Graph-based recommendation

Cristopher Arenas

PUC, Campus San Joaquín

Jueves, 5 de noviembre de 2015

Graph-based recommendation

Marco Gori, Augusto Pucci
ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines.
2007

- Sistemas recomendadores (RS) representan un valor agregado para consumidores.
- RS construye un perfil basado en información implícita o explícita de un usuario que interactúa con el sistema.
- El perfil se genera en base al comportamiento del usuario.
- Se propone un algoritmo de scoring, llamado ItemRank, basado en random-walk para recomenrar un Top-N a usuarios potencialmente interesados por un producto.

- Usuarios u_i, con $i = 1, \ldots, U_n$
- Productos m_j, con $j = 1, \ldots, M_n$

- El objetivo es computar un score $\hat{r}_{i,j}$ para el par u_i m_j que mide interés de un usuario u_i por el producto m_j.

Considerando solo usuarios que valoran 20 o más películas

- 943 usuarios u_i
- 1682 películas m_j
- Tupla $t_{i,j} = (u_i, m_j, r_{i,j})$, con $u_i \in \mathcal{U}$, $m_j \in \mathcal{M}$ y $r_{i,j}$ escala 1-5.
- $\mathcal{L}$: testing (80%), $\mathcal{T}$: training (20%)
- $\mathcal{L}_{u_i} = \{t_{k,j} \in \mathcal{L} : k = i\}$
- $\mathcal{T}_{u_i} = \{t_{k,j} \in \mathcal{T} : k = i\}$

Se define $\mathcal{U}_{i,j} \subseteq \mathcal{U}$, conjunto de usuarios que vieron m_i y m_j:

$$\mathcal{U}_{i,j} = \begin{cases} \{u_k : (t_{k,i} \in L_{u_k}) \wedge (t_{k,j} \in L_{u_k})\} & i \neq j \\ 0 & i = j \end{cases}$$

Se define una matriz $\tilde{\mathcal{C}}$, con coeficientes $\tilde{\mathcal{C}}_{i,j} = |\mathcal{U}_{i,j}|$, como el número de usuarios que vió cada par de películas.
Al normalizar la matriz por la suma de la columna j, se obtiene una matriz de correlación $\mathcal{C}$:

$$\mathcal{C}_{i,j} = \frac{\tilde{\mathcal{C}}_{i,j}}{\omega_j}$$

La matriz de correlación puede ser considerada como una matriz de conectividad con pesos para el grafo de correlación $\mathcal{G}_{\mathcal{C}}$.

- El grafo $\mathcal{G}_{\mathcal{C}}$ es usado en un algoritmo de difusión.
- Debe cumplir con 2 propiedades:
  1. **Propagación**: Si m_k está relacionada con una o más buenas películas con respecto a un usuario u_i, entonces esta será una buena sugerencia para dicho usuario.
  2. **Atenuación**: Buenas películas tienden a repartir su buena influencia si se encuentran relacionadas con muchas películas.

ItemRank es un algoritmo basado en Page Rank, y puede definirse de manera iterativa:

$$IR_{u_i}(0) = \frac{1}{|\mathcal{M}|} \cdot 1_{|\mathcal{M}|}$$
$$IR_{u_i}(t+1) = \alpha \cdot \mathcal{C} \cdot IR_{u_i}(t) + (1-\alpha) d_{u_i}$$

- α es un factor de decaimiento.
- $1_{|\mathcal{M}|}$ es un $|\mathcal{M}|$-vector de unos.
- d es el vector normalizado de $\tilde{d}_{u_i}$. Se define $\tilde{d}^j_{u_i}$ con respecto al componente j-ésimo como:

$$\tilde{d}^j_{u_i} = \left\{ \begin{array}{ll} 0 & t_{i,j} \notin \mathcal{L}_{u_i} \\ r_{i,j} & t_{i,j} \in \mathcal{L}_{u_i} \wedge t_{i,j} = (u_i, m_j, r_{i,j}) \end{array} \right\}$$

- Uso del dataset de MovieLens con 5-fold cross validation.
- La medida de desempeño usada es *degree of agreement* (DOA). Mide que tan bueno es ItemRank para un usuario dado.

$$DOA_{u_i} = \frac{\sum_{(j \in \mathcal{T}_{u_i}, k \in \mathcal{NW}_{u_i})} check\_order_{u_i}(m_j, m_k)}{|\mathcal{T}_{u_i}| \cdot |\mathcal{NW}_{u_i}|}$$

Donde

$$\mathcal{NW}_{u_i} = \mathcal{M} \setminus (\mathcal{L}_{u_i} \cup \mathcal{T}_{u_i})$$

$$check\_order_{u_i}(m_j, m_k) = \begin{cases} 1 & IR_{u_i}^{m_j} \geq IR_{u_i}^{m_k} \\ 0 & IR_{u_i}^{m_j} < IR_{u_i}^{m_k} \end{cases}$$

