PEC 1

Presentación

Nuestra empresa ha creado una aplicación móvil con la que los usuarios pueden solicitar sugerencias de restaurantes. Una vez han visitado el restaurante, los usuarios hacen una valoración de su precio y calidad que se añade a una base de datos que ayuda al sistema a generar nuevas sugerencias por los usuarios, comparando usuarios con preferencias similares.

En esta PEC 1 repasaremos estos conceptos básicos de recomendación siguiendo el hilo de este ejemplo.

Competencias

En este enunciado se trabajan en un determinado grado las siguientes competencias general de máster:

- Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería en informática.
- Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la ingeniería en informática.
- Capacidad para la aplicación de los conocimientos adquiridos y para solucionar problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar estos conocimientos.
- Poseer habilidades para el aprendizaje continuado, autodirigido y autónomo.
- Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- Capacidad para asegurar, gestionar, auditar y certificar la calidad de los desarrollos, procesos, sistemas, servicios, aplicaciones y productos informáticos.

Las competencias específicas de esta asignatura que es trabajan son:

- Entender que es el aprendizaje automático en el contexto de la Inteligencia Artificial.
- Distinguir entre los diferentes tipos y métodos de aprendizaje.
- Aplicar las técnicas estudiadas en un caso concreto.

Objetivos

En este PEC se practicarán los conceptos del temario relacionados con recomendación y clustering mediante su aplicación a un caso concreto.

Descripción de la PEC a realizar

Datos

La web triahotel.com nos da inicialmente un fichero de datos con las opiniones de 100 clientes sobre los 20 hoteles registrados en la web. Tenga en cuenta que los clientes no han visitado todos los hoteles, por lo que no encontrará todas las combinaciones hotel / cliente.

Cada usuario se identifica con un número del 1 al 100. Cada hotel identifica con un número del 1 al 20. Cada usuario emite 8 opiniones sobre cada hotel que visita, que corresponden a valoración general (valores 1 .. 10), habitaciones (1 .. 5), situación (1 .. 5), limpieza (1 .. 5), precio (1 .. 10), recepción (1 .. 5), restaurante (1 .. 10) y otros servicios (1 .. 5).

Entonces cada línea del archivo hotels.data es una valoración de un hotel por un usuario y tiene el siguiente formato (los valores separados por espacios):

idHotel idUsuari valoració1 valoració2 ... valoració8

Los idHotels y idUsuaris están desordenados.

Para el apartado 4 hay otro archivo, favorits.data, con el hotel favorito de cada usuario. Esta pareja usuario / hotel no sale el archivo anterior.

idUsuari idHotel

Actividades

Para resolver esta PAC debe utilizar el código de los programas 2.2-2.7

con las modificaciones apropiadas para trabajar con los nuevos ficheros de datos. También se utilizará el código 4.4 (k-means) y algunas funciones de la biblioteca sklearn. En todas las actividades es necesario que justifique razonadamente su respuesta.

En todas las actividades hay que justificar razonadamente las respuestas.

Actividad 1

El paso previo a cualquier análisis de datos es efectuar un tratamiento para adecuar sus valores. Entre las operaciones habituales encontramos la normalización o estandarización de los datos (ver por ejemplo http://es.wikipedia.org/wiki/Distribución_normal).

Ante todo se pide, pues, que realice el tratamiento previo de los datos que considere necesario (en algún caso podría no ser necesario hacer ningún tratamiento previo).

Como las valoraciones de los diferentes aspectos tienen escalas diferentes, será necesario escalarlas para que se encuentren en el mismo rango de valores, por ejemplo 0..1. Esto será fundamental más adelante para poder calcular distancias entre valoraciones sin que unos aspectos tengan más peso que otros.

También hay que detectar posibles valores erróneos o ausentes. En este caso hay dos valoraciones iguales a cero, cuando el rango empieza en 1. Se podría asumir razonablemente que estos ceros tendrían que ser unos, o también se pueden eliminar estas entradas, que es lo que se hará en este caso.

El código que resuelve este apartado se encuentra en el fichero **activity1.py**. Como se puede ver, dado que en esta PAC se tienen que procesar los datos de dos formas (por hoteles en las actividades 2 y 3 y por usuarios en la 4), se han creado dos funciones diferentes que devuelven la estructura de datos organizada por hoteles o por usuarios.`

Actividad 2

Aplique un agrupamiento k-means (con K = 4) de los hoteles teniendo en cuenta como vector de características de cada hotel las valoraciones medias de cada aspecto del hotel (es decir valor medio asignado valoración general, a habitaciones, situación, limpieza, etc.). Utilice una función de distancia adecuado para comparar hoteles.

`

En primer lugar se tienen que calcular las valoraciones medias de cada hotel. Como cada hotel se ha valorado por muchos usuarios, se calcula la valoración media de cada aspecto de la encuesta por todos los usuarios de cada hotel, por ejemplo la valoración media del aspecto “limpieza” del hotel 2, etc.

Esto genera una estructura de datos {hotel: [valoraciones medias]}. Este vector de valoraciones medias se puede usar como vector de coordenadas del hotel en el espacio de valoraciones, y por tanto k-means se calculará según la distancia entre hoteles en este espacio de valoraciones.

Lo que se obtiene en este apartado es una asignación de hoteles a uno de los cuatro grupos. Nótese que cada vez que se ejecuta k-means el resultado puede cambiar porque comienza eligiendo cuatro hoteles al azar. Un posible resultado es el siguiente:

```
{1: 2, 2: 1, 3: 2, 4: 1, 5: 1, 6: 3, 7: 3, 8: 2, 9: 3, 10: 3,
11: 3, 12: 0, 13: 1, 14: 1, 15: 0, 16: 3, 17: 3, 18: 3, 19: 3,
20: 3}
```

Hay que recordar que el número de *cluster* no tiene ningún significado aparte de agrupar los hoteles, se tiene que entender como una simple etiqueta asignada a cada hotel.

