USO DE CONOCIMIENTO ESTRUCTURADO EN UN SISTEMA DE RECOMENDACIÓN BASADO EN CONTENIDO[1].

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, Miguel A. Rueda-Morales
Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática, Universidad de Granada, 18071 -- Granada, España
{lci,jmfluna,jhg,mrueda}@decsai.ugr.es

Resumen

Un Sistema de Recomendación basado en contenido permite sugerir al usuario nuevos productos en función de su similitud con el contenido (descripción) de otros productos que éste ha juzgado anteriormente. En la vida cotidiana, a la hora de realizar la descripción de un producto complejo, solemos agrupar su contenido en un conjunto de categorías. Así, lo que estamos haciendo es dotar de estructura al contenido del producto. En este trabajo presentamos un Sistema de Recomendación que es capaz de incorporar el conocimiento sobre esa estructura y ayudarse de él para mejorar las recomendaciones. Hemos usado Redes Bayesianas para modelar la estructura de los productos y las relaciones entre éstos y los usuarios del sistema.

Temática

Sistemas de Recomendación basados en Contenido Estructurado, Redes Bayesianas.

1. Introducción

Muchas veces, en nuestra vida cotidiana, se nos presentan situaciones en las que tenemos que tomar decisiones que a simple vista podemos considerar sencillas. Ejemplos claros como elegir una película para ver, un restaurante para cenar, un libro para leer o planear unas vacaciones pueden ser tareas bastante complicadas por una razón principal: la gran cantidad de libros, restaurantes, películas y destinos vacacionales que existen. Los Sistemas de Recomendación (SR) han surgido para intentar paliar las dificultades de tratar con tantas opciones. En términos generales, los SR producen sugerencias (recomendaciones) sobre productos (o acciones) dentro de un determinado dominio en el cuál está interesado el usuario. En concreto, en este trabajo nos centraremos en películas como productos susceptibles de ser recomendados.

Hay muchos tipos de SR [1,2]. En este artículo vamos a tratar con la variante llamada *basada en contenido* [3] (content-based en inglés) cuya finalidad es encontrar productos similares a aquellos que al usuario le han gustado en función del contenido del mismo. El objetivo final será recomendar a un usuario aquellos productos que más se aproximen a sus preferencias (su perfil de usuario) obtenidas bien explícitamente (mediante un formulario o cuestionario) o implícitamente (analizando los registros de compra, enlaces visitados, artículos vistos o votados,…).

Los SR clásicos utilizan una *descripción del contenido plana*, esto es, todo producto es descrito por una lista de características, normalmente representado por un vector en el que cada posición del vector indica una característica y su valor el peso o importancia que tiene para ese usuario la característica [4,5,6]. Sin embargo hoy día, gracias a la proliferación de lenguajes estructurados como XML, podemos encontrar productos que se pueden describir utilizando su *estructura*, esto es, la descripción del producto se expresa como un conjunto de categorías agrupadas por medio de una jerarquía. Por ejemplo, la Figura 1 muestra una representación estructurada del contenido de la película usando el lenguaje XML.

En este trabajo estudiaremos cómo la incorporación del conocimiento sobre la estructura del producto puede ayudar a la hora de obtener recomendaciones. Con una filosofía similar,

[1] Trabajo respaldado por el Ministerio de Educación y Ciencia y la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía bajo los proyectos TIN2005-02516 y TIC-276, respectivamente.

Libra [7], un SR de libros, determina la relevancia de cada término dependiendo de su ubicación en el libro (título, autores, sinopsis, palabras principales,…) y el voto que ha dado el usuario a dicho libro. AVATAR [12], que recomienda programas de TV, usa perfiles de usuario con estructura, de tal forma que hay una entrada en el perfil para cada categoría de programas (deportes, noticias…) y un índice para cada una de ellas que mide el éxito o fallo de las recomendaciones así como el interés del usuario en ellas. Para recomendar un nuevo programa, usa la componente del perfil de usuario asociada a la categoría en la que se encuadra dicho programa.

```
<pelicula>
    <titulo>Batman (1989)</titulo>
    <anio>1989</anio>
    <actores> Keaton_Michael, Nicholson_Jack,
              Basinger_Kim, Cory_Priscilla </actores>
   <directores> Burton_Tim </directores>
   <productores> Peters_Jon, Suzuki_Keiichi </productores>
    <generos> Action, Fantasy, Thriller </generos>
   <keywords> obsessive-love, hostess, dc-comics,
              batman-joker-rivalry </keywords>
   <plot> Gotham City: dark, dangerous, 'protected' only by
          a mostly corrupt police department... </plot>
</pelicula>
```

Figura 1 – XML descriptivo de una película.

