Data & Web Mining

Claudio Lucchese claudio.lucchese@unive.it

Disclaimer

Queste domande non coprono tutti gli argomenti del corso e dell'esame!

K-NN Classifier

- **Cosa succede per k che tende a infinito?**
  a. La predizione non è definita
  b. L'algoritmo non termina
  c. La predizione è la classe maggioritaria del dataset di train

K-NN Classifier

c. La predizione è la classe maggioritaria del dataset

La predizione è sempre uguale alla classe maggioritaria delle istanze del training set selezionate come nearest neighbours.

k tendente a infinito implica selezionare tutte le istanze del training set

K-NN Classifier

- **Cosa fa il classificatore a tempo di training?**
  a. Costruisce un albero di decisione
  b. Niente
  c. Memorizza il Training Set (possibilmente indicizzandolo)

K-NN Classifier

c. Memorizza il Training Set (possibilmente indicizzandolo)

Per questo motivo viene denominato Lazy Learner. In genere, il dataset di training viene indicizzato in una struttura dati che supporti la ricerca k-nn in maniera efficiente.

K-NN Classifier

- **Che impatto ha la “scala” (min/max) delle features?**
  a. Se le features sono su scala diversa la predizione è sicuramente sbagliata
  b. Il calcolo della distanza è dominato dalla features con range [min-max] maggiore
  c. Dipende dalla misura di distanza
  d. Nessun impatto

- b, ma possiamo considerare come vera anche c perchè la misura di distanza potrebbe essere insensibile alla scala

K-NN Classifier

b. Il calcolo della distanza è dominato dalla features con range [min-max] maggiore

Ma potremmo considerare come vera anche c. perchè la misura di distanza potrebbe essere insensibile alla scala

Binary Decision Tree

- **Che impatto ha la “scala” (min/max) delle features?**
  a. Se le features sono su scala diversa la predizione è sicuramente sbagliata
  b. Il calcolo della distanza è dominato dalla features con range [min-max] maggiore
  c. Dipende dalla misura di distanza
  d. Nessun impatto

Binary Decision Tree

d. Nessun impatto

La scelta, in ciascun nodo, del miglior partizionamento D_L e D_R non dipende dalla scala delle features.

Quindi, normalizzare le features non ha senso se intendiamo usare un albero di decision

Binary Decision Tree

- **Perchè “binary”?**
  a. Perché possono gestire solo problemi di classificazione binaria: vero/falso, giallo/blu, cerchio/quadrato
  b. Perché ogni nodo può avere 2 figli
  c. Perché ogni nodo può analizzare solo due features

Binary Decision Tree

b. Perché ogni nodo può avere 2 figli

Binary Decision Tree

- **Quante features possono essere coinvolte nel predicato di un nodo?**
  a. Solo 1 features
  b. Solo 2 features
  c. Solo 1 se numerico, più di una se categorico
  d. Dipende da _____

Binary Decision Tree

d. dipende dalla nostra implementazione

Il predicato di splitting può essere arbitrariamente progettato da noi, e quindi arbitrariamente complicato e coinvolgere un grande numero di features.

Un caso pratico è quello degli alberi obliqui, dove vengono considerate due features per split

Binary Decision Tree

- **Qual è la profondità massima?**
  a. Non si può dire, perchè ?
  b. Pari al numero di features
  c. Pari al numero di istanze
  d. Pari al prodotto tra numero di features e numero di istanze

Binary Decision Tree

c. Pari al numero di istanze

Un albero sensato deve avere almeno un’istanza per foglia, e quindi se un albero avesse una forma a catena (solo figli sinistri) al massimo la profondità sarebbe pari al numero di istanze (meno 1)

Binary Decision Tree

- **Nel caso di variabile categorica con *N* valori distinti possibili, quali affermazioni sono vere?**
  a. Ci sono *N* modi diversi di creare uno Split binario
  b. Ci sono 2^N modi diversi di creare uno Split binario
  c. Ci sono 2^{N-1} modi diversi di creare uno Split binario
  d. Ci sono $2^{N-1}-1$ modi diversi di creare uno Split binario

- d: perchè i sottoinsiemi di N valori distinti sono 2^N, ma metà di questi sono complementare ed equivalenti (se la feature assume i valori ABCD, $x \in \{A\}$ è equivalente a $x \in \{BCD\}$), e infine lo split $x \in \{\}$ non è valido.

Binary Decision Tree

d. Ci sono $2^{N-1}-1$ modi diversi di creare uno Split binario

Perchè in uno split binario, un il criterio può essere della forma $x \in S$ dove S è un sotto-insieme qualsiasi degli N valori distinti.

