Autor: Ítalo Garleni Rodríguez

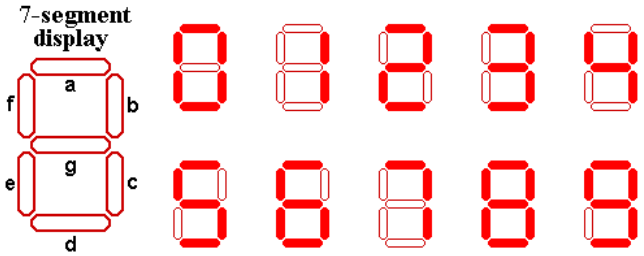
Clasificación con Redes Bayesianas

Autor: Ítalo Garleni Rodríguez

Descripción de la práctica

El problema que vamos a resolver trata de un panel digital de 7 segmentos, en el que se muestran números según los segmentos activados. Existen nueve combinaciones de segmentos, que representan un número del 1 al 9 y, por tanto, una clase.

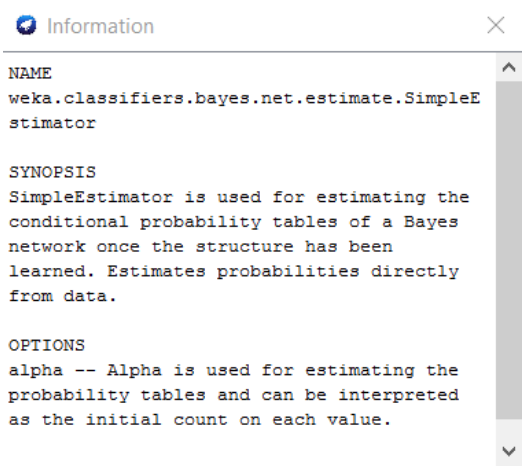
El objetivo es generar diversas Redes Bayesianas con modelos de aprendizaje diferentes, que sean capaz de clasificar las combinaciones de segmentos según el número que representan (1-9), para luego comparar sus tasas de acierto. Para ello, cada segmento se define con 1 ó 0 indicando si está activado o no, respectivamente, con lo cual se posee 7 variables binarias y una variable clase.



Se poseen cuatro datasets: dos datasets pequeños (ledSM.arff y ledSM10.arff) y dos grandes (ledXM.arff y ledXM10.arff). Cada par de datasets posee uno con ruido y otro sin él. Sin embargo, para el análisis de este problema se utilizarán sólo tres de ellos. En el apartado siguiente se explicará el motivo.

Procedimiento y resultados

Como ya se advirtió anteriormente, se usarán tres datasets de los existentes. El dataset principal con el cual se realizará el aprendizaje de la Red Bayesiana será, ledXM10.arff, que posee un 10% de ruido entre sus instancias. Luego, para probar la calidad del modelo generado, se utilizarán dos tipos de test, uno libre de ruido y otro con 10% de ruido, los cuales son la pareja de datasets pequeños nombrados en el apartado anterior.



Para realizar este estudio, se hará uso del entorno WEKA, que es un entorno gráfico para el análisis de datos. El objetivo principal de la práctica es comparar diferentes métodos de aprendizaje de Redes Bayesianas que provee WEKA. Éstos son *GenericSearch*, *HillClimber*, *K2*, *LAGDHillClimber*, *RepeatedHillClimber*, *SimulatedAnnealing*, *TabuSearch* y *TAN*. En todos ellos se utilizará el mismo estimador, **SimpleEstimator**.

Antes de comenzar es necesario realizar un pequeño preprocesamiento de los datos. Existen en el dataset unas 24 variables, de las cuales sólo se usarán las 7 primeras. Por tanto, bastará con eliminar dicho excedente de información y aprender el modelo con las demás variables.

Los algoritmos de aprendizaje que se van a utilizar poseen la siguiente descripción en los documentos de WEKA:

- **HillClimber →** Este método usa un algoritmo de escalada de la montaña, añadiendo, eliminando y revirtiendo arcos hasta obtener una solución. No influye el orden de entrada de variables en el resultado final y considera la direccionalidad de los arcos como una forma de diferenciar dos soluciones.

- **K2→** Este algoritmo es igual que el anterior, pero con la influencia del orden de entrada de variables.

