4. Sistemas de recomendación

- Situaciones con sobrecarga de opciones
  - 1994 → 0.5 millones de productos diferentes a la venta en EE UU
  - 2010 → 24 millones sólo en Amazon

  Sheena Iyengar, The Art of Choosing, 2010
- Recomendación = IR personalizada sin consulta explícita
- Primeras iniciativas publicadas en el 92 (Tapestry en Xerox Parc)
- Precedentes: modelos de usuario basados en estereotipos (finales 70)
- Confluyen otras áreas: Machine Learning (ICML, ECML, IJML, etc.), Data Mining (KDD, etc.), Inteligencia Artificial (IJCAI, AAAI), HCI (ACM IUI)
- Aplicaciones comerciales en Amazon, Last.fm, Pandora, Netflix, Film affinity, etc.; proveedores de soluciones como Strands, ChoiceStream, etc.

1

Bibliografía

Recommender Systems Handbook. F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.). Springer Verlag, ISBN 978-0-387-85819-7, 2011.

G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), June 2005, pp. 734-749.

Recuperación de Información. J. Huete, F. Cacheda, J. M. Fernández (Eds.). RA-MA, en edición, cap. 15.

2

Planteamiento del problema

Items $\mathcal{I}$

Users $\mathcal{U}$

		1	3		2		4			3	
1	2	5		4		1		2	4		5
4		?	3	5			5			2	
	2			5	4	4		5			4
	3	4		5			4	3	5		4
3		2		1	5		3			5	
3			2			3		5		1	

User-item preference data in the recommendation task

3

Planteamiento del problema

La formulación más simple…

- Input
  - Conjunto $\mathcal{U}$ de usuarios, p.e. usuarios de un foro, clientes de una tienda, etc.
  - Conjunto $\mathcal{I}$ de ítems, p.e. películas o cualquier producto a recomendar
  - Conjunto totalmente ordenado $\mathcal{R}$ de valores, p.e. $\mathcal{R}$ = { 1, 2, 3, 4, 5 }
  - Relación funcional $r : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$
  - $r(u,x)$ representa típicamente la valoración del usuario u por el ítem x en la escala $\mathcal{R}$

    Se puede ver como una matriz de ratings

    La mayoría de los valores (p.e. 95%, o generalmente más) son desconocidos
- Objetivo
  - Predecir la valoración $r(u,x)$ de un ítem x por un usuario u que no lo ha puntuado

    Los valores desconocidos $r(u,x) = \varnothing$ dan pie a considerar la recomendación de x a u
  - O en general: generar una lista ordenada de ítems que pueden interesar al usuario

    La predicción de puntuaciones es un caso particular de esta formulación

    Este objetivo se plantea habitualmente como generar un "top k" de recomendaciones

4

Planteamiento del problema

Variantes

- Valoración implícita
  - No se necesita pedir información al usuario
  - $r : \mathcal{U} \times \mathcal{I} \rightarrow$ {0, 1} binario, p.e. "u compra x"

    Se puede tratar como un caso particular con $\mathcal{R}$ = {0, 1}
  - $r : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{N}$ frecuencia de acceso de u a x, p.e. escuchar música

    Se puede tratar binarizando a 1 las frecuencias > 0

    O por una función de conversión frecuencia → puntuación (p.e. percentiles)

    O por métodos propios (p.e. basados en tf-idf)
  - $r : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{P}(\mathcal{T})$, usuarios u etiquetan ítems x, donde $\mathcal{T}$ es un conjunto de tags

    Social tagging, Web 2.0, etc.