En este trabajo presentamos una nueva aproximación que pretende utilizar la información estructurada para mejorar la capacidad predictiva del sistema. De forma muy esquemática, se podría decir que nuestro modelo utiliza no sólo información del contenido, sino también información sobre la estructura a la hora de determinar la similitud entre productos. El sistema propuesto será modelado usando modelos gráficos probabilísticos, y más concretamente, Redes Bayesianas [8]. La razón es que permiten la combinación de una representación cualitativa del problema (la estructura de la información y su relación con los usuarios del sistema) con una representación cuantitativa por medio de un conjunto de distribuciones de probabilidad que miden la fuerza de las relaciones.
La segunda sección del trabajo describe con detalle el modelo que presentamos. El tercer capítulo presenta una serie de resultados experimentales. Finalmente el último capítulo incluye nuestras conclusiones y el posible trabajo futuro.

2. SR Basado en Contenido mediante Redes Bayesianas

El sistema que proponemos hace uso de dos componentes bien diferenciadas. Por una parte tenemos la componente de *contenido* del producto y por otra tenemos de los votos de usuarios. Notar que pese a que la descripción del modelo la vamos a hacer basándonos en los productos a recomendar (películas), nuestro modelo es genérico, con lo que se puede adaptar a cualquier tipo de producto cuyo contenido esté estructurado.
Nuestro primer paso será modelar en la Red Bayesiana las relaciones de dependencia entre variables. Para ello diferenciamos entre cuatro tipos de nodos en la red (ver Figura 2): *a)* Nodos de términos *T* con el contenido las películas (Keaton_Michael, Burton_Tim, Fantasy, obsessive-love,…). *b)* Nodos categoría *C*, que representan la estructura del contenido de la película (actores, directores,…). *c)* nodos ítem *I*, esto es, la película en si. *d)* Nodos voto *V*, que modelan los votos que dan los usuarios a las películas que han visto.
La topología de la red representa cómo se relacionan entre sí las distintas variables del modelo. Así, existe un arco que une cada término con la unidad estructural a la que pertenece. Por ejemplo, existirá un arco entre el término *Keaton_Michael* y la categoría *actores* de la película *Batman*, otro entre el término *Burton_Tim* y la categoría *directores* de la película *Batman*… Mediante este tipo de relaciones modelizamos que la relevancia de cada categoría para una película va a depender de la relevancia o no de los términos que la componen. De igual forma, existe un arco entre cada categoría y su correspondiente nodo ítem y entre éstos y los nodos voto de los usuarios que las han votado.
Para facilitar la comprensión del modelo, a partir de ahora utilizaremos la notación X_i e Y_k para referirnos a cualquier variable y $x_{i,j}$ e $y_{k,l}$ para referirnos al valor j-ésimo y l-ésimo de dichas variables respectivamente.

Cada nodo término, categoría e ítem tiene asociada una variable aleatoria binaria cuyos valores notamos $\{x_{i,0}, x_{i,1}\}$. Esto significa que la variable va a indicar si el nodo es relevante o no. En cambio, a cada nodo voto asociamos una variable aleatoria cuyos posibles valores van desde 1 hasta $\#r\ \{x_{i,1}, x_{i,2,\ldots,}x_{i,r}\}$, es decir, su valor puede ser cualquier voto posible.

