STATISTICAL METHODS FOR MACHINE LEARNING

giosumarin

March 2021

Contents

1 Lezione 1

- clustering: raggruppare punti in accordo alla loro similarità (raggruppare clienti per soldi spesi);
- classification: predirre label semantiche associate ai data points (classificare documenti per argomento);

- planning: vogliamo decidere una sequenza di azioni che devono essere fatte per raggiungere un goal (robot che va da quache parte con ostacoli sul percorso o guida autonoma).
- supervised learning: abbiamo laber per degli esempi e imparo a classificare d questi
- unsupervised learning: clustering (label "attaccata" ai data points)

1.1 Label set

- Y label set
- news classification: Y={sport, politica, business, ... }
- predizione stock price: $Y \in \mathbb{R}$
- classification/categorization: Y insieme finito di simboli, $\hat{y} \stackrel{?}{=} y$, con $\hat{y}$ predizione e y valore reale;
- regression: $Y \in \mathbb{R}$, $|\hat{y} - y|$.

1.2 Loss function

$$l(y, \hat{y}) = \begin{cases} 0 \ se \ y = \hat{y} \\ 1 \ altrimenti \end{cases}$$

$Y = \{spam \ (positivo), nonspam \ (negativo)\}, binary \ classification \ problem$

$$l(y, \hat{y}) = \begin{cases} 2 \ se \ y = nonspam \ e \ \hat{y} = spam \leftarrow \ falso \ positivo \\ 1 \ se \ y = spam \ e \ \hat{y} = nonspam \leftarrow \ falso \ positivo \\ 0 \ altrimenti \end{cases}$$

absolute loss (per regressione): $l(y, \hat{y}) = |\hat{y} - y|$

square loss (per regressione): $l(y, \hat{y}) = (\hat{y} - y)^2$

[ESEMPIO] previsioni meteo: $Y = \{pioggia, asciutto\}$

$\hat{y} = probabilità \ assegnata \ a \ pioggia$; *prediction set:* $Z = \{0, 1\}$

$l(y, \hat{y}) = |\hat{y} - y|$

$$l(y, \hat{y}) = \begin{cases} \ln \frac{1}{\hat{y}} \ se \ y = 1 \\ \ln \frac{1}{1-\hat{y}} \ se \ y = 0 \end{cases}$$

La loss logaritmica ha le seguenti proprietà:

- $\lim_{\hat{y} \to 0^+} l(1, \hat{y}) = \infty$
- $\lim_{\hat{y} \to 1^-} l(0, \hat{y}) = \infty$

2 Lezione 2

2.1 Data Points

X dominio dati, x spesso è codificato convenientemente come vettore di numeri attravero per esempio la one-hot encoding.

$$X = \begin{cases} \mathbb{R}^d \ \textit{attributi numerici} \\ X_1, \ldots, X_d \ \textit{attributi categorici} \end{cases}$$

Possiamo avere anche un mix di diversi attributi.

2.2 Predictor

Un predittore è una funzione che mappa data points in label

$$f : X \to Y, f : X \to \overline{Z}, \overline{Z} \neq Y$$

Dato un ponto x abbiamo quindi

$$\hat{y} = f(x).$$

Quello che vogliore è avere una loss piccola per molti $x \in X$.

2.3 Supervised learning

Abbiamo le coppie (x, y) con x singolo data point e y la sua rispettiva label. Le label possono essere soggettive (annotazioni umane) o ogettive (misurazioni di strumenti).

2.3.1 Training Set

Insieme di esempi su cui effettuiamo l'addestramento; abbiamo quindi un training set in input a un algoritrmo di apprendimento (con la sua loss) e che in output genera un predittore.

2.3.2 Test Set

Insieme di esempi ($\neq$ training set) su cui viene valutata la capacità di generalizzazione di un predittore addestrato sul training set.

2.3.3 Completo

Abbiamo il predittore f uscente dall'algoritmo di apprendimento A usando la funzione di loss l. Abbiamo il test set $(x'_1, y'_1), \ldots, (x'_n, y'_n)$, calcoliamo il nostro test error come

$$\frac{1}{n} \sum_{t}^{n} l(y'_t, f(x'_t)).$$

Il nostro goal è quello di sviluppare una teoria per guidare nel design di A che ci genera predittori con un piccolo test error w.r.t. una loss function.

	x_1	x_2	x_3	x_4	x_5
f^*	-1	1	1	$f^*(x_4)$	$f^*(x_5)$
f^1	-1	1	1	1	1
f^2	-1	1	1	-1	1
f^3	-1	1	1	1	-1
f^4	-1	1	1	-1	-1

2.4 Empirical Risk Minimizer

Fisso un insieme F di predittori e una loss function f. Entra quindi il training set (S) in questo ERM (che ha F e l) e abbiamo in output

$$\hat{f} \in arg\min_{f \in F} \hat{l_S}(f).$$

L'idea è di minimizzare il training error in una classe F di predittori. Se $\min_{f \in F} \frac{1}{n} \sum_{t=1}^{n} l(y'_t, f(x'_t))$ è grande siamo in un caso di underfitting.

