Maestría en Inteligencia Artificial

Aprendizaje Automático Avanzado
Discretización de variables

Dr. Gaddiel Desirena López

Los recuentos brutos que abarcan varios órdenes de magnitud son problemáticos para muchos modelos.

- En un modelo lineal, el mismo coeficiente lineal tendría que funcionar para todos los valores posibles del recuento.
- Los recuentos grandes también podrían causar estragos en los métodos de aprendizaje no supervisados, como el agrupamiento de k–means, que utiliza la distancia euclidiana para medir la similitud entre puntos de datos.
- Un recuento grande en un elemento del vector de datos superaría la similitud en todos los demás elementos, lo que podría desechar toda la medición de similitud.

Una solución es contener la escala cuantificando el recuento. En otras palabras, agrupamos los recuentos en contenedores y nos deshacemos de los valores de recuento reales. La cuantificación asigna un número continuo a uno discreto. Podemos pensar en un conjunto de números discretizados como una **secuencia ordenada de contenedores** que representan una medida de intensidad en el conjunto real.

1. Intervalos de ancho constante

Es el proceso de transformar variables continuas en variables discretas mediante la creación de un conjunto de intervalos contiguos (*bins*). La discretización se utiliza para cambiar la distribución de variables asimétricas y minimizar la influencia de valores atípicos y, por lo tanto, mejorar el rendimiento de algunos modelos de aprendizaje automático.

Con la agrupación de ancho fijo,

- cada bandeja (*bina*) contiene un rango numérico específico.
- Los rangos se pueden diseñar a medida o segmentar automáticamente, y se pueden escalar linealmente o exponencialmente.

Por ejemplo, podemos agrupar a las personas en rangos de edad por década: 0 a 9 años en el contenedor 1, 10 a 19 años en el contenedor 2, etc. Para mapear desde el recuento hasta el contenedor, simplemente dividimos por el ancho del contenedor y toma la parte entera.

Cuando los números abarcan múltiples magnitudes, puede ser mejor agrupar por potencias de 10 (o potencias de cualquier constante): 0–9, 10–99, 100–999, 1000–9999, etc. Los anchos de intervalo crecen exponencialmente, yendo desde O(10), hasta O(100), O(1000) y más allá. Para mapear desde el recuento al contenedor, tomamos el logaritmo del recuento. El agrupamiento de ancho exponencial está muy relacionado con la transformación logarítmica.

2. Intervalos de frecuencia constante

Si clasifica las observaciones en *bins* con la misma frecuencia, la discretización distribuye los valores de una variable sesgada de forma más homogénea en todo el rango de valores.

El agrupamiento de ancho fijo es fácil de calcular. Pero si hay grandes lagunas en los recuentos, habrá muchos contenedores vacíos sin datos. Este problema se puede resolver colocando de forma adaptativa los contenedores en función de la distribución de los datos. Esto se puede hacer usando los **cuantiles** de la distribución. Los cuantiles son valores que dividen los datos en porciones iguales. Por ejemplo, la mediana divide los datos en mitades; la mitad de los puntos de datos son más pequeños y la otra mitad, más grandes que la mediana. Los cuartiles dividen los datos en cuartos, los deciles en décimas, etc.

3. Discretización por K–Means

En la agrupación de k–means

- Se usa el algoritmo de K–means para clasificar.
- Las agrupaciones se definen por los elementos pertenecientes a cada cluster.
- El número de agrupaciones las define el usuario.

3.1. K–means

También conocido como algoritmo de Lloyd, consiste en los siguientes pasos:

1. En el paso de inicialización, se eligen K observaciones al azar como los centroides de los K grupos.
2. Los datos restantes se asignan al grupo **más cercano** de cada centroide.
3. En el paso de iteración, los centroides se vuelven a calcular como los puntos promedio de todas las observaciones dentro del grupo.
4. Las observaciones se reasignan al grupo más cercano recién creado.

