Collaborative filtering

Systemy Rekomendacyjne 2021/2022

Definicja problemu

- Zbiór użytkowników {1, 2, … n_u}
- Zbiór filmów {1, 2, … n_m}
- Użytkownicy oceniają filmy w skali (na przykład) {1, 2, 3, 4, 5}
- r(i, j) = 1 - jeśli użytkownik j ocenił film i
  - r(i, j) = 0 - w p. p.
- $y^{(i, j)}$ - ocena filmu i wystawiona przez użytkownika j
  - Wartość zdefiniowana tylko wtedy, gdy r(i, j) == 1

Definicja problemu

Film	Alice	Bob	Carol	Dave
Listy do M.	5	5	0	0
Notting Hill	5	?	?	0
Wesele	0	0	5	4
Szklana Pułapka	0	0	5	?

Definicja problemu

- Dla każdego filmu i dla każdego użytkownika chcemy wyznaczyć k-elementowy wektor parametrów

Film	Alice	Bob	Carol	Dave
Listy do M.	5	5	0	0
Notting Hill	5	?	?	0
Wesele	0	0	5	4
Szklana Pułapka	0	0	5	?

Film	x1	x2	x3
Listy do M.	0.21	0.05	0.93
Notting Hill	0.45	0.01	0.99
Wesele	0.17	0.76	0.12
Szklana Pułapka	0.98	0.97	0.13

Użytkownik	x1	x2	x3
Alice	0.17	0.92	0.03
Bob	0.33	0.87	0.12
Carol	0.68	0.01	0.79
Dave	0.12	0.21	0.87

Od content-based do collaborative filtering

- Content-based recommender brał pod uwagę wyłącznie oceny konkretnego użytkownika
- Chcemy skorzystać także z danych dostarczonych przez pozostałych użytkowników
- Nie mamy wyliczonych cech filmów - chcielibyśmy je wyznaczyć równocześnie z preferencjami użytkowników

Collaborative filtering optimization objective

→ Given $x^{(1)}, \ldots, x^{(n_m)}$, estimate $\theta^{(1)}, \ldots, \theta^{(n_u)}$:

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

Collaborative filtering optimization objective

- Given $x^{(1)}, \ldots, x^{(n_m)}$, estimate $\theta^{(1)}, \ldots, \theta^{(n_u)}$:

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

- Given $\theta^{(1)}, \ldots, \theta^{(n_u)}$, estimate $x^{(1)}, \ldots, x^{(n_m)}$:

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

Collaborative filtering optimization objective

- Given $x^{(1)}, \ldots, x^{(n_m)}$, estimate $\theta^{(1)}, \ldots, \theta^{(n_u)}$:

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

- Given $\theta^{(1)}, \ldots, \theta^{(n_u)}$, estimate $x^{(1)}, \ldots, x^{(n_m)}$:

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

Minimizing $x^{(1)}, \ldots, x^{(n_m)}$ and $\theta^{(1)}, \ldots, \theta^{(n_u)}$ simultaneously:

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

$$\min_{\substack{x^{(1)},\ldots,x^{(n_m)} \\ \theta^{(1)},\ldots,\theta^{(n_u)}}} J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$$

Collaborative filtering algorithm

1. Initialize $x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)}$ to small random values.
2. Minimize $J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$ using gradient descent (or an advanced optimization algorithm). E.g. for every $j = 1, \ldots, n_u, i = 1, \ldots, n_m$:

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

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

Dotrenowywanie modelu

- Wraz z napływem nowych danych (nowych ocen), model CF można dotrenowywać
- Można optymalizować zarówno oba gradienty na raz (dla filmów i użytkowników) jak i pojedynczo
- "Pojedynczo" najlepiej sprawdzi się w przypadku nowych użytkowników i filmów:
  - Regularnie, np. codziennie, możemy dotrenować model o tych nowych użytkowników, którzy wystawili choć kilka ocen
  - Analogicznie możemy dotrenować model o predykcje premierowych filmów, jeśli choć kilku użytkowników wystawiło oceny

Podobieństwo elementów

Jak zmierzyć podobieństwo filmów lub użytkowników?

Film	x1	x2	x3
Listy do M.	0.21	0.05	0.93
Notting Hill	0.45	0.01	0.99
Wesele	0.17	0.76	0.12
Szklana Pułapka	0.98	0.97	0.13

Użytkownik	x1	x2	x3
Alice	0.17	0.92	0.03
Bob	0.33	0.87	0.12
Carol	0.68	0.01	0.79
Dave	0.12	0.21	0.87

Miary podobieństwa

- Metryka (odległość euklidesowa, taksówkowa, …)
- Odległość cosinusowa
- Iloczyn skalarny

Metryki

- Odległość euklidesowa: $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Odległość taksówkowa: $\sum_{i=1}^{n}|x_i - y_i|$
- Metryka Minkowskiego: $\sqrt[k]{\sum_{i-1}^{n}(x_i - y_i)^k}$

Odległość cosinusowa

$$\frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| * ||\vec{y}||}$$

- Odległość euklidesowa:
  d = sqrt((0.21-0.45)^2 + (0.05-0.01)^2 + (0.93-0.99)^2) ~= 0.25
- Odległość taksówkowa:
  d = abs(0.21-0.45) + abs(0.05-0.01) + abs(0.93-0.99) ~= 0.34
- Odległość cosinusowa:
  d = (0.21*0.45 + 0.05*0.01 + 0.93*0.99) / (0.95*1.09) ~= 0.98

Film	x1	x2	x3
Listy do M.	0.21	0.05	0.93
Notting Hill	0.45	0.01	0.99
Wesele	0.17	0.76	0.12
Szklana Pułapka	0.98	0.97	0.13

Item to item CF

Jak prościej znaleźć podobne elementy?

- Możemy obliczyć podobieństwo użytkowników i filmów na podstawie macierzy wystawionych ocen
  - brakujące oceny można traktować jak 0

Film	Alice	Bob	Carol	Dave
Listy do M.	5	5	0	0
Notting Hill	5	?	?	0
Wesele	0	0	5	4
Szklana Pułapka	0	0	5	?

Item to item filtering

$$y(u_k, m_i) = \frac{\sum_{j:r(k,j)=1}(y(u_k, m_j) * d(m_i, m_j))}{\sum_{j:r(k,j)=1} d(m_i, m_j)}$$

Film	Alice	Bob	Carol	Dave
Listy do M.	5	5	0	0
Notting Hill	5	?	?	0
Wesele	0	0	5	4
Szklana Pułapka	0	0	5	?

Powiązane tematy

- Neural collaborative filtering
- Normalizacja przy użyciu średniej
- Faktoryzacja macierzy

Podsumowanie

- Jak rozszerzyć pomysł stojący za content-based recommender?
- Algorytm collaborative filtering
- Znajdowanie filmów/użytkowników podobnych do danego
- Item-item collaborative filtering