Week 9

Recommender Systems - Collaborative filtering

Problem motivation

feature vector

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	0.9	0
Romance forever	5	?	?	0	1.0	0.01
	?	4	0	?	0.99	0
	0	0	5	4	0.1	1.0
	0	0	5	4	0	0.9

Suppose we have a dataset where we do not know the values of x_1 & x_2 i.e

	x_1 (romance)	x_2
probably	1.0	0.0
	?	?
	?	?
	?	?
	?	?

no idea how romantic each movie is. No idea how action packed each movie is.

$$\theta^1 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \quad \theta^2 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \quad \theta^3 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, \quad \theta^4 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

$x_0 = 1$

Alice & bob both really like romance

Carol & dave both tell us that they really like action

∴ what value of $x^{(1)}$ gives!

$$(\theta^1)^T x^{(1)} \approx 5$$
$$(\theta^2)^T x^{(1)} \approx 5$$
$$(\theta^3)^T x^{(1)} \approx 0$$
$$(\theta^4)^T x^{(1)} \approx 0$$

Week 9

Recommender Systems - Collaborative filtering

one specific movie case:

optimization algorithm

lets say that the users have told us their preferences.

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

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

— regularization term (prevents features from becoming too big

we want to choose values for the feature vector x such that the squared error is minimised.

squared error term

we want to learn the features for all the movies:

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

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

sum over all n_m movies etc

minimise the above will hopefully produce a reasonable set of features for ~~the~~ all the movies.

Week 9

Recommender Systems - Collaborative filtering algorithms

all pairs of i, j

$(i,j): r(i,j) = 1$

where $r(i,j) = 1$

Collaborative filtering optimization objective

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

$$\min_{\theta^{1} \dots \theta^{n_u}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left(\quad \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2$$

all users "j" sum over $\rightarrow$ $j=1$

$i:r(i,j)=1 \rightarrow$ sum over all movies rated by that user.

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

$$\min_{x^{1}, \dots, x^{n_m}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left(\quad \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(i)}\right)^2$$

for every movie i $\rightarrow$ $i=1$

$j:r(i,j)=1 \rightarrow$ sum over all users that have rated that movie

minimising $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ simultaneously!

$$J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)} \dots \theta^{(n_u)}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\quad \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} x_k^2 + \frac{\lambda}{2} \sum \sum \theta$$

$$\boxed{\min_{\substack{x^{(1)}, \dots, x^{(n_m)} \\ \theta^{(1)}, \dots, \theta^{(n_u)}}}} \rightarrow$$

instead of $\theta \rightarrow x \rightarrow \theta \rightarrow$ we are going to minimise with both sets of parameters simultaneously.

$x_0 = 1$ corresponds to an intercept term

doing it this way however yields: $\cancel{x_0 = 1}$; $\boxed{x \in \mathbb{R}^n}$, $x \cancel{\in \mathbb{R}^{n+1}}$

$\theta \in \mathbb{R}^n$

Week 9

Recommender Systems – Collaborative filtering

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

also if we know the features $x^{(1)}, \dots x^{(n_m)}$, we can learn the parameters θ

~~also~~

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

Guess $\theta \xrightarrow[\text{learn features for movies}]{} x \xrightarrow{\text{learn}} \theta \xrightarrow[\text{learn etc}]{} x \to \theta \to x$.

$\boxed{\text{actually works}}$ ✓✓

Week 9

~~$x_0 = 1$~~ $x \in \mathbb{R}^n$, $\Theta \in \mathbb{R}^n$

∴ regularizing everything.

Recommender Systems - Collaborative filtering algorithm

1.) Initialize $x^{(1)}, \dots, x^{(n_m)}, \Theta^{(1)}, \dots \Theta^{(n_u)}$ to small random values

2.) Minimise $J(x^{(1)}, \dots, x^{(n_m)}, \Theta^{(1)}, \dots, \Theta^{(n_u)})$ using gradient descent (or some other optimization algorithm e.g for every $j = 1, \dots, n_u$, $i = 1, \dots, n_m$:

$$x_k^{(i)} := x_k^{(i)} - \alpha \left(\sum_{j:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)}) \Theta_k^{(j)} + \lambda x_k^{(i)} \right)$$

$$\Theta_k^{(j)} := \Theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \Theta_k^{(j)} \right)$$

from the partial derivatives of the cost function

3) for a user with parameters Θ and a movie with (learned) features x, predict a star rating of $\Theta^T x$.

$(\Theta^{(j)})^T (x^{(i)})$ user j is going to rate movie i

Week 9

Recommender Systems: Vectorization low rank matrix factorization

Collaborative filtering

movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
~	?	4	0	?
~	0	0	5	4
~	0	0	5	?

$n_m = 5$ (movies)

$n_u = 4$ (users)

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

$y^{(i,j)}$ — movies of (i), users (j)

Collaborative filtering

$$Y = \begin{bmatrix} \\ \\ \end{bmatrix}$$

$X \Theta^T$ $\quad (\theta^{(j)})^T (x^{(i)})$ — user j, movie i; (i,j)

predicted ratings:

predicted rating user 2 on movie 1

$$\begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & \boxed{(\theta^{(2)})^T(x^{(1)})} & \cdots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \cdots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \cdots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

$$X = \begin{bmatrix} - (x^{(1)})^T - \\ - (x^{(2)})^T - \\ \vdots \\ - (x^{(n_m)})^T - \end{bmatrix} \qquad \Theta = \begin{bmatrix} - (\theta^{(1)})^T - \\ - (\theta^{(2)})^T - \\ \vdots \\ - (\theta^{(n_u)})^T - \end{bmatrix}$$

capital θ

Collaborative filtering is also called Low-rank matrix factorization.

Week 9

Recommender systems: vectorization, low rank matrix factorization

finding related movies:

for each product i, we learn a feature vector $x^{(i)} \in \mathbb{R}^n$

x_1 = romance, x_2 = action, x_3 = comedy, x_4 = ...

How to find movies j related to movie i?

different movie j

movie i has a feature vector

$\|x^{(i)} - x^{(j)}\|$ ← if the difference between movie i & j is minimised, this is a pretty strong indication that they are similar..

eg. 5 most similar movies to movie i:

→ find the 5 movies j with the smallest $\|x^{(i)} - x^{(j)}\|$

Week 9

Recommender Systems - Implementational detail: mean normalization

Users who have not rated any movies

lets say we add a 5^{th} user "eve"...

$n=2$ (romance / action), $\theta^{(s)} \in \mathbb{R}^2$ | ~~th~~ $\theta^5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

the first term of the cost function plays no role because there are no cases where $\boxed{(i,j) ? r(i,j) = 1}$

the ~~t~~ only term that affects is $= \frac{1}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2$

$\hookrightarrow \frac{\lambda}{2} \left[(\theta_1^{(5)})^2 + (\theta_2^{(5)})^2 \right]$

so we want to minimize

$\therefore (\theta^5)^T x^{(i)} = 0$

$\therefore$ eve will rate all movies as 0 stars.

Mean Normalization:

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix}$$

$\hookrightarrow$ eve's unrated movies.

the average value that each movie obtains

$$\mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

subtract off the mean rating.

pretend that this was the data that I got from my users.

stay the same

learn θ^j, x^i

for user j, on movie i predict:

$(\theta^{(j)})^T (x^{(i)}) + \mu_i$

user 5 (eve):

$\theta^5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\underbrace{(\theta^{(5)})^T (x^{(i)})}_{=0} + \mu_i$

$\therefore$ eve's parameters will be the average