    Se puede tratar como "1 tag 1 voto", pero también técnicas más elaboradas y complejas sobre los grafos de tags, items, usuarios…
- Marcas temporales
  - Datos de frecuencia: $r(u,x)$ es un conjunto de marcas temporales
  - Datos de puntuación: $r(u,x)$ es un par puntuación / marca temporal

5

Tipos de estrategias de recomendación

- Recomendación **basada en contenido** (CB)
  - Equivalente a "filtrado de información" en IR
  - Se consideran rasgos de los ítems: palabras (caso texto), señal a/v, descriptores (metadatos), etc.
  - Se comparan con información del usuario recogida en perfil de preferencias
  - El perfil de usuario es de largo plazo, se puede adquirir mediante árboles de decisión, redes neuronales, representación vectorial, etc.
- Filtrado **colaborativo** (CF)
  - Los ítems son opacos
  - Se usa la experiencia de otros usuarios con rasgos similares (gustos, patrones de comportamiento, rasgos demográficos, etc.) para recomendar ítems
- Recomendación **híbrida**: combinación de distintas estrategias de recomendación (CB + CF)
  - Combinar la salida de CF y CB
  - Introducir elementos CB en CF o viceversa
  - Modelos unificados

6

Pablo Castells, Fernando Díez, Estrella Pulido
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Filtrado colaborativo

- Hipótesis: coincidencia en el pasado implica coincidencia en el futuro
- Variantes
  - Basados en memoria (heurísticos)
  - Basados en modelo
- Ventajas
  - Puede aplicarse a cualquier tipo de ítem o producto: documentos, música, películas, libros, etc. –no necesita descripción de los ítems
  - Permite introducir novedad respecto a la experiencia previa del usuario (cross-genre)
  - Similar a popularidad global, pero personalizada al usuario (por afinidad con los "puntuadores", siendo éstos otros usuarios)

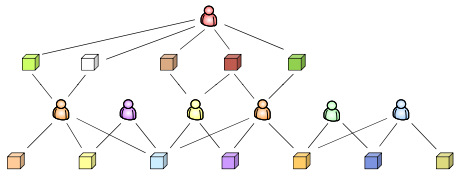
7

Filtrado colaborativo – kNN

- CF por vecinos más próximos (kNN, *k* nearest-neighbors)
- Variantes
  - Basado en usuario
  - Basado en ítem
- Método muy popular originalmente (utilizado p.e. en Amazon)
- Intuitivo y fácil de entender

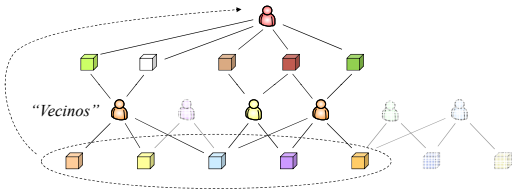
8

kNN basado en usuario



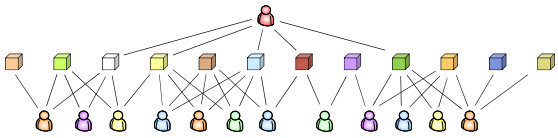
9

kNN basado en usuario



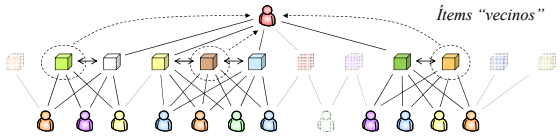
10

kNN basado en ítem



11

kNN basado en ítem



12

Pablo Castells, Fernando Díez, Estrella Pulido
Escuela Politécnica Superior
Universidad Autónoma de Madrid

CF kNN – Predicción de ratings

Basado en usuario: recomendar a *u* los ítems bien puntuados por usuarios *v* "similares" a *u*

$$(a)\quad \hat{r}(u,x) = c \sum_{v \in \eta_k(u), x \in S(v)} sim(u,v)\, r(v,x)$$

$$c = \frac{1}{\sum_{v \in \eta_k(u)} |sim(u,v)|} \qquad \eta_k(u) = top\left(k, sim(u,\cdot), \mathcal{U}\right)$$

$$S(v) = \{x \in \mathcal{I} \mid r(v,x) \neq \varnothing\}$$

$$(b)\quad \hat{r}(u,x) = \bar{r}(u) + c \sum_{v \in \eta_k(u), x \in S(v)} sim(u,v)\left(r(v,x) - \bar{r}(v)\right)$$

$$\bar{r}(u) = \frac{1}{|S(u)|} \sum_{x \in S(u)} r(u,x)$$

13

CF kNN – Comparación de usuarios

(a) Coseno

$$sim(u,v) = \cos\left(r(u), r(v)\right) = \frac{\sum_{x \in \mathcal{I}} r(u,x) r(v,x)}{\sqrt{\sum_{x \in \mathcal{I}} r(u,x)^2 \sum_{x \in \mathcal{I}} r(v,x)^2}}$$

