Ricardo Aler Mur

EVALUACIÓN DE TÉCNICAS DE APRENDIZAJE-I

En esta clase se habla de que no es suficiente con la construcción del modelo sino que es necesario cuantificar sus prestaciones futuras (porcentaje de aciertos, error, etc.).

- El método de evaluación más simple consiste en dividir el conjunto de datos disponibles en entrenamiento y test
- En un problema de clasificación biclase, es posible estimar la incertidumbre alrededor del error en el conjunto de test, mediante la distribución binomial, y esta incertidumbre depende del tamaño de dicho conjunto.
- El conjunto de test tiene que ser representativo del problema. En problemas de muestra desbalanceada es improbable que una selección aleatoria de datos consiga un conjunto representativo, por lo que se utilizan particiones estratificadas.
- El método entrenamiento/test tiene el problema de tener una alta variabilidad si el tamaño del conjunto es pequeño, por lo que se recomienda el uso de la validación cruzada que es una especie de entrenamiento/test repetido en el que las particiones de test nunca solapan.
- El criterio para saber si un modelo tiene unas prestaciones adecuadas es que supere a lo que se podría obtener con una clasificación aleatoria, y en el caso de problemas de

muestra desbalanceada, lo que se podría obtener si se clasificara siempre con la clase mayoritaria.

- En el caso de los problemas de regresión, existen varias maneras de medir las prestaciones de un modelo: error cuadrático, error absoluto, error relativo, etc.
- Además de estimar las prestaciones de un modelo, es interesante poder comparar de manera estadística las prestaciones de dos o más modelos. Se recalcará aquí, que aunque es importante que la prestación media de un modelo sea superior a la del otro, es también importante que la varianza no sea grande. De otra manera, no se podría afirmar rigurosamente que la diferencia en prestaciones no sea debida al azar.

EVALUACIÓN DEL CONOCIMIENTO MINADO

METODOLOGÍA ANÁLISIS DE DATOS

- Recopilación de los datos (tabla datos x atributos)
- Preproceso:
  - **De los datos:**
    - **Normalización**
    - **Para KNN: Edición de Wilson, RNN, Condensación (CNN)**
  - **De los atributos:**
    - **Selección de atributos:**
      - **Ranking: chi-squared, information gain, linear correlation, …**
      - **Subset selection: CFS y WRAPPER**
    - **Transformación / Generación de atributos:**
      - **No supervisada: PCA, random projections, autoencoders**
      - **Supervisada: mediante redes de neuronas**
- **GENERACIÓN DE MODELOS / AJUSTE DE PARÁMETROS / SELECCIÓN DE MODELO**
  - **Clasificación: árboles de decisión, reglas, KNN, prototipos (LVQ)**
  - **Regresión: modelos lineales (lm), árboles de regresión, árboles de modelos, KNN**
- Evaluación: **validación cruzada, matriz de confusión**
- Despliegue y uso del modelo

¿Porqué evaluar modelos?

- Del mismo conjunto de datos disponibles necesitamos obtener:
  - Un modelo
  - Su error futuro
- Fase de evaluación:
  - Queremos conocer el error del modelo en el futuro
  - Queremos comparar el error de dos o mas modelos en el futuro (ej: en una competición) => contraste de hipótesis (diferencias estadísticamente significativas)
- Fase de selección de modelo:
  - Queremos elegir el mejor tipo de algoritmo (ej: árboles de decisión vs. KNN)
  - Queremos elegir el mejor parámetro (o conjunto de parámetros) para un algoritmo

