

Machine Learning

Lo studio sistematico di algoritmi che migliorano con l'esperienza

Lo scopo è usare le funzioni approporziate per creare un modello che risolva un task

Task

- classificazione
- regressione
- clustering

tipi di learning

supervised	predictive	descriptive
unsupervised		

tipi di modello

- geometrici
- probabilistici
- logici

kernel trick

$$x_1 = (x_1^2, y_1^2) \quad x_2 = (x_2^2, y_2^2)$$

$$x_1 \cdot x_2 = x_1^2 x_2^2 + y_1^2 y_2^2$$

prodotti inner prodotti superiori non ricalcolate in vettore

Contingency Table

actual	predicted	
	+	-
+	TP	FN
-	FP	TN

Accuracy

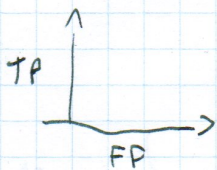
$$P(\hat{c}(x) = c(x))$$

Recall TP/POS

Specificity TN/NEG

Precision $TP / (TP + FP)$

Grafi di copertura (ROC)



Avg recall $(\text{recall} + \text{specificity}) / 2$ confidenza $x \in$
o classe k

Scoring function $\hat{S}(x) \in \{s_1(x), \dots, s_K(x)\}$ $\hat{S} = (s_1(x), \dots, s_K(x))$

Margin $+1$ per ogni x ok in \hat{S}
 -1 altrimenti

Loss function $\text{maggiori margini su una loss}$
 $\text{margini negativi: trans loss più alta}$

Ranking error rate

$1 \cdot \text{errori ranking (neg. > pos)} + \frac{1}{2} \cdot \text{errori margini}$
(pos. = neg)

Class probability estimator ranking classifier con
probabilità

Inquadro errore

$$\frac{1}{2} \|\hat{\eta}(x) - I_C(x)\|_2^2 = \frac{1}{2} \sum (\hat{\eta}(x) - I_C(x))^2$$

$I_C(x)$ = vettore con 1 in posizione $C(x)$ e 0 altrimenti

Mean SE (MSE)

SE medio in tutte le istanze

Probabilità empiriche calcolate da un training set n

esempi clone m

esempi etichettati $|S|$

Correzione di Laplace

m -estimate (Laplace non uniforme)

$$\frac{n_i + 1}{|S| + k}$$

no zero!

$$\frac{n_i + m \cdot \pi_i}{|S| + m}$$

uniforme

probabilità desiderate

prende esempi

Multi class classification

2i combinazioni classificatori binari

one vs rest (ordinato / non ordinato)

one vs one (simmetrico / asimmetrico)

output = matrice

$$\begin{matrix} C_1 & \frac{1}{2} & \frac{1}{2} & \text{ecc} \\ & -1 & \dots & \dots \\ C_2 & 0 & \dots & \dots \\ C_3 & \dots & \dots & \dots \end{matrix}$$

regole = reg. classificatori - w

classificazione = distanza minima = $\# \text{Min} \sum (1 - c_i w_i)^2$

Regresione

$\hat{f} : X \rightarrow \mathbb{R}$

regressor

Imparare un regressor dai esempi

Per evitare ^{overfitting} ~~overfitting~~ bisogna stimare molti
meno parametri che dati

Un modello troppo semplice causa alto bias ma diminuisce
la varianza, uno troppo complesso fa l'opposto

Perdita su $x = \text{Bias}(\hat{f}(x)) + \text{var}(\hat{f}(x))$

Concept Learning - indurre una funzione generale da
dati specifici (es. prezzo auto da k tratti)

Ipotesi congiunzione di limiti negli attributi (val, ?, \emptyset)
_{+ generale}

Dati 2 ipotesi h_1, h_2 $h_1 \geq h_2$ se $\forall x$ $h_2(x) = 1 \rightarrow h_1(x) = 1$

Find -5

spazio ip (molto grande!)

$h =$ ip. più specifica in H'

\forall es. positivo

re constraint $a_i \in h$ vero in x ok

altrimenti $a_i =$ ^{constraint} ~~esempi~~ più generale vero in x .

Problemi!

- non so se ho trovato l'unica ip.

