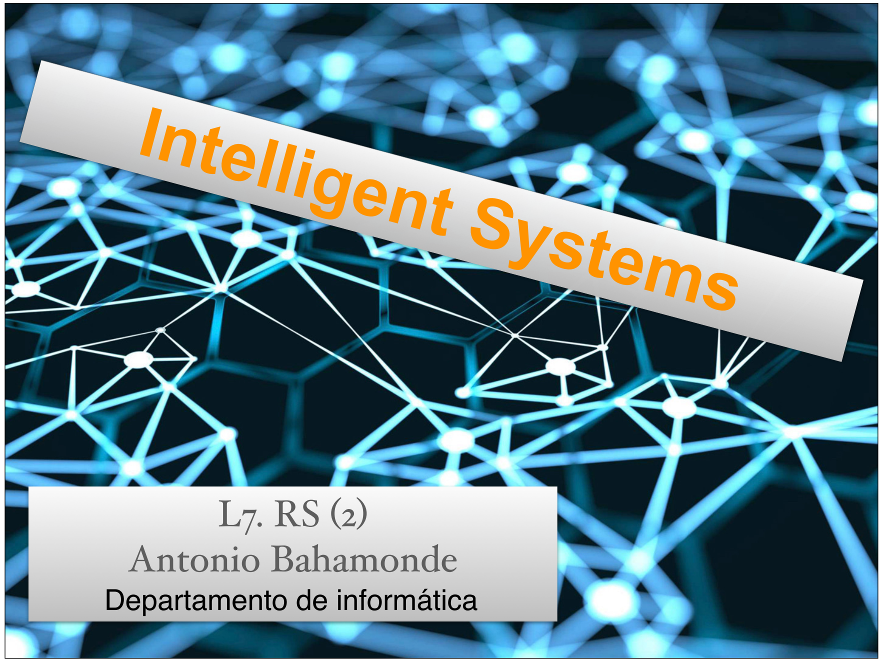


Contents

- Netflix Award
- RS as a learning task
  - regression
  - classification
  - preferences
  - probabilistic preferences
  - matrix factorization

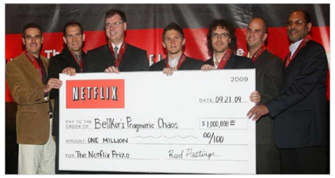
Netflix Award

The New York Times (Sep, 21, 2009):

Netflix Awards $1 Million Prize and Starts a New Contest

[...]try to predict what movies particular customers would prefer

“Accurately predicting the movies Netflix members will love is a key component of our service,” said Neil Hunt, chief product officer (Netflix)



NETFLIX

Netflix Award

Netflix dataset

- Moree than 100 millions movie assessments (1-5 stars)
- From Nov. 11, 1999 to Dec. 31, 2005
- 480189 users
- 17770 movies
- 99% cells are empty
  - Each movie has an average of 5600 assessments
  - Each user has assessed 208 movies (average)
- 2 datasets: train and quiz (test-prize)

Netflix Award

Loss function: root-mean-square error (RMSE)

$$RMSE = \sqrt{\frac{1}{|Quiz|} \sum_{(u,i)\in Quiz} (r(u,i) - b(u,i))^2}$$

Netflix had their own RS, *Cinematch*, with

RMSE = 0.9514.

Winner had to be 10% better than that

Netflix Award

Final score reuslts

	Team	RMSE	Date	Hour
1	BellKor's Pragmatic Chaos	0,8567	26/07/09	18:18:28
2	The Emsemble	0,8567	26/07/09	18:38:22
3	Grand Prize Team	0,8582	10/07/09	21:24:40
4	Opera Solutions and Vandelay United	0,8588	10/07/09	01:12:31

Winner Netflix Award

BellKor's Pragmatic Chaos

Yehuda Koren, Robert M. Bell: Advances in Collaborative Filtering. Recommender Systems Handbook 2011: 145-186

Yehuda Koren, Yahoo! Research

Robert Bell, AT&T Labs – Research

•

How does *Pragmatic Chaos* work?

Titanic y Joe

- Average Netflix: 3.7
- Titanic: 0.5 over the average (all users)
- Joe is quite critic: 0.3 below average

$$\hat{\boldsymbol{r}}_{Joe,Titanic} = 3.7 + 0.5 - 0.3 = 3.9$$

$$\hat{\boldsymbol{r}}_{ui} = \mu + \boldsymbol{b}_i + \boldsymbol{b}_u$$

	baseline	Cinematch	Prize
RMSE	0,9799	0,9514	0,8567

How does *Pragmatic Chaos* work?

