

# COMP 474/6741 Intelligent Systems (Winter 2021)

## Worksheet #5: Recommender Systems

**Task 1.** Let's take some movies that have been #tagged (or categorized) as follows:

	Action	Comedy	Sci-Fi	Horror	Drama	Romance	length
Movie 1	4	8	6	3	0	0	
Movie 2	0	5	0	8	5	0	
Movie 3	1	4	0	3	0	10	

So, each movie becomes a 6-dimensional vector of tags  $t_i$ , e.g.,  $\overrightarrow{\text{Movie}_1} = \langle 4, 8, 6, 3, 0, 0 \rangle$ . Compute the *length* of each movie vector, which is defined as  $\|\vec{m}\| = \sqrt{t_1^2 + \dots + t_n^2}$  (rounded to two digits).

**Task 2.** Now you can *normalize* the vectors, by dividing the raw count of each tag  $t_i$  by the length  $\frac{t_i}{\|\vec{m}\|}$ :

	Action	Comedy	Sci-Fi	Horror	Drama	Romance
Movie 1						
Movie 2						
Movie 3						

Use 4 significant digits for this table (protip: the *length* of each movie vector must now be 1).

此表使用4位有意义的数字 (提示: 现在每个电影向量的长度必须是1)。

**Task 3.** We can now compute the *similarity* between the movies by computing their *cosine similarity*. Since the vectors are normalized, this is simply their dot product:  $\text{sim}(\vec{m}, \vec{n}) = \cos(\vec{m}, \vec{n}) = \vec{m} \cdot \vec{n} = \sum_i m_i \cdot n_i$ :

	Movie 1	Movie 2	Movie 3
Movie 1	1		
Movie 2		1	
Movie 3			1

This is the information we need for an *item-to-item recommendation engine*: Now we can answer the question, which movie is interesting to (buy, watch) for a customer who (bought, watched) Movie 1?

**Task 4.** Now we want to *personalize* the recommendations. We collected the following profiles about the movies watched (bought) by our users in the past:

	Action	Comedy	Sci-Fi	Horror	Drama	Romance	length
Jane	1	2	1	1	1	0	
Joe	0	1	0	1	0	1	

Compute the length of each *user vector* and normalize it like before:

	Action	Comedy	Sci-Fi	Horror	Drama	Romance
Jane						
Joe						

**Task 5.** Now we can answer the question which movie a user is interested in. Compute the cosine similarity between the *user vectors* and the *movie vectors*:

	Movie 1	Movie 2	Movie 3
Jane			
Joe			

**Task 6.** Consider the results from three different recommender systems below: Here, X1–X5 are the items (movies, photos, songs, ...) that the systems should have recommended as relevant for a specific user. The remaining 495 instances are not relevant for the user. A checkmark indicates that a system recommended this item to the user:

	Target	system 1	system 2	system 3
	X1 ✓	X1 ✗	X1 ✓	X1 ✓
	X2 ✓	X2 ✗	X2 ✗	X2 ✓
	X3 ✓	X3 ✗	X3 ✓	X3 ✓
	X4 ✓	X4 ✗	X4 ✓	X4 ✓
	X5 ✓	X5 ✗	X5 ✗	X5 ✓
	X6 ✗	X6 ✗	X6 ✗	X6 ✓
	X7 ✗	X7 ✗	X7 ✗	X7 ✓
	... ✗	...	... ✗	... ✗
	... ✗	...	... ✗	... ✗
	X500 ✗	X500 ✗	X500 ✗	X500 ✗

Evaluate the performance of the three systems using the measures *Precision* and *Recall*:

	Precision	Recall
system 1		
system 2		
system 3		

**Task 7.** Based on the output below, compute  $\text{precision@k} = \frac{1}{k} \cdot \sum_{c=1}^k \text{rel}(c)$  for the three recommender systems (for  $k = 1, 2, 3$ ):

	rel( $k$ )			precision@k			
	1	2	3	1	2	3	AP@3
system 1	1	0	0				
system 2	0	1	0				
system 3	0	0	1				

That is, each system got exactly one recommendation right, but in a different position.

**Task 8.** Moving on to the *average precision*,  $\text{AP@N} = \frac{1}{m} \sum_{k=1}^N \text{precision@k} \cdot \text{rel}(k)$ . Compute the AP@3 and add it to the table above. Here, assume  $m = 3$  (i.e., there could have been 3 correct recommendations in the top-3). Note the difference in the AP@3 for the three systems.

**Task 9.** Create a *document vector* for the movie description  $m_1 = \text{"A comedy with zombies."}$  Start by filling in the tf values below. Then compute  $\text{idf} = \log_{10} \frac{N}{\text{df}}$  (assume  $N = 10,000,000$ ) and  $\text{tf.idf} = \text{tf} \times \text{idf}$  (i.e., no log weighting for tf). Finally, compute the normalized vector  $\vec{q}$  as before.

token	$m_1$				$q_i$
	tf	df	idf	tf.idf	
action		50,000			
comedy		10,000			
zombies		100,000			
romantic		10,000			