Introduzione all'apprendimento automatico

Andrea Asperti

Department of Computer Science, University of Bologna
Mura Anteo Zamboni 7, 40127, Bologna, ITALY
asperti@cs.unibo.it

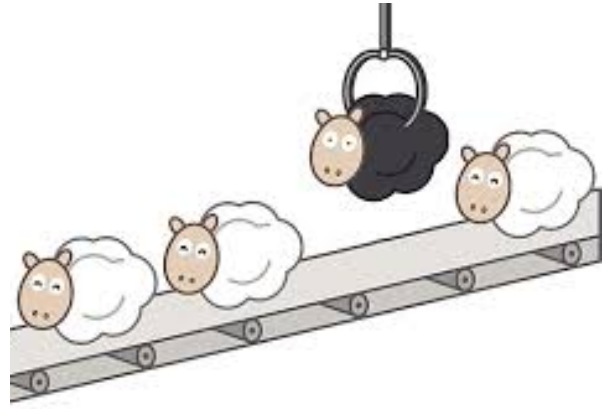
Di cosa si occupa il Machine Learning

Applicazioni

Problemi di difficile soluzione mediante tecniche algoritmiche

- problemi di classificazione

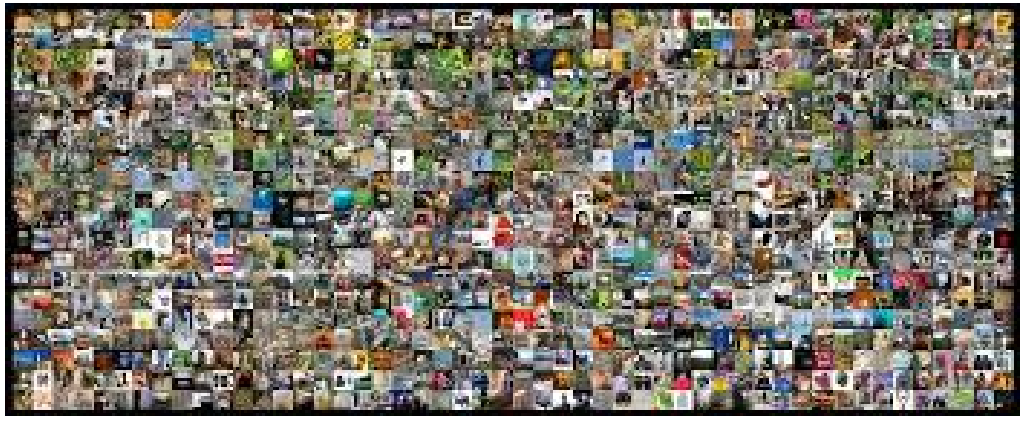
  spam detection, sentiment analysis, transazioni fraudolente, ...
- riconoscimento di immagini/suono
- anomaly detection
- clustering
- techniche generative
- ...



Sempre più problemi sono affrontati con tecniche di Machine Learning

Caratteristiche dei problemi

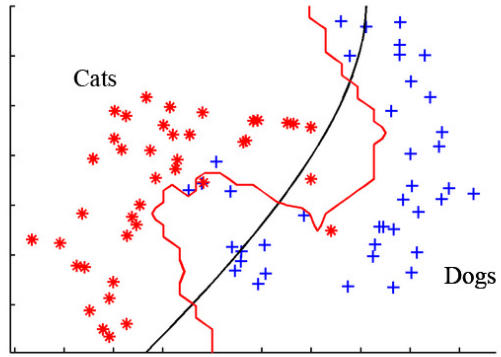
- scarsa conoscenza metateorica
- largo numero di attributi per ogni input
- grande volume di dati disponibili
- evoluzione dinamica del problema



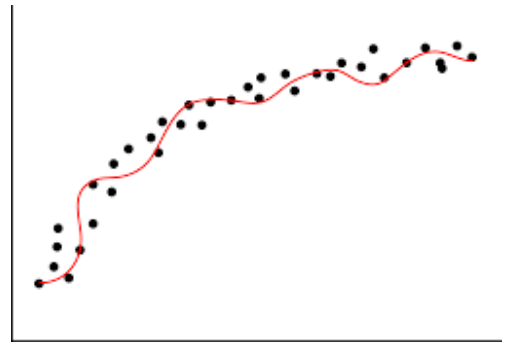
Imagenet database: $\approx$ 10 milioni di immagini

L'approccio del Machine Learning

- definire una classe di **modelli** del problema parametrizzati su di un insieme di **parametri** Θ
- definire una **metrica di valutazione**: una misura dell'**errore** dei vari modelli
- modificare i parametri Θ per **minimizzare** l'errore sui **dati di training**



classification



regression

Il Machine learning è un procedimento di ottimizzazione

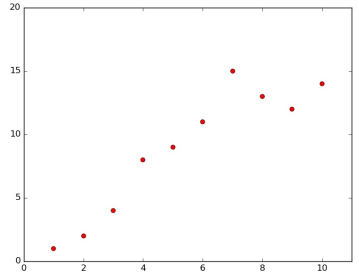
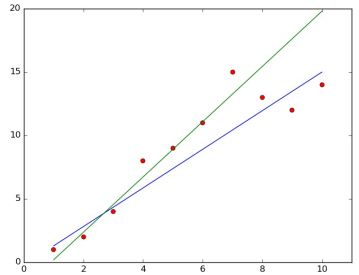
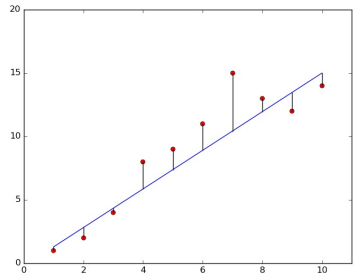
Esempio: regressione

Abbiamo dei punti nel piano e vogliamo tracciare una retta che li approssimi

Step 1 Fissiamo una classe parametrica di modelli.
Per esempio, funzioni lineari: $y = ax + b$;
a e b sono i parametri del modello

Step 2 Scegliamo un modo per confrontare le linee, e decidere quando una è meglio di un'altra
Per esempio, errori quadratici medi (mean squared error - mse)

Step 3 Ottimizziamo i parametri per minizzare l'errore (training).

Perchè parlare di apprendimento?

Il Machine Learning tratta problemi di **ottimizzazione**!
Perchè parliamo di "apprendimento"?