- I sottoinsiemi S di N valori distinti sono 2^N
- Metà di questi sono complementare ed equivalenti (se la feature assume i valori ABCD, $x \in \{A\}$ è equivalente a $x \in \{BCD\}$).
  Quindi scendiamo a $2^N/2=2^{N-1}$
- Infine lo split $x \in \{\}$ non è valido.
  Quindi abbiamo $2^{N-1}-1$

Linear Regression

- **Nel caso di un dataset con *N* features, quanti sono i parametri di una linear regression?**
  a. Uno: la pendenza della retta
  b. *N*: un peso per ciascuna feature
  c. *N+1*: un peso per ciascuna feature + l'offset/bias
  d. *2N*: un peso e un offset per ciascuna feature

Linear Regression

c. *N+1*: un peso per ciascuna feature + l'offset/bias

Se la nostra istanza x ha le features $x_1, x_2, \ldots, x_N$
la regressione lineare consiste nel trovare gli N pesi w_i e l'offset w_0 tali che la combinazione lineare

$$w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$$

approssimi il target y

Linear Regression

- **I pesi/coefficienti della regressione lineare possono essere usati come score di feature importance**

  a. Vero

  b. Falso

Linear Regression

b. Falso

I pesi dipendono dalla scala delle features.

Ad esempio se moltiplicassi una feature x_i per 1000, il suo coefficiente w_i sarebbe diviso per 1000, e quindi la sua importanza sarebbe 1000 volte minore.

Ma visto che l'informazione non è cambiata, la sua importanza non può essere cambiata.

Logistic Regression

- **Con Logistic Regression**
  a. viene imparata una regressione lineare dopo aver applicato il logaritmo alla variabile target
  b. viene risolto in task di regressione
  c. viene risolto un task di classificazione
  d. possono essere risolti accuratamente solo distribuzioni linearmente separabili

Logistic Regression

c. viene risolto un task di classificazione

Il modello di logistic regression predice la probabilità di appartenere ad una classe del problema di classificazione binario

d. possono essere risolti accuratamente solo distribuzioni linearmente separabili

Si, perchè il decision boundary corrispondente è un iperpiano.

Logistic Regression

$$P(y=1|\mathbf{x}) = sig(-z) = \frac{1}{1+e^{-z}}$$

$$z = \mathbf{w}^T\mathbf{x} + b$$

- **Nell'equazione sopra**
  a. L'iperpiano *z* corrisponde al decision boundary
  b. L'iperpiano *z* è ortogonale decision boundary

Logistic Regression

$$P(y=1|\mathbf{x}) = sig(-z) = \frac{1}{1+e^{-z}}$$

$$z = \mathbf{w}^T\mathbf{x} + b$$

b. L'iperpiano *z* è ortogonale decision boundary

All'aumentare di *z*, aumenta la probabilità della classe 1.
Se *z=0*, allora le due classi sono equiprobabili con probabilità ½
Il luogo dei punto dove *z=0*, corrisponde al decision boundary.

Support Vector Machine SVM

- **Le SVM massimizzano il margine, ovvero:**
  a. La distanza media dei punti del training dal decision boundary
  b. Il numero di punti classificati correttamente
  c. La distanza dal decision boundary dei punti a lui più vicini
  d. La complessità del modello

Support Vector Machine SVM

c. La distanza dal decision boundary dei punti a lui più vicini

Questo rende la predizione più robusta rispetto a piccole modificazioni del test set.

Si può dimostrare che la massimizzazione del margine è legata alla minimizzazione della complessità del modello.

Support Vector Machine SVM

- **Le SVM:**
  a. Trovano un iperpiano, quindi si applicano solo a dati linearmente separabili
  b. Trovano un iperpiano in uno spazio ad alta dimensionalità diverso da quello del dataset
  c. Trovano un iperpiano in uno spazio a bassa dimensionalità diverso da quello del dataset
  d. Trovano un iperpiano, e quindi sono equivalenti ad una logistic regression

Support Vector Machine SVM

b. Trovano un iperpiano in uno spazio ad alta dimensionalità diverso da quello del dataset

Lo spazio dipende dalla funzione di kernel che viene usata.

Nel caso di kernel lineare e con un peso C tale da rendere trascurabile il margine, una SVM è di fatto equivalente ad una regressione lineare.

Ensemble Methods

- **Nel caso di Ensemble Methods:**
  a. Il training individua il migliore modello nell'ensemble dopo aver allenato tanti modelli
  b. Genero una predizione usando solo il modello più accurato
  c. Genero una predizione aggregando il risultato prodotto da tutti modelli
  d. I modelli più accurati hanno sempre più importanza

Ensemble Methods

c. Genero una predizione aggregando il risultato prodotto da tanti modelli

Calcolo la moda per classificazione e la media per regressione.

Nel caso del boosting, i modelli hanno un peso, e quindi anche la predizione è media/moda pesata.

