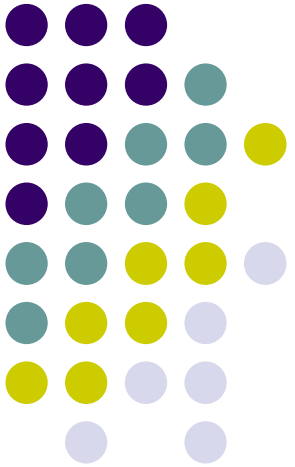
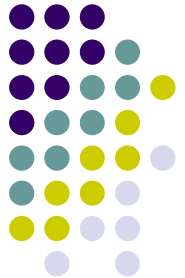
Posibles trabajos HIA

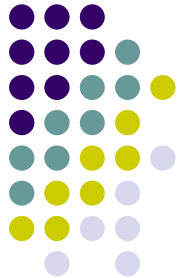


Posibles trabajos



- Comparar otras herramientas de Minería de Datos con Weka
- Estudiar la influencia del ruido en bagging y boosting
- Estudiar la influencia del parámetro de poda en J48 y como cambia el número de nodos y el porcentaje de aciertos
- Comparar la influencia del ruido en dos clasificadores distintos
- Estudiar como evoluciona el porcentaje de aciertos al aumentar la muestra (curva de aprendizaje)
- Influencia de atributos redundantes sobre distintos algoritmos
- Influencia de atributos irrelevantes sobre distintos algoritmos
- Comprobar si el cruce de un punto es mejor que el otro en un dominio (programación genética)

Repositorio UCI

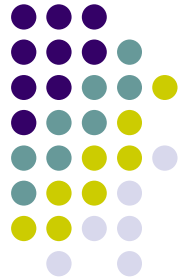


- UCI Machine Learning Repository
- Conjunto de dominios de prueba para minería de datos
- http://archive.ics.uci.edu/ml/
- Los dominios tienen distintas características: atributos numéricos o no, número de clases, número de atributos, número de datos. Los dos últimos pueden tener bastante influencia sobre la velocidad de los algoritmos de aprendizaje.

Comparar otras herramientas de Minería de Datos con Weka

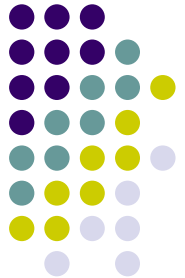
- Consultar “Introducción a la Minería de Datos”, José Hernández Orallo y otros. Prentice Hall. Página 611
- Posibles herramientas a usar: Yale (RapidMiner), DB Miner, R, … y librerias: MLC++, … (estas al menos son de dominio público). Se pueden buscar otras que no estén en esta lista
- El trabajo consistiría en probar la herramienta en algún dominio de minería de datos, mostrar que se puede hacer, proporcionar volcados de pantalla, y si es posible, comparar con lo que se puede hacer con Weka.

Estudiar la influencia del ruido en bagging y boosting



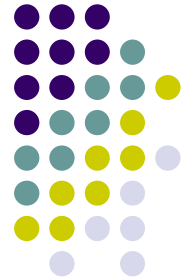
- Bagging y Boosting son meta-algoritmos que suelen mejorar el porcentaje de aciertos de otros clasificadores base. Boosting tiende a sobreadaptarse al ruido, Bagging no. El objetivo del trabajo es comprobarlo.
- Trabajo:
  - Elegir un dominio de UCI
  - Elegir un clasificador base (J48, Decision Stumps, …)
  - Hacer un pequeño programa que permita añadir ruido a la clase del dominio de UCI. Es decir, que con cierta probabilidad, para cada dato, se cambia la clase del dato en otra
  - Con los datos originales, probar Bagging y Boosting con 10, 20 y 30 iteraciones
  - Con los datos con un 10% de ruido (probabilidad de cambiar la clase de un dato = 0.1), probar Bagging y Boosting con 10, 20 y 30 iteraciones
  - ¿Cuál de los dos tiende a obtener peores resultados a medida que se incrementa el número de iteraciones?