- Il tuning dei parametri si basa su osservazioni: il cosiddetto (**training set**). Stiamo imparando dalla esperienza passata

Perchè parlare di apprendimento?

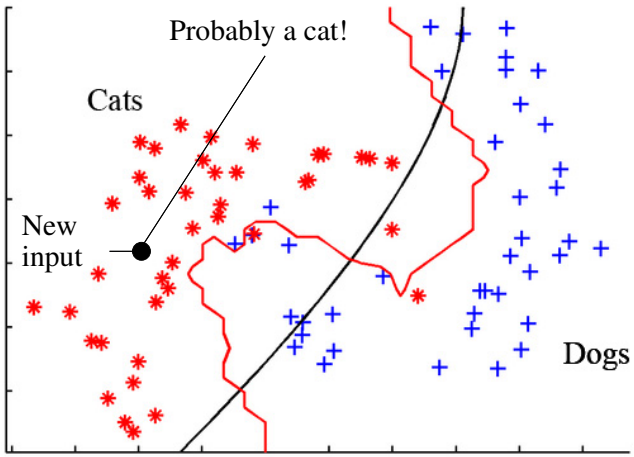
Il Machine Learning tratta problemi di **ottimizzazione**!
Perchè parliamo di "apprendimento"?

- Il tuning dei parametri si basa su osservazioni: il cosiddetto (**training set**). Stiamo imparando dalla esperienza passata

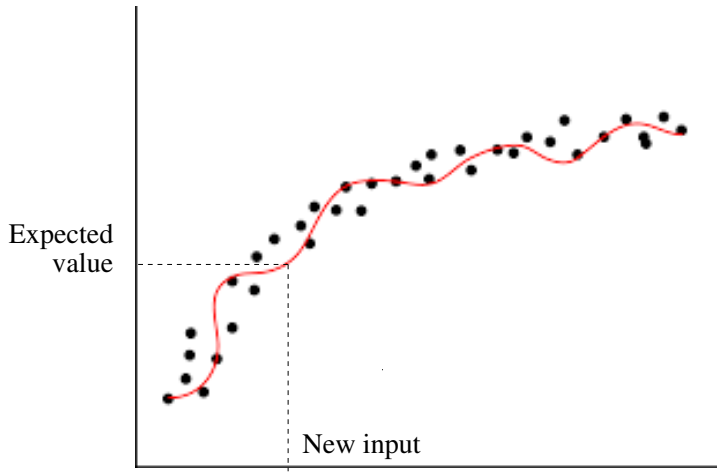
- la soluzione del problema di ottimizzazione non è data in forma analitica (spesso non esiste una soluzione in forma chiusa).

  Si utilizzano invece tecniche **iterative** (ad esempio la discesa del gradiente) per approssimare progressivamente la soluzione. Questo procedimento iterativo di miglioramento può essere inteso come una forma di apprendimento.

Obiettivo: fare predizioni



classificazione

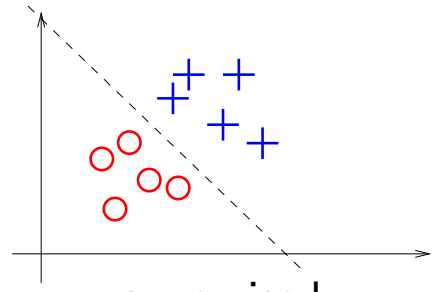


regressione

I modelli possono anche essere utilizzati per acquisire conoscenza sui dati: trovare clusters, stabilire correlazioni, etc.

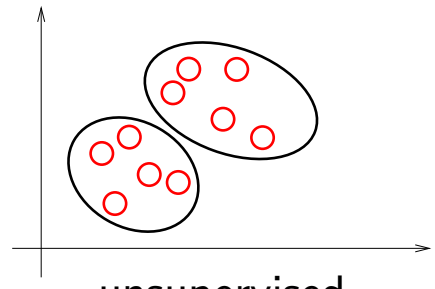
Differenti classi di problemi di apprendimento

- **apprendimento supervisionato**:
  inputs + outputs (etichette)
  - classificazione
  - regressione



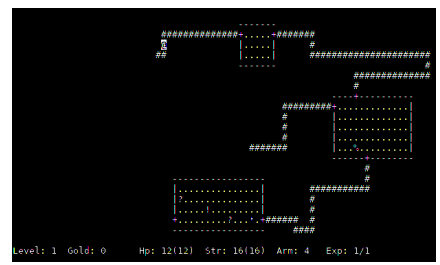
supervised

- **apprendimento non supervisionato**:
  solo inputs
  - clustering
  - analisi delle componenti
  - autoencoding



unsupervised

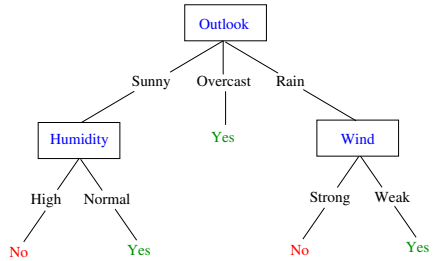
- **apprendimento con rinforzo**
  azioni e ricompense
  - apprendere comportamenti
  - "model-free" planning



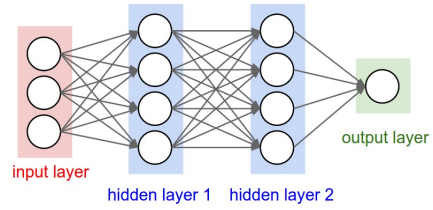
reinforcement

Molte tecniche differenti

- **Modi diversi per definire i modelli**:
  - alberi di decisioni
  - modelli lineari
  - reti neurali
  - ...

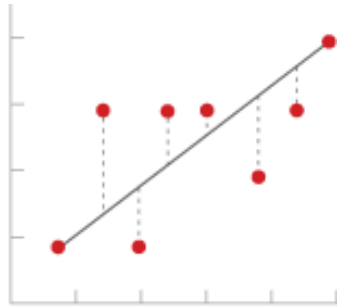


alberi di decisione

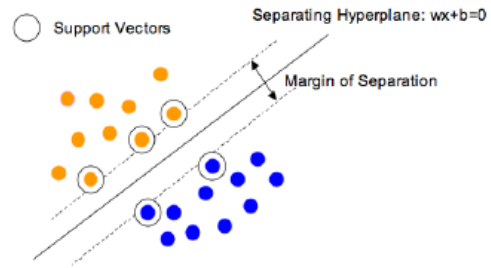


reti neurali

- **Differenti funzioni di loss**:
  - errore quadratico
  - logistic loss
  - cross entropy
  - similarità del coseno
  - margine massimo
  - ...



errore quadratico medio



margine massimo

Features
(Caratteristiche, Attributi)

Feature

Ogni informazione realativa a un dato che ne descrive una sua proprietà è feature (caratteristica).