Ensemble Methods

- **E’ vero che:**
  a. Il Bagging riduce il Bias
  b. Il Bagging riduce la Varianza
  c. Il Boosting riduce il Bias
  d. Il Boosting riduce la Varianza
  e. Il Boosting è più efficace del Bagging

Ensemble Methods

b. Il Bagging riduce la Varianza

c. Il Boosting riduce il Bias

La strategia migliore da utilizzare dipende dal modello base.

Il Boosting è una scelta migliore del Bagging quando il modello base ha un alto Bias (e una bassa varianza).

Ensemble Methods

- **Un Bootstrap Sample di un dataset D:**
  a. E’ un campione di D con dimensione configurabile
  b. Ha la stessa dimensione del dataset D
  c. Non ha doppioni della stessa istanza
  d. Ha un sotto-insieme delle istanze e delle features di D

Ensemble Methods

b. Ha la stessa dimensione del dataset D

E’ un campionamento con replacement, quindi la stessa istanza può essere selezionata più volte. Tutte le features vengono sempre prese in considerazione.

Alcune implementazioni permettono configurazioni aggiuntive, ma queste non fanno parte dell’algoritmo originale.

Ensemble Methods

- **Nel Boosting (AdaBoost), la probabilità per un'istanza di essere campionata:**
  a. Aumenta se il modello predice correttamente
  b. Aumenta se il modello predice in maniera errata
  c. Dipende dall'errore del modello M_i calcolato sul campione D_i
  d. Viene re-inizializzata se il modello ha accuratezza <0.5

Ensemble Methods

b. Aumenta se il modello predice in maniera errata

d. Viene re-inizializzata se il modello ha accuratezza <0.5

Aumenta in funzione dell'errore del modello.

$$w_j = w_j e^{\alpha_i} \qquad \alpha_i = \log \frac{1 - error(M_i)}{error(M_i)}$$

Se il modello M_i ha performance insufficiente:

il modello viene scartato, i pesi re-inizializzati e ne viene allenato uno nuovo su un nuovo campionamento.

L'errore è calcolato sul dataset originale D.

(Disclaimer: a meno di scelte implementative diverse)

Random Forest

- **In Random Forest:**
  a. come in bagging vengono creati dei boostrap sample
  b. la profondità degli alberi va “tunata” sul validation
  c. ad ogni nodo considero un sotto-insieme casuale di features
  d. ogni albero può includere tutte le features del dataset
  e. Le istanze con predizione errata hanno più probabilità di essere selezionate nei sample

Random Forest

a. come in bagging vengono creati dei boostrap sample

c. ad ogni nodo considero un sotto-insieme casuale di features

d. ogni albero può includere tutte le features del dataset

Gli alberi sono fully grown. Random Forest è un'estensione del Bagging e non del Boosting.

Random Forest

- **Una Random Forest può essere usato come stimatore di similarità, dove la similarità di due istanze è:**
  a. pari alla frazione di nodi in comune che attraversano
  b. pari alla frazione di alberi che danno la stessa predizione
  c. dipende dalla similarità media delle predizioni di ciascun albero
  d. pari alla frazione di foglie in comune che attraversano

Random Forest

d. pari alla frazione di foglie in comune che attraversano

Ci sono varianti viene considerata la distanza tra le predizioni di ciascun albero, la percentuale di nodi in comune in ciascun percorso.

Validation Set

- **Il validation set è usato per:**
  a. per stimare l'accuratezza di un modello
  b. per scegliere il modello migliore tra diversi parametri
  c. per scegliere la configurazione di parametri migliore

Validation Set

c. per scegliere la configurazione di parametri migliore

Una volta scelta la configurazione di parametri, si può riallenare su training + validation.

E’ sempre sbagliato stimare l’accuracy su un dataset che abbiamo utilizzato per fare scelte relative al training.

k-fold Cross Validation

- **Nel caso di k-fold cross validation:**
  a. scelgo il modello con performance migliore in uno dei k folds
  b. scelgo il modello con performance migliore in media tra i k folds
  c. ciascuno fold è ottenuto con boostrap sampling
  d. riesco a stimare in maniera più accurata l'accuratezza di un modello

k-fold Cross Validation

nessuna delle risposte!

Il processo di (cross-) validation supporta la scelta degli iper-parametri e non di un modello.

