Aula 21 - Sistemas de Recomendação

João Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
florindo@unicamp.br

Outline

- Bastante populares, especialmente na indústria.
- EX.: recomendação de produtos na Amazon, filmes na Netflix, playlists no Spotify, etc.

Filme	Alice	Bob	Carol	Dave
Love at least	5	5	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	?

- Bastante populares, especialmente na indústria.
- EX.: recomendação de produtos na Amazon, filmes na Netflix, playlists no Spotify, etc.

Filme	Alice	Bob	Carol	Dave
Love at least	5	5	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	?

- Bastante populares, especialmente na indústria.
- EX.: recomendação de produtos na Amazon, filmes na Netflix, playlists no Spotify, etc.

Filme	Alice	Bob	Carol	Dave
Love at least	5	5	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	?

Definimos:

$$
\begin{aligned}
n_u &= \text{número de usuários} \\
n_m &= \text{número de filmes} \\
r(i,j) &= \text{1 se o usuário } j \text{ deu nota para o filme } i\text{, 0 caso contrário} \\
y^{(i,j)} &= \text{nota dada pelo usuário } j \text{ para o filme } i \text{ (definido apenas se } r(i,j) = 1)
\end{aligned}
$$

NOTA

Os 3 primeiros filmes são românticos e os 2 últimos são de ação. Nota-se então uma preferência de Alice e Bob por romance e de Carol e Dave por ação.

Definimos:

$$
\begin{aligned}
n_u &= \text{número de usuários} \\
n_m &= \text{número de filmes} \\
r(i,j) &= \text{1 se o usuário } j \text{ deu nota para o filme } i\text{, 0 caso contrário} \\
y^{(i,j)} &= \text{nota dada pelo usuário } j \text{ para o filme } i \text{ (definido apenas se } r(i,j) = 1)
\end{aligned}
$$

NOTA

Os 3 primeiros filmes são românticos e os 2 últimos são de ação. Nota-se então uma preferência de Alice e Bob por romance e de Carol e Dave por ação.

Outline

- Criar um modelo de regressão linear para cada usuário.
- Um vetor de parâmetros para cada usuário e um vetor de atributos para cada filme.
- Definimos:

$$\begin{aligned} \theta^{(j)} &= \text{vetor de parâmetros para o usuário } j \\ x^{(i)} &= \text{vetor de atributos para o filme } i \\ m^{(j)} &= \text{número de filmes avaliados pelo usuário } j \end{aligned}$$

- Então, para o usuário j e filme i, a nota predita é dada por

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

- Como na regressão linear, convencionamos $x_0 = 1$, de modo que $x^{(i)} \in \mathbb{R}^{n+1}$ e $\theta^{(j)} \in \mathbb{R}^{n+1}$.

- Criar um modelo de regressão linear para cada usuário.
- Um vetor de parâmetros para cada usuário e um vetor de atributos para cada filme.
- Definimos:

$$\begin{aligned} \theta^{(j)} &= \text{vetor de parâmetros para o usuário } j \\ x^{(i)} &= \text{vetor de atributos para o filme } i \\ m^{(j)} &= \text{número de filmes avaliados pelo usuário } j \end{aligned}$$

- Então, para o usuário j e filme i, a nota predita é dada por

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

- Como na regressão linear, convencionamos $x_0 = 1$, de modo que $x^{(i)} \in \mathbb{R}^{n+1}$ e $\theta^{(j)} \in \mathbb{R}^{n+1}$.

- Criar um modelo de regressão linear para cada usuário.
- Um vetor de parâmetros para cada usuário e um vetor de atributos para cada filme.
- Definimos:

$$
\begin{aligned}
\theta^{(j)} &= \text{vetor de parâmetros para o usuário } j \\
x^{(i)} &= \text{vetor de atributos para o filme } i \\
m^{(j)} &= \text{número de filmes avaliados pelo usuário } j
\end{aligned}
$$

- Então, para o usuário j e filme i, a nota predita é dada por

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

- Como na regressão linear, convencionamos $x_0 = 1$, de modo que $x^{(i)} \in \mathbb{R}^{n+1}$ e $\theta^{(j)} \in \mathbb{R}^{n+1}$.

