Filtros basados en contenido (Content-Based Filtering)

En un sistema de recomendación de filtrado de contenido, los elementos se asignan a un espacio de características y las recomendaciones dependen de las características del elemento.

Esto significa que toda la información de características que usamos se deriva sólo de los elementos.

Eso no significa que no dependamos de ninguna información de usuario, sólo que la información se utiliza sólo en el paso de recomendación, y no en el tiempo de computación.

En el filtrado de contenido, las características utilizadas para proporcionar recomendaciones se derivan de los elementos y sólo de los elementos.

Los filtros basados en contenido tienen el producto como base de la predicción, en lugar de tener al usuario. Es decir, utiliza las características del artículo (marca, precio, calificaciones, tamaño, categoría, etc.) para hacer las recomendaciones

Veamos un ejemplo de filtro basado en contenido que usa Machine Learning para hacer las recomendaciones. Pensemos en un sistema de recomendaciones de un servicio de música en streaming. El “producto” en este caso serían las canciones. Los datos de los que disponemos para cada canción son por ejemplo el grupo, el cantante, la discográfica y el género (pop, rock, clásica, banda sonora…). Para enriquecer más al sistema, también vamos a valorar las calificaciones que el usuario ha hecho sobre los temas –calificaciones explícitas, como las puntuaciones con estrellas, o implícitas, como las veces que ha escuchado el tema –, así como las características propias del usuario (edad, sexo y país).

Estos datos, centrados en el producto y aliñados con datos del usuario, serán la materia prima de este sistema de recomendación. Veamos cómo se hace la predicción. El **Machine Learning es una disciplina que hace predicciones en base a preguntas a los datos**. La pregunta que debe responder en este caso es: este usuario al que tengo que hacer una recomendación y que tiene estas características, este comportamiento y que ha calificado previamente estas canciones, ¿qué calificación le daría a esta canción, que es del grupo X, de la discográfica Y y de género Rock? La respuesta del filtro (que hemos entrenado con los datos de cientos de miles de usuarios del sistema) nos daría un número entre 0 y 10, basado en las calificaciones que les han dado otros usuarios que se parecen a él. La pregunta se debe repetir con todas las canciones que se incluyan en el catálogo de recomendaciones y se obtendrá la predicción de las calificaciones de todas ellas. De todas las respuestas, las canciones que obtengan mejor nota serán las que se presenten al usuario.

Los sistemas de recomendación basados en contenido tienen el ítem como base de la predicción, en lugar de tener al usuario. Un sistema de recomendación basado en contenido,básicamente monitorizará los hábitos de consumo de sus usuarios y aprenderá sus preferencias en forma de palabras clave (keywords) o atributos:

- •Los usuarios pueden construir su propio perfil, definir sus preferencias o modificarlas.
- •El sistema puede inferir este perfil a partir de las acciones llevadas a cabo por el usuario(clicks, visualizaciones, compras,etc.) - (Información implícita).
- •El sistema puede inferir el perfil a partir de las votaciones explícitas del usuario sobre los objetos - (Información explícita)
- •El sistema puede inferir el perfil del usuario mezclando las acciones llevadas a cabo por el usuario junto con las votaciones (Implícito y Explícito)
- •Aunque el sistema infiera el perfil del usuario, éste también podrá ver su perfil y modificarlo si cree que algo no es correcto.

Las fuentes que utilizan este tipo de sistemas de recomendación serán fuentes con contenido de calidad y analizables, que generalmente formarán parte de un catálogo. Estos sistemas están muy enfocados a fuentes textuales, puesto que estos se pueden analizar mediante técnicas clásicas de Minería de Datos y Procesamiento del Lenguaje Natural, que permiten extraer de forma sencilla perfiles de usuarios e ítems, pero también está enfocado a cualquier fuente que disponga de información como pueden ser fuentes multimedia, como imágenes o vídeo.

