

## Coursera 16-3 Collaborative-filtering

<https://www.coursera.org/learn/machine-learning/lecture/2WoBV/collaborative-filtering>

### Problem motivation

Movie	Alice (1) $\theta^{(1)}$	Bob (2) $\theta^{(2)}$	Carol (3) $\theta^{(3)}$	Dave (4) $\theta^{(4)}$	$x_1$ (romance)	$x_2$ (action)
$x^{(1)}$ Love at last	5	5	0	0	1.0	0.0
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

$x^{(1)} = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}$   
 $\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$   
 $\theta^{(j)}$   
 $(\theta^{(1)})^T x^{(1)} \approx 5$   
 $(\theta^{(2)})^T x^{(1)} \approx 5$   
 $(\theta^{(3)})^T x^{(1)} \approx 0$   
 $(\theta^{(4)})^T x^{(1)} \approx 0$

Andrew Ng

- feature learning이라 부른다. 분석에 사용하는 feature들을 스스로 학습할 수 있는 알고리즘
- 얼마나 romantic한지, action요소가 많은지 측정하는 것은 어렵고 비용이 많이 든다.
- $(\theta^{(j)})^T x^{(i)}$  식을 이용하여  $x_1, x_2$ 를 유추한다.

## Optimization algorithm

Given  $\theta^{(1)}, \dots, \theta^{(n_u)}$ , to learn  $x^{(i)}$ :

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{k=1}^n (x_k^{(i)})^2$$

Given  $\theta^{(1)}, \dots, \theta^{(n_u)}$ , to learn  $x^{(1)}, \dots, x^{(n_m)}$ :

$$\min_{x^{(1)}, \dots, x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2$$

- 최적화 알고리즘은 Content Based Recommendations에서 배운 것과 동일하다.

## Collaborative filtering

Given  $x^{(1)}, \dots, x^{(n_m)}$  (and movie ratings),  
can estimate  $\theta^{(1)}, \dots, \theta^{(n_u)}$

$r^{(i,j)}$   
 $y^{(i,j)}$

Given  $\theta^{(1)}, \dots, \theta^{(n_u)}$ ,  
can estimate  $x^{(1)}, \dots, x^{(n_m)}$

Guess  $\Theta \rightarrow x \rightarrow \Theta \rightarrow x \rightarrow \Theta \rightarrow x \rightarrow \dots$

- 1번식,  $x^{(1)} \dots x^{(n_m)}$  영화의 특성과 평점이 주어지면 다른 사용자들의 성향(세타)를 구할 수 있다.
- 2번식, 사용자의 성향(세타)이 주어지면 영화들의 특성을 구할 수 있다.
- 1번을 먼저 학습하는지, 2번을 먼저 학습하는지 의문이 든다. 하지만 무엇이 먼저인지가 중요한 것이 아니라 사용자의 성향(세타), 영화의 특성( $x_i$ )를 찾는 과정을 반복하여 합리적인 값을 유추해야 한다.
- 정리하면, 협업 필터링은 알고리즘을 실행할 때 모든 사용자가 영화평점을 잘 예측하기 위한 작업이라 할 수 있다. 자신이 평가한 영화평점이 다른 사람들이 영화 평점을 예측하는 시스템에 사용될 수 있다.
- 다음 비디오에선 좀더 효율적인 협업 필터링 알고리즘을 배울 예정이다.

\*단어공부\*

Pose : (문제,질문) 제기하다, 포즈를 취하다

Iterating: 반복적

Simultaneously : 동시에

Computationally : 계산적으로