Herramientas de la IA. Práctica de Minería de Datos. El Brain Computer Interface.

1. Introducción

Cada año, se celebra una competición sobre Brain-Computer Interfaces. En ellas, algunos laboratorios suministran datos EEG, a los cuales se les pueden aplicar algoritmos de aprendizaje automático. Nosotros trabajaremos con los datos de la III competición (año 2005) (http://www.bbci.de/competition/iii/#data_set_v), y en concreto, con el Data set V ‹mental imagery, multi-class›, cuya descripción se puede encontrar aquí:

http://www.bbci.de/competition/iii/desc_V.html

Por favor consultad ese enlace para entender los datos. En resumen, se dispone de datos de 3 sujetos. Cada sujeto realizó cuatro sesiones en momentos distintos. Las tres primeras sesiones de cada sujeto generaron los datos que se suministraron para la competición. Los datos de la última sesión eran desconocidos para los participantes de la competición y se utilizaron para comprobar cual era el mejor algoritmo durante la competición.

Los datos originales están en el dominio del tiempo, utilizan 32 electrodos y han sido muestreados con una frecuencia de 512Hz. Eso quiere decir que cada 1/512 segundos tenemos una muestra. En cada instante de tiempo, el sujeto puede estar imaginando una de entre tres posibilidades: mover la mano izquierda, la derecha, o pensando en una palabra. Los datos en el dominio del tiempo son complicados de utilizar. Nosotros trabajaremos con los datos denominados "precomputed features". Esos datos han sido generados pasándolos al dominio de la frecuencia mediante la transformada de Fourier. Además, sólo utilizan 8 canales (electrodos), los más relevantes de entre los 32. La banda de frecuencias que se considera varía entre 8Hz y 32Hz, con una resolución de 2Hz. Así pues, por cada canal, tendremos 12 atributos: frecuencia de 8-10Hz, 10Hz-12Hz, …, 30Hz-32Hz. Como hay 8 canales, y 8 bandas de frecuencia por canal, cada dato tendrá 12*8 = 97 atributos, mas la clase.

Paso a describir los ficheros ya en formato Weka con los que trabajaremos (se pueden encontrar en “Datos EEG para la practica de Weka (segunda practica)”). Para cada sujeto i, tenemos dos ficheros:

A) train_test1_subject**i**_psd010203.arff
B) test2_subject**i**_psd04.arff

El fichero A contiene los datos de las tres primeras sesiones, y el B contiene la cuarta sesión. El fichero A se llama train_test1 porque contiene los datos que vamos a usar para el entrenamiento y el test de nuestros algoritmos. Usaremos para entrenamiento las dos primeras sesiones (que constituyen el 66% de los datos, situados al comienzo del fichero) y para test la tercera sesión (que constituye el 34% de los datos, situados al final del fichero).

El fichero B (la cuarta sesión) sólo será utilizado al final de todo el proceso, para obtener una medida no sesgada del algoritmo que hayamos seleccionado.

Nota importante: aunque hay datos sobre tres sujetos, cada grupo de prácticas trabajará sólo con los datos de un sujeto. Si los NIA de los dos miembros del grupo son los dos pares o los dos impares, el grupo de prácticas trabajará con el sujeto número uno. En caso contrario, trabajará con el sujeto número dos. En esta práctica no utilizaremos al sujeto número tres.

2. Qué hay que hacer en la práctica

En esta práctica realizaremos los siguientes pasos (2 puntos).

1. Exploración de los datos (1.5 puntos)
2. Experimentación (0.25 puntos)
3. Test (0.25 puntos)
4. Discipulus (0.5 puntos adicionales sobre la nota final)

Exploración de los datos

Para el sujeto que le haya tocado al grupo (el uno o el dos), **y utilizando los ficheros A y el explorer,** comprobar las siguientes cuestiones, detallando los resultados obtenidos:

1. **Pruebas iniciales**:
   - Probar (al menos) dos algoritmos de clasificación lineal y dos no lineal. Ajustar parámetros de los algoritmos en la medida de lo posible. ¿Merece la pena utilizar clasificadores no lineales?. A partir de este punto, siempre se utilizará el mejor algoritmo, y con los mejores parámetros, obtenido en este paso.
   - ¿Se puede considerar que los resultados obtenidos son buenos?
2. **Clases desbalanceadas:**
   - ¿Están las clases desbalanceadas?
   - ¿Está muy desequilibrado el porcentaje de aciertos de cada una de las clases?
   - Intentar balancear los porcentajes de aciertos de las clases, ya sea replicando datos, ya sea utilizando algún meta-algoritmo apropiado. Hacer varias pruebas y representar los resultados en una tabla. ¿En algún caso mejorar en una clase implica empeorar en alguna otra?

