# Introducción al Reconocimiento de Patrones 2018
# IIE, Facultad de Ingeniería, UdelaR

Práctico 1

Entrega: martes 4 de setiembre (eva); jueves 6 de setiembre en clase (versión papel)

## Árboles de decisión

### Ejercicio 1

Durante la construcción de un árbol de decisión en un problema de dos clases (w1 y w2), un nodo recibe los siguientes patrones de seis dimensiones binarias:

| w1 | w2 |
|---|---|
| 1 1 0 1 0 1 | 0 1 1 1 0 0 |
| 1 0 1 0 0 1 | 0 1 0 1 0 0 |
| 1 0 0 0 0 1 | 0 1 1 0 1 0 |
| 1 0 1 1 0 1 | 0 1 0 0 0 0 |
| 0 1 0 1 0 1 | 0 0 1 0 0 0 |
| 1 1 1 0 0 1 | 0 1 0 1 0 0 |
| 1 0 0 1 0 1 | 1 1 1 0 0 0 |
| 0 1 1 0 0 0 | 1 1 0 1 0 1 |

1. ¿Qué característica debería usarse para hacer la próxima ramificación considerando impureza basada en entropía?
2. Considere el uso de un test de hipótesis para detener el crecimiento. ¿Cuál es la hipótesis nula en este caso?
3. Calcule $\chi^2$ para la decisión de la parte 1. ¿Difiere significativamente de la hipótesis nula para un nivel de confianza de 0.01? ¿Debería detenerse el crecimiento?
4. Repita la parte 3 para un nivel de confianza de 0.05.

Los siguientes dos ejercicios tienen como objetivo familiarizarse y utilizar el árbol de decisión sklearn.tree.DecisionTreeClassifier provisto por módulo scikit-learn de PYTHON.

### Ejercicio 2

1. ¿Cuál es aproximadamente la profundidad de un árbol de decisión entrenado (sin restricciones) en un conjunto de entrenamiento de 1 millón de muestras?
2. Si se utiliza impureza de Gini ¿La impureza de un nodo hijo es mayor o menor que la de su nodo padre? ¿Es generalmente menor/mayor o siempre menor/mayor?
3. Si un árbol de decisión se sobre ajusta al conjunto de entrenamiento, ¿Es una buena idea intentar decrementar `max_depth`?
4. Si el tiempo de entrenamiento de un árbol es de 1 hora en un conjunto de 1 millon de muestras, ¿Cual es el tiempo estimado de entrenamiento sobre un confjunto de 10 millones de muestras?

5. Entrenamiento y ajuste de un árbol de decisión al dataset moon:

   a) Utilizar el comando `make_moonss(n_samples=10000, noise=0.4)` para generar el dataset.

   b) Dividir el dataset en un conjunto de entrenamiento y uno de test utilizando `train_test_split()`.

   c) Utilizar grid search y validación cruzada (con ayuda de la clase `GridSearchCV`) para encontrar un conjunto adecuado de parámetros para un árbol decisión. Sugerencia: Probar con distintos valores de `max_leaf_nodes`.

   d) Entrenar un árbol con los parámetros encontrados en el conjunto completo de entrenamiento. Obtener el desempeño en el conjutno de test (Se debería alcanzar un accuracy mayor a 84 %).

### Ejercicio 3

1. Continuando el ejercicio anterior, generar 1000 subconjuntos del conjutno de entrenamiento, cada uno con 100 muestras elegidas de forma aleatoria. Sugerencia: utilizar la clase `ShuffleSplit`

2. Entrenar un árbol de decisión en cada subconjutno de entrenamiento utilizando los parámetros encontrados en el ejercicio anterior. Evaluar el desempeño de cada árbol en el conjutno de test. Justificar los resultados obtenidos.

3. Para cada muestra del conjutno de test, obtenga su predicción a parir del voto mayoritario de los 1000 árboles.

4. Evaluar el desempeño en todo el conjunto de test y comparar el resultado con el obtenido en el ejercicio anterior. Justificar lo observado.

## Weka

### Ejercicio 4: Clasificación

El objetivo de este ejercicio es familiarizarse con la herramienta `Weka` (`www.cs.waikato.ac.nz/~ml/weka/`) desde su interfaz gráfica y aplicar las diferentes técnicas de clasificación vistas hasta el momento. Se trabaja sobre un subconjunto de dígitos de la base MNIST disponibles en la página del curso.