- non rilevo inconsistenze (guarda solo es. positivi)

- perché preferire la ip. più specifica?

Version Space Spazio delle ipotesi valide

List then Eliminate (la ricetta)

Lista tutte le ipotesi valide

\forall es. $x \in$ Training Set

elimina le ip. inconsistenti

Ottieni il Version space

General boundary

Insieme di ip. più generali

es. + forma riduce 5

es - "aumenta 6

Specific boundary

Insieme di ip. più specifiche

Condolote Elimination

* $d \in \text{Training set}$

re $d +$

L toglia da G le ipr. inconsistenti con d

L $\forall \text{ ipr. } \in S$: ipr. inconsistenti con d

L toglia ipr. da S

L aggiungi ogni generalizzazione minima h di ipr

L toglia ogni s da S : $s, i \in S$ $s > i$

h ok per d
| $\exists i \in G: i > h$

re $d -$

L toglia da S le ipr. inconsistenti con d

L $\forall \text{ ipr. } \in G$: ipr. inconsistenti con d

L toglia ~~ipr.~~ da G

L aggiungi ogni specializzazione minima h di ipr

L toglia ogni g da G : $g, i \in G$ $g < i$

h ok per d
| $\exists s \in S: s < h$

Se esempi sufficienti con verze altrimenti l'ordine degli esempi potrebbe cambiare il risultato

Il bias free learning fa schifo!

Non potendo fare osservazioni non è in grado di lavorare su istanze mai viste

Prior Induttivo

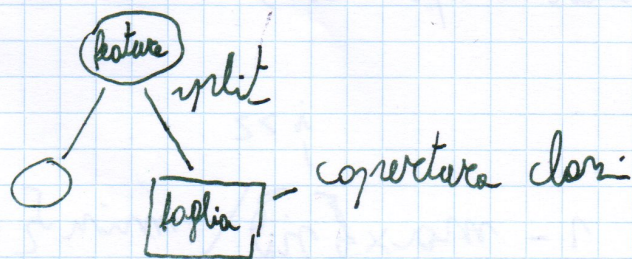
Insieme minimo di assunzioni: $\forall c, \forall$ concetto c e

training set $D_c \forall x \in X \quad B \wedge D_c \wedge x \vdash L(x, D_c)$

implica algorithms learning

Nota ipotesi = curva ROC

Alberi di decisione



Se le foglie non hanno label allora è solo un feature tree
Gli alberi di decisione rappresentano i concetti in
forma normale ~~espressiva~~ disgiuntiva $(A \vee B) \vee (C \wedge D) \dots$
Possono soffrire di overfitting se $\#$ foglie = $\#$ esempi

Generare un feature tree

Se D ^{esempi} omogeneo label (0)

altrimenti

split (0)

* subset r se $r \neq \emptyset$

costruirli in un albero (ricorsione)

altrimenti

ritorna una foglia

combina le foglie in un albero

Purezza di uno split

$$p_i = \frac{n^+}{n^+ + n^-} \text{ (purezza empirica)}$$

Impurità

$i > 2$
L'indice minoritario $1 - \max\{p_i\} \setminus \min\{p_i, 1 - p_i\}$

L'indice di Gini (errore se il labeling fosse casuale)

$$2p(1-p) \setminus \sum_i p_i(1-p_i)$$

L'entropia (vedi avanti)

Impurità insieme di foglie indipendenti

$$\sum \frac{|D_j|}{|D|} \text{imp}(D_j)$$

Entropia

confusione in bit

$$n \text{ possibilità} = \log_2 n \text{ bit}$$

Se si verifica un evento E con probabilità p , E
ci ha guadagnato I informazioni

$$I(E) = \log_2 \frac{1}{p} = -\log_2 p \quad \left(\begin{array}{l} p(E) \approx 0 \quad I \text{ alta} \\ p(E) \approx 1 \quad I \text{ bassa} \end{array} \right)$$

