Trabajo Aprendizaje Activo 2019

El trabajo consiste en ensayar los métodos vistos en clase sobre el dataset Semeion Handwritten Digit, disponible en `https://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit`. El dataset contiene 1592 imágenes digitalizadas de tamaño 16x16, es decir, 256 dimensiones.

Hemos dividido los datos en tres ficheros para facilitar la tarea:

- `semeion_labeled.csv`: contiene el conjunto inicial de muestras etiquetadas.
- `semeion_unlabeled.csv`: contiene el conjunto inicial de muestras no etiquetadas.
- `semeion_test.csv`: contiene el conjunto de validación para evaluar los algoritmos. Este conjunto no debe usarse para entrenar los modelos de ninguna forma.

El clasificador a utilizar en los experimentos serán máquinas vectores soporte (SVM). En los ficheros CSV que os hemos dejado la primera columna es la etiqueta y las 256 restantes el dígito (matriz 16x16 como un vector de 256 dimensiones). Se pueden ver los dígitos haciendo un `reshape(16,16)`.

Pese a su nombre, el conjunto `semeion_unlabeled.csv` contiene etiquetas, pero a efectos prácticos no las debemos usar. Solo podremos utilizarlas cuando 'movamos' estas muestras al conjunto de datos etiquetados.

Hay dos tareas a realizar:

1. La primera tarea consiste en realizar un programa que permita combinar cualquiera de las estrategias de active learning (AL) y diversidad vistas en clase. Como criterios de AL dispondremos MS (margin sampling, o most uncertain), MCLU (multi-class label uncertainty), SSC (significance space construction) y nEQB (normalized entropy query bagging). Como criterios de diversidad implementaremos MAO (most ambiguous and orthogonal), lambda, y diversity by clustering.

   Nuestro programa tomará como entradas:

   - El conjunto reducido de datos etiquetados (L).
   - El conjunto grande de datos no etiquetados (U).
   - El conjunto de test, que no se podrá utilizar más que para validar los resultados, nunca para entrenar.
   - Un criterio AL a aplicar.
   - Un criterio de diversidad.

   El programa devolverá dos conjuntos de resultados de validación, uno con el resultado de aplicar un algoritmo completamente aleatorio de selección de muestras, y otro con el resultado de aplicar la combinación de los criterios AL y diversidad escogidos. En ambos resultados se mostrará el acierto en clasificación (accuracy o f1-score) versus el número de muestras empleado en el conjunto de entrenamiento. La figura 1 muestra un ejemplo del tipo de gráficas que hemos de generar.

   Las muestras se escogerán y pasarán del conjunto de no etiquetadas al de etiquetadas en grupos de 10. Es decir, en cada iteración se seleccionarán 10 muestras, bien de forma aleatoria, bien

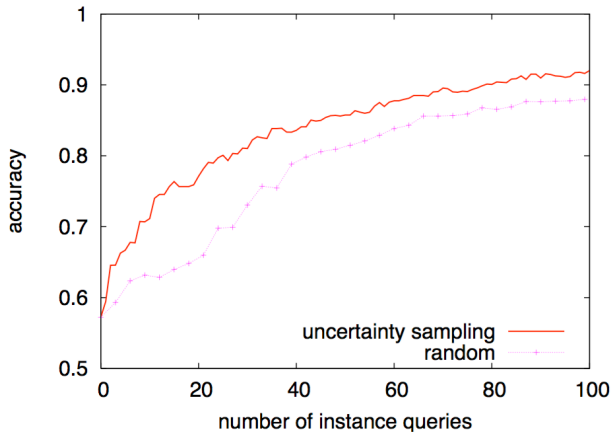

Figura 1: Ejemplo de comparación del algoritmo *uncertain sampling (o SVM margin sampling)* vs *random sampling*.

mediante el criterio AL + diversidad, que se moverán del conjunto de no etiquetadas al de etiquetadas.

2. Una vez programado lo anterior, la segunda parte consistirá en comparar los resultados de aplicar distintas combinaciones de métodos AL y diversidad. La comparación se realizará contra el muestreo aleatorio, el cual en teoría deberíamos superar (aunque quizás no siempre sea así).

   Dado que hay 4 métodos de AL y 3 de diversidad, debemos generar 12 gráficas comparativas. Cada una de ellas mostrará un combinación frente a muestreo aleatorio. Fijaros que no será necesario repetir el experimento de muestreo aleatorio más que una vez, será el mismo para todas las comparaciones.

   Para obtener datos estadísticamente signifcativos, cada experimento deberá repetirse entre 50 y 100 veces. Las gráficas mostrarán los resultados promedio junto con 2 desviaciones estándar.

Notas adicionales:

- El trabajo se puede realizar de manera individual o en grupos de dos personas como máximo.
- Se pueden realizar los programas en Python o R.
- Se deben presentar los programas desarrollados listos para funcionar, acompañados de una memoria analizando y discutiendo los resultados obtenidos. Una buena opción es presentar un notebook Jupyter.