

Apprendimento basato sulle istanze

Apprendimento basato sulle istanze

- Apprendimento: semplice memorizzazione di tutti gli esempi $\langle x_i, f(x_i) \rangle$
- Classificazione di una nuova istanza x_q : reperimento degli esempi piu' simili e classificazione sulla base di questi

Nearest neighbour

- Nearest neighbour (o 1-nearest neighbour): data una nuova istanza di query x_q , si cerca prima l'esempio piu' vicino x_n e poi si stima $f(x_q)$

$$f'(x_q) = f(x_n)$$

- dove f' è la stima di f

K-nearest neighbour

- k-nearest neighbour: si considerano i k esempi piu' vicini a x_q
 - Se la funzione f e' discreta, allora si restituisce il valore v di f piu' frequente tra i k esempi
 - Ogni esempio x_i assegna un voto al valore v per il quale $f(x_i)=v$

$$f'(x_q) = \arg \max_{v \in V} \sum_{i=1}^k \delta(v, f(x_i))$$

- dove $f: \mathfrak{R}^n \rightarrow V$ e $\delta(a,b)=1$ se $a=b$ altrimenti $\delta(a,b)=0$

K-nearest neighbour

- Se la funzione f è continua, si restituisce la media tra i valori di f per i k esempi

$$f'(x_q) = \frac{\sum_{i=1}^k f(x_i)}{k}$$

K-nearest neighbour

- Vantaggi: non c'è bisogno di inferire una descrizione globale della funzione f , si generano solo approssimazioni locali, utile se f è molto complessa
- Svantaggi:
 - alto tempo per la classificazione (devono venire considerati tutti gli esempi)
 - Tutti gli attributi sono considerati per reperire i casi dalla base di conoscenza: se il concetto target dipende solo da pochi attributi, gli esempi effettivamente più simili possono essere a grande distanza

K-nearest neighbour

- Assunzione: le istanze sono punti di \mathbb{R}^n
- Come distanza si può utilizzare la distanza euclidea:
- $x = \langle a_1(x), a_2(x), \dots, a_n(x) \rangle$

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2}$$

Altre distanze

- Distanza di Minkowski di ordine p

$$d(x_i, x_j) = \left(\sum_{r=1}^n |a_r(x_i) - a_r(x_j)|^p \right)^{1/p}$$

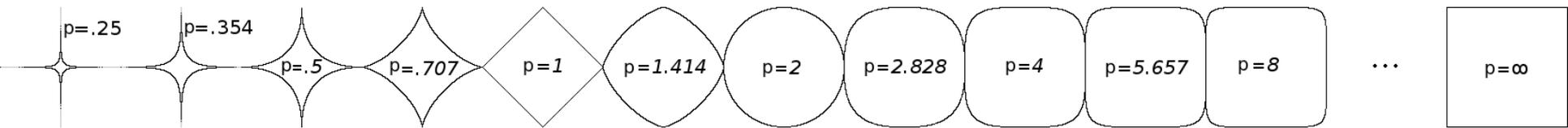
- p=2 => distanza euclidea
- p=1 => distanza Manhattan

$$d(x_i, x_j) = \sum_{r=1}^n |a_r(x_i) - a_r(x_j)|$$

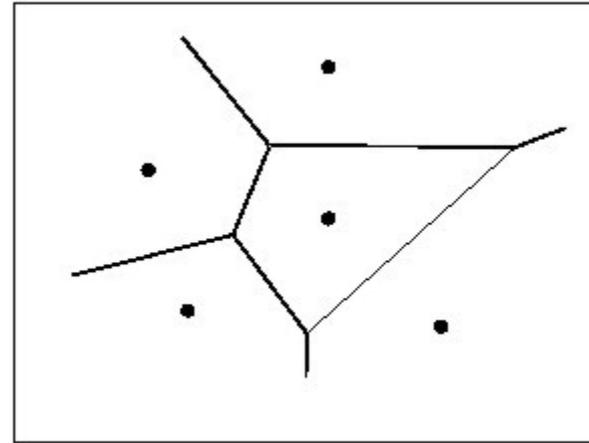
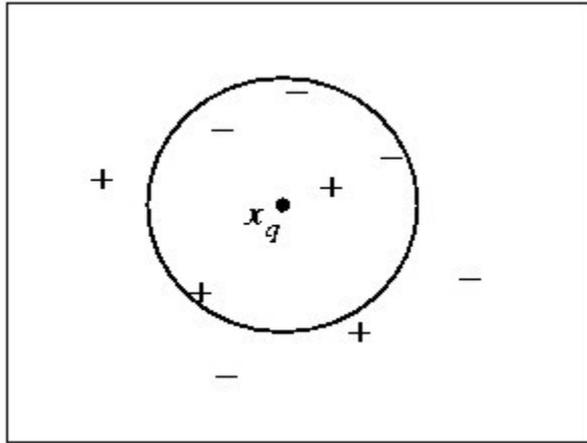
- p= ∞ => distanza di Chebyshev

