Machine Learning - Appunti

Leonardo Polidori
Edoardo Marchionni

Anno Accademico 2024/2025

Contents

1 Introduzione

1.1 Machine Learning

Programmare un algoritmo affinché impari dai dati o dall'esperienza. Molto simile alla statistica:

- Entrambi i campi attingono al calcolo, alla probabilità e all'algebra lineare.
- Entrambi tentano di scoprire modelli nei dati.

Diversità dalla statistica:

- Il Machine Learning è più interessato alla costruzione di agenti autonomi.
- Pone maggiore enfasi sulle prestazioni predittive e sulla scalabilità.

1.1.1 Workflow

1. Scegliere se utilizzare il Machine Learning.
2. Raccogliere e organizzare i dati.
3. Stabilire una baseline.
4. Scegliere il modello.
5. Ottimizzare.
6. Ricercare gli iperparametri.
7. Analizzare performance ed errori e iterare.

1.1.2 Tipi di Machine Learning

- **Supervised Learning**: Esempi etichettati con comportamento corretto.
- **Reinforcement Learning**: L'agente impara massimizzando un segnale di ricompensa.
- **Unsupervised Learning**: Non viene etichettato nulla; si cerca di trovare pattern nei dati.

1.1.3 Vettori di input

Gli algoritmi di Machine Learning utilizzano vari tipi di dati. Gli input sono spesso rappresentati come vettori nel piano reale $\mathbb{R}^2$.

2 Supervised Learning

2.1 Nearest Neighbors

Per classificare un vettore x, possiamo cercare il vettore più vicino nel training set e copiare la sua label.

$$\|x^{(a)} - x^{(b)}\| = \sqrt{\sum_{j=1}^{d}(x_j^{(a)} - x_j^{(b)})^2} \tag{1}$$

L'algoritmo trova la coppia più vicina a tale vettore di input ed assegnerà come label di x proprio quella del suo vicino.

Algorithm

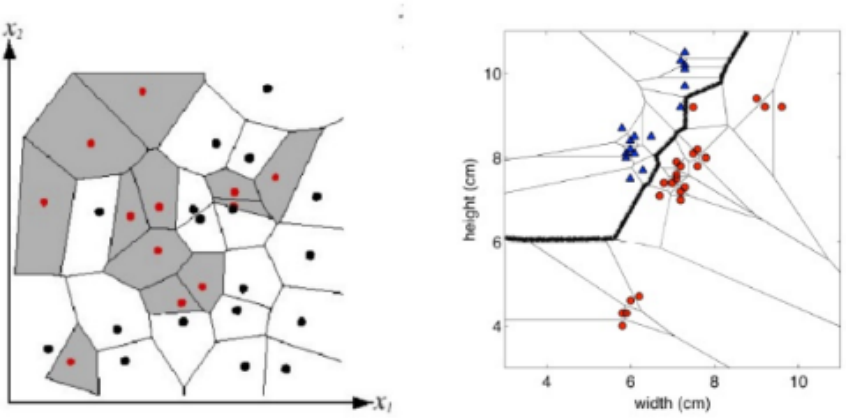
Find example (x^*, t^*) (from the stored training set) closest to $\mathbf{x}$. That is:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}^{(i)} \in \text{training set}} distance(\mathbf{x}^{(i)}, \mathbf{x})$$

Output $y = t^*$

2.1.1 Voronoi Diagram

Data una serie di punti nello spazio, il diagramma di Voronoi suddivide lo spazio in celle o regioni, dove ogni cella contiene i punti dello spazio che sono più vicini a uno specifico punto rispetto a tutti gli altri punti. Questo concetto è usato nel **Nearest Neighbors**, un algoritmo di Supervised Learning. Nel contesto del KNN, un Voronoi Diagram mostra come lo spazio di input è suddiviso in base alla distanza dai punti di training. Ogni regione rappresenta un'area in cui tutti i punti sono classificati come appartenenti alla stessa classe del vicino più prossimo. il Voronoi viene rappresentato grazia al decision boundary ovvero i confini che separano le diverse classi in uno spazio di input, cioè le linee o superfici che dividono lo spazio in regioni classificate diversamente.



(a) Voronoi Diagram (b) Decision Boundary

2.1.2 Rumore

Una delle problematiche di tale algoritmo è la sensibilità al rumore generato da possibili dati etichettati male che creano un effetto a cascata per i futuri input.
Per ovviare a ciò si può far considerare i K punti del training set più vicini **(KNN)**.
inserire immagine

Scelta di K

- K piccolo:
  - Ottimo per catturare pattern a grana fine;
  - Può portare all'overfit, cioè la troppa sensibilità alle caratteristiche del training set.
- K grande:
  - Effettua previsioni stabili calcolando la media su molti esempi;
  - Può portare all'underfit, cioè non riuscire a catturare dettagli importanti del traning set portando a generalizzazioni dei dati.
- K bilanciato:
  - La scelta ottimale di K dipende dal numero di punti dati n;
  - Buone proprietà teoriche se $K \to \infty$ e $K/n \to 0$;
  - Regola empirica: scegliere $K < \sqrt{n}$;
  - Possiamo scegliere K usando il validation set.