- Criar um modelo de regressão linear para cada usuário.
- Um vetor de parâmetros para cada usuário e um vetor de atributos para cada filme.
- Definimos:

$$\begin{aligned} \theta^{(j)} &= \text{vetor de parâmetros para o usuário } j \\ x^{(i)} &= \text{vetor de atributos para o filme } i \\ m^{(j)} &= \text{número de filmes avaliados pelo usuário } j \end{aligned}$$

- Então, para o usuário j e filme i, a nota predita é dada por

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

- Como na regressão linear, convencionamos $x_0 = 1$, de modo que $x^{(i)} \in \mathbb{R}^{n+1}$ e $\theta^{(j)} \in \mathbb{R}^{n+1}$.

- Criar um modelo de regressão linear para cada usuário.
- Um vetor de parâmetros para cada usuário e um vetor de atributos para cada filme.
- Definimos:

$$
\begin{aligned}
\theta^{(j)} &= \text{vetor de parâmetros para o usuário } j \\
x^{(i)} &= \text{vetor de atributos para o filme } i \\
m^{(j)} &= \text{número de filmes avaliados pelo usuário } j
\end{aligned}
$$

- Então, para o usuário j e filme i, a nota predita é dada por

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

- Como na regressão linear, convencionamos $x_0 = 1$, de modo que $x^{(i)} \in \mathbb{R}^{n+1}$ e $\theta^{(j)} \in \mathbb{R}^{n+1}$.

- Suponha 2 atributos x_1 (nível de romance) e x_2 (nível de ação):

Filme	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (ação)
Love at least (1)	5	5	0	0	0.9	0
Romance forever (2)	5	?	?	0	1.0	0.01
Cute puppies of love (3)	?	4	0	?	0.99	0
Nonstop car chases (4)	0	0	5	4	0.1	1.0
Swords vs. karate (5)	0	0	5	?	0	0.9

- Vamos estimar a nota dada pelo usuário 1 (Alice) para o filme 3 (Cute puppies of love). Temos:

$$x^{(3)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}.$$

Suponha que

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix},$$

então $y^{(31)}$ será predito por

$$(\theta^{(1)})^T x^{(3)} = 5 \cdot 0.99 = 4.95.$$

- Suponha 2 atributos x_1 (nível de romance) e x_2 (nível de ação):

Filme	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (ação)
Love at least (1)	5	5	0	0	0.9	0
Romance forever (2)	5	?	?	0	1.0	0.01
Cute puppies of love (3)	?	4	0	?	0.99	0
Nonstop car chases (4)	0	0	5	4	0.1	1.0
Swords vs. karate (5)	0	0	5	?	0	0.9

- Vamos estimar a nota dada pelo usuário 1 (Alice) para o filme 3 (Cute puppies of love). Temos:

$$x^{(3)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}.$$

Suponha que

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix},$$

então $y^{(31)}$ será predito por

$$(\theta^{(1)})^T x^{(3)} = 5 \cdot 0.99 = 4.95.$$

- Parâmetros $\theta^{(j)}$ aprendidos por regressão linear:

$$\underset{\theta^{(j)}}{\operatorname{argmin}} \frac{1}{2m^{(j)}} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T(x^{(i)}) - y^{(i,j)})^2 + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

- Em sistemas de recomendação não se inclui $m^{(j)}$ no denominador (para efeitos da minimização não faz diferença):

$$\underset{\theta^{(j)}}{\operatorname{argmin}} \frac{1}{2} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

- Parâmetros $\theta^{(j)}$ aprendidos por regressão linear:

$$\underset{\theta^{(j)}}{\text{argmin}} \frac{1}{2m^{(j)}} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T(x^{(i)}) - y^{(i,j)})^2 + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

- Em sistemas de recomendação não se inclui $m^{(j)}$ no denominador (para efeitos da minimização não faz diferença):

$$\underset{\theta^{(j)}}{\text{argmin}} \frac{1}{2} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

- Porém, temos n_u usuários e a minimização deve ser feita sobre todos eles, de modo que aprendemos $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$ por

$$\underset{\theta^{(1)},\cdots,\theta^{(n_u)}}{\text{argmin}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

- A função minimizada é a nossa função de custo (objetivo) $J(\theta^{(1)}, \cdots, \theta^{(n_u)})$ e podemos minimizá-la por gradiente descendente:

$$\begin{aligned} \theta_k^{(j)} &:= \theta_k^{(j)} - \alpha \textstyle\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} && (k = 0) \\ \theta_k^{(j)} &:= \theta_k^{(j)} - \alpha \left(\textstyle\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right) && (k \neq 0). \end{aligned}$$