- DOA_{u_i} mide para el usuario u_i el porcentaje de pares de películas que están rankeados en el orden correcto con respecto al número total de pares.
- Se establecen dos métricas globales considerando la métrica individual: Macro-averaged DOA y micro-average DOA (Macro DOA y Micro DOA).
- Macro DOA mide el promedio de los DOA por cada usuario:

$$\text{Macro DOA} = \frac{\sum_{u_i \in \mathcal{U}} DOA_{u_i}}{|\mathcal{U}|}$$

- Micro DOA mide el promedio ponderado entre los valores DOA individuales:

$$\text{Micro DOA} = \frac{\sum_{u_i \in \mathcal{U}} \left(\sum_{(j \in \mathcal{T}_{u_i}, k \in \mathcal{NW}_{u_i})} check\_order_{u_i}(m_j, m_k) \right)}{\sum_{u_i \in \mathcal{U}} \left(|\mathcal{T}_{u_i}| \cdot |\mathcal{NW}_{u_i}| \right)}$$

	ItemRank		ItemRank (binary graph)	
	micro DOA	Macro DOA	micro DOA	Macro DOA
SPLIT 1	87.14	87.73	71.00	72.48
SPLIT 2	86.98	87.61	70.94	72.91
SPLIT 3	87.20	87.69	71.17	72.98
SPLIT 4	87.08	87.47	70.05	71.51
SPLIT 5	86.91	88.28	70.00	71.78
Mean	**87.06**	**87.76**	**70.63**	**72.33**

Table 1: Performance comparison between ItemRank and its simplified version with binary Correlation Graph.

	MaxF	CT	PCA CT	One-way	Return	L^+	ItemRank	Katz	Dijkstra
Macro DOA	84.07	84.09	84.04	84.08	72.63	87.23	**87.76**	85.83	49.96
difference with MaxF (in %)	0	+0.02	-0.03	+0.01	-11.43	+3.16	**+3.69**	+1.76	-34.11
STD of the difference with MaxF	0	0.01	0.76	0.01	1.06	0.84	**0.31**	0.24	1.52

Table 2: Comparison among different scoring algorithm applied to MovieLens data set.

- Se presentó un algoritmo de scoring que puede ser usado para recomendar productos de acuerdo a preferencias de un usuario.
- Se comparó con algunos métodos que establecen rankings.
- Como trabajo a futuro se espera experimentación del algoritmo en diferentes aplicaciones.
- También se espera que una nueva versión de ItemRank pueda predecir la satisfacción esperada.

Graph-based recommendation

Youwei Wang, Weihui Dai, Yufei Yuan
Website browsing aid: a navigation graph-based recommendation system.
2008

Website browsing aid

Motivación

- Sitios web son una fuente importante de todo tipo de información.
- Es complejo encontrar información relevante que refleje un interés particular de un usuario.
- Muchos sitios web no proveen guías de navegación hoy en día.

Posibles soluciones

1. Mapas del sitio
2. Motores de búsqueda
3. Herramientas de ayuda inteligentes: Técnicas adaptativas

Website browsing aid

Propuesta

- Desarrollo de un asistnte de navegación online, usando CF basado en grafos.
- Considerar secuencias de navegación y similaridad.
- Uso de conocimiento de grupos.
- Uso de clustering para extraer grupos de usuarios.
- Métricas clásicas para medir desempeño.

Website browsing aid

Grafos de navegación

- Fuentes de información: log files (dirección IP, dirección URL, tiempo)
- Representación del grafo de navegación:
  - nodos: páginas visitadas
  - navigation paths: clics de hipervínculos
- Noción de comunidad: hábitos de navegación similares entre miembros de una comunidad.

Grafo g

Es una 2-tupla (v, e), con v conjunto finito de vértices y $e \subseteq v \times v$ conjunto de arcos. El número de nodos en g es $|g|$.

Sub-grafo g'

Es un grafo $g' = (v', e')$ tal que $v' \subseteq v$ y $e' = e \cap (v' \times v')$. Se usa la notación $g' \subseteq g$.

Sub-grafo común

Sean g_1 y g_2 grafos. g es sub-grafo común de g_1 y g_2 si $g \subseteq g_1$ y $g \subseteq g_2$. Se usa la notación $g \subseteq cs(g_1, g_2)$.

Sub-grafo común maximal

Sean g_1 y g_2 grafos. g es sub-grafo común maximal de g_1 y g_2 si no existe otro sub-grafo g' de g_1 y g_2 que tenga más nodos que g. Se usa la notación $g \subseteq mcs(g_1, g_2)$.