$$\begin{aligned} d(x_i, x_j) &= \lim_{p \rightarrow \infty} \left(\sum_{r=1}^n |a_r(x_i) - a_r(x_j)|^p \right)^{1/p} = \\ &= \max_{r=1}^n |a_r(x_i) - a_r(x_j)| \end{aligned}$$

Luoghi dei punti a distanza di Minkowski 1



Esempi in R_2



- 1-NN classifica l'esempio come positivo
- 5-NN classifica l'esempio come negativo

• Diagramma di Voronoi (1-NN): il poligono convesso che circonda ciascun punto indica la regione più vicina al punto
(<http://www.cs.cornell.edu/Info/People/chew/Delaunay.html>)

K-nearest neighbour

- Gli attributi possono avere diverse scale e quindi la loro importanza nella misura della distanza può essere diversa
- Per risolvere questo problema, prima di calcolare la distanza, si cambia la scala di ciascun attributo in modo che i valori dell'attributo varino tra 0 e 1
 - Siano $a_{i\min}$ $a_{i\max}$ rispettivamente i valori minimo e massimo dell'attributo a_i nel training set
 - Il valore dell'attributo $a'_i(x)$ scalato tra 0 e 1 è dato

da

$$a'_i(x) = \frac{a_i(x) - a_{i\min}}{a_{i\max} - a_{i\min}}$$

K-nearest neighbour

- Se i punti non appartengono a R^n ma gli attributi (alcuni o tutti) sono nominali allora è necessario definire una distanza tra i valori di tali attributi
 - Ad esempio, qual'è la distanza tra basso, medio e alto?
- Nel caso più semplice si assegna distanza 0 se i valori sono identici, altrimenti distanza 1
 - La distanza tra basso e basso è 0, tra basso e alto è 1
- In altri casi si possono utilizzare distanze definite ad hoc, basate su informazione extra disponibile sull'attributo (scale ordinali)
 - Ad esempio si può assegnare una distanza inferiore alla coppia basso e medio rispetto alla coppia basso e alto

Esempio

| No | Outlook | Temp | Humid | Windy | Class |
|-----|----------|------|--------|-------|-------|
| D1 | sunny | mild | normal | T | P |
| D2 | sunny | hot | high | T | N |
| D3 | sunny | hot | high | F | N |
| D4 | sunny | mild | high | F | N |
| D5 | sunny | cool | normal | F | P |
| D6 | overcast | mild | high | T | P |
| D7 | overcast | hot | high | F | P |
| D8 | overcast | cool | normal | T | P |
| D9 | overcast | hot | normal | F | P |
| D10 | rain | mild | high | T | N |
| D11 | rain | cool | normal | T | N |
| D12 | rain | mild | normal | F | P |
| D13 | rain | cool | normal | F | P |
| D14 | rain | mild | high | F | P |

Esempio

- Caso da classificare:
- $\langle \text{rain, cool, high, T} \rangle$
- Distanza dagli esempi (senza usare informazione extra sugli attributi):

D1: $\sqrt{3}$

D2: $\sqrt{2}$

D3: $\sqrt{3}$

D4: $\sqrt{3}$

D5: $\sqrt{3}$

D6: $\sqrt{2}$

D7: $\sqrt{3}$

D8: $\sqrt{2}$

D9: 2

D10: 1

D11: 1

D12: $\sqrt{3}$

D13: $\sqrt{2}$

D14: $\sqrt{2}$

Esempio

- 1-NN: in questo caso ci sono due esempi che sono a minore distanza (1) dal caso: D10 e D11, entrambi appartengono alla classe N
- Quindi il caso è classificato come N
- Se gli esempi a minore distanza avessero avuto classe diversa, si sarebbe scelta la classe di maggioranza