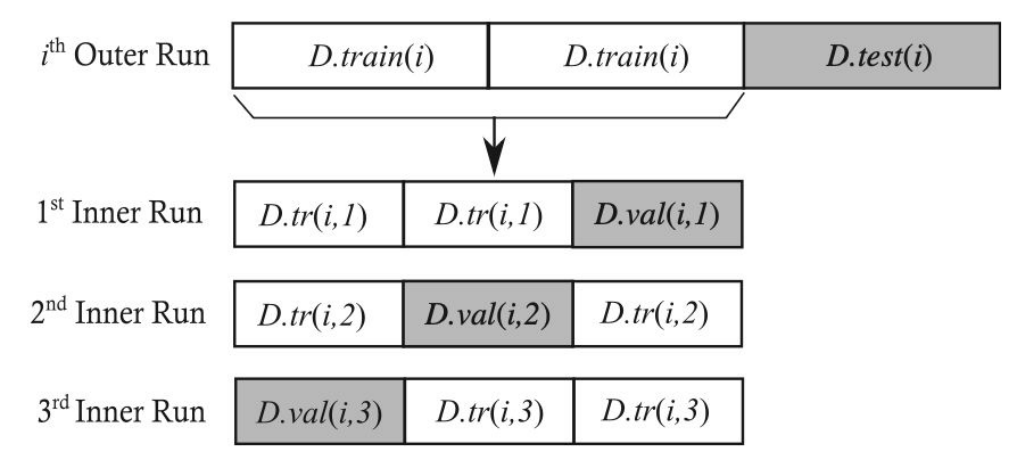
I k fold sono partizioni della stessa dimensione del training set.

Nested Cross Validation

- **La Nested Cross Validation:**
  a. è un sinonimo di k-fold cross-validation
  b. è utile per stimare in maniera più accurata l'accuratezza di un modello
  c. è utile per stimare in maniera più accurata la bontà di un setting di parametri
  d. non identifica né un modello né un setting di parametri ottimale

Nested Cross Validation

d. non identifica né un modello nè un setting di parametri ottimale



restituisce una stima dell'accuracy di un "algoritmo e del suo processo di training"

Feature Importance and Selection

- **La Feature importance è naturale in una foresta:**
  a. gli ultimi alberi sono più importanti dei primi
  b. i primi alberi sono più importanti degli ultimi
  c. ogni nodo ha associato un gain
  d. si, ma non in una random forest per via della selezione casuale delle features

Feature Importance and Selection

c. ogni nodo ha associato un gain

si, è il gain calcolato a tempo di costruzione dell'albero.

Accumulando i vari gain su tutti i nodi che usano una data feature, e pesati per il numero di istanze corrispondenti, si ottiene la feature importance.

Se gli alberi sono pesati, dovremmo tener conto del peso.

Feature processing

- **Vogliamo predire il rischio di default di uno stato con il seguente dataset. Come gestire le features?**

X= {Population, Continent, is_BRICS} -> Y = {Risk: alto, medio, basso}

	POPULATION	CONTINENT	IS_BRICS	RISK (Y)
a.	Standardize	Standardize	Standardize	One-Hot-Enc.
b.	Nothing	One-Hot-Enc.	Nothing	Nothing
c.	Nothing	One-Hot-Enc.	Nothing	Ordinal Enc.
d.	Nothing	One-Hot-Enc.	Binary Enc.	Ordinal Enc.

Feature processing

X= {Population, Continent, is_BRICS} -> Y = {Risk: alto, medio, basso}

	POPULATION	CONTINENT	IS_BRICS	RISK (Y)
b.	Nothing	One-Hot-Enc.	Nothing	Nothing

Population non ha bisogno di processing se usiamo un albero, potrebbe aver bisogno di normalizzazioni per altri modelli.

Continente ha 6 valori distinti

is_BRICS è binaria, va bene che assuma i valori 1 e 0

La variabile Y (target) non ha bisogno di processing

Precision / Recall / etc.

- **Se voglio massimizzare il numero di istanze di classe *c* individuate dal mio classificatore:**
  a. voglio massimizzare la precision
  b. voglio massimizzare la recall
  c. voglio massimizzare il true positive rate
  d. voglio massimizzare il true negative rate

Precision / Recall / etc.

- **Se voglio massimizzare il numero di istanze di classe *c* individuate dal mio classificatore:**

  ~~a. voglio massimizzare la precision~~

  b. voglio massimizzare la recall

  c. voglio massimizzare il true positive rate

  d. voglio massimizzare il true negative rate

Precision of class c: $\frac{\#\text{ instances correctly classified as } c}{\#\text{ instances predicted as class } c}$

Recall of class c: $\frac{\#\text{ instances correctly classified as } c}{\#\text{ instances with true label } c}$

True Negative Rate: $\frac{\#\text{ True Negatives}}{\#\text{ Total Negatives}}$

True Positive Rate: $\frac{\#\text{ True Positives}}{\#\text{ Total Positives}}$

Average Precision / Recall / etc.

- **Per calcolare la media delle precisioni rispetto alle diverse classi di un dataset**
  a. posso usare Macro average
  b. posso usare Weighted average
  c. posso usare F-Measure
  d. posso usare la media armonica

Average Precision / Recall / etc.

a. posso usare Macro average
   è esattamente la media delle precisioni
b. posso usare Weighted average
   è la media per sta per il numero di istanze nelle classi