

Multi dimensional Space

→ we have 18 genres, so in order to know the angle b/w the movies

so we need 18-dimensional space

(above, Comedy & action → 2-D)

→ In above, we find similarity like this
we will find the ~~cosine~~ angle (θ) b/w ~~books~~ movies

$$\cos \theta = x \Rightarrow x \text{ is similarity.}$$

Multi Dimensional cosine

$$\text{CosSim}(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

Time Similarity

• Same years of Movie

↳ Because 1800's movies & 2000's movies

are different right. (some people like latest movies)

↳ So we will find the similarity the release years.

Time Similarity

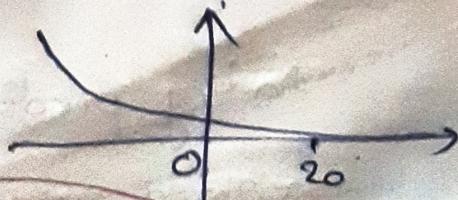
def YearSimilarity(self, movie1, movie2,

diff = abs(years[movie1] - years[movie2])

To Convert the similarity score to 0 & 1
similarity score ← $\text{Sim} = \text{math.exp}(-\text{diff}/10)$

to 0 & 1;

So, we have chosen exponential function



at 0 → similarity score is at 20 → similarity is very less

Q) How do we turn these similarities into actual rating predictions
(blu movies based on their attributes)

↳ Super lib will only find the actual rating of a user for a given movie

↳ So, how to include these content based similarities to get effective actual rating in order

Ans:- KNN

↳ In KNN we will select some number (k) of the nearest neighbours; & we will predict ratings for these " k " members

↳ we can define k nearest neighbours means highest Content similarity scores.

↳ To turn these top k^{nearest} movies into actual rating prediction, we can just take a weighted average of their similarity scores to the movie.

Similarity scores
b/w this movie & all others, the user rated

Sort

TOP K nearest Movies

weighted average

Rating Prediction

Selecting

Content based {
top K movies which are having high similarity scores
called as nearest neighbors }

~~weighted average of top K nearest movies.~~