**Building Recommendation Systems**

1. Access movie recommendation system:

<http://www.movielens.org/>

username: [cenlkang@gmail.com](mailto:cenlkang@gmail.com), password: butterfly (Create your own account)

self rate a list of movies <so the system may match me to similar users later>

exam the movies recommended to me

1. How to build recommendation system?

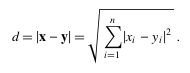
* User based approach
* Item based approach

1. User based approach

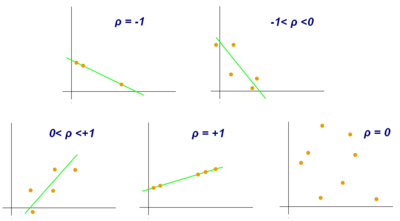
* Compute how similarity people are in their taste
* How to represent each person? What does the data look like?
  + - <http://www.grouplens.org/> (Data Sets 🡪 ML 100k)
    - README, u.item, u.user, u.data

Each person is represented with

* Compute similarity between pair-wise people
  + Similarity measures
    - Euclidean distance



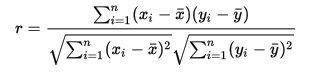
* + - Pearson correlation
      * measure of the linear dependence ([correlation](https://en.wikipedia.org/wiki/Correlation)) between two variables *X* and *Y*. It has a value between +1 and −1 inclusive, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation.



* + - * Pearson's correlation coefficient between two variables is defined as the [covariance](http://en.wikipedia.org/wiki/Covariance) of the two variables divided by the product of their [standard deviations](http://en.wikipedia.org/wiki/Standard_deviations):

:::Desktop:Screen shot 2012-01-22 at 11.44.17 AM.png

The above formula defines the *population* correlation coefficient, commonly represented by the Greek letter *ρ* (rho). Substituting estimates of the covariance and variances based on a [sample](http://en.wikipedia.org/wiki/Statistical_sample) gives the *sample correlation coefficient*, commonly denoted *r* :



Or the following used in the programming:



* + Load (movie) preference data
  + Compute the top 5 matched person from a given user
    - Compare the results from Euclidean distance and Pearson Coefficient
  + Get the recommendations
* Project Questions:
  + Can Jaccard coefficient, Manhattan distance, Tanimoto score be used to compute similarity in these cases? Write python functions to compute each of these similarity values. Which top 5 matches are returned in each case?
  + Currently, all, except for those with negative similarity value, user’s ratings are used for computing the ranking. How can this be modified to improve the way ranking maybe computed? (only the top n most similar user’s rating should be used)
* Python notes
  + 2 spaces per indentation
  + nested dictionary
  + setdefault usage

DATA\_SOURCE = (('movieA', '4'), ('movieA', '3.4'), ('movieB', '3'), ('movieB', '5'), ('movieB', '4.4'),)

**version1: test for empty key explicitly in code:**

newdata = {}

for k, v in DATA\_SOURCE:

if newdata.has\_key(k):

newdata[k].append(v)

else:

newdata[k] = [v]

print newdata

**version2: use setdefault to test for empty key implicitly**

newdata = {}

for k, v in DATA\_SOURCE:

newdata.setdefault(k, []).append(v)

print newdata

result from the two versions are the same:

{'movieB': ['3', '5', '4.4'], 'movieA': ['4', '3.4']}