El paso de iteración continúa hasta que se encuentran los k centros óptimos.

4. Discretización por árboles de decisión

Consiste en utilizar un árbol de decisión para identificar los bins óptimos en los que ordenar los valores de las variables.

- El árbol de decisión se construye utilizando la variable a discretizar y el objetivo.
- Cuando un árbol de decisión hace una predicción, asigna una observación a una de las N hojas finales, por lo tanto, cualquier árbol de decisión generará una salida discreta, cuyos valores son las predicciones en cada una de sus N hojas.
- La discretización con árboles de decisión crea una relación monótona entre los contenedores y el objetivo.

4.1. Árboles de decisión

El proceso de construcción de un árbol de predicción (regresión o clasificación) se divide en dos etapas [1]:

- División sucesiva del espacio generando regiones disjuntas R_1, R_2, R_3, ... R_j. Aunque, desde el punto de vista teórico las regiones podrían tener cualquier forma, si se limitan a regiones rectangulares (de múltiples dimensiones), se simplifica en gran medida el proceso de construcción y se facilita la interpretación.

- Predicción de la variable respuesta en cada región.

El objetivo es encontrar las J regiones (R_1, ... R_j) que minimicen la distancia entre la media de las respuestas de las j observaciones $\hat{y}_{R_j}$ y su objetivo y_i, esto es

$$\text{mín} \sum_{j=1}^{J} d(y_i, \hat{y}_{R_j})$$

Infortunadamente, no es posible considerar todas las posibles particiones del espacio, por esta razón, se recurre a lo que se conoce como Partición binaria recursiva (*recursive binary splitting*).

4.2. Partición binaria recursiva

El objetivo del método es encontrar en cada iteración el predictor X_j y el umbral s tal que, si se distribuyen las observaciones en las regiones $\{X \mid X_j < s\}$ y $\{X \mid X_j \geq s\}$, se consigue la mayor reducción posible entre el objetivo y el promedio de la observación. El algoritmo es:

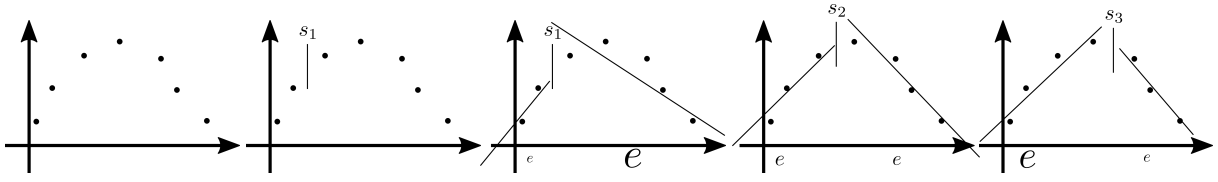
1. El proceso se inicia en lo más alto del árbol, donde todas las observaciones pertenecen a la misma región.
2. Se identifican todos los posibles puntos de corte s para cada uno de los predictores $(X_1, \ldots, X_p)$.
   - En el caso de predictores cualitativos, los posibles puntos de corte son cada uno de sus niveles.
   - Para predictores continuos, se ordenan de menor a mayor sus valores, el punto intermedio entre cada par de valores se emplea como punto de corte.
3. Se calcula la distancia total que se consigue con cada posible división.

   $$d(y_i, \hat{y}_{R_1}) + d(y_i, \hat{y}_{R_2}), \qquad \forall\, i : x_i \in R_j, \;\; j = \{1, 2\}$$

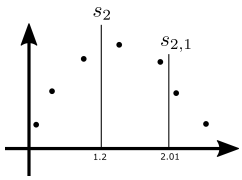
   cada término representa el resultado de separar las observaciones acorde al predictor j y valor s.
4. Se selecciona el predictor X_j y el punto s que resulta en la menor distancia total, es decir, que da lugar a las divisiones más homogéneas posibles. Si existen dos o más divisiones que consiguen la misma mejora, la elección entre ellas es aleatoria.
5. Se repiten de forma iterativa los pasos 1 a 4 para cada una de las regiones que se han creado en la iteración anterior hasta que se alcanza algún criterio de paro. Algunos de ellos son:
   - Que ninguna región contenga un mínimo de n observaciones.
   - Que el árbol tenga un máximo de nodos terminales.
   - Que la incorporación del nodo reduzca el error en al menos un porcentaje mínimo.

