IIC2433 — Minería de Datos

Apuntes: Aprendizaje no supervisado, Clustering (primera parte)

1. Aprendizaje no supervisado

Hasta ahora todos nuestros problemas consideraban un *entrenamiento* con datos rotulados: set X de datos para los cuales ya conocía la respuesta a la pregunta, almacenada en un vector y. ¿Pero qué pasa si no disponemos de datos rotulados para empezar? Es difícil pensar que podremos llegar a algo tan fino como una regresión lineal si no tenemos datos con los cuales aprender, pero sin embargo hay harto que si podemos hacer, solo basta un poco de trabajo y (a veces) mucha creatividad.

1.1. Clustering

El problema de aprendizaje no-supervisado más conocido es el de *Clustering*. En este problema, somos entregados un set de datos, y nos interesa generar un agrupamiento de estos datos, de forma de que los datos más cercanos queden en el mismo grupo, y los datos más lejanos queden en otro grupo.

Nota que esta definición requiere de varias precisiones.

- **Noción de distancia**. ¿Qué significa datos más cercanos o lejanos? Cuando los datos son vectores en algún espacio métrico (como números reales o naturales), tenemos a disposición las distancias asociadas a ese espacio: por ejemplo, en vectores sobre los reales podemos usar la distancia euclidiana. Pero otras veces vamos a requerir otras nociones de distancia: si clusterisamos en un grafo vamos a usar como distancia el camino más corto entre los nodos, si clusterizamos sobre secuencias quizás queramos usar la distancia Levenshtein, y asi. La noción de distancia va a determinar qué puntos están cerca, y cuales están lejos.

- **Noción de agrupación**. En su forma más básica, queremos producir grupos que sean una partición del conjunto X: cada grupo pertenece solo a una clase. Pero puede que sea útil, por ejemplo, producir una agrupación en distintos niveles, donde un cluster sea dividido en varios clústeres más pequeños. Esto se conoce como clustering jerárquico.

- **Número de clústeres**. Esta pregunta es importante. Algunos algoritmos van a necesitar un número fijo de clusters, pero otros van a ir produciendo ese número a medida que van maximizando ciertos criterios.

1.2. Otras formas de aprendizaje no-supervisado

Si bien clustering es el más usual, hay muchas otras formas. En algunas clases más vamos a comenzar a estudiar el problema de reducción de dimensionalidad, o en una forma más general, el problema de cómo generar un espacio latente que sea más manejable para alguna otra tarea, o que haga que la siguiente tarea pueda ser más eficiente aún. Trabajando con texto, es muy usual realizar trabajos de modelado de topicos (o topic modelling), en donde tratamos de encontrar cuáles son los tópicos más prevalentes en el texto. O la generación de resúmenes automáticos, donde tratamos de identificar (o generar) las frases más relevantes en un cuerpo de texto. Un poco más avanzado, hay algunos problemas de aprendizaje que hoy en día son resueltos por técnicas más avanzadas: modelos generativos de redes profundas, modelos basados en energía, etc.

2. Kmeans: una forma simple de hacer clustering

2.1. Definición

Kmeans es un algoritmo para clusterizar datos numéricos. Sus inputs son entonces un conjunto $X = x_1, \ldots, x_n$ de vectores numéricos (todos de la misma dimensionalidad), y un número K de clusters. Podemos pensar en Kmeans como un algoritmo que trata de ubicar K vectores adicionales en el dataset: estos K vectores van a ser los centros de los clusters, y cada vector en X va a estar clasificado de acuerdo al centro de clúster en el que quede más cerca. Entonces:

1. En este algoritmo $d(a, b)$ es la distancia usual (euclideana) entre vectores.
2. Generar $K = \{m_1, \ldots, m_k\}$ vectores nuevos al azar. Estos vectores son los centros.
3. Inicializar $S_1, \ldots, S_k$ vacíos, son los puntos asociados a cada uno de los K centros.
4. Repetir hasta que $S_1, \ldots, S_k$ no cambie entre una iteración y otra:
   a) Para cada vector $x \in X$, asignar x al conjunto S_j de tal forma que se minimiza $d(x, k_j)$. En caso de empate, tomar un conjunto S_j al azar dentro de los que empatan.
   b) Recalcular cada k_j como el centro del conjunto S_j:

$$k_j = \frac{1}{|S_j|} \sum_{x \in S_j} x$$

2.2. ¿Número de clústeres?

Un problema usual en Kmeans es que la elección de clústeres depende del usuario. En pocos casos (como en la tarea) ya vamos a saber la respuesta. Pero en la gran mayoría,

no. La forma usual es fijarse en diferentes medidas de distorsión entre los clústeres. Pero el problema es que la distorsión siempre va a tener a disminuir mientras aumenta el número de clústeres. Mal que mal, ¡la clusterización perfecta es asignarle un clúster a cada punto en el espacio! Por eso se habla del método del codo: graficar la medida de distorsión contra el número de clústeres, y tomar nota del punto en el que la distorsión comienza a disminuir de forma más lenta. Ese punto representa una buena cantidad de clústeres.

Como medidas de distorsión, sugerimos (entre otras):

- El promedio de las distancias cuadradas de cada punto a su centro de clúster.
- El promedio de las distancias entre cada par de centros del clúster.