Le features sono gli input del processo di apprendimento.

L'apprendimento è molto sensibile alla scelta delle features.

Determinare delle buone caratteristiche è complesso (tipicamente richiede una buona conoscenza del dominio).

Esempi di features

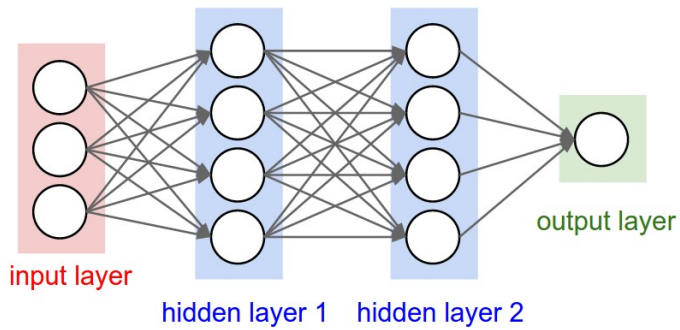
diagnosi medica	profilazione di utenti	previsioni meteorologiche
sintomi	dati demografici	temperatura
condizione del paziente	interessi personali	umidità
cartella clinica	social communities	pressione
risultato di esami	stile di vita	precipitazione
...	...	...

Deep learning

Approccio tradizionale: determinare durante il preprocessing delle buone features, e applicare un metodo semplice ma robusto di apprendimento, ad esempio qualche tecnica lineare.

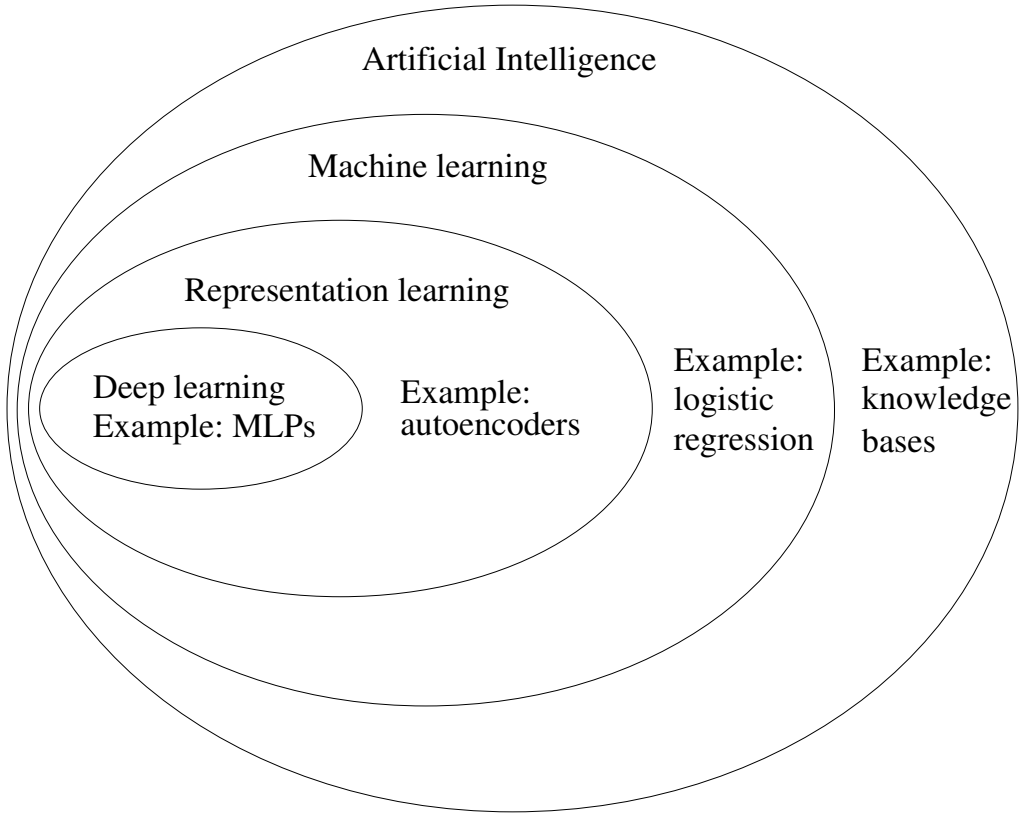
Approccio "deep": fornire in input dati grezzi e delegare alla macchina il compito di sintetizzare nuove features (rappresentatione interna).

Deep learning è implementato attraverso reti neurali profonde
La rete neurale è profonda quando si hanno layers multipli:



Ogni layer sintetizza nuove features in funzione delle precedenti.