Previous equation look nice, but it is too simple

Koren & Bell proposed a new (and more complex) version to estimate the mark given by a user u of and item i,

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

μ	General average in Netflix
b_i	bias (item i)
b_u	bias (user u)
q_i, p_u	Are (column) vectors with k components (columns of matrices Q & P)

How does *Pragmatic Chaos* work?

Vectors q_i and p_u are representing k features of items and users respectively

They are learned (as bias)

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

μ	General average in Netflix
b_i	bias (item i)
b_u	bias (user u)
q_i, p_u	Are (column) vectors with k components (columns of matrices Q & P)

Matrix factorization

step by step

6 = 3 * 2
M1 = M2 * M3

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

$$q_i^T p_u = \begin{pmatrix} | \\ q_i \\ | \end{pmatrix}^T \begin{pmatrix} | \\ p_u \\ | \end{pmatrix} = \begin{pmatrix} - & q_i & - \end{pmatrix} \begin{pmatrix} | \\ p_u \\ | \end{pmatrix}$$

$$= \langle q_i, p_u \rangle$$

$$\underbrace{\left(\begin{pmatrix} | \\ q_1 \\ | \end{pmatrix} - \begin{pmatrix} | \\ q_i \\ | \end{pmatrix} - \right)}_{Q} \underbrace{\begin{pmatrix} 0 \\ | \\ 1 \\ | \\ 0 \end{pmatrix}}_{it} - o_i = Q \cdot it$$

$$\underbrace{\left(\begin{pmatrix} | \\ p_1 \\ | \end{pmatrix} - \begin{pmatrix} | \\ p_u \\ | \end{pmatrix} \cdots \right)}_{P} \underbrace{\begin{pmatrix} 0 \\ | \\ 1 \\ | \\ 0 \end{pmatrix}}_{u} - o_u = P \cdot u$$

$$q_i^T p_u = \langle q_i, p_u \rangle = \langle Q \cdot it, P \cdot u \rangle$$

$$= (Q \cdot it)^T \cdot P u = it \, Q^T P u$$

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

RS as machine learning tasks

- Fill matrix of assessments
  - Regression
  - Classification
  - Preferences

RS as a machine learning task

	p1	p2	p3	p4
u1	4	*	*	9
u2	*	7	4	*
u3	*	*	3	8
u4	5	5	*	*

The matrix of assessments contains the marks given by users (rows) to the items (columns)

The goal is to fill the matrix ***according*** to the available scores

Thus we must ***learn*** a function

$$f_\theta(\boldsymbol{u}, \boldsymbol{p})$$

that depends on a set of parameters

$$\theta$$

RS as a machine learning task

The meaning of the word ***according*** is of key importance. It will define the kind of learning task

If we think that cell scores are

- Exact values: *regression*
- Labels of a finite set: *classification*
- Clues to order: *preferences*, *ranking*

Fill matrix with regression

	p1	p2	p3	p4
u1	4	*	*	9
u2	*	7	4	*
u3	*	*	3	8
u4	5	5	*	*

- Assuming that the available scores are numeric and reliable
- Regression may fill the matrix trying to minimize the difference from predictions and real values
- Netflix award

Fill matrix with regression

The learning task is defined by the dataset

$$D = \{(\boldsymbol{u}, \boldsymbol{p}; \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p})) : \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p}) \quad \text{Available} \quad \}$$

The aim is to solve

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{D} \left(f_\theta(\boldsymbol{u}, \boldsymbol{p}) - \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p})\right)^2 + \nu r(\theta)$$

where the last summand is a *regularization* parameter

Fill matrix with a classifier

	p1	p2	p3	p4
u1	👎	*	*	👍
u2	*	👍	👎	*
u3	*	*	👎	👍
u4	👎	👎	*	*

- Assuming reliable labels of a finite set in cells
- Matrix can be filled using a classifier

Fill matrix with a classifier

The learning task is defined by the dataset

$$D = \{(\boldsymbol{u}, \boldsymbol{p}; \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p})) : \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p}) = +1, -1\}$$

The aim is to solve

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{D} \max\{0, 1 - \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p}) f_\theta(\boldsymbol{u}, \boldsymbol{p})\} + \nu r(\theta)$$

Fill matrix learning preferences

	p1	p2	p3	p4
u1	4	*	*	9
u2	*	7	4	*
u3	*	*	3	8
u4	5	5	*	*

$$\boldsymbol{M}(\boldsymbol{u_1}, \boldsymbol{p_4}) > \boldsymbol{M}(\boldsymbol{u_1}, \boldsymbol{p_1})$$
$$\boldsymbol{M}(\boldsymbol{u_2}, \boldsymbol{p_2}) > \boldsymbol{M}(\boldsymbol{u_2}, \boldsymbol{p_3})$$
$$\boldsymbol{M}(\boldsymbol{u_3}, \boldsymbol{p_4}) > \boldsymbol{M}(\boldsymbol{u_3}, \boldsymbol{p_3})$$

