

Practica 2 – Sistema de Recomendación

Alumno: Fco Javier Vargas García-Donas

Resolución

La resolución del ejercicio se ha planteado en el desarrollo de los siguientes ficheros

cofiCostFunc.m

Para la resolución de esta función se ha traducido la notación usada en el código a la notación usada en clase.

```
% Notation
% W = Theta'
% H = estimated_Y = X W
W = Theta';

H = X * W;      % Y_est = X * W
J = 1/2 * sum(sum( R .* (X*W - Y).^2 )) + lambda/2 * sum(sum( X.^2 )) +
    lambda/2 * sum(sum( W.^2 ));

Theta_grad = (X*W - Y).*R.'*X + lambda*W';
X_grad      = (X*W - Y).*R * W' + lambda*X;
```

Luego se ha implementado la función de coste con regularización (inclusión del parámetro lambda para evitar que los valores de los parámetros a aprender “x” e “w” no tomen un rango de valores muy amplio).

Este código corresponde a las expresiones analíticas:

$$J = \frac{1}{2} \sum_{j=1}^{n_y} \sum_{i:j:r(i,j)=1} \left(\mathbf{x}^{(i)} \mathbf{w}^{(j)} - y(i,j) \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_y} \sum_{k=1}^n (w_k^{(j)})^2$$

$$\begin{aligned} \forall j, \quad \mathbf{w}_k^{(j)} &:= \mathbf{w}_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} (\mathbf{x}^{(i)} \mathbf{w}^{(j)} - y(i,j)) \mathbf{x}_k^{(i)} + \lambda \mathbf{w}_k^{(j)} \right) \\ \forall i, \quad \mathbf{x}_k^{(i)} &:= \mathbf{x}_k^{(i)} - \alpha \left(\sum_{j:r(i,j)=1} (\mathbf{x}^{(i)} \mathbf{w}^{(j)} - y(i,j)) \mathbf{w}_k^{(j)} + \lambda \mathbf{x}_k^{(i)} \right) \end{aligned}$$

recsys.m

Se ha ido siguiendo paso a paso las diferentes partes del código y comprobando que las modificaciones realizadas pasan los test de comprobación incluidos en el código. A continuación se muestran algunos trozos del output del script.

```
If your backpropagation implementation is correct, then  
the relative difference will be small (less than 1e-9).  
Relative Difference: 1.45112e-12
```

```
Cost at loaded parameters (lambda = 1.5): 31.344056  
(this value should be about 31.34)
```

```
If your backpropagation implementation is correct, then  
the relative difference will be small (less than 1e-9).
```

```
Relative Difference: 1.82229e-12
```

A continuación, se ha puntuado alguna de las películas que venían en la base de datos en función de los gustos del autor:

New user ratings:

```
Rated 4 for Twelve Monkeys (1995)  
Rated 4 for Usual Suspects, The (1995)  
Rated 1 for Taxi Driver (1976)  
Rated 5 for Star Wars (1977) % I love it!  
Rated 4 for Pulp Fiction (1994)  
Rated 4 for Forrest Gump (1994) % I almost cry with it  
Rated 5 for Terminator 2: Judgment Day (1991) % Amazing!  
Rated 5 for Silence of the Lambs, The (1991) % Breathtaking  
Rated 4 for Jungle Book, The (1994)  
Rated 1 for Lawrence of Arabia (1962)
```

Para realizar una comparación intuitiva de los resultados del algoritmo se expone que al autor:

- Le gustan películas con acción, futuristas o con buen argumento
- No le gustan películas lentas (Taxi Driver o Lawrence de Arabia)

Tras entrenar al sistema recomienda una serie de películas (se añaden solo las que el autor ha visto)

```
Predicting rating 8.3 for movie Titanic (1997)  
Predicting rating 8.0 for movie Schindler's List (1993)  
Predicting rating 7.9 for movie Godfather, The (1972)  
Predicting rating 7.8 for movie Empire Strikes Back, The (1980)
```

De las cuales el autor puede confirmar que todas las películas recomendadas le gusta, aunque:

- La película del imperio contraataca estaría por encima de todas
- El padrino tendría mejor puntuación: aunque es sorprendente como el algoritmo recomienda El Padrino (a pesar de ser una película lenta) y aun así coincide con los gustos del autor.
- Titanic, tendría una nota un poco mas baja por ser una película un poco lenta aunque al autor le encanta.