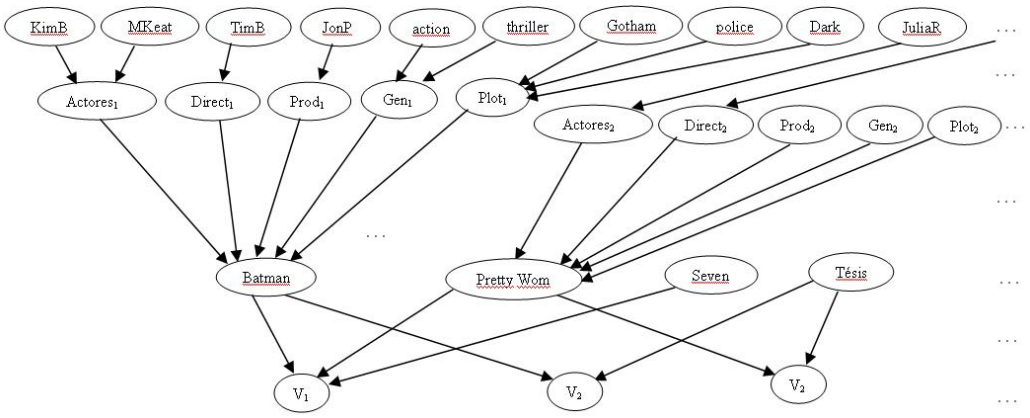

Fig.2 – Ejemplo de Estructura de la Red

Faltarían para completar el modelo, la definición de las distribuciones de probabilidad marginal $P(X_i)$ para los términos, y las probabilidades condicionadas $P(X_i|pa(X_i))$ para el resto de nodos, siendo $pa(X_i)$ todas las posibles configuraciones de los padres de X_i.
Mientras que la estimación de la distribución de probabilidad en los nodos término es bastante simple, la situación para el resto de distribuciones es más compleja debido al gran número de padres que puede tener cada nodo. Así, proponemos el uso de modelos canónicos para representar las probabilidades condicionadas que nos van a permitir usar procedimientos de inferencia muy eficientes. La definición de las probabilidades en los distintos nodos es:

- Para el caso de los nodos término, T, es similar a la dada en [3]:

$$P(x_{i,1})=\frac{1}{M},\ P(x_{i,0})=1-P(x_{i,1}), \quad \textbf{(1)}$$

siendo M el número total de términos en la colección.

- Para el caso de los conjuntos de nodos C, I y V utilizamos dos modelos canónicos distintos dependiendo de la información que representa el nodo:

  - Para los nodos *argumento* (plot en inglés), *ítem* y *usuario,* consideramos que la relevancia de nodo se puede determinar como una suma ponderada de los valores de relevancia de cada uno de sus padres [9], esto es:

$$P(x_{i,j} \mid pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) \quad \textbf{(2)}$$

donde $y_{k,l}$ es el valor que el nodo Y_k toma en la configuración $pa(X_i)$ y $w(y_{k,l}, x_{i,j})$ son pesos que miden cómo el l-ésimo valor de la variable Y_k describe el j-ésimo estado del nodo X_i, con $w(y_{k,\bullet},x_{i,j}) \geq 0$ y $\sum_{Y_k \in Pa(X_i)} w(y_{k,\bullet}, x_{i,j}) \leq 1$.

La definición de los pesos varía dependiendo del tipo de nodo que consideremos:

- Para los nodos *argumento:*

$$\forall X_i \in argumento, \forall Y_k \in Pa(X_i), w(y_{k,l}, x_{i,j}) = \frac{tf_k \cdot idf_k}{\sum_{h=1}^{N} tf_h \cdot idf_h} \quad \textbf{(3)}$$

donde tf_k es la frecuencia (número de veces que aparece) del término Y_k en la descripción del argumento, idf_k es la frecuencia documental inversa de Y_k en la colección completa, siendo N el número total de padres de la categoría *argumento*.

- Para los nodos *ítem* tenemos los pesos: $w(y_{k,l}, x_{i,j}) = b_k$, donde b_k depende del tipo de nodo *categoría* con el que estemos tratando. En esencia, estos pesos se corresponden con la importancia que el usuario da a la categoría.
- Para los nodos *voto*, necesitamos medir la influencia del nodo *ítem* Y_j en el patrón de voto del usuario X_i. Proponemos definir los pesos como:

$$w(y_{j,1}, x_{i,s}) = 1/|Pa(X_u)|, \qquad w(y_{j,0}, x_{u,0}) = 1/|Pa(X_u)|,$$
$$w(y_{j,1}, x_{i,t}) = 0, \text{ con } t \neq s,\ 0 \leq t \leq \#r, \qquad w(y_{j,0}, x_{u,t}) = 0,\ 1 \leq t \leq \#r. \quad \textbf{(4)}$$

siendo $|Pa(X_u)|$ el número de películas previamente votadas por el usuario *u* y *s* el voto real del usuario a la película Y_j.

- Para el resto de nodos *categoría* (*actores, directores, géneros*...), definimos las probabilidades implementando una puerta Noise-OR ya que consideramos que el hecho de que un solo término sea relevante nos da la suficiente información sobre la relevancia de la categoría. Así, usando esta puerta, minimizamos la pérdida de peso que se daría en el caso de un gran tamaño en el conjunto de padres si usáramos una puerta sumatoria. Las probabilidades son:

$$P(x_{i,j} \mid pa(X_i)) = 1 - \prod_{Y_k \in R(pa(X_i))} w(y_{k,l}, x_{i,j}), \quad \textbf{(5)}$$

con $R(pa(X_i))$ aquellos términos relevantes en la configuración $pa(X_i)$, $w(y_{k,l}, x_{i,j}) \geq 0$ y $\prod_{Y_k \in pa(X_i)} w(y_{k,l}, x_{i,j}) \leq 1$.