Una métrica de distancia d debe cumplir:

1. $0 \leq d(g_1, g_2) \leq 1$
2. $d(g_1, g_2) = d(g_2, g_1)$
3. $d(g_1, g_2) \leq d(g_1, g_2) + d(g_2, g_3)$

Se define la distancia entre g_1 y g_2 como:

$$d(g_1, g_2) = 1 - \frac{|mcs(g_1, g_2)|}{|g_1| + |g_2| - |mcs(g_1, g_2)|}$$

Website browsing aid

Simplificación de grafo de navegación

- Una buena medida de similaridad entre grafos corresponde a la métrica de distancia planteada anteriormente.
- Se desea *linealizar* el grafo de navegación para convertirlo en una *secuencia* de navegación.
- Se eliminan nodos repetidos en la navegación.
- Se consigue reducir tiempos de cálculo de distancia.

Website browsing aid

Simplificación de grafo de navegación

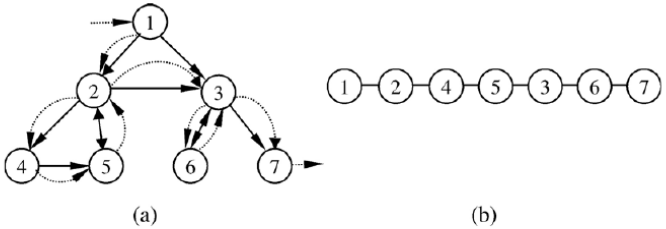

Fig. 1. An example of navigation graph simplification: (a) original navigation graph (dotted lines); (b) navigation sequence after simplification.

Se propone un sistema recomendador compuesto por dos etapas:

1. Clustering Offline
2. Recomendación Online

Website browsing aid

Sistema recomendador propuesto

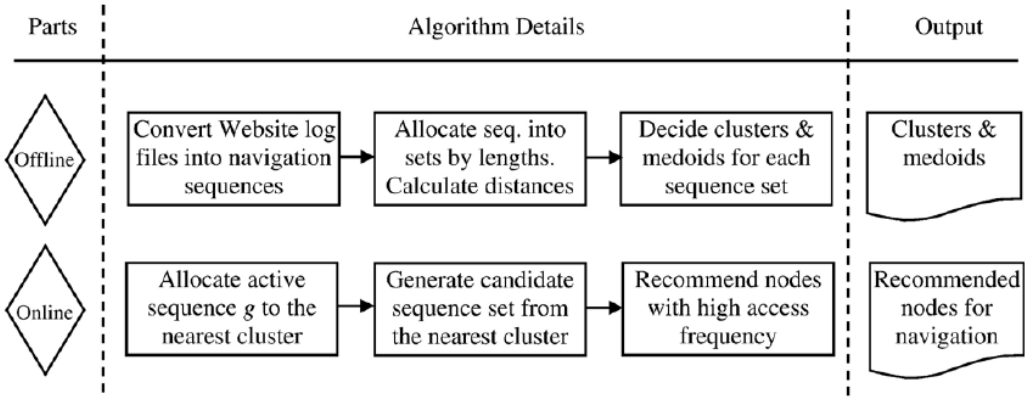

Fig. 2. Recommendation algorithm scheme.

Website browsing aid

Sistema recomendador propuesto

Algorithm for offline clustering

Input: Website log files.

Output: A set of sequence clusters $\{C_{j,k}\}$ and corresponding medoids $\{c_{j,k}\}$.

Step 1. In the website's user access log file, convert each user's navigation path to a graph and then simplify it into a navigation sequence. Remove sequences with length shorter than s or longer than S and denote the sequence set as $G = \{g_i | s \leq |g_i| \leq S\}$.

Step 2. Allocate navigation sequences into a series of sequence set $G_j (j \in [s,S])$, where $G_j = \{g_i | \ |g_i| \geq j\}$. Compute the distance between each couple of sequences within each sequence set G_j.

Step 3. Clustering sequences in each set G_j and output clusters $C_{j,k} (k \in [1,c])$. Determine the medoid $c_{j,k}$ for each cluster $C_{j,k}$.

Algorithm for online recommendation

Input:
g: an active website navigation sequence.
m: the maximal number of nodes for recommendation.
$d_{\min}$: the minimal distance threshold.
Output:
A set of nodes as recommendation for the current visitor.
Step 1. Allocate the active sequence g to the sequence set G_j, where $j = |g|$. Assign g to the cluster $C_{j,k}$ with the greatest possibility according to Eq. (2).
Step 2. Generate candidate sequence set $C_{j,k}$' from cluster $C_{j,k}$ by excluding any sequence with distance $d(g, g_{j,k,l}|_j) \geq d_{\min}$, where $g_{j,k,l} \in C_{j,k,l}$.
Step 3. Sort the nodes in $[C_{j,k}']_j^+ - g$ in descending order according to their access frequency. Recommend at most m nodes with higher access frequency.