(b) Pearson correlation

$$sim(u,v) = \frac{\sum_{x \in \mathcal{I}} \left(r(u,x) - \bar{r}(u)\right)\left(r(v,x) - \bar{r}(v)\right)}{\sqrt{\sum_{x \in \mathcal{I}} \left(r(u,x) - \bar{r}(u)\right)^2 \sum_{x \in \mathcal{I}} \left(r(v,x) - \bar{r}(v)\right)^2}}$$

(En la práctica se suma en $x \in S(u) \cap S(v)$ en lugar de $x \in \mathcal{I}$)

Se puede calcular de antemano toda la matriz de similitudes

14

CF kNN – Predicción de ratings

Basado en ítem: recomendar a *u* los ítems "similares" a los mejor puntuados por *u*

$$\hat{r}(u,x) = c \sum_{y \in S(u)} sim(x,y)\, r(u,y)$$

$$c = \frac{1}{\sum_{y \in S(u)} |sim(x,y)|}$$

Ver p.e. *G. Linden et al, Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7 (1), 2003, pp. 76-80*

15

CF kNN – Comparación de ítems

(a) Coseno

$$sim(x,y) = \cos\left(r(x), r(y)\right) = \frac{\sum_{u \in \mathcal{U}} r(u,x) r(u,y)}{\sqrt{\sum_{u \in \mathcal{U}} r(u,x)^2 \sum_{u \in \mathcal{U}} r(u,y)^2}}$$

(b) Pearson correlation

$$sim(x,y) = \frac{\sum_{u \in \mathcal{U}} \left(r(u,x) - \bar{r}(x)\right)\left(r(u,y) - \bar{r}(y)\right)}{\sqrt{\sum_{u \in \mathcal{U}} \left(r(u,x) - \bar{r}(x)\right)^2 \sum_{u \in \mathcal{U}} \left(r(u,y) - \bar{r}(y)\right)^2}}$$

(c) Coseno ajustado

$$sim(x,y) = \frac{\sum_{u \in \mathcal{U}} \left(r(u,x) - \bar{r}(u)\right)\left(r(u,y) - \bar{r}(u)\right)}{\sqrt{\sum_{u \in \mathcal{U}} \left(r(u,x) - \bar{r}(u)\right)^2 \sum_{u \in \mathcal{U}} \left(r(u,y) - \bar{r}(u)\right)^2}}$$

(En la práctica se suma en $u \in S(x) \cap S(y)$ en lugar de $u \in \mathcal{U}$)
Se puede calcular de antemano toda la matriz de similitudes

16

Variantes y mejoras

- Control de mínimo solapamiento
  - ¡Ojo! Mínimo solapamiento o mínimo vecindario pueden dar lugar a anomalías
  - Exigir mínimo solapamiento para la similitud, y mínimo nº de vecinos para la predicción
  - Multiplicar *sim* (*u*,*v*) por $|S(u) \cap S(v)| / m$ (análogamente para ítems), donde p.e. $m = 50$, para penalizar similitudes con poca base de comparación
- Normalización de los ratings: centrar en la media (Pearson), igualar el rango o la varianza, amplificar los ratings más altos, etc.
- Al calcular *sim* (*u*,*v*), asignar valores por defecto a los ratings que faltan para $x \in S(u) \cup S(v) - S(u) \cap S(v)$
- Multiplicar *sim* (*u*,*v*) por *idf* (*x*) para reducir la influencia de ítems que tienen ratings de muchos usuarios (i.e. son poco discriminatorios)
- Formar vecindarios por otras técnicas de clustering alternativas a kNN
- Etc.