A continuación se muestran una serie de gráficos que ejemplifican el proceso:

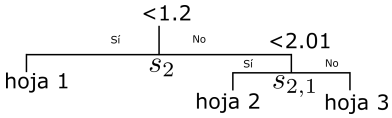
1. Se tienen una serie de datos relacionados
2. Se separan los datos en dos grupos limitados por s_1
3. Se realiza una regresión (o promedio) en cada grupo y se mide el error de aproximación
4. Se siguen separando (s_2, s_3, etc.) y ajustando los datos para comparar el error



5. Se selecciona la combinación de menor error (s_2), y se sigue particionando el espacio. Por el lado izquierdo los datos ya no se pueden particionar, por el lado derecho solo hay una posible separación ($s_{2,1}$).



El árbol queda de la siguiente forma:



donde las hojas corresponden a los segmentos R_1, R_2, ..., R_j del algoritmo general.

5. Métricas

- **Residual Sum of Squares (RSS):**

$$RSS = \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

- **Distancia de Minkowski:** [2]

$$d(y, \hat{y}_{R_j}) = \left(\sum_{i=1}^{J} |y_i - \hat{y}_{R_j}|^p \right)^{1/p},$$

  - cuando $p = 2$, se tiene la distancia **Euclideana**.
  - cuando $p = 1$, se tiene la distancia **Manhattan**.
  - cuando $p = \infty$, se tiene la **máxima distancia**. En este caso, la distancia corresponde a la componente $|y_i - \hat{y}_{R_j}|$ con el valor mas alto.

Para construir un árbol de clasificación se emplea el mismo método recursive binary splitting, sin embargo, como la variable respuesta en cualitativa, no es posible emplear un criterio de selección continuo. Para ello, existen varias alternativas:

- **Gini index:** Es una medida de la varianza total en el conjunto de las K clases del nodo m. Se considera una medida de pureza del nodo.

$$G_m = \sum_{k=1}^{K} \hat{p}_{m,k}(1 - \hat{p}_{m,k}),$$

donde $\hat{p}_{m,k}$ es la proporción de observaciones del nodo m que pertenecen a la clase k. Cuando $\hat{p}_{m,k}$ es cercano a cero o a uno, el término $\hat{p}_{m,k}(1 - \hat{p}_{m,k})$ es pequeño. Como consecuencia, cuanto mayor sea la pureza del nodo, menor el valor del índice Gini G.

- **Information gain: Cross entropy:** Si un nodo es puro, contiene únicamente observaciones de una clase, su entropía es cero. Por el contrario, si la frecuencia de cada clase es la misma, el valor de la entropía alcanza el valor máximo de uno

$$D = -\sum_{k=1}^{K} \hat{p}_{m,k} \log(\hat{p}_{m,k}).$$

- **Chi–cuadrada χ^2:** cuanto mayor sea este valor, mayor sera la evidencia de que existe una diferencia.

$$\chi^2 = \sum_k \frac{(y_k - \hat{y}_k)^2}{\hat{y}_k}$$

Referencias

[1] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer.

[2] Igual, L., & Seguí, S. (2017). Introduction to Data Science. In Introduction to Data Science (pp. 1-4). Springer, Cham.

[3] Galli, S. (2020). Python feature engineering cookbook: over 70 recipes for creating, engineering, and transforming features to build machine learning models. Packt Publishing Ltd. 166–195.