Aprendizaje y Razonamiento Automático - Práctica de Modelos Probabilísticos - Predicción de la valoración de alojamientos de AirBnB

Ramón Béjar

Mayo de 2022

Objetivo de la práctica

Considera el siguiente dominio de aplicación para inferencia y aprendizaje con modelos probabilísticos basados en redes bayesianas. Se trata de conseguir predecir la valoración media que tiene un alojamiento de AirBnB, en el rango [1,2,3,4,5], a partir de diferentes características del alojamiento. El conjunto de datos con el que tenéis que trabajar para aprender un modelo, se encuentra en esta web: `https://www.kaggle.com/fermatsavant/airbnb-dataset-of-barcelona-city` Pero también os he dejado el fichero CSV correspondiente junto con el enunciado de la práctica.

Cada alojamiento se representa mediante un conjunto de 9 atributos:

- `room_type`
- `neighborhood`
- `reviews`
- `overall_satisfaction`
- `accommodates`
- `bedrooms`
- `price`
- `latitude`
- `longitude`

En la página web de descarga tenéis la descripción del significado de cada atributo.

El atributo `overall_satisfaction` es el que queremos ser capaces de predecir a partir de los otros (será el atributo que usamos como `class variable` para cada instancia). No todos los atributos son del mismo tipo. Algunos toman valores en conjuntos discretos, pero otros varían en rangos de números reales.

Observar que el fichero con los datos para aprender un modelo no se encuentra en formato ARFF:

`https://waikato.github.io/weka-wiki/formats_and_processing/arff_stable/`

(el que usa weka por defecto), sino que se encuentra en formato csv. Por tanto, antes de poder usarlos con el algoritmo de aprendizaje K2 de weka tendréis que transformarlos a ficheros ARFF donde todos los atributos tengan un rango discreto de valores. Eso quiere decir que para aquellos atributos que son números reales, tendréis que transformarlos a un conjunto de valores discretos, escogiendo un tamaño para ese conjunto discreto. Ese tamaño del conjunto discreto puede ser un mismo parámetro para cada uno de esos atributos, o podéis pensar en alguna función que escoja automáticamente un tamaño adecuado en función del número de valores reales diferentes del atributo que aparecen en el dataset. Pensar que si el conjunto discreto es muy pequeño respecto al rango original de valores, seguramente estaréis comprimiendo demasiado los valores (muchos valores reales se estarán mapeando a un mismo valor discreto), aunque esto no tiene porque ser necesariamente algo malo.

Vuestro trabajo

Modelos a aprender con K2

El objetivo es obtener tres modelos Bayesianos, considerando tres clases de modelos diferentes:

1. Redes bayesianas obtenidas con K2 a partir de un modelo inicial vacio (sin aristas iniciales) y con un orden entre las variables escogido al azar y con un valor para el número máximo de padres por variable (parámetro U en el algoritmo K2) igual a 3.
2. Red bayesiana naive (modelo único), siendo la variable `overall_satisfaction` la variable independiente (y padre de todas las otras). Por tanto, el valor de U en este caso tendrá que ser 0.
3. Redes bayesianas obtenidas con K2 a partir de un modelo inicial que sea la red bayesiana naive del punto 2, pero pudiendo añadir dos aristas adicionales para cada nodo (y por tanto U tendrá que ser igual a 3).

Para obtener un buen modelo para cada caso (excepto para el caso 2 que sólo tiene un modelo posible), ejecuta el algoritmo K2 de `Weka` 10 veces, cambiando en cada ejecución el orden entre las variables (esto se consigue simplemente indicando que el orden inicial es aleatorio), con la idea de maximizar las probabilidades de obtener un buen modelo.

Selección del mejor modelo para las clases 1 y 3

Para las clases 1 y 3, ya que en cada una generamos 10 modelos diferentes, tendremos que escoger uno de esos modelos como el mejor para cada una. Para escoger el mejor modelo de esos 10, utilizaremos el data set de testing que generáis con vuestro script python. Es decir, escogeremos como mejor modelo el que tenga un error menor a la hora de clasificar las instancias en el conjunto de testing [1]. Por clasificar una instancia nos referimos a la predicción que hacemos de su atributo de clase (`overall_satisfaction`) a partir de los otros atributos. Pero tenéis que mostrar igualmente el error que se obtiene con los 10 modelos y su UPSM score (Bayes Score en Weka).