1. Analice la distribución de los datos en el espacio de características usando las herramientas de visualización disponibles (histogramas y gráficas de pares de características). ¿Considera que las características permiten una buena discriminación de los datos?

2. Utilice el clasificador J48 (`weka.classifiers.trees.J48`) con los parámetros por defecto y evalúe su desempeño usando validación cruzada en 10 particiones. Estudie los restantes parámetros del clasificador y realice modificaciones con el objetivo de disminuir el sobreajuste a los datos de entrenamiento.

3. Aplique el clasificador k-NN (`weka.classifiers.lazy.IBk` ) y evalúe su desempeño usando validación cruzada en 10 particiones. Encuentre el k óptimo utilizando alguna de las posibilidades que ofrece Weka para automatizar dicho parámetro.

4. Compare los resultados obtenidos para ambos clasificadores indicando ventajas y desventajas comparativas.

## Ejercicio 5: Selección de características

En este ejercicio estudiaremos el impacto que tiene el conjunto de características en la clasificación y aplicaremos diferentes técnicas de selección de características provistas por `Weka`. El conjunto de datos son 2400 electrocardiogramas que se encuentran disponibles en la página del curso y las características son directamente las muestras de la forma de onda.

1. Estime el desempeño del clasificador C4.5 (con los parámetros por defecto), usando validación cruzada con 10 particiones (10-fold CV). Luego utilice el filtro `Resample`[1] para sortear una muestra con el 10 % de los datos originales y vuelva a hacer la estimación de desempeño. Observe que el número de patrones $n$ y la cantidad de características $d$ no cumplen con el criterio de buena práctica $n/d > 10c$. Indique las razones que considera explican los resultados obtenidos[2].

2. Selección individual de características.

   *a*) Utilice Análisis de Componentes Principales (PCA) como técnica de selección de características. Varíe el porcentaje de varianza acumulada y registre el número de características que se obtienen. Estime el desempeño del clasificador luego de aplicar PCA sobre los datos (mediante 10-fold CV). Para ello puede utilizar el clasificador compuesto `AttributeSelectedClassifier`[3] que aplica una técnica de selección de características antes de entrenar el clasificador.

   *b*) Aplique el criterio de ganancia de información `InfoGainAttributeEval`[4] y razón de ganancia de información `GainRatioAttributeEval`[5] para ordenar las características. Compare el ordenamiento y explique en qué difieren ambos criterios de evaluación.

   *c*) ¿Qué desventaja puede tener la selección individual de características ? ¿Cómo se compara PCA con los métodos basados en ganancia de información en este caso?

   *d*) *Opcional:* Explore la herramienta `Experimenter` de Weka para facilitar las pruebas. Se recomienda consultar el tutorial incluído en la distribución del programa.

3. Selección de subgrupos de características.

   *a*) Considere el criterio de evaluación de subgrupos `CfsSubsetEval`[6] basado en correlación. Explique cuál es la heurística usada para la evaluación. Aplique este criterio junto a la búsqueda `BestFirst`[7] para seleccionar un conjunto reducido de características. ¿Qué ventajas tiene BestFirst sobre la búsqueda secuencial clásica? Estime el desempeño del clasificador para este conjunto de características.

   *b*) Realice la selección de características utilizando el enfoque `wrapper`. Utilice el criterio de evaluación `WrapperSubsetEval`[8] y seleccione un conjunto de características para cada clasificador. Estime el desempeño de cada clasificador con esta técnica de selección. ¿Qué ventajas y desventajas tiene este enfoque?

[1] `weka.filters.supervised.instance.Resample`

[2] Puede resultar de utilidad consultar la sección 7.1 del libro "Data Mining: Practical machine learning tools and techniques", Ian H. Witten and Eibe Frank (2005).

[3] `weka.classifiers.meta.AttributeSelectedClassifier`

[4] `weka.attributeSelection.InfoGainAttributeEval`

[5] `weka.attributeSelection.GainRatioAttributeEval`

[6] `weka.attributeSelection.CfsSubsetEval`

[7] `weka.attributeSelection.BestFirst`

[8] `weka.attributeSelection.WrapperSubsetEval`