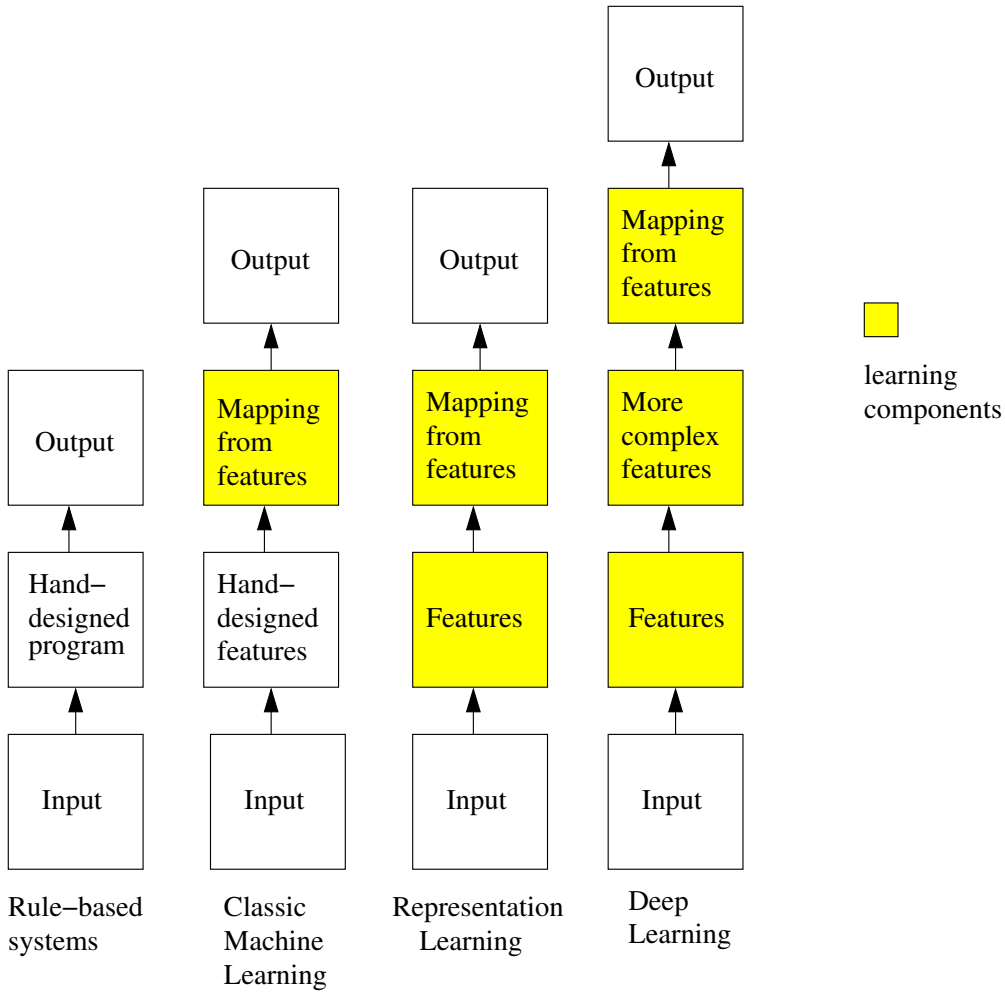
Relazione tra le aree di ricerca



Da "Deep Learning" di Y.Bengio, I.Goodfellow, A.Courville, MIT Press.

Per una prospettiva storica sui different ambiti di ricerca potete consultare il mio blog.

Componenti addestrati mediante apprendimento



Da "Deep Learning" di Y.Bengio, I.Goodfellow, A.Courville, MIT Press.

AI, machine learning, deep learning

- **Sistemi basati su conoscenza**: prendete un esperto, chiedetegli come risolverebbe il problema e poi provate a mimare il suo comportamento mediante regole logiche

AI, machine learning, deep learning

- **Sistemi basati su conoscenza**: prendete un esperto, chiedetegli come risolverebbe il problema e poi provate a mimare il suo comportamento mediante regole logiche

- **Machine-Learning tradizionale**: prendete un esperto, chiedetegli di quali features avrebbe bisogno per risolvere un dato problema, raccogliete un insieme di osservazioni e lasciate alla macchina il compito di apprendere l'associazione

AI, machine learning, deep learning

- **Sistemi basati su conoscenza**: prendete un esperto, chiedetegli come risolverebbe il problema e poi provate a mimare il suo comportamento mediante regole logiche
- **Machine-Learning tradizionale**: prendete un esperto, chiedetegli di quali features avrebbe bisogno per risolvere un dato problema, raccogliete un insieme di osservazioni e lasciate alla macchina il compito di apprendere l'associazione
- **Deep-Learning**: liberatevi dell'esperto

Informazioni Didattiche

- struttura del corso
- libri, tutorials and blogs
- software
- esame
- ricevimento

Struttura del corso

Part I: Machine Learning

- Alberi di decisione
- Metodi Probabilistici
- Naif Bayes
- Maximum Likelihood
- Regressione logistica
- La tecnica del gradiente
- Generativo vs Discriminativo
- Regressione Multinomiale

Part II: Deep Learning

- Neurone Artificale
- Architettura di reti neurali
- Espressività
- Backpropagation
- Reti Convoluzionali
- Autoencoders
- Problemi di visione
- Tecniche generative

Libri

Machine Learning

- C.M.Bishop. Pattern Recognition and Machine Learning. Springer 2006.
- E. Alpaydin, Introduction to Machine Learning, Cambridge University Press, 2010.

Deep Learning

- Y.Bengio, I.Goodfellow and A.Courville. Deep Learning, MIT Press to appear.
- Dive into deep learning (D2L)

Online Tutorials e Blogs

Possibile documentarsi on line (aggiornamento veloce):

- Tensorflow tutorials
- Deep Learning Tutorial. LISA lab. University of Montreal.
- Deep Mind blog
- Open AI blog
- Keras blog
- towardsdatascience
- Machine learning tutorial with Python
- moltissime lezioni e seminari on line su youtube
- moltissimo codice opens source su github
- ...

Software suggerito

Tecniche di Machine learning tradizionali:
- Scikit-learn

Reti Neurali e Deep Learning:
- TensorFlow, Google Brain
- Keras, F.Chollet
- PyTorch, Facebook
- MXNET, Apache
- ...

Image/signal processing:
- OpenCV
- Scipy

Esame

- quiz individuale (40%)
- progetto individuale (60%)

Le specifiche del progetto saranno fornite su virtuale una decina di giorni prima della data del quiz di ogni appello. Dovete caricare la vostra soluzione entro una settimana (le dealines precise saranno fornite assieme alla specifica).

L'argomento del progetto è diverso per ogni appeelo.

Il test e il progetto possono essere affrontati in modo asincrono. L'esito sia per il test che per il progetto resta valido fino a che non si sostiene e consegna una nuova prova.

Ricevimento e altro

Prof.Andrea Asperti
andrea.asperti@unibo.it
Via Zanolini 41 (ex Stazione Veneta)

Ricevimento su appuntamento

- homepage:https://www.unibo.it/sitoweb/andrea.asperti
- my own deeplearningblog
- deepfridays seminars

Alberi di decisione

Approssimazione di funzioni

Train set: un insieme di of esempi di allenamento

$$\langle x^{(i)}, y^{(i)} \rangle$$

dove

- $x^{(i)} \in X$ (insieme degli inputs)
- $y^{(i)} \in Y$ (insieme degli outputs - ground truth)
- i indice dell'istanza dell'esempio di training

Problema: "apprendere" la funzione che mappa $x^{(i)}$ a $y^{(i)}$

Y discreto: problema di classificazione (predizione di una classe)
Y continuo: problema di regressione (predizione di un valore).