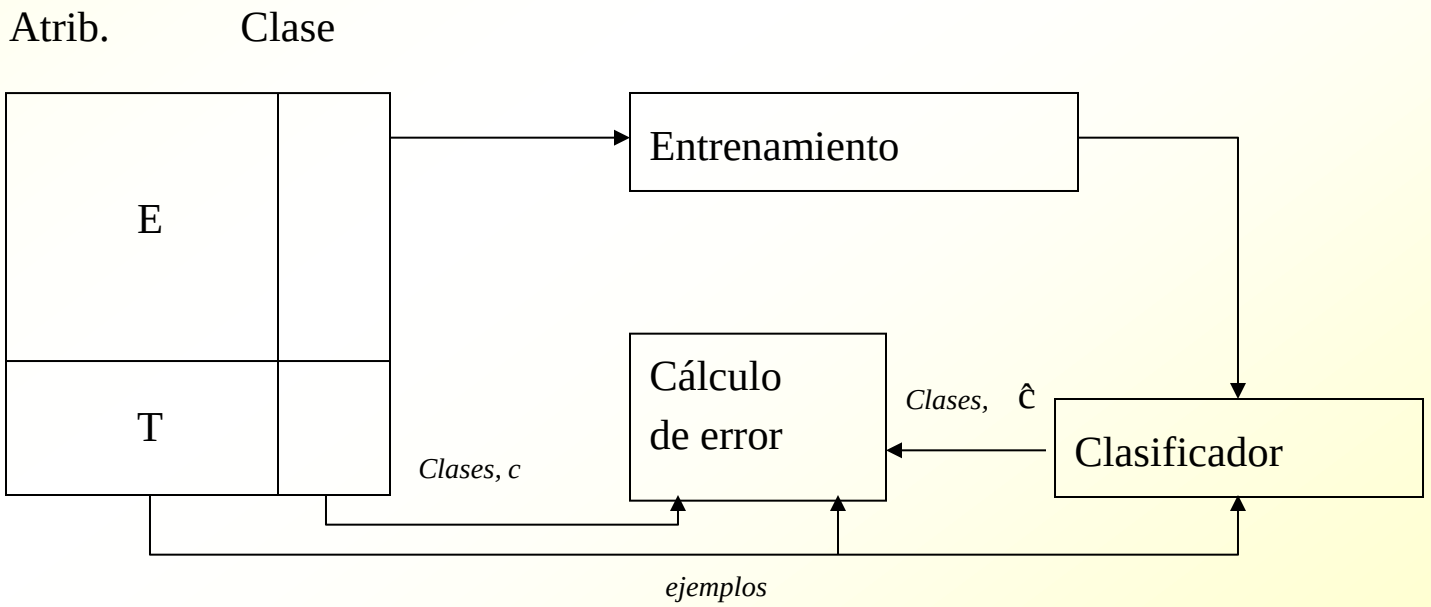
¿Porqué evaluar modelos?

- Notar que cuando hay pocos datos, un algoritmo puede errar bastante en cuanto a la frontera de separación correcta (en clasificación)
- Notar que un algoritmo lo puede hacer mal incluso con muchos datos:
  - Ej: Naive Bayes en Checkerboard con 500 o 1000 datos
- Notar que en el checkerboard los decision trees lo hacen bien porque dividen el espacio con fronteras paralelas
- Notar que en parity, el que las SVMs lo hagan bien depende mucho de ajustar correctamente el parámetro gamma

http://home.comcast.net/~tom.fawcett/public_html/ML-gallery/pages/

Evaluación de modelos

- El conjunto de ejemplos se divide en dos partes: entrenamiento (E) y test (T)
- Se aplica la técnica (p.e. árboles de decisión) al conjunto de entrenamiento, generando un clasificador
- se estima el error (o tasa de aciertos) que el clasificador comete en el conjunto de test



Evaluación

Evaluación de modelos

- Del mismo conjunto de datos disponibles necesitamos obtener:
  - Un modelo
  - Su precisión futura
- Dividir los datos disponibles en entrenamiento/train (2/3) y test (1/3)
  - Test es independiente de train y representativo puesto que train y test vienen de la misma distribución subyacente
- Condiciones que debe cumplir el conjunto de evaluación (test):
  - Independiente del conjunto usado para construir el modelo
  - Pero representativo del conjunto de entrenamiento
  - Lo mas grande que podamos para que sea preciso

Sobre el tamaño del conjunto de test

- La división train 2/3 test 1/3 es algo arbitraria, pero común
- Tenemos un dilema:
  - Cuanto mas grande sea el conjunto de test, mas preciso será el cómputo del error de test
  - Pero tendremos menos datos en train para construir el modelo
- Opciones:
  - Construir el modelo con muchos datos (train) pero tener poca seguridad sobre si el modelo es bueno o malo
  - Construir el modelo con pocos datos (será malo), pero tendremos gran seguridad sobre que el modelo es, efectivamente, malo

Incertidumbre sobre el error en test

- ¿Podemos saber hasta que punto el error que computemos con el conjunto de test es incierto?
- Si, porque el error sobre el conjunto de test es una media y la incertidumbre podemos determinarla estimando la varianza o desviación típica
- Supongamos que tenemos N datos en el conjunto de test y que $\{x_i \mid i = 1:N\}$ representa los aciertos del modelo para cada dato ($x_i == 1$ si acierto, $x_i == 0$ si fallo)
- La tasa de aciertos **para este conjunto de test** podremos **estimarla** como:

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Incertidumbre sobre el error en test

- La tasa de aciertos **para este conjunto de test** T podremos estimarla como:

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Pero el conjunto de test T es una muestra y podría haber sido distinto. Es decir, los x_i podrían ser distintos y eso daría lugar a una estimación distinta
- Supongamos que la tasa de aciertos real (pero desconocida) del modelo es f
- Eso quiere decir que las x_i son variables aleatorias que pueden valer 0 (fallo) o 1 (acierto) con probabilidad f

Incertidumbre sobre el error en test

- Eso quiere decir que las x_i son variables aleatorias que pueden valer 0 (fallo) o 1 (acierto) con probabilidad f
- Obviamente, cuanto mayor sea N, mejor será la estimación de f:

$$N \to \infty \Rightarrow \hat{f} \to f$$

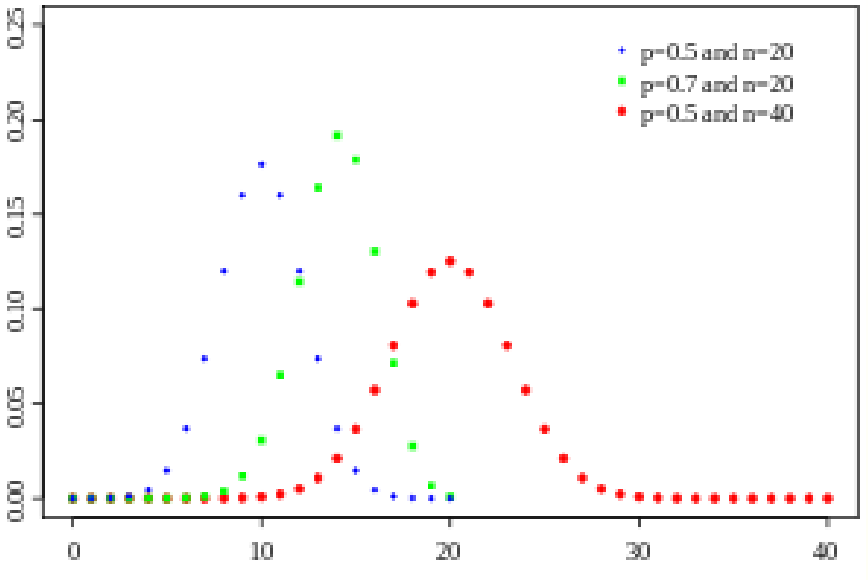
- Las x_i no son distintas a tirar una moneda trucada N veces
- En distintas secuencias de N tiradas, el número de caras (aciertos) será distinto
- Puesto que la estimación de f depende del conjunto de test T que se use, sería más correcto escribirla como

$$\hat{f}_T = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad x_i \in T$$

Incertidumbre sobre el error en test

- El número de caras (aciertos) en N tiradas (N datos de test) sigue una distribución binomial de media N*f y varianza N*f*(1-f)
- Es decir, que ocurran n aciertos de N posibles es:

$$p\left(\sum_{i=1}^{N} x_i = n\right) = \left(\frac{N!}{n!(N-n)!}\right) f^n (1-f)^{N-n}$$



Incertidumbre sobre el error en test

- Si en lugar de contar el número de aciertos contamos la proporción de aciertos

$$\hat{f}_T = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad x_i \in T$$

- Entonces sigue una distribución de media f y varianza f*(1-f)/N
- Bajo la suposición de que N*f>=0.5 y N*(1-f)>=0.5 y N>=30, podemos aproximar una binomial por una normal con las mismas medias y varianzas: N(f, f*(1-f)/N)
- Recordemos que lo queremos saber es, si estimamos f mediante $\hat{f}_T$ ¿hasta que punto podemos equivocarnos en el valor real de f?

