# Sistemas de Recomendación

## Filtrado colaborativo

# Filtrado colaborativo

- Basado en usuarios
- Basado en ítems
- Híbrido

# Basado en usuarios

- Identificar los usuarios más parecidos a un usuario en particular calculando una medida de similitud entre pares de usuarios
  - Combinar las preferencias en una predicción
  - Supuesto: Las coincidencias en preferencias pasadas predicen las coincidencias futuras
  - Ver (Herlocker, 1999)

# Problemas computacionales

- Pero para *m* usuarios y *n* ítems…
  - Correlación entre dos usuarios: *O(n)*
  - Todas las correlaciones para un usuario: *O(mn)*
  - Todas las correlaciones de a pares: $O(m^2n)$
  - Recomendaciones: *O(mn)* por lo menos

# Algoritmo no personalizado

- El rating del usuario *u* para el ítem *i* es…

$$r_{u,i} = \frac{1}{N} \sum_{u' \in U} r_{u',i}$$

- Donde:
  - *U* es el conjunto de los *N* usuarios más parecidos al usuario *u* y que opinaron sobre el ítem *i*

# Algoritmo personalizado (1)

- Mejora: no todos los usuarios son igual de similares

$$r_{u,i} = \frac{\sum_{u' \in U} \text{simil}(u, u')\, r_{u',i}}{\sum_{u' \in U} \left|\text{simil}(u, u')\right|}$$

- Donde:
  - *simil* es una función de similitud entre usuarios
  - Denominador: factor de normalización

# Algoritmo personalizado (2)

- Mejora: no todos los usuarios usan la misma escala

$$r_{u,i} = \bar{r_u} + \frac{\sum_{u' \in U} \text{simil}(u, u')(r_{u',i} - \bar{r_{u'}})}{\sum_{u' \in U} \left|\text{simil}(u, u')\right|}$$

- Donde:
  - $\bar{r_u}$ es el promedio de los ratings del usuario *u*
- Cuidado: hay que aplicar máximo y mínimo

# Medidas de similitud

- Coeficiente de correlación de Pearson ($R^2$):

$$\text{simil}(x,y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2 \sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

- Similitud coseno:

$$\text{simil}(x,y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}}$$

# Problemas

- Dos usuarios tienen pocos ítems en común
  - Pongo un mínimo
- Cuántos usuarios tomo como vecindario
  - Verifico para distintos *k* (usualmente entre 25 y 100)
  - Límite de distancia mínima
- Para hacer top *N* calcula todos los ratings y elige los N mejores

# Recomendadores híbridos

- Combinar 2 o más algoritmos
- Es una técnica general, no es específica a ningún algoritmo
- Es similar a *stacking* o *boosting* en aprendizaje automático

# Combinar algoritmos

- Regresión lineal
  - $p(u, i) = a_1\, p_1(u, i) + a_2\, p_2(u, i) + \ldots + b$
  - Donde $a_j$ es el peso del algoritmo $j$ y $p_j(u, i)$ es la predicción del algoritmo $j$ para el usuario $u$ y el ítem $i$
- Jerárquico
  - $p(u, i) = f_1(u, i)\, p_1(u, i) + f_2(u, i)\, p_2(u, i) + \ldots + b$
    - Donde $f_j(u, i)$ es una función del usuario u y el ítem i

# Cambio de algoritmos

- Cambiar de algoritmo según las características de los usuarios y los ítems
  - Ítems nuevos
  - Ítems muy populares
  - Usuarios con pocos ratings
  - Usuarios con ratings muy similares (*grey sheep*)

# Implementación en Python

- Surprise (http://surpriselib.com)
- Algoritmos no personalizados
- Filtrado colaborativo basado en ítems y en usuarios
- Factorización de matrices

# Implementación en R

- recommenderlab (http://lyle.smu.edu/IDA/recommenderlab/)
- rrecsys (https://cran.r-project.org/web/packages/rrecsys/index.html)
- Algoritmos no personalizados
- Filtrado colaborativo basado en usuarios y en ítems
- Factorización de matrices (ya lo veremos)

# Próxima clase (práctica)

- Aprender a usar surprise
- Entrega de predicción usando filtrado colaborativo

# ¿Preguntas?