Comparar la influencia del ruido en dos clasificadores distintos



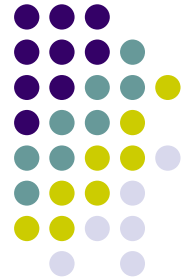
- Parecido al anterior. Elegir un dominio y dos algoritmos y comparar cual de ellos es más sensible al ruido. Probar con:
  - los datos originales
  - con los datos con 10% de ruido
  - con los datos con 20% de ruido
  - …
- Comprobar cual de los algoritmos tiende a empeorar su porcentaje de aciertos a medida que se incrementa el ruido

Estudiar la influencia del parametro de poda en J48 y como cambia el número de nodos y el porcentaje de aciertos



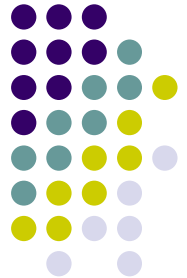
- J48 tiene un parámetro que controla el crecimiento del árbol (su complejidad), llamado confidence factor. Cuanto mas grande es, más tiende a podarse el árbol (menos nodos tiene)
- Elegir un dominio de UCI
- Estudiar el comportamiento de J48 para distintos valores del parámetro, mirando el número de nodos y el porcentaje de aciertos.
- ¿Existe algún valor del parámetro para el que los resultados son óptimos (en términos de porcentaje de aciertos?

Estudiar como evoluciona el porcentaje de aciertos al aumentar la muestra (curva de aprendizaje)



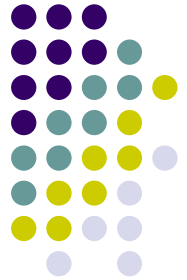
- Lo mejor es utilizar siempre todos los datos de entrenamiento disponibles
- En ocasiones hay demasiados y el algoritmo tarda bastante
- Es interesante ver cual es el tamaño ideal de la muestra de entrenamiento
- Elegir un dominio de UCI con bastantes datos y un algoritmo y obtener el porcentaje de aciertos para muestras aleatorias conteniendo el 10%, el 20%, …, el 90%, el 100% de los datos originales. A esto se le denomina “curva de aprendizaje”
- ¿Se observa que hay un momento en el que añadir más datos no aporta demasiado al incremento en el porcentaje de aciertos del clasificador?

Influencia de atributos redundantes sobre distintos algoritmos



- Los atributos redundantes contienen parecida información a otros ya presentes y tienden a reducir el porcentaje de aciertos de algunos algoritmos
- Elegir un algoritmo y un dominio con atributos numéricos
- Transformar el dominio, creando nuevos atributos redundantes:
  - Ej: simplemente copiar el mismo atributo dos o mas veces
  - Ej: tomar dos atributos aleatoriamente y añadir uno nuevo que sea la media de los dos
- Trabajo: introducir 1, 2, 3, …, 10 atributos redundantes y observar si el porcentaje de aciertos se deteriora o no

Influencia de atributos irrelevantes sobre distintos algoritmos



- Los atributos irrelevantes no aportan ninguna información para llevar a cabo la clasificación
- Elegir un algoritmo (ej: IB1, Naive Bayes, …) y un dominio
- Transformar el dominio creando nuevos atributos irrelevantes. Esto se puede hacer por ejemplo creando un atributo con valores aleatorios
- Trabajo: introducir 1, 2, 3, …, 10 atributos aleatorios y observar si el porcentaje de aciertos se deteriora o no

Comprobar si el cruce de un punto es mejor que el otro en un dominio

- Trabajo:
  - Lanzar 5 ejecuciones de PG con operador de cruce estándar
  - Lanzar 5 ejecuciones de PG con operador de cruce de 1 punto
  - Usar la función de Rastrigin
  - ¿Cuál de los dos operadores optiene mejores resultados? (calcular la media de las cinco ejecuciones)

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2).$$