L'informazione media per un insieme di eventi

$$i = \sum p_i \cdot \log_2 \frac{1}{p_i} = H(E) \Rightarrow \text{entropia}$$

Best split impurità minima

Ranking trees

Gli alberi dividono lo spazio delle istanze in segmenti

I segmenti possono essere ordinati per g_i

L'ordinamento crea una curva ROC convessa

Le classi vengono etichettate in base al costo

$$c = CFN / CFP$$

$$cbr = P/N$$

Threshold = punto ROC con curvatura $1/c \cdot cbr$

Preming semplificare il modello se possibile

Per semplificare usando c si usa $\boxed{n^+ n^-}$ foglia

$$n^- \frac{n^-}{n^+} \frac{1}{c} = \frac{n^-}{n^+} \frac{CFP}{CFN}$$

Ogni es. positivo viene considerato negativo

(FN) e viceversa (FP)

Se $\frac{n^-}{n^+} \cdot \frac{CFP}{CFN} > 1$ allora si predice - perché il

costo d'errore sarebbe troppo alto nell'altro caso

Come scegliere dove fare preming

Reduced error

* l'albero controlla l'accuratezza su tutto lo spazio D_m

che copre

Se accuratezza \leq majority class in D_m allora

semplificare il decision tree con una foglia c

* Errore di Generalizzazione

Calcolato sul training set durante la costruzione dell'albero

* foglia si aggiunge una generalità k

vengono create solo foglie che riducono l'errore di

$k+1$ o più

training error

$$e'(T) = e(T) + \underbrace{N \cdot 0,5}_{\text{* foglie}} - \text{generalità}$$

√ Gini

Il contrario di Gini ed Entropia NON è influenzato dalla distribuzione fra classi

Overfitting

Sequasi la distribuzione più/meno omogenea

Se si aumenta di c si può evitare di usare per determinate le classi

Tempi di training più lunghi!

2 Riduzione di varianza

$$\text{Varianza di un Bool} = \text{Gini}/2 = p_i(1-p_i)$$

$$\text{Deviazione standard} = \sqrt{\text{Gini}}$$

Learning = ridurre \uparrow

Alberi di regressione

$$\text{Var} = \frac{1}{|Y|} \sum (y - \bar{y})^2$$

target values

$$\text{Se insieme esclusivo} \quad \sum \frac{|Y_i|}{|Y|} \text{var}(Y_i) =$$

$$= \underbrace{\frac{1}{|Y|} \sum y^2}_{\text{costante}} - \sum \frac{|Y_i|}{|Y|} \bar{y}_i^2 - \max$$

L'albero è uguale ma

└ var al posto di imp per split

└ Label(Y) = \bar{y} nella foglia

└ Omogeneo se var in foglia \leq threshold

Alberi di clustering

$Dis(x_1, x_2)$ - distanza

$Dis(D)$ - distanza insieme = $\frac{1}{D^2} \sum_{x_1} \sum_{x_2} Dis(x_1, x_2)$

$\{D_1, \dots, D_k\}$ - split possibili

Il migliore è $\min \left\{ \sum_D \frac{|D_j|}{|D|} \cdot Dis(D_j) \right\}$

come
 Dis distanza euclidea

Se un oggetto ha d feature allora $Dis(x_1, x_2) = \sum_d (x_{1i} - x_{2i})^2$

$$Var = \frac{1}{N} \sum_n (x_{ji} - \bar{x}_i)^2$$

Σ stare delle feature = distanza euclidea fra vettori a d dimensioni $\frac{1}{N} \sum_n (\vec{x}_j - \vec{\bar{x}})^2$

Diminimilarity per dataset

distanza media $\sim 2 = 2 \text{ var}(x) = Dis(x)$

$$Var(x) = \sum_d Var_i(x) \quad \mathcal{O}(|D|)$$

Nota

cluster troppo piccoli = overfitting

sensibili ad outliers

per label si usa la classe più rappresentativa
(istanza con Dir minore o medoide)