Spazio delle ipotesi

Le tecniche di Machine Learning richiedono la scelta preliminare di uno spazio di funzioni H, all'interno del quale cercare la soluzione che meglio approssima i dati di training.

Un modello è un modo di specificare e calcolare una funzione nello spazio delle ipotesi H.

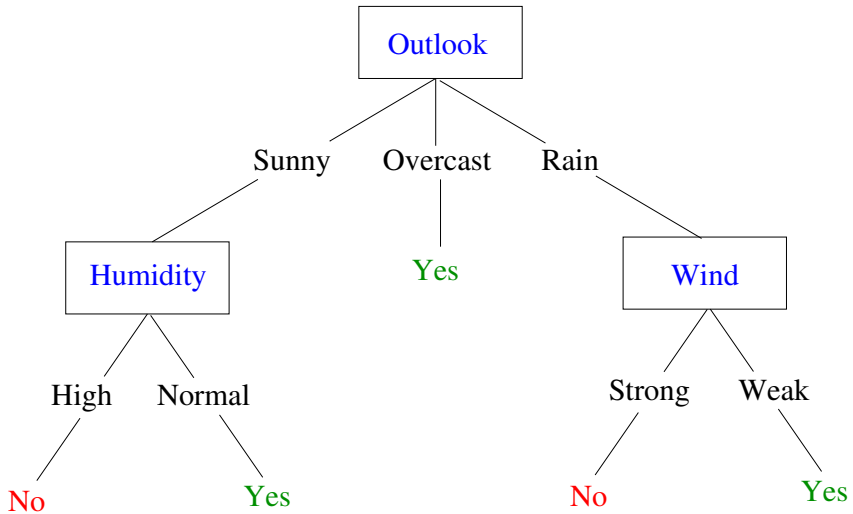
Questioni relative alla espressività:

- abbiamo un modello per ogni funzione h nello spazio delle ipotesi?
- se il modello esiste, è unico?
- se non è unico, abbiamo una preferenza?

Alberi di decisione: un esempio

È un buon giorno per giocare a tennis?

H : Outlook $\times$ Humidity $\times$ Wind $\times$ Temp $\rightarrow$ Play Tennis?

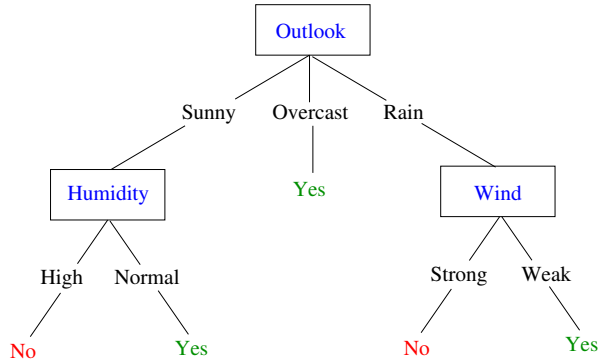


Ogni nodo testa un attributo (feature) X

Ogni arco uscente da un nodo corrisponde a uno dei possibili valori discreti di X

Ogni foglia predice la riposta Y (o un probabilità $P(Y|X)$)

Configurazione del problema



- Insieme di input X:
  ogni istanza $x \in X$ è un vettore di features del seguente tipo:
  $\langle$ Humidity=high, Wind=weak, Outlook=rain, Temp=hot $\rangle$
- Funzione target $f : X \to Y$
  Y assume valori discreti (booleani)
- Spazio delle ipotesi $H = \{h \mid h : X \to Y\}$ (nessuna restrizione)
  - vogliamo modellare $h \in H$ con un albero di decisione
  - ogni istanza $x \in X$ definisce un cammino nell'albero che conduce a una foglia etichettata con $y \in Y$, corrispondente alla soluzione predetta

Play-tennis training set

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Expressivitá del modello

Siano date $X = X_1 \times X_2 \cdots \times X_n$ dove $X_i = \{\text{True,False}\}$

Possiamo esprimere $Y = X_2 \wedge X_5$? oppure $Y = X_4 \vee X_1$?

Possiamo esprimere $Y = X_2 \wedge X_5 \vee (\neg X_3) \wedge X_4 \wedge X_1$?

Expressività del modello (2)

Questioni importanti

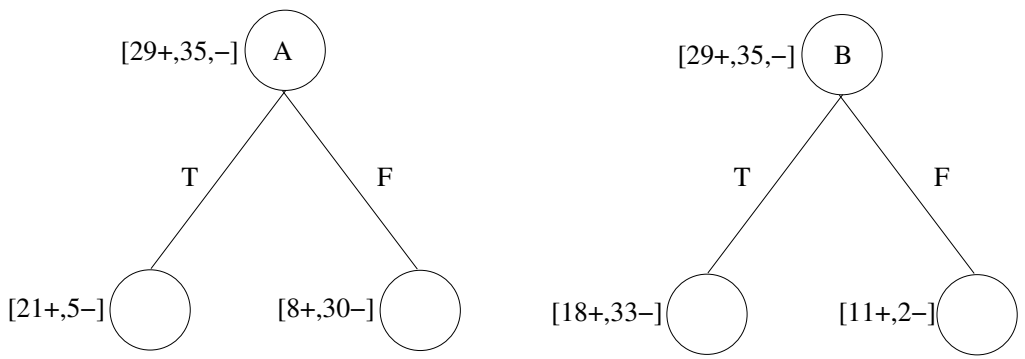
- abbiamo un albero di decisione per ogni funzione h nello spazio delle ipotesi?
- se l'albero esiste è unico?
- se non è unico, abbiamo preferenze?

Costruzione Top-down induttiva

Main loop:

1. assegnare al nodo corrente il "miglior" attributo X_i
2. creare un nodo figlio per ogni possibile valore di X_i e propagare i dati verso i figli a seconda del loro valore
3. per ogni nodo figlio, se tutti i dati del training set associati al nodo hanno una stessa etichetta y, marcare il nodo come foglia con etichetta y, altrimenti ripetere dal primo punto

Ma quale è il "miglior" attributo?