Incertidumbre sobre el error en test

- Vamos a trabajar directamente con la binomial en lugar de aproximarla por una normal (R permite calcular la binomial)
- Supongamos que:
  - Tenemos 1000 datos, y usamos 666 para entrenar y 334 para test
  - El porcentaje de aciertos en test es del 80%. O sea 267 aciertos: $\hat{f}_T = 0.8$
  - ¿Cuál es el intérvalo de confianza alrededor de $\hat{f}_T = 0.8$ en el que está el verdadero valor f?
  - O sea, que la probabilidad de encontrar f en ese intérvalo sea un valor alto (95%)

$$p(|\ f - \hat{f}_T\ |<= z) = 0.95$$

Incertidumbre sobre el error en test

```
install.packages("binom")
library(binom)
binom.confint(267, 334)[5,]
method x n mean lower upper
5 exact 267 334 0.7994012 0.7523801 0.841022
```

Es decir, el valor puede ser un 5% mayor o menor al observado (0.8)

Incertidumbre sobre el error en test

- Si usaramos 9/10 para entrenar y 1/10 para hacer el test:
- binom.confint(0.8*100, 100)[5,]
- method x n mean lower upper
- 5 exact 80 100 0.8 0.7081573 0.8733444

- El valor real podría ser entre un 10% menor y un 7% mayor

Incertidumbre sobre el error en test

- Con 1000 datos para hacer el test:

```
binom.confint(0.8*1000, 1000, tol = 1e-8)[5,]
method x n mean lower upper
5 exact 800 1000 0.8 0.7738406 0.8243794
```

3% por debajo y 2% por encima

Incertidumbre sobre el error en test

- Con 10000 datos para hacer el test

```
binom.confint(0.8*10000, 10000, tol = 1e-8)[5,]
method x n mean lower upper
5 exact 8000 10000 0.8 0.7920233 0.8078016
```

1% por debajo y 1% por encima

Problemas train/test

- Problema 1: es posible que por azar, los datos de entrenamiento y/o test estén sesgados, sobre todo si hay pocos datos.
  - Dicho de otra manera, que el conjunto de test no sea representativo del de entrenamiento (es fácil que ocurra si test es pequeño)
- Problema 2: los resultados que proporcionamos no son repetibles (¿y si otro investigador divide los datos en train y test de otra manera?)

Entrenamiento y test repetido

- Consiste en partir el conjunto de datos totales múltiples veces y calcular el porcentaje de aciertos medio
- La idea es que los sesgos de unas y otras particiones se cancelen
- **Método:**
  - **Repetir múltiples veces:**
    1. **Desordenar el conjunto de datos total aleatoriamente**
    2. **Escoger los primeros 2/3 para entrenamiento y construir el modelo con ellos**
    3. **Escoger los últimos 1/3 para el test y estimar el porcentaje de aciertos**
  - **Calcular el porcentaje de aciertos medio**

Particiones estratificadas

- Para que el test sea **mas representativo**, es conveniente que las particiones sean **estratificadas**
- La proporción entre las clases que existe en el conjunto de datos original, se intenta mantener en los conjuntos de train y test
  - Ejemplo: si en el conjunto original un 65% de los datos pertenecen a la clase positiva, la estratificación intentará que esa proporción se mantenga en train y test

Entrenamiento y test repetido

- Problema: las distintas particiones de test no son independientes (pueden solaparse unas con otras por casualidad)
- Explicación: en el caso extremo, si por casualidad todas las particiones de test contuvieran exactamente los mismos datos, el repetir muchas veces el cálculo en test no nos aportaría ninguna información adicional
- El caso extremo no ocurre, pero siempre hay algún solape entre las particiones de test
- Lo ideal es que las particiones de test no solapen

Validación cruzada (crossvalidation)

- Solución: dividir **varias veces** el mismo conjunto de datos en entrenamiento y test y calcular la media. Así, las particiones de test no solaparán.
- Se divide el conjunto de datos original en k partes. Con k=3 tenemos los subconjuntos A, B, y C.
- Tres iteraciones:
  - Aprender con A, B y test con C (T1 = % aciertos con C)
  - Aprender con A, C y test con B (T2 = % aciertos con B)
  - Aprender con B, C y test con A (T3 = % aciertos con A)
  - % aciertos esperado T = (T1+T2+T3)/3
- El clasificador final CF se construye **con todos los datos (los tres conjuntos** A, B y C). Se supone que T es una estimación del porcentaje de aciertos de CF
- Se suele utilizar k=10

Validación cruzada (crossvalidation)

- El método de validación cruzada utiliza muy bien los datos al calcular el porcentaje de aciertos esperado, porque todos ellos se utilizan para test (en alguna partición).
- De hecho, todos los datos figuran como entrenamiento o test en alguno de los ciclos de validación cruzada.
- Las particiones de test de los distintos ciclos son independientes (no solapan)
- Nota: a cada una de las k divisiones de los datos de entrenamiento se la denomina *fold*