- Porém, temos n_u usuários e a minimização deve ser feita sobre todos eles, de modo que aprendemos $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$ por

$$\underset{\theta^{(1)},\cdots,\theta^{(n_u)}}{\text{argmin}} \; \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

- A função minimizada é a nossa função de custo (objetivo) $J(\theta^{(1)}, \cdots, \theta^{(n_u)})$ e podemos minimizá-la por gradiente descendente:

$$\begin{aligned} \theta_k^{(j)} &:= \theta_k^{(j)} - \alpha \textstyle\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} \qquad (k = 0) \\ \theta_k^{(j)} &:= \theta_k^{(j)} - \alpha \left(\textstyle\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right) \quad (k \neq 0). \end{aligned}$$

Outline

- Como calculamos os valores dos atributos $x_1, x_2, \cdots$?

- Pedir para cada usuário rotular todos os filmes como de romance, ação, comédia, etc. não é viável na prática.

- Podemos aprender esses atributos automaticamente.

- Se conhecermos os vetores de parâmetros $\theta^{(j)}$, podemos obter $x^{(i)}$ minimizando o erro da previsão.

- EX.: $x^{(1)}$ deve satisfazer

$$\begin{aligned}(\theta^{(1)})^T x^{(1)} &\approx 5\\ (\theta^{(2)})^T x^{(1)} &\approx 5\\ (\theta^{(3)})^T x^{(1)} &\approx 0\\ (\theta^{(4)})^T x^{(1)} &\approx 0\end{aligned}.$$

- Como calculamos os valores dos atributos $x_1, x_2, \cdots$?
- Pedir para cada usuário rotular todos os filmes como de romance, ação, comédia, etc. não é viável na prática.
- Podemos aprender esses atributos automaticamente.
- Se conhecermos os vetores de parâmetros $\theta^{(j)}$, podemos obter $x^{(i)}$ minimizando o erro da previsão.
- EX.: $x^{(1)}$ deve satisfazer

$$\begin{aligned}
(\theta^{(1)})^T x^{(1)} &\approx 5 \\
(\theta^{(2)})^T x^{(1)} &\approx 5 \\
(\theta^{(3)})^T x^{(1)} &\approx 0 \\
(\theta^{(4)})^T x^{(1)} &\approx 0
\end{aligned}.$$

- Como calculamos os valores dos atributos $x_1, x_2, \cdots$?
- Pedir para cada usuário rotular todos os filmes como de romance, ação, comédia, etc. não é viável na prática.
- Podemos aprender esses atributos automaticamente.
- Se conhecermos os vetores de parâmetros $\theta^{(j)}$, podemos obter $x^{(i)}$ minimizando o erro da previsão.
- EX.: $x^{(1)}$ deve satisfazer

$$
\begin{array}{l}
(\theta^{(1)})^T x^{(1)} \approx 5 \\
(\theta^{(2)})^T x^{(1)} \approx 5 \\
(\theta^{(3)})^T x^{(1)} \approx 0 \\
(\theta^{(4)})^T x^{(1)} \approx 0
\end{array}.
$$

- Como calculamos os valores dos atributos $x_1, x_2, \cdots$?
- Pedir para cada usuário rotular todos os filmes como de romance, ação, comédia, etc. não é viável na prática.
- Podemos aprender esses atributos automaticamente.
- Se conhecermos os vetores de parâmetros $\theta^{(j)}$, podemos obter $x^{(i)}$ minimizando o erro da previsão.
- EX.: $x^{(1)}$ deve satisfazer

$$\begin{aligned}(\theta^{(1)})^T x^{(1)} &\approx 5\\(\theta^{(2)})^T x^{(1)} &\approx 5\\(\theta^{(3)})^T x^{(1)} &\approx 0\\(\theta^{(4)})^T x^{(1)} &\approx 0\end{aligned}.$$

- Como calculamos os valores dos atributos $x_1, x_2, \cdots$?

- Pedir para cada usuário rotular todos os filmes como de romance, ação, comédia, etc. não é viável na prática.

- Podemos aprender esses atributos automaticamente.

- Se conhecermos os vetores de parâmetros $\theta^{(j)}$, podemos obter $x^{(i)}$ minimizando o erro da previsão.