Website browsing aid

Experimentos

- Dataset de Music Machines
  - Fechas: 01-01-1999 al 31-01-1999.
  - Log records de más de 4000 visitantes.
  - Training: 24 días consecutivos
  - Testing: Día 25
  - 7 experimentos desplazando en 1 día training y testing.

- Data pre-processing:
  - Uso de programa en JAVA para parsear páginas web: 916 páginas y 7630 links identificados.
  - 65% de visitantes navegaron por 1 o 2 páginas.
  - 34% de visitantes navegaron entre 3 a 20 páginas.
  - 1% visitó más de 20 páginas.

Website browsing aid

Experimentos

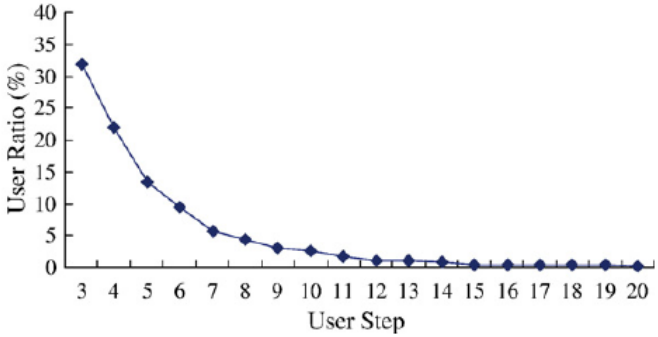

Fig. 3. Ratio of visitors browsing 3 to 20 web pages.

Performance:

$$precision = \frac{|\{Relevant\}| \cap |\{Retrieved\}|}{|\{Retrieved\}|}$$

$$recall = \frac{|\{Relevant\}| \cap |\{Retrieved\}|}{|\{Relevant\}|}$$

Se asume que un sitio web es **relevante** cuando está hacia el final de una secuencia de navegación.

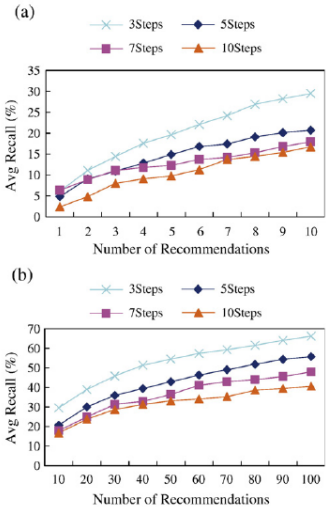

Fig. 4. Recalls under different parameter settings: (a) prediction number from 1 to 10; (b) prediction number from 10 to 100.

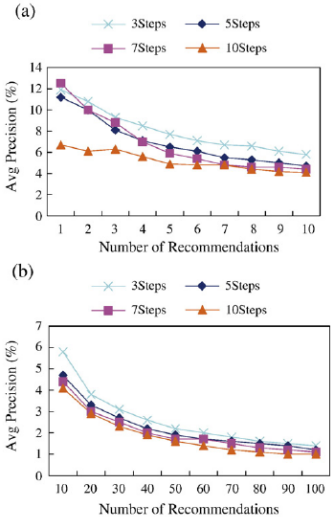

Fig. 5. Precisions under different parameter settings: (a) prediction number from 1 to 10; (b) prediction number from 10 to 100.

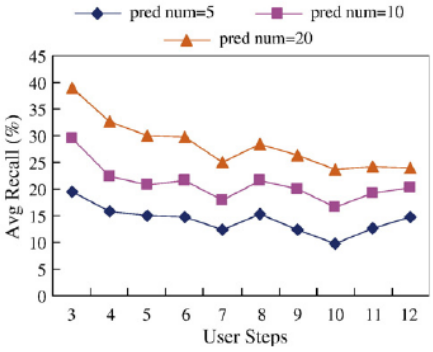

Fig. 6. Average recalls for various user steps.

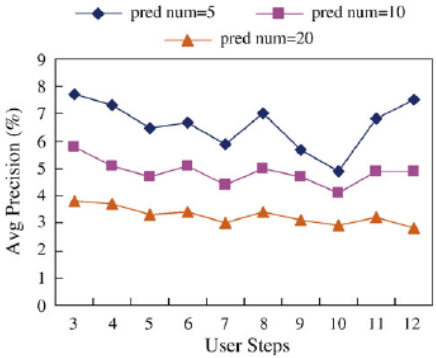

Fig. 7. Average precisions for various user steps.

- Se propuso un sistema de asistencia de navegación usando clustering y modelado mediante grafos.
- Se utilizaron dos etapas, una offline y otra online.
- A futuro se espera validar la definición de sitio web interesante.
- También se espera probar con otras técnicas de clústering y otras métricas de distancia.