17

Detalles y opciones de implementación

- ¿Tomar vecinos o tomar todo el conjunto de usuarios (análogamente, ítems)?
- ¿Tomar *k* vecinos fijos en $\mathcal{U}$ de forma que para predecir $\hat{r}(u,x)$ actúen sólo los vecinos *v* con $x \in S(v)$, o tomar los *k* vecinos en $S(x)$?
- ¿Qué se hace para predecir $\hat{r}(u,x)$ si *u* no tiene vecinos? ¿Cómo debe tenerse en cuenta al evaluar el acierto global? (MAE, RMSE, etc. – ver más adelante)
- ¿Pueden salir predicciones fuera de rango? ¿Cuándo, y qué hacer en tal caso?
- En Pearson y coseno centrado, ¿tomar $\bar{r}(u)$ sobre todo el perfil de *u* o sólo sobre $S(u) \cap S(v)$?
- La eficacia del recomendador puede depender sensiblemente de estas decisiones

18

CF basado en factorización de matrices

- Descomposición con k factores: $A \approx A_k = U_k S_k V_k^t$
- Vectores-fila de usuario: $U_k S_k^{\frac{1}{2}} \longrightarrow p$
- Vectores-columna de ítem: $S_k^{\frac{1}{2}} V_k^t \longrightarrow q$
- Predicción de puntuaciones: $\hat{r}(u,i) = q_i^t p_u = \sum_{j=1}^{k} q_{i,j} p_{u,j}$
- Cómputo de p, q: $\min_{p,q} \left[\sum_{u,i} \left(r(u,i) - q_i^t p_u \right)^2 + \lambda \left(\|q_i\|^2 + \|p_u\|^2 \right) \right]$

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, 42(8), August 2009, pp. 30-37.

19

CF basado en modelos

- Por ejemplo, suponiendo que los ratings son valores numéricos $r(u,x) \in \mathcal{R}$, la predicción de ratings se define mediante:

$$\hat{r}(u,x) = \sum_{r \in \mathcal{R}} r \cdot \mathrm{P}(r|u,x)$$

- Diversas técnicas de estimación de $\mathrm{P}(r|u,x)$: redes y modelos bayesianos, métodos de clustering, machine learning (p.e. redes neuronales), regresión, máxima entropía, cadenas de Markov, pLSA, y un largo etc.

20

CF basado en pLSA

T. Hofmann. Latent semantic models for collaborative filtering. ACM TOIS 22(1), 2004, pp. 89-115.

- Versión para datos "implícitos" → $\mathrm{P}(x|u) = \sum_z \mathrm{P}(x|z)\mathrm{P}(z|u)$
  - La variable z representa factores latentes no observados
  - Se escoge el nº de factores
  - Se calculan $\mathrm{P}(x|z)$ y $\mathrm{P}(z|u)$ como parámetros por máxima verosimilitud con Expectation Maximization sobre las observaciones de la co-ocurrencia entre usuarios e items
- Hoffman propone otra variante para ratings explícitos
- pLSA se propuso originalmente como alternativa a LSA para IR en texto

T. Hofmann, Probabilistic Latent Semantic Indexing. SIGIR 1999, pp. 50-57.

- Latent Dirichlet Allocation (LDA), variante que parece mejorar los resultados –equivalente a pLSA con priori Dirichlet

David M. Blei, Andrew Y. Ng, Michael I. Jordan, Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 2003, pp. 993-1022.

21

Recomendación basada en contenido

- La recomendación se hace en base a una comparación $sim(u,x)$
- Ejemplo:
  - $\mathcal{I}$ = documentos de texto → modelo vectorial
  - x = vector de keywords (p.e. utilizando tf-idf)
  - u = promedio de los vectores de los ítems puntuados por u, ponderado por sus ratings

$$\hat{r}(u,x) = sim(u,x) = \cos(u,x) = \frac{\sum_{k \in \mathcal{K}} u_k x_k}{|u||x|}$$

$$u_k = c \sum_{y \in \mathcal{I}} r(u,y) \cdot y_k \qquad c = \frac{1}{\sum_{y \in \mathcal{I}} r(u,y)}$$

- Alternativas con modelos probabilísticos, etc.

