TEMA 7: MÉTODOS DE EVALUACIÓN DE MODELOS

Metodología de evaluación

- Resubstitution error (accuracy)
  - Error en los datos de entrenamiento
- No es una buena estimación de la calidad de un modelo en datos futuros
  - Es muy optimista

Metodología de evaluación

- Para realizar la validación de los sistemas de predicción se crean conjuntos de ejemplos de entrenamiento y de test
  - Se asume que ambos conjuntos representan bien el problema
- El conjunto de entrenamiento se utiliza para realizar el aprendizaje del modelo
- Evaluación del modelo
  - Obtener el rendimiento con los ejemplos de entrenamiento
  - Obtener el rendimiento con los ejemplos de test
    - Estimación de la calidad ante nuevas situaciones
  - Diferencia entre entrenamiento y test
    - Capacidad de generalización

Metodología de evaluación

- Situación ideal
  - Muestra de datos grande para entrenar el modelo
  - Otra muestra de datos grande e independiente para evaluar el modelo
  - Normalmente
    - Cuanto más grande sea el conjunto de entrenamiento mejor será el modelo generado
    - Cuanto más grande sea el conjunto de test mejor será la estimación de la calidad del modelo

¿Qué hacer si la cantidad de datos es limitada?

Metodología de evaluación

- Existen distintas técnicas para obtener los conjuntos de entrenamiento y de test, entre ellas
  - Hold-out
  - Validación cruzada
  - Leave-one-out
  - Boostraping

Hold-out

- Consiste en dividir la BD en dos conjuntos independientes
  - Conjunto de entrenamiento (CE)
  - Conjunto de test (CT)
- El tamaño del CE normalmente es mayor que el del CT
  - 2/3 vs. 1/3
  - 4/5 vs. 1/5
- Los ejemplos del CE suelen obtenerse mediante muestreo sin reemplazamiento de la BD inicial
  - El CT está formado por los elementos no incluidos en el CE

Hold-out

- Problema
  - Los ejemplos puede que no sean representativos
    - Una clase puede no estar presente en uno de los conjuntos
- Solución
  - Estratificación: técnica que asegura que las distribución de las clases en los conjuntos de entrenamiento y de test sea similar a la del conjunto original
- Hold-out con estratificación
  - Problema: puede haber sesgos (bias) ya que se utiliza muestreo para generar los subconjuntos
    - Hay menos datos para aprender el modelo: puede que no sea tan bueno como si se utilizaran todos los datos

Hold-out con repetición

- La estimación hold-out puede ser más fiable si se repite el proceso varias veces
  - Con diferentes subconjuntos de entrenamiento y test
  - El error (accuracy) de todas las iteraciones se promedia para calcular la estimación del error (accuracy) del modelo
- Problema
  - Los diferentes conjuntos de test puede tener overlap
    - Algunos ejemplos pueden no aparecer nunca para aprender el modelo
    - ¿Se puede prevenir el overlap?

Validación cruzada

- La validación cruzada previene el overlap
  - Se dividen todos los ejemplos de la base de datos en k subconjuntos
    - Los subconjuntos deben tener el mismo número de ejemplos (o similar)
    - Los ejemplos se asignan a los subconjuntos aleatoriamente
    - Cada ejemplo se asigna a un solo subconjunto
      - Subconjuntos de intersección vacía
    - Los subconjuntos se crean habitualmente utilizando estratificación

Validación cruzada de k particiones

- Para mejorar la validación cruzada
  - Validación cruzada de k particiones con repetición

