

Minería de Datos

Tema 10.- Predicción numérica o regresión

Julia Flores

Departamento de Informática
 Universidad de Castilla-La Mancha
 EPSA

Contenidos

1. Definición del problema
2. Validación de algoritmos de regresión
3. Técnicas de clasificación válidas para regresión
4. Análisis de regresión
5. Árboles de regresión
6. Árboles de modelos
7. Sistemas basados en reglas difusas (repass)

Predicción numérica o *regresión*

- **Objetivo:** predecir el valor numérico para una variable a partir de los valores conocidos para otras.
- La definición del problema es por tanto parecido al de la clasificación, tendremos **variables predictoras** o atributos y una variable "clase" o de **regresión** que en este caso es numérica.
- Ahora, la mayoría (o todas) las **variables predictoras son numéricas**.
- Ejemplos:
 - ▶ ¿qué consumo tendrá un coche en autovía en función de su peso, cilindrada, potencia, ...?
 - ▶ ¿Qué número de artículos tendremos en el próximo pedido?
 - ▶ ¿Cuántos meses necesitaremos para desarrollar un proyecto software?
 - ▶ ¿Cuál es la probabilidad de que un cliente determinado sea receptivo a un envío publicitario?
 - ▶ ¿Cuántos enfermos tendremos en urgencias la próxima nochebuena?
 - ▶ ¿Qué nota sacaré en Minería de Datos sabiendo la de Ing. del Conocimiento?

Validación en predicción numérica

- Todas las técnicas de validación estudiadas en clasificación son válidas para predicción numérica.
- La diferencia está en que ahora debemos medir el error de otra forma.
- Debemos medir el error cometido al aproximar un conjunto de valores $\{v_1, \dots, v_n\}$ por su estimación $\{\hat{v}_1, \dots, \hat{v}_n\}$.

Error Cuadrático Medio (ECM) $ECM = \frac{\sum_{i=1}^n (v_i - \hat{v}_i)^2}{n}$	ECM estandarizado (ECME) $ECME = \sqrt{\frac{\sum_{i=1}^n (v_i - \hat{v}_i)^2}{n}}$
Error Medio Absoluto (EMA) $EMA = \frac{\sum_{i=1}^n v_i - \hat{v}_i }{n}$	Error Absoluto Relativo (EAR) $EAR = \frac{\sum_{i=1}^n v_i - \hat{v}_i }{\sum_{i=1}^n v_i - \bar{v} }$
Coeficiente de correlación $r_{v\hat{v}} = \frac{\sum_{i=1}^n (v_i - \bar{v})(\hat{v}_i - \bar{\hat{v}})}{(n-1)\sigma_v\sigma_{\hat{v}}}$	

Métodos basados en ejemplos/instancias:

Al usar kNN , si los k vecinos más próximos $\{e_1, \dots, e_k\}$ tienen valores $\{v_1, \dots, v_k\}$ para la variable objetivo, entonces el valor a devolver para el objeto siendo analizado e' sería:

$$v = \begin{cases} \frac{\sum_{i=1}^k v_k}{k} & \text{si todos cuentan igual} \\ \frac{\sum_{i=1}^k w_k \cdot v_k}{\sum_{i=1}^k w_k} & \text{si se hace un voto ponderado} \end{cases}$$

por ejemplo con $w_i = \frac{1}{d(e_i, e')}$.

Métodos basados en redes neuronales:

- ▶ La capa de salida sería una única neurona.
- ▶ Como los pesos se adaptan en función del error cometido, es suficiente con medir de forma adecuada el error.

- Resultados sobre la base de datos CPU (sin tener en cuenta la variable nominal fabricante) y usando 5-cv:

▶ 1-NN:

EMA: 15.1292

EAR: 17.3137%

▶ 5-NN:

EMA: 17.5316

EAR: 20.0630%

▶ 5-NN (pesando por 1/distancia):

EMA: 13.1456

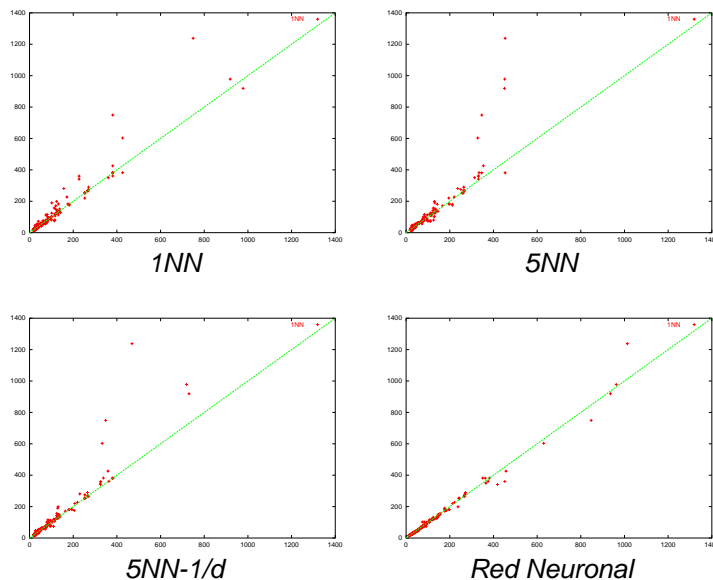
EAR: 15.0437%

▶ Red neuronal (valores por defecto excepto 100 epoch):