2.4.1 Esempio

Prendiamo F grande e vediamo cosa succede.

$$X = \{x_1, \ldots, x_5\}, Y = \{-1, 1\}, F \textit{ contiene tutti i classificatori binari}$$
$$|F| = 2^5 = 32, \exists f^* \textit{ t.c. } y_t = f^*(x_t) \textit{ con } t = \{1, \ldots, 5\}$$

Se il training set è formato dai primi 3 data point tutti e 4 i predittori hanno lo stesso training error uguale a 0. In questo caso non possiamo decidere quale predittore usare. Chiamo questo caso overfitting.

Possiamo estrapolare la seguente regola da questo esempio (quando F è finito):

$$m \geq \log_2 |F|$$

3 Lezione 3

3.1 Overfitting e Underfitting

Overfitting: Training error basso ma test error alto (nel caso di ERM $|F|$ grande).

Underfitting: Training error alto e test error vicino al training error.

3.2 Noisy label

In pratica non c'è $f^* : X \to Y$ tale che $f^*(x) = y$ per tutti gli esempi (x, y). y ha un disturbo dato x: stessi datapoint possono avere diverse label.

- Human in the loop: persone categorizzano diversamente i dati;

- Lack of information: le informazioni contenute nei datapoint X non sono sufficienti per determinare un'unica label

Noisy label provoca overfitting.

3.3 Nearest Neighbor

Classifica i punti usando la label del training point più vicino. Averno $x = (x_1, \ldots, x_d)$ con d coordinate per punto calcoliamo la distanza euclidea per ogni dimensione: $||x - x'|| = \sqrt{\sum_i^d (x_i - x'_i}$. NN genera un predittore con training error sempre a 0, devo salvare tutto il training set per fare la predizione.

k-NN: come NN ma usa la maggioranza delle label dei k vicini data points. k viene scelta dispari per non avere un numero pari di vicini nel punto del test che voglio classificare. Quindi guardo i k più vicini al punto che sto valutando e vince la maggioranza. Per classificazione uso più label (il caso generico è spiegato su classificazione binaria), per regressione faccio la media dei k più vicini.

4 Lezione 4

4.1 Tree Predictor

NN funziona solo per attributi numerici. Abbiamo un albero in cui i nodi interni sono taggati come test e i nodi foglia solo label. Per esempio abbiamo

$$X_i = \{a, b, c, d\}$$
$$f(X_i) = \begin{cases} 1 \ se \ x_1 = a \\ 2 \ se \ x_2 = b \\ 3 \ altrimenti \end{cases}$$

I valori del sistema indicano a quale nodo devo andare. Posso avere confronti su attributi categorici o soglie suattributi $\in \mathbb{R}$.

Dato un training set S, come costruisco un tree classificator?

- $Y = -1, 1$;
- albero binario compreto (0 o 2 figli);
- $0 - 1$ loss;
- $S = (x_1, y_1), \ldots, (x_m, y_m)$.

Partiamo da un classificatore binario costante che classifica tutti allo stesso modo, ovvero uso la maggioranza per assegnare la label alla foglia. Splitto ora questa foglia in due foglie e pongo un test nel classificatore costante iniziale. Ricordiamo che un test è fatto su una sola dimensione dell'input. Partiziono il

training set S in due: S_l e S'_l a cui corrispondono rispettivamente le labl y_l e y'_l. Defininiamo $N_l = |S_l|$ dove y_l è in maggioranza in S_l.

Definiamo ora $S_l^+ = \{(x_t, y_t) \in S_l : y_t = +1\}$ e $S_l^- = \{(x_t, y_t) \in S_l : y_t = -1\}$.

$$y_l = \begin{cases} +1 & \textit{se } N_l^+ \geq N_l^- \\ -1 & \textit{altrimenti} \end{cases}$$

$$\mathbb{I} = \begin{cases} 1 & \textit{se vero} \\ 0 & \textit{altrimenti} \end{cases}$$

$$\hat{l}_s(h_T) = \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}\{h_T(x_t) \neq y_t\} =$$

$$= \frac{1}{m} \sum_{l} \min\{N_l^+, N_l^-\} =$$

(l'ultimo passaggio è perchè alla label associo la maggioranza, con l nella sommatoria ciclo sui nodi foglia. Considerando che $\frac{N_l^+}{N_l} + \frac{N_l^-}{N_l} = 1$ e che $\sum_l (N_l^+ + N_l^-) = m = |S|$)

$$= \frac{1}{m} \sum_{l} \min$$