- **LAGDHillClimber →** este método se denomina "escalada mirando hacia arriba". A diferencia de los dos anteriores, no busca el mejor candidato siguiente, sino que analiza los posibles pasos futuros para escoger por dónde "escalar". No le influye el orden de las variables y también considera la direccionalidad de los arcos.

- **RepeatedHillClimber →** Para llevar a cabo su aprendizaje, genera aleatoriamente un grafo, para luego modificarlo a través de un HillClimber. Realiza esta operación un número de veces para finalmente devolver el mejor resultado.

- **SimulatedAnnealing →** Utiliza el bien conocido método SimulatedAnnealing, en el que se va iterando un número definido de veces y evaluando los nodos y arcos del grafo, hasta obtener una solución. En cada iteración se reduce la probabilidad de modificar los arcos, simulando así el enfriamiento del metal cuando se forja.

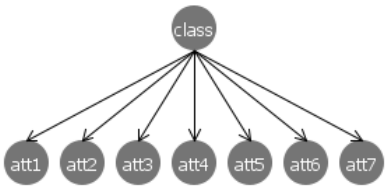
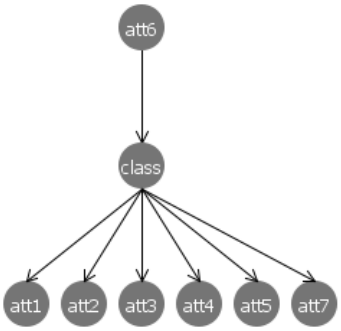
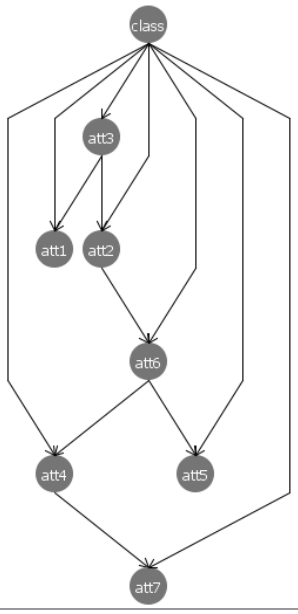
- **TabuSearch →** Es un método metaheurístico por búsqueda en los vecinos, en la que se van recorriendo éstos hasta establecer el arco más probable. Es un algoritmo parecido al HillClimber pero guardando las soluciones rechazadas anteriormente. De este modo no se repetirán las "malas decisiones" del pasado.

- **TAN →** Este método determina el peso máximo de cada nodo sobre la clase para generar el grafo. Con esto se obtiene una estructura de árbol que describe con relativa fiabilidad la influencia de los nodos sobre la clase.

Los resultados obtenidos en el entorno WEKA usando los algoritmos de aprendizaje descritos son los siguientes:

ACCURACY	Test sin ruido	Test + 10% ruido
HillClimber	1	0.735
K2	1	0.735
LAGDHillClimber	1	0.735
RepeatedHillClimber	1	0.735
SimulatedAnnealing	1	0.735
TabuSearch	1	0.735
TAN	1	0.746

Los grafos obtenidos de los métodos de aprendizaje son idénticos en varios casos. Sin embargo, en otros distan bastante del grafo común que se ha obtenido. Éstos son:

HillClimber, K2, RepeatedHillClimber, SimulatedAnnealing	*LAGDHillClimber, TabuSearch*	*TAN*
		

Conclusiones

Como podemos observar en los resultados obtenidos, las redes bayesianas poseen una tasa de acierto idéntica en las que se genera el mismo grafo. Además, en el grafo TabuSearch, que difiere en un nodo, también posee exactamente la misma tasa de acierto que el grafo más común. El grafo común describe el problema de la forma más sencilla posible, poniendo a la clase como consecuencia directa de sus atributos. Sin embargo, esto no signifca que sea el grafo más acertado, sino el más fácil de comprender.

El último grafo generado por el algoritmo de aprendizaje TAN es muy diferente y más complejo que los anteriores, e incluso supera la tasa de acierto de éstos. Sin embargo, puede no ser el grafo que se busca, ya que añade un ligero aumento de la tasa de acierto a costa de una complejidad mucho mayor. Debido a su complejidad, describe mejor la realidad, ya que indica que con conocer unos pocos atributos se puede predecir la clase sin necesidad de conocer el resto.