EMA: 6.1005

EAR: 6.9813%

Visualizando el error cometido:



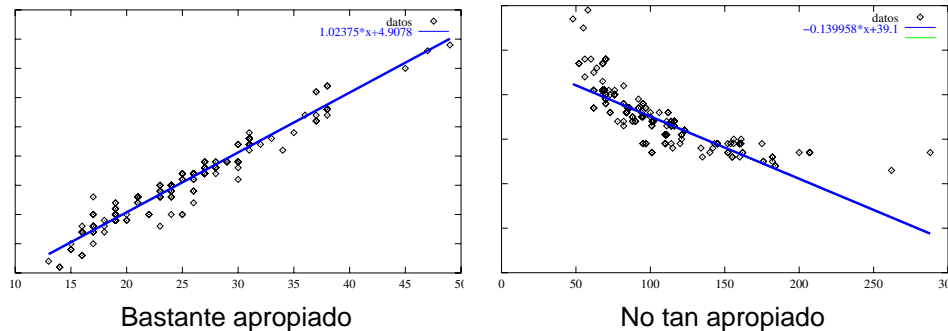
- Sin duda el método más utilizado para realizar la tarea de predicción numérica.
- La idea es estimar la variable objetivo (y) como una ecuación que contiene como incógnitas al resto de las variables (x_1, \dots, x_n) .
- El modelo más simple es la **Regresión lineal** que reducida a una sola variable predictora tiene la forma:

$$y = a + b \cdot x$$

- Estos coeficientes pueden obtenerse fácilmente mediante el método de los mínimos cuadrados:

$$b = \frac{\sum_{i=1}^s (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^s (x_i - \bar{x})^2} \quad a = \bar{y} - b \cdot \bar{x}$$

Regresión lineal: ejemplo



- Para estimar curvas es necesario usar otra regresión, por ejemplo, **regresión exponencial**: $y = a \cdot e^{bx}$
- ¿cómo estimamos ahora a y b ? Tomando logaritmos:

$$\ln(y) = \ln(a \cdot e^{bx}) \Rightarrow \ln(y) = \ln(a) + \ln(e^{bx}) \Rightarrow y^* = a^* + bx$$

- Es decir, tenemos un problema de regresión lineal entre $y^* = \ln(y)$ y x . Una vez estimados a^* y b podemos calcular $a = e^{a^*}$

M. Julia Flores - EPSA/UCLM

Minería de Datos - p.9/15

Regresión lineal múltiple

- Cuando hay más de una variable predictora, la ecuación de predicción se transforma en:

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Este problema es conocido como **Regresión Lineal Múltiple**.

- La estimación de los coeficientes es algo más compleja y requiere operar con matrices (y sus inversas).
- Por ejemplo, continuando con nuestro fichero CPU.arff tendríamos:

$$\begin{aligned} \text{class} = & 0.066 * \text{MYCT} + 0.0143 * \text{MMIN} \\ & + 0.0066 * \text{MMAX} + 0.4945 * \text{CACH} \\ & - 0.1723 * \text{CHMIN} + 1.2012 * \text{CHMAX} - 66.4814 \end{aligned}$$

$$\begin{aligned} \text{EAM} = & 34.31 \\ \text{EAR} = & 39.26\% \end{aligned}$$

- Existen técnicas más complejas de regresión que evidentemente aproximan mucho mejor los datos de entrada.

M. Julia Flores - EPSA/UCLM

Minería de Datos - p.10/15

Árboles de regresión

- La estructura es idéntica a un árbol de decisión, pero ahora **las hojas contienen un valor numérico**.
- Ese valor numérico se calcula como la media del valor para la variable *clase* de todos los ejemplos que han llegado a esa hoja durante el proceso de construcción del árbol.
- La evaluación de un nuevo ejemplo es idéntico a los árboles de decisión.
- Durante el proceso de predicción es posible usar un **suavizado** de los valores del ejemplo a tratar, con el fin de salvar las posibles discontinuidades presentes en los datos (en WEKA lo controla el parámetro *useUnsmoothed*).
- El **criterio de selección de una variable** en la construcción del árbol está basado en una reducción del error esperado: reducción de la desviación/varianza en la variable objetivo:

$$SDR = sd(T) - \sum_i \frac{|T_i|}{|T|} \cdot sd(T_i)$$

SDR = *Standard Deviation Reduction*

- Finalmente, el árbol es podado para evitar el sobreajuste.