Definimos $w(y_{k,l}, x_{i,j}) = 1 - \alpha$, siendo α una variable que mide la importancia del término en la categoría correspondiente.

2.1 Prediciendo el voto del usuario

Una vez que hemos completado la Red Bayesiana, la podemos utilizar para predecir el voto que un usuario daría a una película que no ha visto. Para ello bastaría con introducir como evidencia en la red la información de contenido asociada a la nueva película y propagar hacia el nodo *voto*, esto es, consideramos como evidencia, *ev,* el conjunto de términos que componen una película. Esta propagación la podemos hacer de forma eficiente siguiendo una filosofía descendente (*top-down*): Comenzamos calculando las probabilidades en los nodos *término*. Una vez completadas, pasamos a calcular las de los nodos *categoría* utilizando las probabilidades ya obtenidas en los nodos *término* y así hasta completar el modelo usando las probabilidades de los nodos *categoría* para hallar las de los nodos *ítem* y éstas para obtener las probabilidades de los nodos *voto*.

Inicialmente, las probabilidades en los nodos *término* se calculan: $P(x_{i,j} \mid ev) = 1$ si $X_i \in ev$ y 0 en otro caso. Para el resto de nodos del modelo usamos:

$$P(x_{i,j} \mid ev) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) \cdot P(y_{k,l} \mid ev) \quad \textbf{(6)}$$

$$P(x_{i,j} \mid ev) = 1 - \prod_{Y_k \in pa(X_i)} w(y_{k,l}, x_{i,j}), \text{ con } w(y_{k,l}, x_{i,j}) = 1 - (\alpha \cdot P(y_{k,l} \mid ev)) \quad \textbf{(7)}$$

usando la *Ecuación 6* en el caso de que la categoría sea de tipo *argumento*, nodos *ítem* y nodos *voto,* y la *Ecuación 7* para el resto de nodos *categoría* (*actores, directores, productores, géneros* y *keywords*).

3. Experimentación

Esta sección presenta algunos resultados experimentales sobre el funcionamiento del modelo. Con respecto a la Base de Datos (BD), hemos utilizado dos fuentes de información

distintas: Movielens[2] en la que tenemos los votos que han dado 943 usuarios a 1682 películas. Esta información la hemos cruzado con las distintas BD en IMDB[3] para obtener la descripción en contenido (actores, directores, productores, géneros, keywords y plot) de dichas películas. El resultado es una BD de 1389 películas en formato XML (uno para cada película). El número total de votos de los usuarios a dichas películas es 70813.
Nuestro SR parte de un sistema de Recuperación de información llamado Garnata [10] que nos va a permitir medir la similitud entre películas considerando su estructura.
Hemos realizado las experimentaciones sobre cuatro modelos distintos, dos que aprovechan la estructura de la información para obtener las recomendaciones y dos que no:
Los modelos que no aprovechan la estructura son:

- *Baseline (BL)*: Hemos creado un modelo *baseline* que considera la descripción de la película como texto plano. Para ello enlazamos directamente los términos con los nodos *item* correspondientes (cada película con todos los términos que la describen) y éstos con los usuarios que han visto dichas películas. Usamos las ecuaciones *1* y *2* para definir las probabilidades en el modelo. La definición de los pesos para los nodos *ítem* y *usuario* se hace mediante las ecuaciones *3* y *4* respectivamente.
- *Naive Bayes (NB):* Hemos implementado un modelo Naive Bayes considerando como clase la variable que representa el voto de un usuario y como hijos todas las características de las películas que ha visto el usuario. Como las películas que ha visto cada usuario son distintas, evaluaremos un NB distinto para cada usuario.

Los modelos que usan la estructura son:

- *Estructurado con perfil de usuario genérico (E+PG)*: Sobre el modelo estructurado presentado en la Sección 2, hemos utilizado un criterio genérico para determinar el grado de similitud entre películas, es decir, todos los usuarios utilizan el mismo conjunto de valores en los nodos *ítem* y *categorías*.
- *Estructurado con perfil de usuario individual (E+PI)*: Aplicamos un criterio específico (los valores usados en los nodos *ítem* se obtienen de la experimentación) para determinar el grado de similitud entre películas, es decir, si por ejemplo, un usuario sólo ve películas románticas sin importarle los *actores*, *directores*,…, el peso en su perfil del *género* será bastante alto con respecto al resto de categorías y si otro usuario sólo ve películas en las que actúen una serie de actores determinados, será la categoría *actores* la que obtendrá un peso elevado.