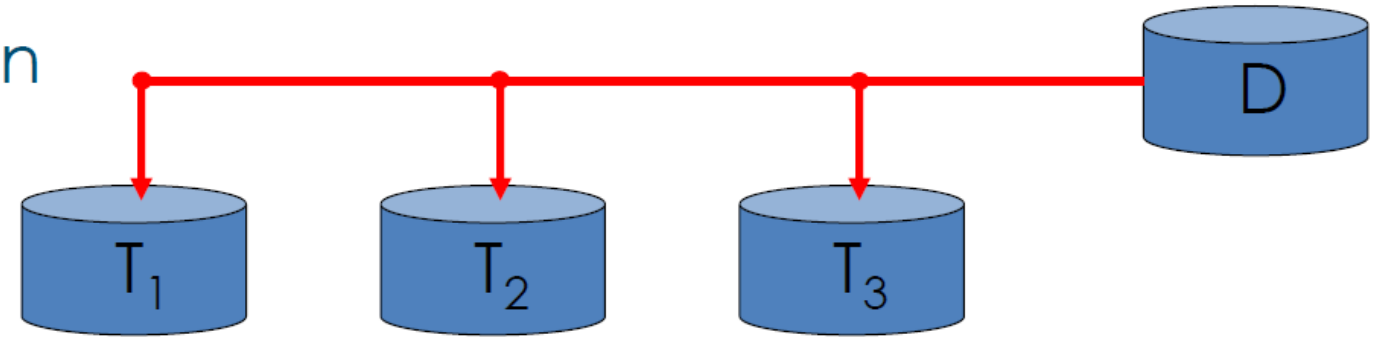
Validación cruzada de k particiones

Proceso de validación cruzada

- Generación de los conjuntos de entrenamiento y test
  - Como conjunto de entrenamiento se toman (k-1) subconjuntos
  - Como conjunto de test se toma el subconjunto restante
- Aprendizaje y evaluación
  - Se realiza el aprendizaje del modelo con el conjunto de entrenamiento
  - Se obtiene el rendimiento en ambos conjuntos
- Realizar el proceso de aprendizaje y evaluación *k* veces
  - Iteración *i*
    - El subconjunto *i* se escoge como conjunto de test
    - La unión de los *(k-1)* restantes como conjunto de entrenamiento
  - Devolver como rendimiento la media aritmética obtenida en las *k* iteraciones
- Valores típicos de *k*
  - 5
  - 10