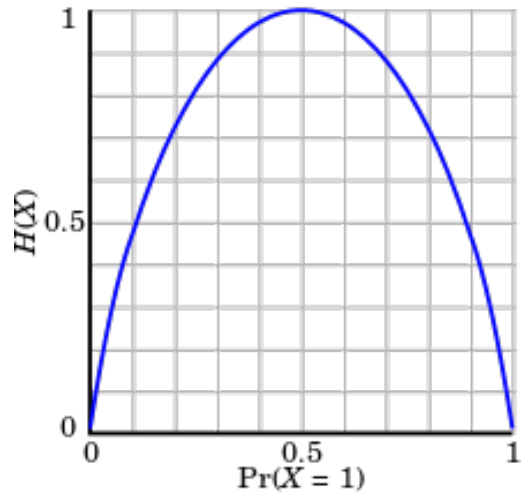
Entropia

L'entropia $H(X)$ di una variabile aleatoria X è

$$H(X) = -\sum_{i=1}^{n} P(X = i) log_2 P(X = i)$$

dove n è il numero dei possibili valori di X.

L'entropia misura il grado di impurità (disordine) della informazione. È massima ($\log n$) quando X è uniformemente distribuita tra tutti i suoi n valori, e minima (0) quando è concentrata su un singolo valore.



Teoria della Informazione (Shannon)

L'Entropia è la quantità media di informazione prodotta da una sorgente stocastica di dati.

L' *informazione* è associata alla *probabilità* di ogni dato (la "sorpresa" associata all'evento):

- un evento con probabilità 1 non trasmette informazione:

$$I(1) = 0$$

- dati due eventi indipendenti con probabilità p_1 e p_2, la probabilità congiunta è $p_1 p_2$ ma l'informazione acquisita è la somma delle informazioni dei due eventi indipendenti:

$$I(p_1 p_2) = I(p_1) + I(p_2)$$

Ci aspettiamo che l'informazione sia antimonotona rispetto alla probabilità ed è quindi naturale definire

$$I(p) = -log(p)$$

Teoria dei codici (Shannon)

L'Entropia misura il numero medio di bits richiesti per trasmettere il valore prodotto da una sorgente stocastica X.

Supponiamo di avere n eventi con la stessa probabilità. Quanto bits occorrono per codificare ogni possibile risultato?

Teoria dei codici (Shannon)

L'Entropia misura il numero medio di bits richiesti per trasmettere il valore prodotto da una sorgente stocastica X.

Supponiamo di avere n eventi con la stessa probabilità. Quanto bits occorrono per codificare ogni possibile risultato?

$$log(n)$$

Teoria dei codici (Shannon)

L'Entropia misura il numero medio di bits richiesti per trasmettere il valore prodotto da una sorgente stocastica X.

Supponiamo di avere n eventi con la stessa probabilità. Quanto bits occorrono per codificare ogni possibile risultato?

$$log(n)$$

Calcoliamo l'entropia. In questo caso,

$$\begin{aligned} H(X) &= -\sum_{i=1}^{n} P(X=i) log_2 P(X=i) \\ &= -\sum_{i=1}^{n} 1/n \, log_2(1/n) \\ &= log(n) \end{aligned}$$

Teoria dei codici (Shannon)

L'Entropia misura il numero medio di bits richiesti per trasmettere il valore prodotto da una sorgente stocastica X.

Supponiamo di avere n eventi con la stessa probabilità. Quanto bits occorrono per codificare ogni possibile risultato?

$$log(n)$$

Calcoliamo l'entropia. In questo caso,

$$\begin{aligned} H(X) &= -\sum_{i=1}^{n} P(X=i) log_2 P(X=i) \\ &= -\sum_{i=1}^{n} 1/n \, log_2(1/n) \\ &= log(n) \end{aligned}$$

Se gli eventi non hanno la stessa probabilità possiamo migliorare la codifica!!

Guadagno Informativo

Entropia di X

$$H(X) = -\sum_{i=1}^{n} P(X = i) log_2 P(X = i)$$

Entropia Condizionale di X dato uno specifico $Y = v$

$$H(X|Y = v) = -\sum_{i=1}^{n} P(X = i|Y = v) log_2 P(X = i|Y = v)$$

Entropia Condizionale di X dato Y
(media ponderata sugli m possibili valori di Y)

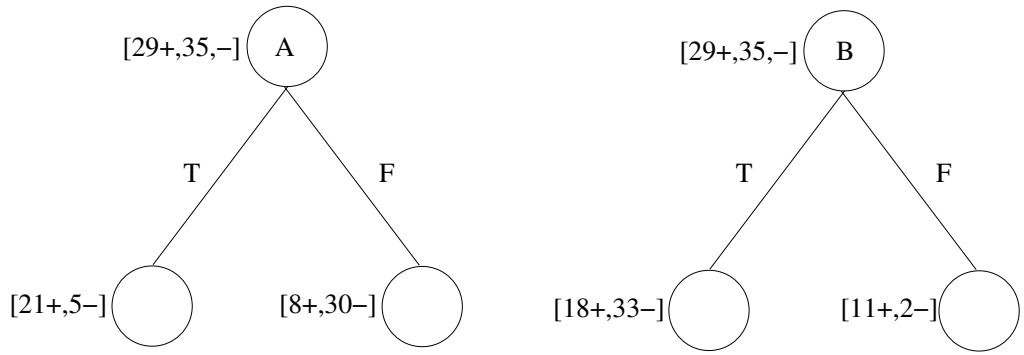
$$H(X|Y) = \sum_{v=1}^{m} P(Y = v) H(X|Y = v)$$

Guadagno Informativo tra X e Y:

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Meglio A o B?

Misuriamo la riduzione di entropia della variabile target Y conseguente alla conoscenza di un qualche attributo X, cioè il guadagno informativo $I(Y, X)$ tra Y e X.