Modelli a regole

Liste ordinate

Regole per regole in ordine

if n class n

else if n class n

else default class

Impartire una regola

accresci il corpus aggiungendo congiunzioni di letterali

si trova il segmento delle istanze

si seleziona una label come testo

Differenza con alberi

Nelle regole ci interessa la purezza di uno solo dei

segmenti, quindi niente media per selezionare

lo split migliore

Rule list learning

$R = \emptyset$ $D = \text{data}$

finché $D \neq \emptyset$

imposta una regola π

$$R \leftarrow R \cup \{\pi\}$$

$$D \leftarrow \{x \mid x \text{ coperto da } \pi\}$$

Rule learning

$L =$ insieme literali $D = \text{dati}$ body

finché D non omogeneo

$l =$ best literal (es. purezza max)

body $\Lambda = l$

$$D \leftarrow \{x \mid x \text{ coperto da } \Lambda\}$$

$$L \leftarrow L \cup \{l\}$$

label (es. majority class)

res. = if body then label

Liste di regole = alberi

Le liste di regole sono rappresentabili come alberi e ereditano la convessità della curva ROC

Rule list come Rankers

In base all'ordine delle regole si può indurre un ordine sulle classi

Regole non ordinate

\exists : seleziona la classe

\exists : costruiscono regole con massima precisione

Ordine indifferente - la lista copre una classe solo

Rule learning

finché $\exists x \mid x$ coperto da $\pi \wedge x \notin C$ (D omogeneo)

$\left\{ \begin{array}{l} \text{h} = \text{best literal (max precision)} \\ \text{body} = \text{body} \wedge \text{h} \end{array} \right.$

$\left\{ \begin{array}{l} \text{D} += x : x \text{ coperto da } \pi \\ \text{L} -= l \end{array} \right.$

$\left\{ \begin{array}{l} \text{D} += x : x \text{ coperto da } \pi \\ \text{L} -= l \end{array} \right.$

$\left\{ \begin{array}{l} \text{D} += x : x \text{ coperto da } \pi \\ \text{L} -= l \end{array} \right.$

$\left\{ \begin{array}{l} \text{D} += x : x \text{ coperto da } \pi \\ \text{L} -= l \end{array} \right.$

$\pi = \text{if } h \text{ then } C$

Problema

la precisione porta a scartare regole quasi
pure specializzabili in regole più generali
(overfitting)

Soluzioni: beam search (k candidati per step)
distribuzione di Laplace

Copertura per regole non coperte

$$p(A) = 1/4 \quad \# p(AB) = 6/12$$

$$p(B) = 5/8$$

Y rule set sono sempre migliori delle rule list

Y rule set coprono massimo 2^n segmenti le liste
massimo n ($n = \#$ regole)

Predizione con rule set

Si sceglie la regola con copertura maggiore

Weighted covering

Col clustering si vogliono impostare più regole per esempio

⇓
NON possiamo eliminare regole coperte

⇓
Se ne abbassa il peso (es. 1 \rightarrow $\frac{1}{2}$ per ogni regola)

Regole di associazione

item set

transazioni che coinvolgono 1 o più item

rappresentabili come matrice / grafo

La copertura lungo il grafo (# oggetti coinvolti) è monotona \Rightarrow il grafico è ~~convesso~~ (convesso più di n volte)
l'insieme degli item set frequenti è convesso

Item possono essere trovati secondo il grafo

Maximal item set rispetto monotono

Item set chiusi

copertura verso disgiunte (elementi diversi)

Nota in introduzione

ret frequenti frequenti - NO riduzione informazione
└ tanti dati

ret non-mali - perdita informazione max
└ pochi dati (min)

ret chiusi - media info No loss
└ dati pochi

Regole associazione

if B then H B, H item set

prima si estraggono i ret frequenti (altrimenti le associazioni potrebbero essere servite al corso)

~~le regole B, H come sotto da un set~~

~~di dati~~

rule mining

~~rule mining~~ $\# \text{transazioni}$
 $\forall B, H : B \cap H = \emptyset$ \uparrow *arbitrario*
 $L \text{ e } \text{supp}(B \cup H) / \text{supp}(B) \geq n'$
allora B \rightarrow H

Nota

$\text{supp}(B \cup H) / \text{supp}(B) = \text{confidence} = \text{prob. condizionata}$

Lift

misura per vedere regole valide

$$\text{lift}(R(B, H)) = \frac{n \cdot \text{supp}(B \cup H)}{\text{supp}(B) \cdot \text{supp}(H)}$$

$$n = \# \text{transazioni} (\text{supp}(B) + \text{supp}(H))$$

$\text{lift} = 1 \rightarrow B, H$ statisticamente indipendenti