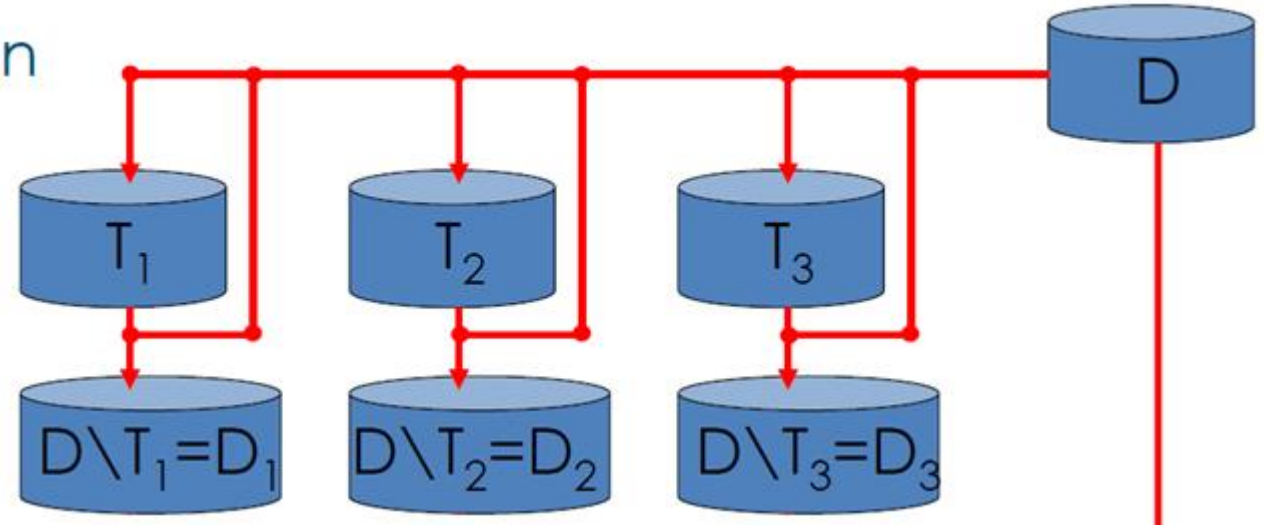
Validación cruzada de k particiones

- Partition



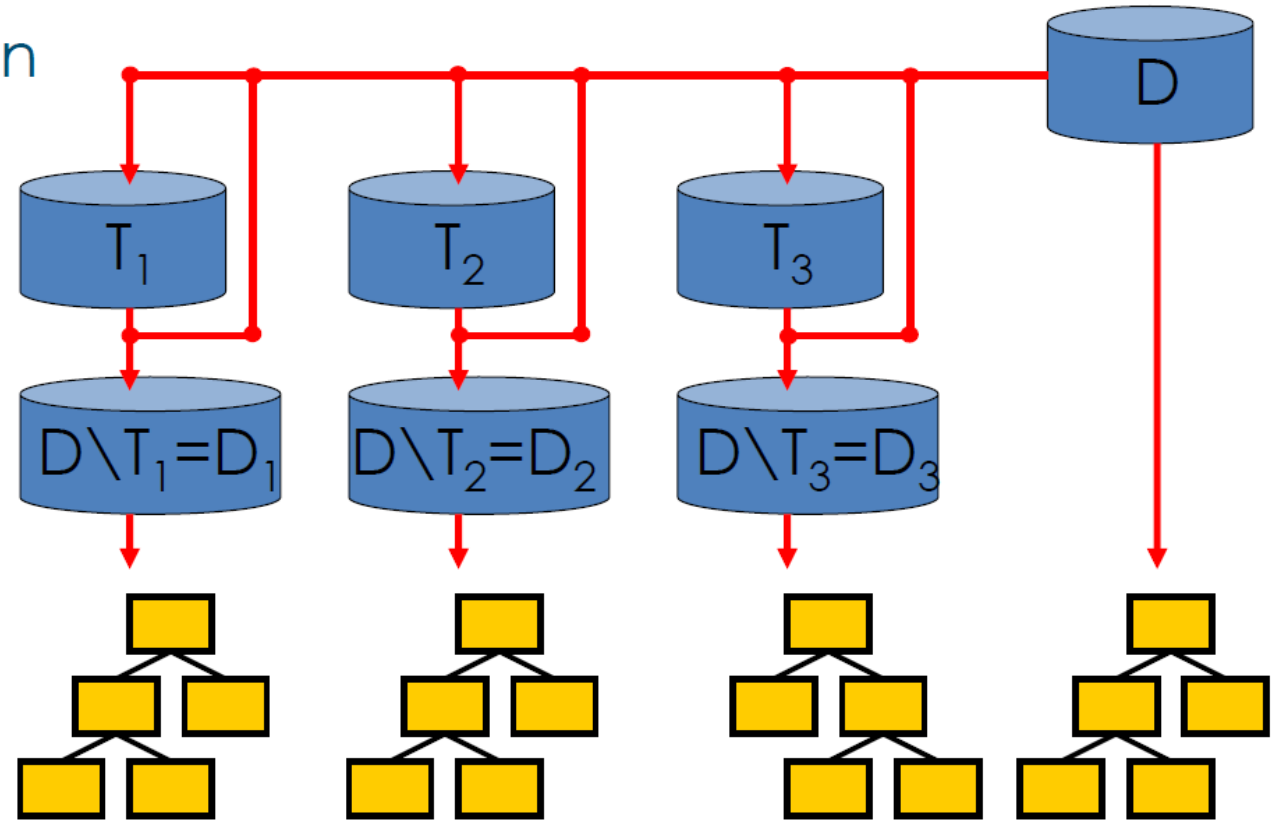
Validación cruzada de k particiones

- Partition
- Train



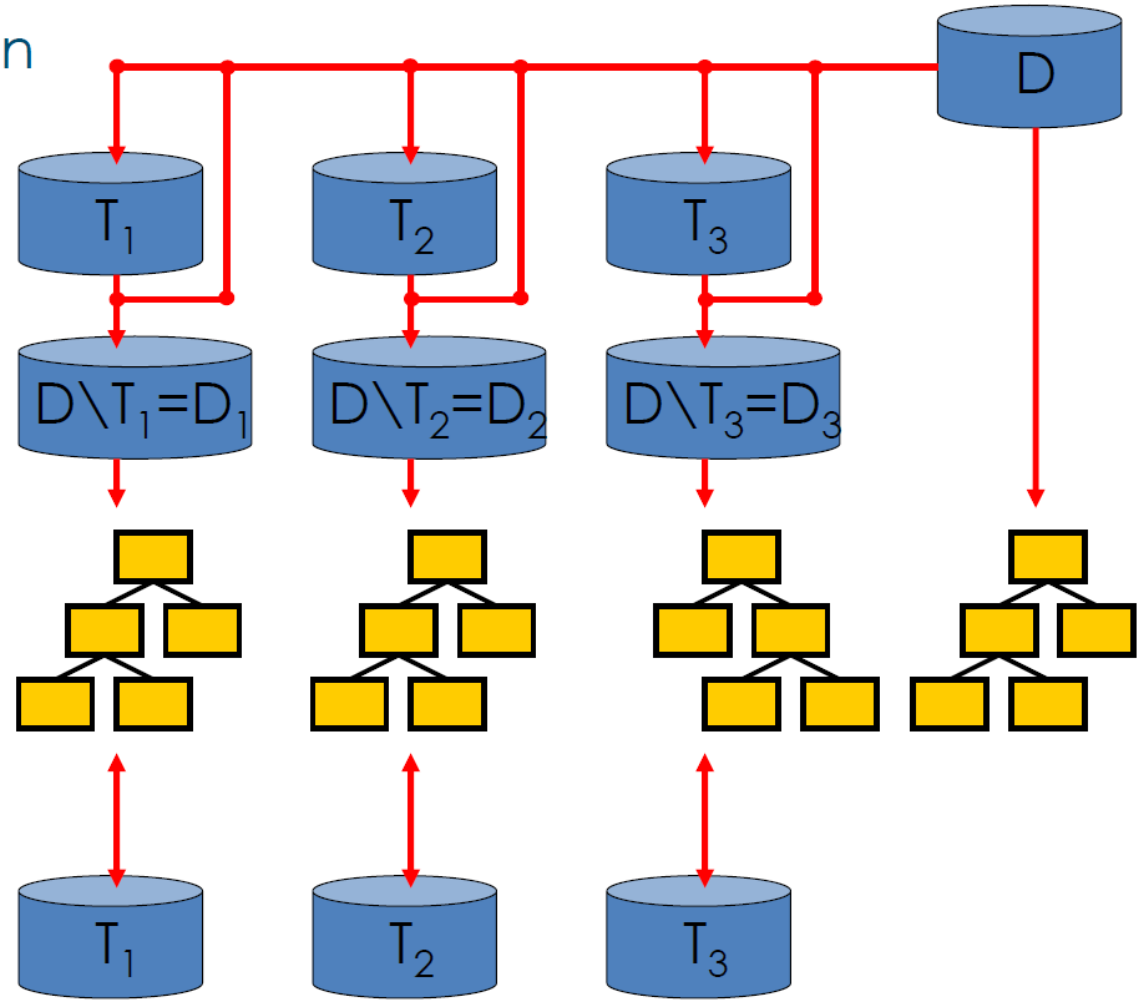
Validación cruzada de k particiones

- Partition
- Train



Validación cruzada de k particiones

- Partition
- Train
- Test



Leave-one-out

- Es un caso especial de validación cruzada
  - *k* es igual al número de ejemplos
- Ventajas
  - El proceso es determinista (no hay aleatoriedad)
  - Se utiliza el máximo posible de datos para la inducción del clasificador
- Desventajas
  - Alto coste computacional
  - No es posible aplicar la estratificación

- Se utiliza en BBDD muy pequeñas

Boostraping

- Está basado en el proceso de muestreo con reemplazamiento
  - A partir de una BD con N ejemplos se obtiene un CE con N ejemplos
    - Un ejemplo puede escogerse más de una vez para entrenar el modelo
  - Como CT se utilizan los ejemplos de BD que no hayan sido elegidos en CE
- ¿Cuántos ejemplos habrá en CT? ¿Qué porcentaje respecto a N?
  - La probabilidad de que se elija un ejemplo es $\frac{1}{N}$
    - Por tanto, la probabilidad de que no sea elegido es $1 - \frac{1}{N}$
  - Se hacen N extracciones. Por tanto la probabilidad de que un ejemplo no sea elegido es

$$\left(1 - \frac{1}{N}\right)^N \approx e^{-1} = 0.368$$

  - El CE tendrá aproximadamente el 63.2% de los ejemplos y el CT el 36.8%
- Esta técnica se conoce como 0.632 bootstrap

Boostraping

- El error sobre el CT suele ser bastante pesimista
  - Se entrena el modelo solamente con un 63.2% de los ejemplos
- Por tanto, se combina con el error de entrenamiento

$$error_{CT} = 0.632 * error_{CT} + 0.368 * error_{CE}$$

  - El error de entrenamiento tiene menor peso que el de test

- Para mejorar el proceso
  - Boostraping con repetición