El código que resuelve este apartado se encuentra en el fichero **activity2.py**, que llama a las funciones de **kmeans_list**.

Actividad 3

Utilice la medida de calidad de agrupamiento Adjusted Rand Index para evaluar el agrupamiento del apartado anterior. Este índice compara dos agrupamientos para ver cómo de similares son; puede dar un valor entre -1 y 1, donde un 1 significa que dos agrupamientos son idénticos y un valor negativo significa que son muy diferentes. Este índice se puede calcular usando una función de sklearn, como por

ejemplo:

>>> From sklearn importe metrics

>>> Labels_true = [0, 0, 0, 1, 1, 1]

>>> Labels_pred = [0, 0, 1, 1, 2, 2]

>>> Metrics.adjusted_rand_score (labels_true, labels_pred)

12:24 ...

Calcular el Adjusted Rand Index sabiendo que el agrupamiento real de los hoteles debería ser [1 .. 5], [6 .. 11], [12 .. 16] y [17 .. 20].

En su ejemplo, labels_true debería ser el agrupamiento que ha obtenido por k-means y labels_pred el agrupamiento real de los hoteles.

La función que calcula el Rand Index espera un vector con las asignaciones de los hoteles, es decir una lista con 20 elementos (uno por hotel) que diga a qué grupo 0..3 pertenece cada hotel. Como referencia (labels_true), se tiene que definir otra lista que codifique el agrupamiento que indica el enunciado:

```
labels_true = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3]
```

En general la coincidencia no es muy grande. Como k-means devuelve un resultado diferente cada vez que se ejecuta, el valor del Rand Index también es diferente. Los valores obtenidos varían entre 0.05 y 0.25, valores que en general indican que el agrupamiento no se parece mucho. Posiblemente el agrupamiento de referencia se fundamenta en criterios predefinidos que no tienen en cuenta las valoraciones de los usuarios.

Un experimento interesante es calcular el Rand Index de dos ejecuciones de k-means. Eso puede dar una idea de si realmente hay clusters en los datos o no. En este caso, en general el Rand Index entre k-means tienden a ser más altos que respecto a la asignación del enunciado, lo que significa que posiblemente hay una cierta estructura de grupos pero no es la propuesta.

El código de esta actividad se encuentra en el fichero **activity2.py**, dado que es continuación natural del apartado anterior.

Actividad 4

Genere un recomendador ponderado basado en memoria y, para cada usuario, encuentre el hotel más recomendable. Coinciden las recomendaciones con el hotel favorito de cada usuario listado en el archivo favorits.data?.

Para cada usuario hay un diccionario {hotel: [valoraciones]} y se debe calcular la similitud entre usuarios como la correlación entre valoraciones del mismo hotel. Teniendo en cuenta esta similitud se estiman las preferencias del usuario mediante las de los otros usuarios, dando más peso a los usuarios con gustos más parecidos. Se usará la correlación de Pearson como medida de similitud porque en general funciona mejor, sobre todo si los valores son relativos.

El código de la correlación de Pearson que hay en los materiales no se puede usar directamente porque espera una única valoración por usuario y hotel, y en este problema hay 8.

Una solución a esto es que la función que calcula el coeficiente de Pearson se aplique a la correlación entre valoraciones y después se calcule la correlación media. En la función **weightedRating** también hay un cambio en una linea (comentario *# PAC1 2014*) para poder tratar muchas valoraciones por usuario (suma todas las valoraciones).

Al ejecutar este recomendador se obtiene que la posición media de los hoteles favoritos en la lista de recomendaciones es 3.96, lo que quiere decir que de 10 hoteles que un usuario no había valorado la recomendación dada se encuentra hacia la mitad de la lista. Es decir, la calidad de la recomendación es media.

El código que resuelve este apartado es **activity4.py**.

Recursos

Este PEC requiere de los siguientes recursos:

Básicos: Ficheros de datos adjuntos al enunciado

Complementarios: Manual de teoría de la asignatura. En especial las tablas de código 2.2-2.7 y 4.4.

Criterios de valoración

Los ejercicios tendrán la siguiente valoración asociada:

Actividad 1: 1 punto

Actividad 2: 3 puntos

Actividad 3: 2.5 puntos

Actividad 4: 3.5 puntos

Es necesario razonar las respuestas en todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.

Formato y fecha de entrega

La PEC debe entregarse antes del **próximo 5 de Abril** (antes de las 24h).

La solución a entregar consiste en un informe en formato PDF usando la plantilla colgada en el tablón de la asignatura más los archivos de código (*. Py) que usó para resolver la prueba. Estos archivos deben comprimir en un archivo ZIP.

Adjuntar el fichero a un mensaje en el apartado de **Entrega y Registro de AC (RAC)**. El nombre del archivo debe ser ApellidosNombre_IA_PEC1 con la extensión. zip.

Para dudas y aclaraciones sobre el enunciado, diríjase al consultor responsable de su aula.

Nota: **Propiedad intelectual**
A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por tanto comprensible hacerlo en el marco de una práctica de los estudios del Grado en Informática, siempre y esto se documente claramente y no suponga plagio en la práctica.
Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se presentará junto con ella un documento en el que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y el su estatus legal: si la obra está protegida por copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL ...). El estudiante deberá asegurarse de que la licencia que sea no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente deberá asumir que la obra está protegida por copyright. Deberán, además, adjuntar los archivos originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.