The scores of items are only considered as relative comparisons for each user

- For instance, u1 prefers item p4 over p1. However, we are not sure about the absolute scores
- When users are not professionals, the marks assigned are not trustable, but they are reliable as relative comparisons

Fill matrix learning preferences

dataset $\mathcal{D}$

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{u_1}, \boldsymbol{p_4}) &> \boldsymbol{M}(\boldsymbol{u_1}, \boldsymbol{p_1}) & & (\boldsymbol{u_1}, \boldsymbol{p_4}, \boldsymbol{p_1}) \\ \boldsymbol{M}(\boldsymbol{u_2}, \boldsymbol{p_2}) &> \boldsymbol{M}(\boldsymbol{u_2}, \boldsymbol{p_3}) & \Rightarrow \quad & (\boldsymbol{u_2}, \boldsymbol{p_2}, \boldsymbol{p_3}) \\ \boldsymbol{M}(\boldsymbol{u_3}, \boldsymbol{p_4}) &> \boldsymbol{M}(\boldsymbol{u_3}, \boldsymbol{p_3}) & & (\boldsymbol{u_3}, \boldsymbol{p_4}, \boldsymbol{p_3}) \end{aligned}$$

$$\boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p_b}) > \boldsymbol{M}(\boldsymbol{u}, \boldsymbol{p_w}) \Rightarrow [\boldsymbol{u}, \boldsymbol{p_b}, \boldsymbol{p_w}] \in \mathcal{D}$$

Triples of

- user
- 2 items with different marks:
  - one is better (b) than the other (w)

Fill matrix learning preferences

In this case, we must solve

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{D} \max\{0, 1 - f_\theta(\boldsymbol{u}, \boldsymbol{p}_b) + f_\theta(\boldsymbol{u}, \boldsymbol{p}_w)\} + \nu r(\theta)$$

Matrix factorization

step by step

6 = 3 * 2
M1 = M2 * M3

Factorization

- When viewing RSs as a Machine Learning task, no indication was made about the form that the function f should have, capable of filling in the matrix of evaluations.
- In any case, we will always assume that both products and users can be represented by vectors.

Factorization

A general expression for f (the filling function) is this

$$M(\boldsymbol{u},\boldsymbol{p}) \cong f(\boldsymbol{u},\boldsymbol{p}) = \sum_{i,j} \boldsymbol{x}_{ij}\boldsymbol{u}_i\boldsymbol{p}_j$$

Then, learning **f** is learning **X** whose components are the weights (xij) of each pair of user item.

But this matrix can have an unmanageable dimension

To overcome this problem, we can determine **f** by means of two matrices that **factor X**.

$$f(\boldsymbol{u},\boldsymbol{p}) = \boldsymbol{u}^T\boldsymbol{X}\boldsymbol{p} = \boldsymbol{u^T}\boldsymbol{W^T}\boldsymbol{V}\boldsymbol{p} = \langle \boldsymbol{W}\boldsymbol{u}, \boldsymbol{V}\boldsymbol{p}\rangle$$

Factorization: geometric interpretation

The equation $f(\boldsymbol{u},\boldsymbol{p}) = \langle \boldsymbol{W}\boldsymbol{u}, \boldsymbol{V}\boldsymbol{p}\rangle$

means that we are **embedding** users and items into a common Euclidean space

$$\mathbb{R}^{|\mathcal{U}|} \to \mathbb{R}^k, \qquad \boldsymbol{u} \mapsto \boldsymbol{W}\boldsymbol{u};$$
$$\mathbb{R}^{|rep(\mathcal{P})|} \to \mathbb{R}^k, \qquad \boldsymbol{p} \mapsto \boldsymbol{V}\boldsymbol{p}.$$

and then **f** is proportional to the distance to an hyperplane

$$\begin{aligned} f(\boldsymbol{u},\boldsymbol{p}) &= \langle \boldsymbol{W}\boldsymbol{u}, \boldsymbol{V}\boldsymbol{p}\rangle \\ &= \|\boldsymbol{W}\boldsymbol{u}\|\|\boldsymbol{V}\boldsymbol{p}\|cos(\boldsymbol{W}\boldsymbol{u}, \boldsymbol{V}\boldsymbol{p}) \\ &= \|\boldsymbol{W}\boldsymbol{u}\|d(hyp(\boldsymbol{W}\boldsymbol{u}, \boldsymbol{V}\boldsymbol{p})) \end{aligned}$$

Factorization: geometric interpretation