Apunte sobre FunkSVD

Denis Parra, IIC3633, PUC Chile

Marzo 2025

El método de predicción de ratings basado en descenso de gradiente sobre la matriz R (ratings o calificaciones de usuarios sobre ítems) fue introducido por Simon Funk[1] para el Netflix prize. A partir de ahí varias estrategias se derivaron que concluyeron en los métodos ganadores del Netflix prize. El artículo de Koren et al. (2009) resume estas estrategias, mencionando trabajos que se aproximan y derivan soluciones similares (Koren, 2008; Takács et al., 2007; Paterek, 2007).

Función objetivo de FunkSVD

La función objetivo para el modelo FunkSVD sesgado es:

$$\underset{q_*,p_*,b_*}{\operatorname{argmin}} \sum_{(u,i)\in K} \left(r_{ui} - (\mu + b_u + b_i + p_u \cdot q_i)\right)^2 + \lambda\left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2\right) \quad (1)$$

donde r_{ui} representa la calificación (*rating*) que el usuario u dio al item i, y $\hat{r}_{ui} = \mu + b_u + b_i + p_u \cdot q_i$ corresponde a la predicción que nuestro modelo de recomendación hace de esa calificación. Los elementos $\left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2\right)$ corresponden a la regularización (para evitar sobreajuste), y λ corresponde al factor de regularización.

En la expresión, p_u y q_i son los factores latentes de usuario e item, respectivamente, y μ, b_u, b_i corresponden a los sesgos general, del usuario y del item, respectivamente. Nuestro objetivo es entonces encontrar los parámetros que nos permiten minimizar la función objetivo.

Calculando gradientes

Vamos a calcular la derivada parcial de L respecto de los parámetros para luego obtener las reglas de actualización, donde L se refiere a una predicción en particular:

$$L = \left(r_{ui} - (\mu + b_u + b_i + p_u \cdot q_i)\right)^2 + \lambda\left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2\right) \quad (2)$$

[1] https://sifter.org/~simon/journal/20061211.html

Derivada parcial de L con respecto a p_u

$$\frac{\partial L}{\partial p_u} = -2\left(r_{ui} - \mu - b_u - b_i - p_u \cdot q_i\right) q_i + 2\lambda p_u \tag{3}$$

Derivada parcial de L con respecto a q_i

$$\frac{\partial L}{\partial q_i} = -2\left(r_{ui} - \mu - b_u - b_i - p_u \cdot q_i\right) p_u + 2\lambda q_i \tag{4}$$

Derivada parcial de L con respecto a b_u

$$\frac{\partial L}{\partial b_u} = -2\left(r_{ui} - \mu - b_u - b_i - p_u \cdot q_i\right) + 2\lambda b_u \tag{5}$$

Derivada parcial de L con respecto a b_i

$$\frac{\partial L}{\partial b_i} = -2\left(r_{ui} - \mu - b_u - b_i - p_u \cdot q_i\right) + 2\lambda b_i \tag{6}$$

Reglas de actualización

Queremos actualizar los parámetros θ moviéndonos en la ***dirección contraria*** al gradiente $\Delta\theta$, de forma proporcional a un factor γ, la tasa de aprendizaje (en inglés, *learning rate*). En términos generales:

$$\theta' \leftarrow \theta - \gamma \Delta\theta \tag{7}$$

Regla de actualización para p_u

Podemos simplificar la ecuación (3) reemplazando $e_{ui} = r_{ui} - \mu - b_u - b_i - p_u \cdot q_i$, donde e_{ui} le llamamos *error de predicción*, y multiplicando γ por una constante $1/2$, quedando la regla de actualización como:

$$p'_u \leftarrow p_u - \frac{1}{2}\gamma\left(-2e_{ui}q_i + 2\lambda p_u\right) \tag{8}$$

simplificando nuevamente, nos queda:

$$p'_u \leftarrow p_u + \gamma\left(e_{ui}q_i - \lambda p_u\right) \tag{9}$$

Si lees el artículo de Koren et al. (2009), encontrarás exactamente la regla de actualización descrita.

Regla de actualización para q_i

$$q'_i \leftarrow q_i + \gamma\left(e_{ui}p_u - \lambda q_i\right) \tag{10}$$

Regla de actualización para b_u

$$b'_u \leftarrow b_u + \gamma\left(e_{ui} - \lambda b_u\right) \tag{11}$$

Regla de actualización para b_i

$$b_i' \leftarrow b_i + \gamma \left(e_{ui} - \lambda b_i\right) \tag{12}$$

References

Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Major components of the gravity recommendation system. *Acm Sigkdd Explorations Newsletter*, 9(2):80–83, 2007.