Estos sistemas utilizan técnicas de recuperación de la información (similares a las que utilizan los buscadores en Internet) **o de clasificación** en base a un criterio y utilizan **aprendizaje máquina supervisado** para inducir un clasificador que pueda discriminar entre aquellos ítems que puedan ser de interés para un determinado usuario de aquellos que no son de su interés, por ello, a la hora de construir un perfil de este tipo, es importante conocer la importancia de un atributo a la hora de definir un objeto/ítem (relevancia de un atributo).

Dado un vector de preferencias de usuario, y dado un nuevo ítem, el sistema será capaz de determinar la probabilidad de que a un usuario le guste este nuevo ítem.

Se dice que un sistema de recomendación es del tipo *Content-Based* cuando está basado únicamente en las características del producto y no en la valoración del usuario al producto.

Para ello, tenemos que describir el producto de una manera en la que se pueda realizar una relación entre productos, es decir, tenemos que vectorizar el producto (extracción de tags o atributos) para luego medir la similitud apoyándonos en los atributos extraídos.

Extracción de atributos de un document

Lo primero que necesitamos es extraer los atributos de un determinado documento.

Lo podemos hacer en 2 pasos.

- **Calculamos las ocurrencias:** Extraemos las ocurrencias y eliminamos las palabras carentes de importancia (stopwords) además de los números, puesto que pierden el significado fuera de su contexto.

- **Desde las ocurrencias a las frecuencias:** Normalizamos y ponderamos cada ocurrencia mediante técnicas de análisis del texto.

- http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

Term frequency Inverse Document Frequency (TFIDF)

Esta técnica es muy usada en el área de Information Retrieval (IR). Esta técnica ayuda a entender qué términos son más relevantes o menos relevantes en una fuente textual, buscando con qué frecuencia ocurre cada término en el conjunto entero de ítems. Si un término ocurre con mucha frecuencia en un conjunto de ítems indica que no es relevante para discriminar un objeto de otro. El parámetro **'stop_words** le dice al módulo TF-IDF que ignore las palabras comunes en inglés como 'the'...

En sciki-learn tenemos la clase **TfidfVectorizer** dentro del paquete:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

Esta clase permite normalizar cada uno de los tags que aparecen en un texto determinado y asignarle una importancia dentro del mismo.

```
tfidf_vectorizer = TfidfVectorizer( stop_words=cachedStopWords, token_pattern='(?u)\\b[a-zA-Z]\\w\\w+\\b')
tfidf_vectorizer.fit(datos.text)
```

```
TfidfVectorizer(analyzer='word', binary=False, charset=None,
        charset_error=None, decode_error='strict',
        dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
        lowercase=True, max_df=1.0, max_features=None, min_df=1,
        ngram_range=(1, 1), norm='l2', preprocessor=None, smooth_idf=True,
        stop_words=['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours',
 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'i
tself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', '...ave
n', 'isn', 'ma', 'mightn', 'mustn', 'needn', 'shan', 'shouldn', 'wasn', 'weren', 'won', 'wouldn'],
        strip_accents=None, sublinear_tf=False,
        token_pattern='(?u)\\b[a-zA-Z]\\w\\w+\\b', tokenizer=None,
        use_idf=True, vocabulary=None)
```

En la documentación oficial se pueden ver todos los parámetros que soporta,en el ejemplo anterior sólo se utilizan el de stop_words y el token_pattern

http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Ventajas

1. A diferencia del Filtrado Colaborativo, si los ítems tienen descripciones suficientes, nos evitamos el “new-item problem/cold-start”

2. Las representaciones del contenido son variadas y permiten utilizar diversas técnicas de procesamiento del texto, uso de información semántica, inferencias, etc.

3. Es sencillo hacer un sistema más transparente: usamos el mismo contenido para explicar las recomendaciones.

Inconvenientes

1. Tienden a la sobre-especialización: va a recomendar items similares a los ya consumidos. Este problema puede ser resuelto agregando a la búsqueda aleatoria (por ejemplo mediante algoritmos genéticos)

2. No evitan el “new-user problem”. Es necesario disponer de información o puntuaciones de los distintos usuarios para poder realizar una predicción.