The movie recommender system

Rodion Efremov a.k.a. Machine Funkeehs

Project in Practical Machine Learning, spring 2015, Department of Computer Science, University of Helsinki

Objective of the machine learning system

Objective of the machine learning system

- Let a user rate some movies of her/his choice. A rating is an integer within range $[1, 5]$.

Objective of the machine learning system

- Let a user rate some movies of her/his choice. A rating is an integer within range $[1, 5]$.
- After rating, recommend some movies to the user taking her/his ratings into account.

Basic “training data”

Basic “training data”

- The smallest **movielens** data package.

Basic “training data”

- The smallest **movielens** data package.
- Contains 943 users, 1682 movies and 1e5 ratings.

Basic “training data”

- The smallest **movielens** data package.
- Contains 943 users, 1682 movies and 1e5 ratings.
- Of course, the system supports adding more users and ratings to the database.

Basic “training data”

- The smallest **movielens** data package.
- Contains 943 users, 1682 movies and 1e5 ratings.
- Of course, the system supports adding more users and ratings to the database.
- Ratings may be updated or removed.

Basic “training data”

- The smallest **movielens** data package.
- Contains 943 users, 1682 movies and 1e5 ratings.
- Of course, the system supports adding more users and ratings to the database.
- Ratings may be updated or removed.
- It is not, however, possible to modify the movie set in our implementation.

Solving recommendation problem

Solving recommendation problem

- How do we recommend movies to a user U_0?

Solving recommendation problem

- How do we recommend movies to a user U_0?
- Find, say k, other users $U_1, \ldots, U_k$ that act **like** U_0 and recommend U_0 whatever $U_1, \ldots, U_k$ tend to like!

Solving recommendation problem

- How do we recommend movies to a user U_0?
- Find, say k, other users $U_1, \ldots, U_k$ that act **like** U_0 and recommend U_0 whatever $U_1, \ldots, U_k$ tend to like!
- So we use the famous k-nearest neighbor algorithm.

Solving recommendation problem

- How do we recommend movies to a user U_0?
- Find, say k, other users $U_1, \dots, U_k$ that act **like** U_0 and recommend U_0 whatever $U_1, \dots, U_k$ tend to like!
- So we use the famous k-nearest neighbor algorithm.
- What does **like** means? We need a **similarity measure** here...

Similarity measure

Similarity measure

- The basic Jaccard-coefficient is applicable here:

$$d(U_i, U_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}},$$

Similarity measure

- The basic Jaccard-coefficient is applicable here:

$$d(U_i, U_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}},$$

where

Similarity measure

- The basic Jaccard-coefficient is applicable here:

$$d(U_i, U_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}},$$

  where
  - f_{01} is the amount of all movies seen by the user U_j,

Similarity measure

- The basic Jaccard-coefficient is applicable here:

$$d(U_i, U_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}},$$

where

  - f_{01} is the amount of all movies seen by the user U_j,
  - f_{10} is the amount of all movies seen by the user U_i,

Similarity measure

- The basic Jaccard-coefficient is applicable here:

$$d(U_i, U_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}},$$

where

  - f_{01} is the amount of all movies seen by the user U_j,
  - f_{10} is the amount of all movies seen by the user U_i,
  - $f_{11} = |M|$ is the amount of all movies seen by **both** U_j and U_i.

The actual similarity measure used

The actual similarity measure used

We used the following measure:

$$d(U_i, U_j) = \frac{f_{11} - \sigma(U_i, U_j)}{f_{01} + f_{10} + f_{11}},$$

The actual similarity measure used

We used the following measure:

$$d(U_i, U_j) = \frac{f_{11} - \sigma(U_i, U_j)}{f_{01} + f_{10} + f_{11}},$$

where

$$\sigma(U_i, U_j) = \sum_{m \in M} \frac{|r_i(m) - r_j(m)|}{5},$$

The actual similarity measure used

We used the following measure:

$$d(U_i, U_j) = \frac{f_{11} - \sigma(U_i, U_j)}{f_{01} + f_{10} + f_{11}},$$

where

$$\sigma(U_i, U_j) = \sum_{m \in M} \frac{|r_i(m) - r_j(m)|}{5},$$

where

- M is the set of movies which both the users U_i and U_j have seen,

The actual similarity measure used

We used the following measure:

$$d(U_i, U_j) = \frac{f_{11} - \sigma(U_i, U_j)}{f_{01} + f_{10} + f_{11}},$$

where

$$\sigma(U_i, U_j) = \sum_{m \in M} \frac{|r_i(m) - r_j(m)|}{5},$$

where

- M is the set of movies which both the users U_i and U_j have seen,
- $r_x(m)$ gives the rating score for the movie m by user U_x.

The actual similarity measure used

We used the following measure:

$$d(U_i, U_j) = \frac{f_{11} - \sigma(U_i, U_j)}{f_{01} + f_{10} + f_{11}},$$

where

$$\sigma(U_i, U_j) = \sum_{m \in M} \frac{|r_i(m) - r_j(m)|}{5},$$

where

- M is the set of movies which both the users U_i and U_j have seen,
- $r_x(m)$ gives the rating score for the movie m by user U_x.
- $|r_i(m) - r_j(m)|$ is the integer within interval $[0, 4]$.

The actual similarity measure used

So we use the Jaccard coefficient, but we penalize the similarity at those movies that have drastically different rating scores and we do not penalize at all those movies that have the same scores (as given by the two users, whose similarity is measured).