



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
Departamento de Ciencias de la Computación
IIC3633 - Sistemas Recomendadores
Segundo Semestre 2018

Review 3

Matrix Factorization Techniques for Recommender Systems

Genaro Laymuns

Este paper comienza hablando acerca de las distintas estrategias (a nivel global) para diseñar sistemas recomendadores. El primer punto importante que destacan es el de la diferencia entre los dos tipos de técnicas para el diseño de Sistemas recomendadores, la cual vale la pena mencionar. Estas técnicas consisten en *content-based filtering* y *collaborative filtering*. En técnicas *content-based filtering*, el sistema solamente se fija en los atributos del producto, estableciendo un ranking fijo que no depende del usuario. Por otro lado, *collaborative filtering* incorpora a la técnica anteriormente utilizada el comportamiento del usuario. Dada la alta cantidad de variables que considera el segundo modelo con respecto al primero, en general se pueden obtener muchos mejores resultados que simplemente utilizando información acerca de los productos. Sin embargo, no es una técnica que haya que dejar de lado, pues el *collaborative filtering* presenta una gran desventaja, y es que no es capaz de agregar nuevos usuarios o productos al sistema. Esto es conocido como *cold start problem*.

Suponiendo entonces que cada usuario y cada producto tiene un vector indicando sus características, es posible ver la correlación entre un usuario y un producto viendo qué tan parecidos son dichos vectores. Dado que ambas partes son representadas de forma vectorial, la correlación entre ellos puede ser obtenida mediante el producto interno entre estos vectores. Así, el principal supuesto de este modelo es que el rating que cada usuario le dará a cada producto se encuentra dentro de una gran matriz R cuyas coordenada r_{ij} corresponden al producto punto entre el vector de atributos del ítem y el usuario. La recomendación que se entrega finalmente corresponde a la proyección del usuario sobre las direcciones principales de dicha matriz, las cuales representan distintas tendencias entre los usuarios.

Dado este modelo, el paper propone distintas maneras de mejorar esta recomendación, y me gustaría proponer una mejora a una de ellas en particular. Al estimar los vectores de atributos, se resuelve un problema de optimización, donde se minimiza

$$\sum (r_{ij} - q_i^T p_j)^2 + \|q_i\|_2^2 + \|p_j\|_2^2.$$

El hecho de que se minimice la norma-2 al cuadrado es que dichos vectores son por lo general vectores sparse. Sin embargo, para recuperar el sparsity de la matriz, conviene utilizar la norma-1, la cual corresponde a la suma del valor absoluto de sus coordenadas, $\|x\|_1 = \sum |x_i|$. Se intentaría minimizar entonces

$$\sum (r_{ij} - q_i^T p_j)^2 + \|q_i\|_1^2 + \|p_j\|_1^2.$$

Este tipo de problemas son conocidos como problemas de regularización ℓ_1 . En relación al modelo de regularización ℓ_2 (el utilizado en el paper), este presenta algunas diferencias en cuanto a los resultados que se puedan obtener, las cuales mencionamos a continuación.

- ℓ_2 (norma usual) penaliza *ponderacion*², mientras que ℓ_1 penaliza $|ponderacion|$.
- Las derivadas parciales tanto en ℓ_1 y ℓ_2 son fáciles de obtener. Sin embargo, la norma en ℓ_1 no siempre corresponde a una función derivable, pues presenta singularidades en el cero.
- Ambas funciones de error son convexas. Sin embargo, el regularizador ℓ_2 corresponde a una función fuertemente convexa, la cual puede presentar ventajas en términos de poder computacional necesario. El regularizador ℓ_1 puede ser minimizado de manera eficiente con métodos de primer orden

La idea de cambiar los distintos tipos de regularizadores es recuperar la naturaleza de un vector que se quiere estimar. Calcular el error con la norma en ℓ_2 entrega por lo general vectores de baja longitud, pero con pequeños valores en cada una de sus coordenadas, mientras que calcular el error con la norma en ℓ_1 entrega en general vectores sparse (o bien con coordenadas con un peso considerablemente mayor). Esto podría resultar en mejoras de la predicción del rating de parte de los usuarios, pues al entregar vectores que tiendan a ser sparse, el perfil de cada usuario tendrá finalmente gustos mejor definidos, y por lo tanto podría resultar en obtener tendencias más claras entre los distintos tipos de usuarios del sistema.