22

Evaluación de sistemas de recomendación

- Online
  - Usuarios reclutados al efecto
  - Costosa
  - No reproducible – no permite comparar con recomendadores que no se incluyeran en el experimento
  - Máxima flexibilidad: permite evaluar en principio cualquier tipo de aspecto o técnica de recomendación
- Offline
  - Colecciones públicas
  - Económica y reproducible
  - Restrictiva en cuanto a los aspectos que permite evaluar – no se pueden evaluar aspectos que necesiten datos no recogidos en la colección
- Cabe evaluar diferentes dimensiones e indicadores de calidad de un recomendador
  - Acierto, novedad, ventas…

23

Evaluación de sistemas de recomendación

- Es común (según el tipo de recomendador) dividir el conjunto de ratings S en dos subconjuntos ("leave n out"): entrenamiento $S_{training}$ y prueba (ground truth) S_{test}
  - Aleatoriamente (p.e. 5 ó 10-fold) un porcentaje de ratings para training/test
  - Corte temporal (p.e. Netflix)
  - "Leave one out"

$0 \le \mathrm{MAE} \le \mathrm{RMSE} \le \max(\mathcal{R})$

- Mean Average Error $\mathrm{MAE} = \frac{1}{|S_{test}|} \sum_{(u,x) \in S_{test}} \left| \hat{r}(u,x) - r_{test}(u,x) \right|$
  - $r_{test}(u,x)$ = valor de rating en S_{test}, $\hat{r}(u,x)$ se computa usando $S_{training}$
- Root Mean Squared Error $\mathrm{RMSE} = \sqrt{\frac{1}{|S_{test}|} \sum_{(u,x) \in S_{test}} \left(\hat{r}(u,x) - r_{test}(u,x) \right)^2}$
- Otras medidas de correlación global entre predicciones y ratings
- Otras métricas de IR: nDCG, precisión $\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$, recall $\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$, MAP, F harmónica, etc.
- ROC: contrapartida TP vs. FP, gráfica $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$ vs. $\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
- Otras más subjetivas: cobertura, novedad, diversidad, confianza…
- Eficacia comercial: incremento en clickthrough, conversion rate, retorno de clientes, incremento de ventas, de ingresos…

24

Evaluación de sistemas de recomendación Un problema abierto en el área

- Las métricas de error no necesariamente determinan la satisfacción del usuario (i.e. la efectividad práctica)
  - El error o acierto en las puntuaciones bajas es irrelevante
  - Los recomendadores que definen un ránking sin predecir ratings no se pueden evaluar así
  - La efectividad de una recomendación es una cuestión de ránking
- Las métricas de evaluación de ránking no son fáciles de aplicar
  - Las suposiciones de la metodología Cranfield no se cumplen del todo en los experimentos de recomendación
  - Gran divergencia entre autores, difícil comparación entre experimentos
- El acierto no es el único factor de utilidad de una recomendación y la efectividad del sistema
  - Novedad, diversidad
  - Cobertura
  - Confianza
  - Etc.
- Efectividad para el usuario vs. para el proveedor (vendedor) de ítems

25

Limitaciones de los recomendadores

- Varias relacionadas con data sparsity: necesidad de masa crítica y solapamiento de datos
  - CB:
    - Sobre-especialización: encasillamiento y falta de novedad/diversidad
    - Necesidad de disponer de descriptores del contenido
    - Portfolio effect: redundancia, duplicados que no se detectan
  - CF:
    - Nuevo usuario
    - Nuevo ítem
    - Early rater
    - Oveja negra
    - Ruido (inconsistencias) en el input del usuario ("rate it again")
    - Coste computacional ~ O($|\mathcal{U}|$ $|\mathcal{I}|$) similitud y ~ O($|\mathcal{U}|$ + $|\mathcal{I}|$) predicción
- Soluciones
  - Métodos híbridos CB + CF, suelen funcionar mejor
  - Uso de información demográfica de los usuarios, conocimiento del dominio, etc.
  - Captar ratings "implícitos"
  - Técnicas de relleno de ratings desconocidos (promedio, etc.)
  - Modelos unificadores que comportan probabilidades a priori, suavizado, etc.
  - Otras técnicas: SVD, etc.