Pruebas	% clase 1	% clase 2	% clase 3	% global
1				
2				
3				
4				

   - Una vez balanceados los porcentajes de aciertos por clase, ¿se consigue mantener el porcentaje de aciertos global del apartado 1, o se empeora?

3. **Selección de atributos**:

- Utilizar varias técnicas de selección de atributos. ¿Merece la pena en este problema utilizar Wrapper?
- Utilizando Ranker, realizar una gráfica que muestre como varía el porcentaje de aciertos en función del número de atributos considerados, entre 1 y 16 atributos (o sea, la gráfica tendrá 16 puntos).
- ¿Se consigue mejorar el porcentaje de aciertos disminuyendo el número de atributos?
- Los atributos más importantes ¿tienen algún sentido neurológico? (sensory homunculus)

4. **Visualización:**
   - ¿Se puede ver de manera gráfica la separación entre las clases utilizando alguna pareja de los mejores atributos seleccionados?

Experimentación:

Utilizando **el experimenter con los ficheros A**, hacer una experimentación con los mejores algoritmos encontrados en la fase de exploración. El objetivo es comprobar si alguno de los algoritmos es mejor de manera significativa que los demás. Por supuesto, se puede probar el mismo algoritmo cambiando sus parámetros, y especialmente, utilizándolo junto con algoritmos de selección de atributos (si es que esto funcionó bien en la fase de exploración)

Test:

Ahora por fin utilizaremos **los ficheros B** para hacer el test. Para ello volveremos a utilizar el explorer, de la siguiente manera. Para el mejor algoritmo encontrado en la fase de experimentación (experimenter), aprenderemos con el fichero A y haremos el test con el B. Obtendremos, por ejemplo, una tabla como la siguiente:

Algoritmo	**% aciertos en el experimenter**	**% aciertos con el conjunto B (el test)**
J48 con selección de atributos	70%	65%

Por fin podemos responder a la pregunta:

- Si en la competición hubieramos seleccionado nuestro algoritmo disponiendo sólo del conjunto A, ¿qué porcentaje de aciertos hubieramos obtenido en la competición (sobre el conjunto B)?.
- Podemos comparar con los resultados obtenidos en la competición (dataset V) (no es de esperar que consigamos superar los mejores resultados porque algunos de ellos son algoritmos especializados) : http://www.bbci.de/competition/iii/results/index.html#martigny

Discipulus (opcional):

Discipulus es una herramienta de Programación Genética que evoluciona código máquina, y que está especializado en problemas de clasificación y regresión. La versión de demostración y la documentación, se pueden encontrar aquí:

http://www.rmltech.com/DemoDown01.htm

Usar Discipulus considerando el mejor número de atributos según la tabla (ordenados según Ranker) construida en la fase de exploración (pero **nunca mas de 50**, pues ese es el número máximo de atributos que admite Discipulus). Además, Discípulus sólo permite aprendizaje **biclase**, por lo que nos vamos a limitar a las dos clases que le resulte mas complicado de separar al mejor algoritmo de Weka del apartado anterior (lo podemos ver en la matriz de confusión). Seguiremos los siguientes pasos:

1. Cogeremos el fichero A y crearemos otro A' donde sólo estén las dos clases mas complicadas (se puede hacer con un filtro de Weka)
2. Haremos lo mismo con el B (generaremos B')
3. Convertiremos A' y B' al formato de Discipulus (basicamente, hay que quitar la cabecera de Weka)
4. Crearemos el fichero A_1' con los primeros 66% de los datos de A' (equivale a las sesiones uno y dos). Utilizaremos ese fichero como **train**
5. Crearemos el fichero A_2' con los siguientes 34% de los datos de A' (equivale a la sesión tres). Utilizaremos ese fichero como **validation**
6. Utilizaremos el fichero B' como **application**

Ahora estamos en disposición de utilizar Discipulus. ¿Consigue igualar o superar el porcentaje de aciertos del mejor algoritmo elegido anteriormente (**para las dos clases consideradas**)?. ¿Lo consigue rápidamente?. Comentar cualquier otro detalle relevante.

3. Apéndice:

El software de Weka, tanto para Linux como para Windows, se puede obtener aquí:

http://www.cs.waikato.ac.nz/ml/weka/

Weka se autoinstala.

Aquí se puede encontrar un tutorial corto sobre Weka (con leer hasta el punto 3.2 es mas que suficiente):

http://www.dsic.upv.es/%7Ecferri/weka/CursDoctorat-weka.pdf

Mas información sobre Weka, aquí: http://www.dsic.upv.es/%7Ecferri/weka/