[1] El error será simplemente el porcentaje de instancias clasificadas incorrectamente.

Script python para transformación y partición de datos

Para poder trabajar con weka a partir del dataset original (que se encuentra en formato csv), tendréis que desarrollar un pequeño script en python para transformar los datos a ARFF y con atributos discretos. El script deberá recibir, **al menos**, estos argumentos:

1. Nombre del fichero csv donde se encuentran los datos originales.
2. Nombre del fichero ARFF donde queremos guardar los datos para aprendizaje.
3. Nombre del fichero ARFF donde guardaremos los datos para evaluar los modelos que aprendemos.

Tener en cuenta que python tiene librerias disponibles para trabajar con ficheros csv.

Vuestro script debe enviar un 75 % de los registros de entrada a los datos de aprendizaje (primer fichero ARFF) y el 25 % restante al fichero con los datos para evaluación. **Esta selección debe de hacerse de forma aleatoria** usando las funciones del modulo random de python, y tenéis que usar como *seed* del generador

aleatorio los cinco últimos digitos del DNI de alguno de los miembros del grupo, valor que tendréis luego que entregarme, para que yo pueda reproducir la generación de vuestros ficheros ARFF a partir del fichero csv (poniendo la misma *seed* que vosotros habéis usado). Evidentemente, en función del tamaño original del dataset, puede no ser posible seleccionar tamaños para los dos datasets de salida que tengan **exactamente** esas proporciones de 3/4 en el de aprendizaje y 1/4 para el de testing, pero siempre es posible escoger los tamaños de forma que la fracción que representan se aproxime lo más posible a estos valores.

Tal como hemos comentado antes, este script también deberá transformar todos los atributos que no tengan originalmente un dominio discreto, a una versión con un dominio discreto. Es decir, mapear un rango de números reales a un rango discreto, usando aquellos parámetros que vosotros consideréis oportunos. Por ejemplo, si para el atributo `overall_satisfaction` observáis que los únicos valores reales que aparecen son estos:

$$[1,\ 1.5,\ 2,\ 2.5,\ 3,\ 3.5,\ 4,\ 4.5,\ 5]$$

podéis decidir mapearlos al conjunto discreto $[1, 2, 3, \ldots, 9]$. O si queréis *perder* la información de todos los valores decimales, mapearlo simplemente al rango $[1, 2, 3, 4, 5]$. O podéis usar una función genérica que haga esto para cualquier conjunto de entrada, pudiendo escoger el tamaño del conjunto discreto resultante.

Por otro lado, el fichero csv usa caracteres con codificación UTF-8, de forma que al pasarlos a ARFF, comprobar que no causen problemas a weka. Siempre podéis decidir transformarlos a valores en ASCII, usando algún tipo de diccionario python.

Entrega de la práctica

Deberéis entregarme:

1. El script en python (versión 2 o 3) usado para convertir el fichero csv de entrada en los dos ficheros ARFF para weka: el de aprendizaje y el de evaluación. Este script debe de estar **suficientemente comentado** para poder entender su funcionamiento. Poner instrucciones claras sobre si es python 2 o 3, y si hay requerimientos de librerías adicionales para poder usarlo. **No aceptaré ningún programa que no explique con detalle todas las funciones que se usan, tanto si son vuestras como de alguna librería externa**.
2. Un documento PDF que contenga:
   a) Una explicación de como ejecutar vuestro script y el valor que poner en la función random.seed() (los cinco últimos dígitos de vuestro DNI) para obtener los dos ficheros ARFF que habéis usado para aprender y evaluar vuestros modelos.
   b) La puntuación log(UPSM) y el error de clasificación obtenido **en todos los modelos** para cada una de las tres clases de modelos.
   c) El mejor modelo seleccionado de cada clase y su justificación de porque es el mejor. Mostrar también el dibujo de la red bayesiana para cada uno de estos tres modelos (incluso para el caso de la red bayesiana naive). Recordar que weka te permite visualizar la red bayesiana del modelo aprendido.

Importante: El que quiera obtener una nota superior al 8.5, deberá de explicar con detalle de que forma ha escogido el tamaño de los conjuntos discretos de valores para transformar los atributos de entrada con valores reales. Y explicar como cambia la calidad de la predicción de los modelos al probar con diferentes tamaños de los conjuntos discretos, y con que tamaños os habéis quedado finalmente para obtener el mejor resultado.