M. Julia Flores - EPSA/UCLM

Minería de Datos - p.11/15

Árboles de regresión: ejemplo con CPU

```

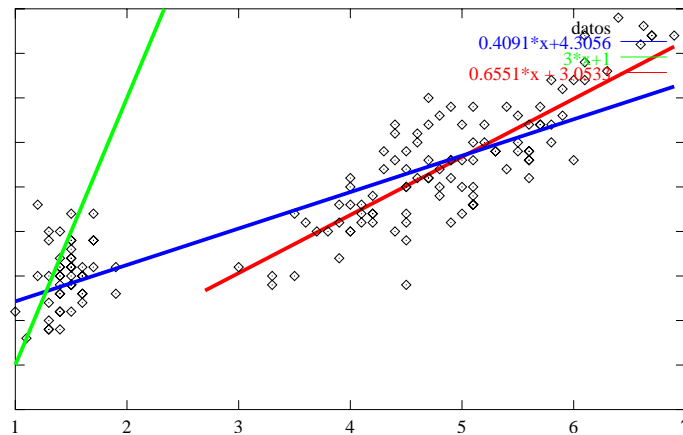
MMAX <= 14000 :
|  CACH <= 8.5 :
|  |  MMAX <= 6100 : LM1 (75/5.4%)
|  |  MMAX > 6100 :
|  |  |  MYCT <= 83.5 :
|  |  |  |  MMAX <= 10000 : LM2 (8/4.5%)
|  |  |  |  MMAX > 10000 : LM3 (3/3.78%)
|  |  |  |  MYCT > 83.5 : LM4 (22/6.73%)
|  |  CHMIN <= 7 :
|  |  |  MYCT <= 95 : LM5 (7/13.3%)
|  |  |  MYCT > 95 :
|  |  |  |  CACH <= 28 : LM6 (12/7.85%)
|  |  |  |  CACH > 28 : LM7 (6/7.27%)
|  |  CHMIN > 7 : LM8 (8/45%)
|  MMAX > 14000 :
|  |  MMAX <= 22500 :
|  |  |  CACH <= 27 :
|  |  |  |  CHMIN <= 5 : LM9 (14/6.92%)
|  |  |  |  CHMIN > 5 : LM10 (5/12.8%)
|  |  |  CACH > 27 : LM11 (18/25.1%)
|  |  |  |  |  |  LM1: class = 24.3
|  |  |  |  |  |  LM2: class = 45.3
|  |  |  |  |  |  LM3: class = 62.7
|  |  |  |  |  |  LM4: class = 39.6
|  |  |  |  |  |  LM5: class = 63.6
|  |  |  |  |  |  LM6: class = 43.9
|  |  |  |  |  |  LM7: class = 55
|  |  |  |  |  |  LM8: class = 104
|  |  |  |  |  |  LM9: class = 75.4
|  |  |  |  |  |  LM10: class = 94.4
|  |  |  |  |  |  LM11: class = 128
|  |  |  |  |  |  LM12: class = 262
|  |  |  |  |  |  LM13: class = 174
|  |  |  |  |  |  LM14: class = 355
|  |  |  |  |  |  LM15: class = 971
|  |  |  |  |  |  EAM 18.1275
|  |  |  |  |  |  EAR 20.7449%
|  |  |  |  |  |  WEKA: sin suavizar
|  |  |  |  |  |  poda=1
    
```

M. Julia Flores - EPSA/UCLM

Minería de Datos - p.12/15

Árboles de Modelos

- Son árboles de regresión en los que la poda se realiza en mayor medida y en las hojas en lugar de un valor numérico contienen una ecuación de regresión local a esa partición del espacio.



Si $x \leq 3$ entonces modelo1 (—) sino modelo2 (—)

Árboles de modelos: ejemplo con CPU

```
MMAX <= 14000 :
| CACH <= 8.5 : LM1 (108/3.99%)
| CACH > 8.5 : LM2 (33/3.89%)
MMAX > 14000 :
| MMAX <= 22500 : LM3 (37/4.73%)
| MMAX > 22500 : LM4 (31/69.2%)

LM1: class = 15.9 - 0.00453MYCT + 0.00327MMAX
LM2: class = -0.609 + 0.004MMAX + 0.59CACH + 1.57CHMIN
LM3: class = 1.64 - 0.0266MYCT + 0.00485MMIN + 0.00346MMAX + 0.627CACH
      + 1.43CHMIN + 0.127CHMAX
LM4: class = -350 - 0.843MYCT + 0.0183MMAX + 1.62CACH

EMR      10.0402
EAR 11.4899 %

WEKA: sin suavizar
poda=0
```

Conjuntos difusos

- La idea es aprender un SBR difusas a partir de los datos, y aplicarlo para predecir el valor de los nuevos casos.
- Podemos aplicar algoritmos como Wang y Mendel, Bardossy y Duckstein, evolutivos, ...
- P.e. para el segundo caso visto de regresión (en el que iba mejor la exponencial), si usamos 5 etiquetas difusas [NM,NS,ZR,PS,PM] para ambas variables y el algoritmo de Wang y Mendel (que no es el mejor) obtenemos el siguiente conjunto de reglas:

PS \Rightarrow NS
NM \Rightarrow PM
ZR \Rightarrow NS
NS \Rightarrow NS
PM \Rightarrow NM

- En este problema se obtienen las siguientes correlaciones (valor real, valor predicho):
Regresión lineal: 0.77
Conjunto difuso: 0.86