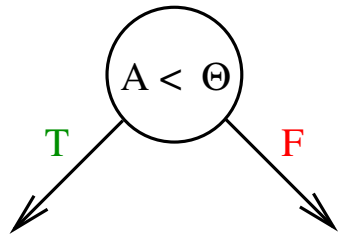
$H(Y) = -(29/64) \cdot log_2(29/64) - (35/64) \cdot log_2(35/64) = .994$
$H(Y|A = T) = -(21/26) \cdot log_2(21/26) - (5/26) \cdot log_2(5/26) = .706$
$H(Y|A = F) = -(8/38) \cdot log_2(8/38) - (30/38) \cdot log_2(30/38) = .742$
$H(Y|A) = .706 \cdot 26/64 + .742 \cdot 38/64 = .726$
$I(Y, A) = H(Y) - H(Y|A) = .994 - .726 = .288$

Per B otteniamo $H(Y|B) = .872$ e $I(Y, B) = .122$

Quindi, A è migliore!

Il caso continuo

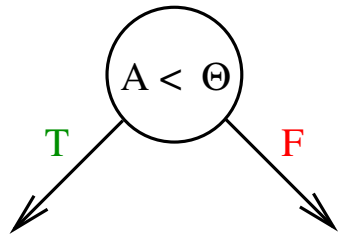
Quando gli attibuti sono continui, prendiamo le decisioni in base a opportune soglie (thresholds):



- confrontiamo le varie soglie in base al loro guadagno informativo
- ma come scegliere le soglie candidate?

Il caso continuo

Quando gli attibuti sono continui, prendiamo le decisioni in base a opportune soglie (thresholds):



- confrontiamo le varie soglie in base al loro guadagno informativo
- ma come scegliere le soglie candidate?
  - sampling a intervalli discreti prefissati
  - ordinare il train set rispetto a un dato attributo e scegliere come soglie i valori medi tra dati consecutivi

Un esempio

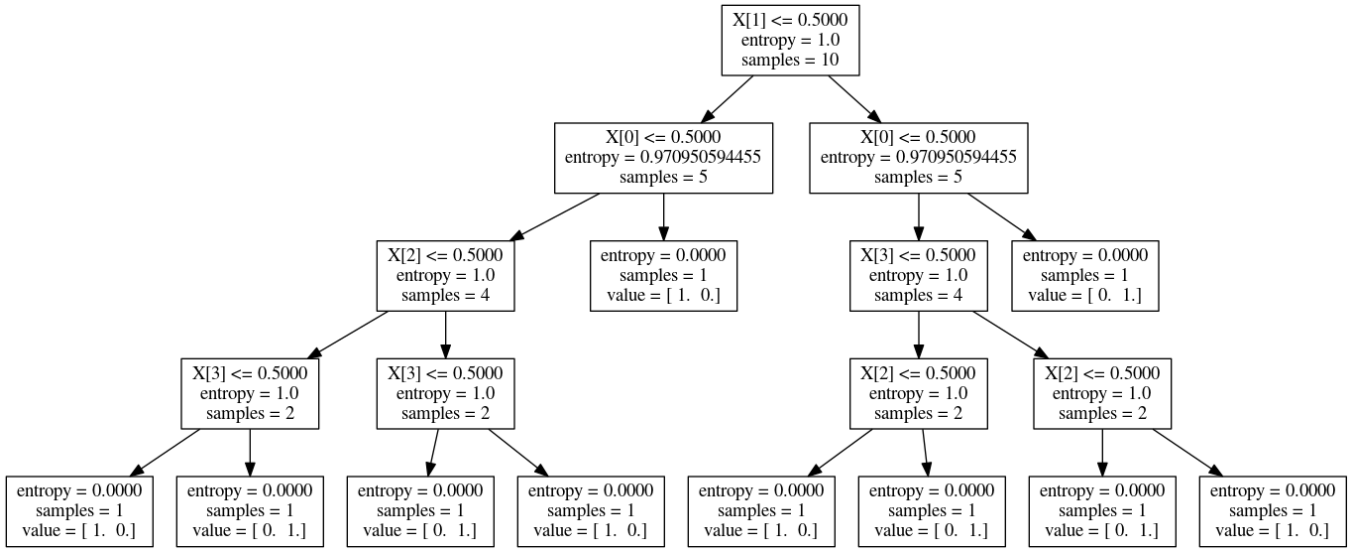
```
from sklearn import tree
X = [[0,0,0,0], [1,0,0,0], [0,0,0,1], [0,0,1,0], [0,0,1,1],
     [0,1,0,0], [0,1,0,1], [1,1,0,1], [0,1,1,0], [0,1,1,1]]
Y = [0,0,1,1,0,0,1,1,1,0]
#label is 0 if X[2]==X[3], 1 otherwise

clf = tree.DecisionTreeClassifier(criterion='entropy',
                                  random_state=0)
clf = clf.fit(X, Y)

print(clf.predict([1,1,1,1]))   #> [1]
print(clf.feature_importances_) #> [0.17, 0.029, 0.4, 0.4]

tree.export_graphviz(clf, out_file='tree.dot')
#dot -T png tree.dot -o tree.png
```

Esempio: albero di decisione



In questo caso, il contenuto informativo delle features individuali non è una buona tecnica di selezione!

Tuttavia, l'"importanza" restituita dal metodo è abbastanza precisa, in quanto questa viene calcolata "ex post" sull'intero albero di decisione.

Overfitting

Overfitting

Consideriamo l'errore relativo all'ipotesi $h \in H$

- sul training set, $error_{train}(h)$
- sull "intero" data set $\mathcal{D}$, $error_{\mathcal{D}}(h)$