Suddivisione del dataset :

- Training Set: Usato per addestrare il modello con un set di dati già classificati.
- Validation Set: set di dati di cui si conosce la classificazione ma usati per calcolare l'errore preliminare del modello. Tuning degli iperparametri (sisceglie la versione con l'errore più basso) e per prevenire fenomeni come overfitting.
- Test Set: ultimo set di dati usato dopo aver ottimizzato il modello per valutare in modo definitivo le prestazioni di esso su dati completamente nuovi di cui si conosce la classificazione. Serve a calcolare l'errore finale del modello(ci si aspetta che sia simile a quello del validation set).

K-Fold Cross-Validation :
il K-Fold Cross-validation è una tecnica di tuning degli hyperparameter robusta. Essa consiste nel dividere il Data set in Training e test set; per poi suddividere a sua volta training set in K-folder di cui uno farà da validation set per poi ruotare a turno per tutti i folder. Quindi si calcola l'errore per ogni combinazione ottenendo k errori diversi, successivamente si calcola la media di questi che fungerà da errore totale dell'istanza del modello di KNN. Infine si fa una valutazione dell'errore finale con il test data per le performance del sistema. (si parla di K diversi uno per il modello e uno per la validazione)

inserire immagine

2.2 Regressione KNN

La differenza principale con il KNN classico è che le etichette non sono più un Vero o Falso o più in generale discrete ma si adotta un etichetta rappresentata da un numero reale

$$t \in \mathbb{R}$$

.

insert image

2.2.1 Problema della dimensionalità

Le visualizzazioni a bassa dimensionalità sono fuorvianti. In grandi dimensioni, la maggior parte dei punti sono molto distanti tra loro.
Punti necessari affinché il vicino risieda ad una distanza $< \epsilon$:

- Volume di un cubo di lato 1 è pari a 1^d
- Volume di un cubetto di lato $\epsilon < 1$: $O(\epsilon^d)$;
- Allora: $(\frac{1}{\epsilon})^d$ probabilità per un punto di finire all'interno di ϵ.

2.2.2 Normalizzazione

I vicini più prossimi possono essere sensibili agli intervalli di diverse features.
Soluzione: normalizzare ogni dimensione ad avere media nulla e varianza unitaria.

- Calcolare media: μ_j;
- Calcolare deviazione standard: σ_j;
- Calcolare:
$$\tilde{x}_j = \frac{x_j - \mu_j}{\sigma_j}$$
- Utilizzare $\tilde{x}_j$.

2.2.3 Costo computazionale

- Numero di operazioni al training time: 0;
- Numero di operazioni al test time, per query:
  - Calcolare le distanze euclidee D-dimensionali con N punti dati: $O(ND)$;
  - Ordinare le distanze: $O(N \log(N))$.
- Operazioni da eseguire ad ogni query: costoso;
- Dati da memorizzare: l'intero dataset in memoria.

3 Regressione Lineare

Tale algoritmo ha come principale obiettivo quello di predire target scalari come funzione lineare degli input. Mentre KNN è un algoritmo completo, la regressione lineare esemplifica un approccio modulare:

- Scegliere un modello che descriva le relazioni tra le variabili di interesse;
- Definire una loss function che quantifica quanto male sta andando il fitting dei dati;
- Scegliere un regolarizzatore che definisca quanto si preferiscono modelli differenti;
- Scegliere un modello che minimizzi la loss function e soddisfi i vincoli imposti dal regolarizzatore, possibilmente usando anche un algoritmo di ottimizzazione.

3.0.1 Modello

La predizione del valore di target è una funzione lineare delle features:

$$y = f(x) = \sum_{j=1}^{D} w_j x_j + b$$

- Features: $x = (x^{(1)}, x^{(2)}, ..., x^{(D)}) \in \mathbb{R}^D$;
- Predizione: $y = t \in \mathbb{R}$;
- Vettore di Pesi: w;
- Bias: b.

3.0.2 Loss Function e Funzione di Costo

La loss function definisce quanto è dannosa una predizione y rispetto al target t.

- **Squared Error Loss Function**: errore quadratico tra predizione e target:

$$L(y, t) = \frac{1}{2}(y - t)^2$$

  - Residuo da minimizzare: $y - t$.

- **Funzione di costo**: loss function mediata su tutti i campioni del training set:

$$J(x, t) = \frac{1}{2N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)})^2 = \frac{1}{2N} \sum_{i=1}^{N} \left(w^T x^{(i)} + b - t^{(i)}\right)^2$$

- Costo empirico.

4 Decision Trees

Gli alberi di decisione suddividono lo spazio delle feature tramite una struttura ad albero. Ogni nodo interno effettua un test su un attributo, mentre ogni foglia predice un output.

4.1 Algoritmo ID3

L'algoritmo ID3 utilizza un approccio top-down per costruire l'albero.

1. Inizia con l'intero training set e un albero vuoto.
2. Seleziona la miglior feature/attributo.
3. Effettua lo split e ripete il processo in modo ricorsivo.