El método de validación usado es el *leave-one-out* [11] ya que podemos encontrar usuarios que han visto un número reducido de películas como para separarlas en conjunto de entrenamiento y prueba. Se han utilizado dos tipos de medidas para determinar la capacidad de recomendación de los modelos [13]: el porcentaje de acierto (%) que mide la frecuencia con la que el sistema hace predicciones correctas y el error medio absoluto (MAE) que mide la desviación media absoluta entre el voto real y el voto predicho.

BL		**NB**		**E+PG**		**E+PI**	
%	%	%	MAE	%	MAE	%	MAE
40.07	0.819	39.32	0.842	40.3	0.817	**41.69**	**0.792**

Tabla 2. Resultados Experimentales

Como se puede observar en la Tabla 2, nuestros modelos estructurados tienen el mejor comportamiento y entre éstos, obtenemos los mejores resultados con el que utiliza un perfil de usuario individual lo cual nos da una idea de lo importante que es tener un perfil cercano al gusto de cada usuario. Con el uso de un perfil común, la ganancia que se pueda obtener de los usuarios que se adapten a ese perfil se pierde en los usuarios que no lo hacen. Para el modelo *E+PG*, hemos observado que las categorías que tienen más peso a la hora de obtener

[2] http://www.movielens.org
[3] http://www.imdb.com

buenos resultados son los actores, productores y géneros. Con respecto al valor α, obtenemos los mejores resultados siempre con un valor (0.1). Éste valor para α, nos lleva a pensar en la necesidad de hacer filtrado de términos (quedarnos los más relevantes en cada película) ya que provocan bastante ruido para valores altos de α. Debido a la extensión del trabajo, no presentamos un análisis más detallado de los resultados obtenidos en el modelo *E+PI*.

4. Conclusiones

Hemos presentado un modelo de Sistema de Recomendación basado en contenido que usa una representación de la información de forma estructurada para mejorar las recomendaciones. Mediante la experimentación realizada hemos demostrado que nuestro modelo estructurado se comporta mejor que los modelos *Baseline* y *Naive Bayes.* Como trabajo futuro nos plantearemos el estudio de métodos automáticos para el aprendizaje de perfiles de usuario. Por otra parte, pensamos hacer un filtrado de los datos de las películas para quitarnos términos poco relevantes (como actores extras) que, por la estructura de nuestro modelo, lo que realmente hacen es meter ruido en las predicciones de los votos.

5. Bibliografía

[1] P. Resnick and H.R. Varian. 1997. Recommender Systems. *Communications of the ACM,* 40(3):56-58.

[2] S. Kangas. 2002. Collaborative filtering and recommendation systems. VTT Information Technology, Research Report TTE4-2001-35.

[3] M. Balabanovic and Y. Shoham. 1997. Fab: Content-based, collaborative recommendation. *Communications of the ACM,* 40(3):66-72.

[4] L.M. de Campos, J. M. Fernández-Luna and J. F. Huete. (2003a). The BNR model: foundations and performance of a Bayesian network-based retrieval model, International Journal of Approximate Reasoning, 34, 265-285.

[5] Szomszor, M., Cattuto, C., Alani, H., O'Hara, K., Baldassarri, A., Loreto, V. and Servedio, V. D. P. (2007) Folksonomies, the Semantic Web, and Movie Recommendation. In *Proceedings of 4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0* (in press), Innsbruck, Austria.

[6] Mak, H. Koprinska, I. Poon, J. INTIMATE: a Web-based movie recommender using text categorization. Web Intelligence, 2003. Proc. IEEE/WIC International Conference.

[7] R.Mooney and L. Roy. 2000. Content-based book recommending using learning for text categorization. In *Proc. Of the Pacific Rim Int. Conf. on Artif. Intell.,* 679-689.

[8] Finn V. Jensen. Bayesian Networks and Decision Graphs, Springer-Verlag 2001.

[9] L.M. de Campos, J. M. Fernández-Luna and J. F. Huete. 2006. A bayesian Network approach to Hybrid Recommending Systems. In *Int. Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 2158–2165, 2006.

[10] Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, Alfonso E. Romero, *Garnata: An Information Retrieval System for Structured Documents based on Probabilistic Graphical Models*, Proc. of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006),Paris (France).

[11] M. Stone. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society, B, 36(1):111-147, 1974.

[12] AVATAR: A Multi-Agent TV Recommender System using MHP Applications. Yolanda Blanco Fernández, José J. Pazos Arias, Alberto Gil Solla, Manuel Ramos Cabrer, Martín López Nores, Ana Belén Barragáns Martínez. IEEE International Conference on E-Technology, E-Commerce and E-Service (EEE), page 660--665 - mar 2005.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.