26

¿Qué método funciona mejor?

- Depende ampliamente del problema, experimentos publicados… pros y contras
- Generalmente, los métodos basados en modelo alcanzan menor nivel de acierto que los heurísticos; coste offline alto (construcción del modelo) pero más rápidos en tiempo de recomendación
- Las técnicas de vecindario basadas en clustering pueden tener ventajas en coste computacional (al llevarse offline), pero parecen tener menor tasa de acierto
- Pearson (especialmente con el ajuste $|S(u) \cap S(v)| / m$) tiende a ser más estable y fiable que el coseno
- Las técnicas kNN basadas en ítem tienen ventajas en coste y acierto cuando $|\mathcal{I}| << |\mathcal{U}|$
  - Matriz de similitud más pequeña para mantenerla en RAM
  - Mayor tasa de solapamiento entre ítems → similitudes más significativas
  - Menor necesidad de actualización: los ítems tienen perfil más estable que los usuarios
  - P.e. Amazon describe IB kNN con 29M usuarios y varios millones de ítems (Linden 2003)
- Pros y contras de recomendación basada en contenido vs. filtrado colaborativo
  - En general las soluciones híbridas son una buena opción
- Las técnicas basadas en factorización están obteniendo los mejores niveles de acierto en experimentos y competiciones recientes (p.e Netflix prize)

27

Sistemas de recomendación en el mercado

- Comercialización de la tecnología
  - Amazon, Barnes & Noble
  - Last.fm, Pandora, iTunes
  - Netflix (2006-09 prize, 20.000 equipos en competición)
  - Más películas: Film affinity
  - TV: TiVo
  - Gmail ads, Google news recommendation
  - Mercados online: eBay
  - Online retail: Walmart, etc.
  - Strands, ChoiceStream
  - Y cada día más…
- Impacto
  - 60% de las películas alquiladas en Netflix procede de recomendaciones (NYT, 2008)
  - 35% de las ventas de Amazon procede de recomendaciones (datos 2006)
  - 38% más clicks en Google news con la recomendación de noticias (Das et al., WWW 2007)

28

Sistemas y recursos experimentales

- MovieLens (películas), GoupLens (noticias), Jester (chistes), etc.
- Librerías: Mahout (Taste), CoFi, etc.
- Datasets
  - Netflix: 100M ratings, 480K users, 17K movies
  - KDD Cup: 300M ratings, 1M users, 600K items de diferentes tipos en una taxonomía: canciones, discos, artistas, géneros
  - MovieLens: 10M ratings, 71K users, 10K movies (IMDb), 100K tags
  - Epinions: 13M votos de ±132.000 usuarios a 1.560.144 *reviews*, junto con relaciones de confianza/desconfianza entre usuarios
  - EachMovie: 2M ratings, 73K users, 1.6K movies
  - Jester Joke: 4.1M ratings, 73K users,100 jokes
  - …

29

Líneas abiertas de investigación

- Extensiones sobre la base de formulación $\hat{r}(u,x) = f(\mathcal{U},\mathcal{I},\mathcal{S})$, donde $\mathcal{U}$ = usuarios, $\mathcal{I}$ = ítems, $\mathcal{S}$ es una relación en $\mathcal{U} \times \mathcal{I} \times \mathcal{R}$, siendo $\mathcal{R}$ el rango de valores de rating y f un predictor
- Recomendación contextual (Adomavicius et al)
  - Espacios multidimensionales: $r : \mathcal{U} \times \mathcal{I} \times \mathcal{C}_1 \times \mathcal{C}_2 \times \cdots \rightarrow rating$
- Estrategias para el sparsity problem
  - Factorización (p.e.) SVD
  - Algoritmos de propagación (CSA)
  - → similitudes indirectas (transitivas)
  - Bi-clustering → partial matching (similitudes especializadas)
- Toma en cuenta del eje temporal
- Ratings multicriterio
- Robustez a ratings ruidosos (spam, manipulación, etc.)
- Novedad y diversidad

30