Estensioni di k-NN

- NN pesato sulla base della distanza: il contributo di ciascuno dei k vicini e' pesato sulla base della distanza dal punto da classificare x_q
- Funzione discreta: si pesa ciascun voto sulla base dell'inverso del quadrato della distanza

$$f'(x_q) = \arg \max_{v \in V} \sum_{i=1}^k w_i \delta(v, f(x_i))$$

$$w_i = \frac{1}{d(x_q, x_i)^2}$$

Estensioni di k-NN

- Funzione continua: si pesa ciascun contributo alla media con l'inverso del quadrato della distanza

$$f'(x_q) = \frac{\sum_{i=1}^k w_i f(x_i)}{\sum_{i=1}^k w_i}$$

$$w_i = \frac{1}{d(x_q, x_i)^2}$$

Osservazione

- Aggiungendo un coefficiente basato sulla distanza, si potrebbero considerare tutti gli esempi per la classificazione invece dei k più vicini
- In questo caso si parla di un metodo globale invece di locale come il k -NN

Osservazioni

- k-NN e' robusto agli errori perche' si prende la media di k esempi
- Bias induttivo: assunzione che la classificazione di una istanza x_q sia simile a quella dei suoi k vicini

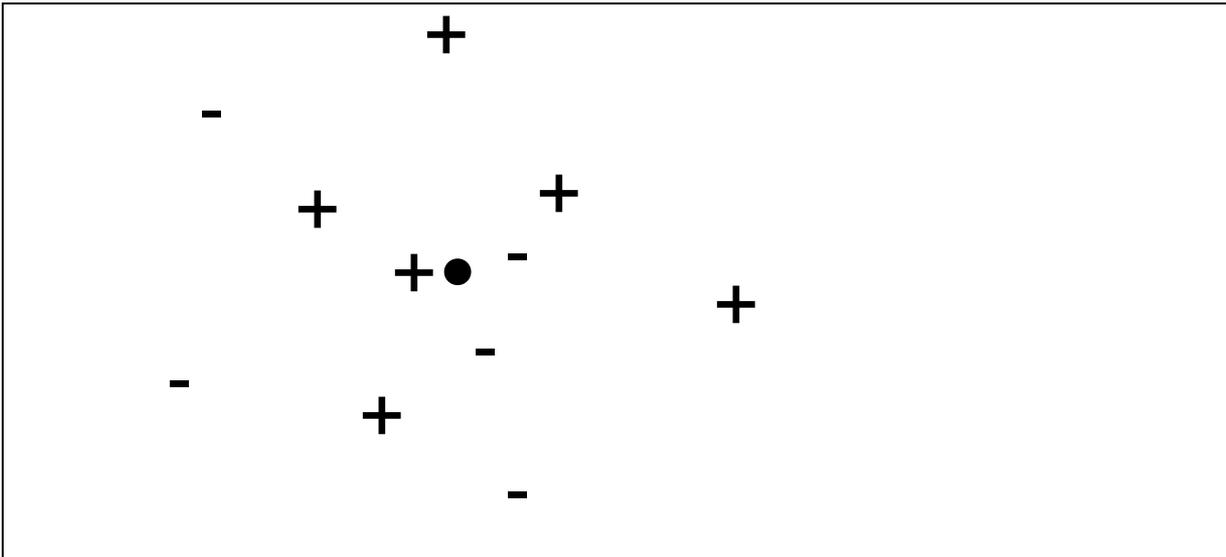
Curse of dimensionality

- k-NN considera tutti gli attributi degli esempi, in contrasto con gli algoritmi per l'apprendimento di regole e alberi di decisione che selezionano solo gli attributi piu' rilevanti
 - Esempio: 20 attributi di cui solo 2 rilevanti per la classificazione delle istanze
- Soluzioni:
 - Allunga ciascun asse di un peso w_r ,
 - w_1, \dots, w_n sono scelti mediante cross-validation: un sottoinsieme degli esempi e' scelto come insieme di training, i pesi sono scelti in modo da minimizzare l'errore nel classificare i rimanenti esempi
 - Questo processo va ripetuto piu' volte in modo da affinare i pesi w_r

Curse of dimensionality

- Un approccio alternativo consiste nel porre alcuni pesi a 0, sempre utilizzando la cross validation

Esercizio



- Si indichi la classificazione del punto • con 1-NN, 3-NN e 5-NN
- 1-NN=+, 3-NN=-, 5-NN=+