- EX.: $x^{(1)}$ deve satisfazer

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

- Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, obtemos $x^{(i)}$ por

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

- Mas temos n_m filmes e os vetores de atributo são aprendidos simultaneamente:

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

- Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, obtemos $x^{(i)}$ por

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

- Mas temos n_m filmes e os vetores de atributo são aprendidos simultaneamente:

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

- Ideia geral da **filtragem colaborativa**:
  - Dado $x^{(1)}, \cdots, x^{(n_m)}$ (e as notas dos filmes), estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$
  - Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$
- Porém, uma coisa depende da outra! Como resolver isso?
- R.: Chutar θ inicial pequeno aleatório e ir iterativamente calculando $\theta \to x \to \theta \to x \to \theta \to x \to \cdots$.
- Todos os usuários **colaboram** para a indicação de determinado filme.

- Ideia geral da **filtragem colaborativa**:
  - Dado $x^{(1)}, \cdots, x^{(n_m)}$ (e as notas dos filmes), estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$
  - Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$
- Porém, uma coisa depende da outra! Como resolver isso?
- R.: Chutar θ inicial pequeno aleatório e ir iterativamente calculando $\theta \to x \to \theta \to x \to \theta \to x \to \cdots$.
- Todos os usuários **colaboram** para a indicação de determinado filme.

- Ideia geral da **filtragem colaborativa**:
  - Dado $x^{(1)}, \cdots, x^{(n_m)}$ (e as notas dos filmes), estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$
  - Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$
- Porém, uma coisa depende da outra! Como resolver isso?
- R.: Chutar θ inicial pequeno aleatório e ir iterativamente calculando $\theta \to x \to \theta \to x \to \theta \to x \to \cdots$.
- Todos os usuários **colaboram** para a indicação de determinado filme.

- Ideia geral da **filtragem colaborativa**:
  - Dado $x^{(1)}, \cdots, x^{(n_m)}$ (e as notas dos filmes), estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$
  - Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$

- Porém, uma coisa depende da outra! Como resolver isso?

- R.: Chutar θ inicial pequeno aleatório e ir iterativamente calculando $\theta \to x \to \theta \to x \to \theta \to x \to \cdots$.

- Todos os usuários **colaboram** para a indicação de determinado filme.

- Ideia geral da **filtragem colaborativa**:
  - Dado $x^{(1)}, \cdots, x^{(n_m)}$ (e as notas dos filmes), estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$
  - Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$

- Porém, uma coisa depende da outra! Como resolver isso?

- R.: Chutar θ inicial pequeno aleatório e ir iterativamente calculando $\theta \to x \to \theta \to x \to \theta \to x \to \cdots$.

- Todos os usuários **colaboram** para a indicação de determinado filme.

- Ideia geral da **filtragem colaborativa**:
  - Dado $x^{(1)}, \cdots, x^{(n_m)}$ (e as notas dos filmes), estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$
  - Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$

- Porém, uma coisa depende da outra! Como resolver isso?

- R.: Chutar θ inicial pequeno aleatório e ir iterativamente calculando $\theta \rightarrow x \rightarrow \theta \rightarrow x \rightarrow \theta \rightarrow x \rightarrow \cdots$.

- Todos os usuários **colaboram** para a indicação de determinado filme.

Algoritmo

Função objetivo:

- Dado $x^{(1)}, \cdots, x^{(n_m)}$, estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$:

$$\underset{\theta^{(1)},\cdots,\theta^{(n_u)}}{\operatorname{argmin}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2$$

- Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$:

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

Podem ser juntados em um só:

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

$$\underset{\substack{x^{(1)},\cdots,x^{(n_m)}\\ \theta^{(1)},\cdots,\theta^{(n_u)}}}{\operatorname{argmin}} J(x^{(1)}, \cdots, x^{(n_m)}, \theta^{(1)}, \cdots, \theta^{(n_u)})$$

Algoritmo

Função objetivo:

- Dado $x^{(1)}, \cdots, x^{(n_m)}$, estimar $\theta^{(1)}, \cdots, \theta^{(n_u)}$:

$$\underset{\theta^{(1)},\cdots,\theta^{(n_u)}}{\operatorname{argmin}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2$$

- Dado $\theta^{(1)}, \cdots, \theta^{(n_u)}$, estimar $x^{(1)}, \cdots, x^{(n_m)}$:

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

Podem ser juntados em um só:

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

$$\underset{\substack{x^{(1)},\cdots,x^{(n_m)}\\ \theta^{(1)},\cdots,\theta^{(n_u)}}}{\operatorname{argmin}} J(x^{(1)}, \cdots, x^{(n_m)}, \theta^{(1)}, \cdots, \theta^{(n_u)})$$

Algoritmo

Algoritmo:

1. Inicializar $x^{(1)}, \cdots, x^{(n_m)}, \theta^{(1)}, \cdots, \theta^{(n_u)}$ com valores aleatórios pequenos (quebra de simetria como nas redes neurais)
2. Minimizar $J(x^{(1)}, \cdots, x^{(n_m)}, \theta^{(1)}, \cdots, \theta^{(n_u)})$ usando gradiente descendente (ou um algoritmo mais avançado de otimização). Assim, para todo $j = 1, \cdots, n_u, i = 1, \cdots, n_m$:

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

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

3. Para um usuário com parâmetro $\theta^{(j)}$ e um filme com atributos (aprendidos) $x^{(i)}$, prever a nota $(\theta^{(j)})^T (x^{(i)})$.

Algoritmo

NOTA

Aqui não temos mais a convenção $x_0 = 1$. Nosso parâmetro x_0 será aprendido por nosso algoritmo como todos os demais.

Outline

Nossa matriz de notas

Filme	Alice	Bob	Carol	Dave
Love at least	5	5	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	?

pode ser representada pela matriz Y:

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

- Notas preditas matricialmente:

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

  Definimos então as matrizes X com $(x^{(i)})^T$ em cada linha e Θ com $(\theta^{(j)})^T$ em cada linha:

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

  A matriz de predição pode ser representada vetorialmente por

$$X\Theta^T.$$

- O nome **fatorização de matriz de posto baixo** se deve ao fato de a matriz $X\Theta^T$ ter posto baixo.

- Notas preditas matricialmente:

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

  Definimos então as matrizes X com $(x^{(i)})^T$ em cada linha e Θ com $(\theta^{(j)})^T$ em cada linha:

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

  A matriz de predição pode ser representada vetorialmente por

$$X\Theta^T.$$

- O nome **fatorização de matriz de posto baixo** se deve ao fato de a matriz $X\Theta^T$ ter posto baixo.

Filmes Relacionados

- Dois filmes i e j são relacionados ("similares") se

$$\|x^{(i)} - x^{(j)}\|$$

  for pequeno.

- Se queremos encontrar, por exemplo, os 5 filmes mais relacionados ao filme i, buscamos pelos 5 filmes j com os menores $\|x^{(i)} - x^{(j)}\|$.

Filmes Relacionados

- Dois filmes i e j são relacionados ("similares") se

$$\|x^{(i)} - x^{(j)}\|$$

  for pequeno.

- Se queremos encontrar, por exemplo, os 5 filmes mais relacionados ao filme i, buscamos pelos 5 filmes j com os menores $\|x^{(i)} - x^{(j)}\|$.

Normalização pela Média

- Suponha que um usuário 5 (Eve) não deu nota para nenhum filme:

Filme	Alice	Bob	Carol	Dave	Eve
Love at least	5	5	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	?	?

- Até agora fizemos:

$$\underset{\substack{x^{(1)},\cdots,x^{(n_m)}\\ \theta^{(1)},\cdots,\theta^{(n_u)}}}{\operatorname{argmin}} \frac{1}{2}\sum_{(i,j):r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}(x_k^{(i)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2.$$

Normalização pela Média

- Suponha que um usuário 5 (Eve) não deu nota para nenhum filme:

Filme	Alice	Bob	Carol	Dave	Eve
Love at least	5	5	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	?	?

- Até agora fizemos:

$$\underset{\substack{x^{(1)},\cdots,x^{(n_m)} \\ \theta^{(1)},\cdots,\theta^{(n_u)}}}{\operatorname{argmin}} \frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2.$$

Normalização pela Média

- Note que este usuário influenciaria apenas no termo regularizador e $\theta^{(5)}$ otimizado seria o vetor nulo.
- Teríamos então
$$(\theta^{(5)})^T x^{(i)} = 0,$$
ou seja, o usuário 5 atribuiria 0 estrela para todos os filmes, o que não é nada realista.

Normalização pela Média

- Note que este usuário influenciaria apenas no termo regularizador e $\theta^{(5)}$ otimizado seria o vetor nulo.

- Teríamos então
$$(\theta^{(5)})^T x^{(i)} = 0,$$
ou seja, o usuário 5 atribuiria 0 estrela para todos os filmes, o que não é nada realista.

Normalização pela Média

- Solução: definir um vetor médio de nota por filme:

$$\mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix}$$

e subtrair o vetor μ de cada coluna de Y:

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

Então para o usuário j e filme i predizer

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