Leave-one-out

- Es una validación cruzada con k = número de datos de entrenamiento
- Si hay N datos de entrenamiento, repetir k=n veces:
  - Reservar el dato número N para test
  - Entrenar con los N-1 datos restantes
  - Hacer el test con el dato N (el resultado sólo puede ser acierto o fallo)
- El porcentaje de aciertos esperado será:
  - (aciertos/N)*100
- Es preciso porque se usan casi todos los datos para entrenar, y a la vez todos los datos figuran como test en alguno de los ciclos
- Pero es tremendamente costoso en tiempo (hay que lanzar el algoritmo de aprendizaje N veces)

Criterios básicos para evaluar

- En problemas de clasificación, si tenemos 2 clases (o M), el porcentaje de aciertos a superar es el 50% (o 100*1/M).
  - De otra manera, sería mejor tirar una moneda (azar) que utilizar el clasificador para predecir

- En problemas de clasificación, si tenemos una clase con muchos más datos que otra, el porcentaje de aciertos a superar es el porcentaje de datos de la clase mayoritaria
  - Ej: Sean dos clases (+ y -). Hay 90 datos + y 10 -. Un clasificador que prediga siempre + (independientemente de los atributos), ya acertará en un 90%. Hay que hacerlo mejor que eso.

Evaluación de predicción numérica

- Valores reales: $\{a_1, \ldots, a_n\}$, valores predichos: $\{p_1, \ldots, p_n\}$

$$\text{MSE}: \frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}; \quad \text{RMSE} = \sqrt{\text{MSE}}$$

$$\text{RSE}: \frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}; \quad \text{RRSE} = \sqrt{\text{RSE}}; \quad \bar{a} = \frac{a_1 + \ldots + a_n}{n}$$

$$\text{MAE}: \frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{n}; \quad \text{RAE} = \frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|a_1 - \bar{a}| + \ldots + |a_n - \bar{a}|}$$

Mean Squared Error, Root-Mean Squared Error

Relative Squared Error

Mean Absolute Error, Root Absolute Error

Evaluación de predicción numérica

- Valores reales: $\{a_1, \ldots, a_n\}$, valores predichos: $\{p_1, \ldots, p_n\}$

$$\text{correlacion}: \ \rho_{PA} = \frac{S_{PA}}{S_P S_A} \in [-1, 1];$$

$$S_{PA} = \frac{\sum_{i=1}^{n} (p_i - \bar{p})(a_i - \bar{a})}{n-1}$$

$$S_P = \sqrt{\frac{\sum_{i=1}^{n} (p_i - \bar{p})^2}{n-1}}; \quad S_A = \sqrt{\frac{\sum_{i=1}^{n} (a_i - \bar{a})^2}{n-1}}$$

Performance measure	Formula
mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
mean absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
relative squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$, where $\bar{a}=\frac{1}{n}\sum_i a_i$
root relative squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
relative absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
correlation coefficient	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_p=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, and $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$

* p are predicted values and a are actual values.

Comparación de dos modelos

- Ejemplo, sobre un conjunto de datos E, J48 puede obtener un 90% de aciertos (en 10-fold crossvalidation) y NN 92%. ?Podemos asegurar que NN es mejor que J48 en este dominio?
- No necesariamente, si usaramos otro conjunto de datos E', puede que J48 sacara 92% Y NN 89%
- Existe variabilidad, debido a que no disponemos del conjunto total de datos (que puede ser infinito), sino muestras finitas y pequeñas E,E', …
- Necesitamos saber como de grande es esa variabilidad (varianza)

Comparación de dos modelos

- Necesitamos saber como de grande es esa variabilidad (varianza)
- Hacemos la validación cruzada muchas veces

- Para cada algoritmo, repetir 10 veces
  - Desordenar los datos de entrenamiento
  - Calcular P_i de validación cruzada (de por ejemplo, 10 folds)
- Realizar test estadístico (t-test) para ver si las diferencias son significativas. Si la varianza es pequeña es más fácil que la diferencia sea significativa

Comparación de dos modelos

- ?Cuál de estos dos casos es mas probable que corresponda a una diferencia significativa?. Hacemos para A y B 10 crossvalidations de 10 folds cada una. (media, desviación)
  - A = (90%, 8%), B=(94%, 7%)
  - A = (90%, 0.001%), B=(91%, 0.002%)

Comparación de dos modelos

- Idea importante: el que la diferencia sea significativa depende mas de que la varianza sea pequeña que de que las medias estén muy separadas.