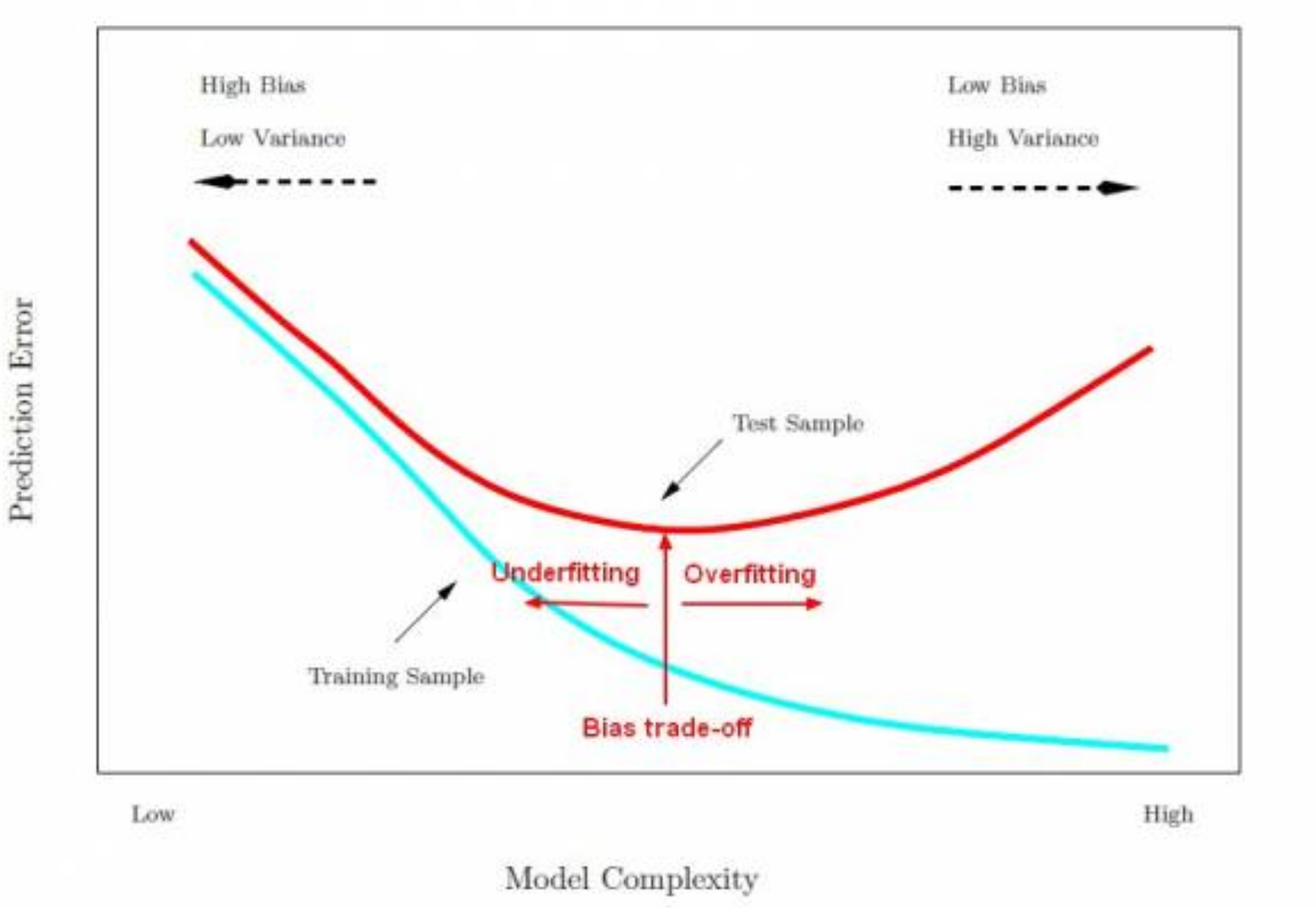
Diciamo che h overfitta il training set se esiste un'altra ipotesi h' such that

$$error_{train}(h) < error_{train}(h')$$

ma

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

Overfitting e complessità del modello



Controllare e evitare overfitting

Problema: non conosciamo $\mathcal{D}$!

Dividere i dati disponibili in due insiemi disgiunti:

training set utilizzato per scegliere l'ipotesi h

validation set utilizzato per misurare accuracy e overfitting h

Tipicamente si riserva una ulteriore porzione, il **test set**, per la valutazione finale.

Evitare overfitting sugli alberi di decisione

- early stopping: terminare la costruzione dell'albero non appena il miglioramento del modello non è più statisticamente significativo. Ad esempio, quando il guadagno informativo è inferiore ad una certa soglia, o il numero dei dati relativo al nodo è troppo piccolo.
- post-pruning: si svilluppa l'intero albero e quindi si procede a potarlo all'indietro.

post-pruning finalizzato alla riduzione dell'errore

Costruire un albero di decisione completo per il training set.

Ripetere la seguente operazione finch´è ogni ulteriore pruning non migliora l'accuratezza della predizione:

1. per ogni sottoalbero, valutare (sul validation set) l'impatto della sua rimozione sulla accuratezza della classificazione
2. effettuare (in modo greedy) il pruning del sottoalbero che ottimizza l'accuracy

Varianti: Gini's impurity

Gini's impurity (coefficiente di Gini) misura la probabilità che un generico elemento sia mal classificato in base alla classificazione corrente (è una possibile alternativa alla nozione di entropia).

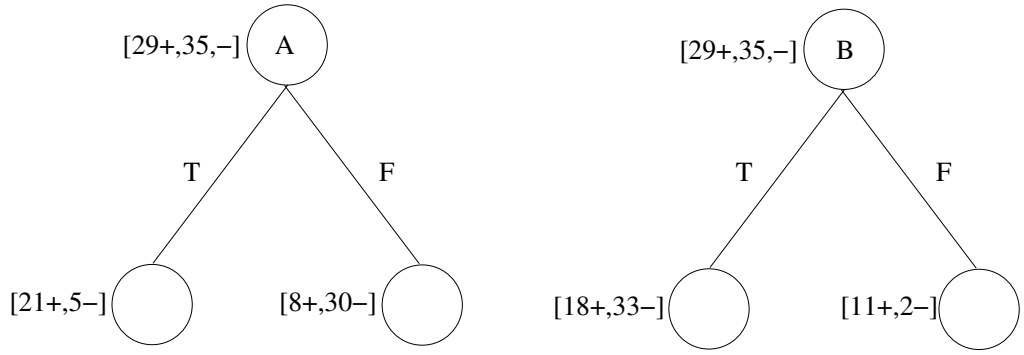
Date m categorie, sua f_i la frazione dei dati con etichetta i. Questa è pari alla probabilità che un qualche input appartenga alla categoria i. La probabilità che venga classificato in modo errato è dunque $1 - f_i$, e il coefficiente di Gini è semplicemente la media pesata di questa quantità su tutte le possibili categorie, cioè

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m}(f_i - f_i^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} f_i^2 = 1 - \sum_{i=1}^{m} f_i^2$$

La metrica è applicata su ogni nodo figlio, e i valori sono sommati in modo pesato, in modo del tutto analogo a quanto avviene per l'entropia, al fine di ottenere una misura della qualità dell'attributo.

Gini: Meglio A o B?

Valutiamo gli split utilizzando il coefficiente di Gini.



Per A:
$I_G(A = T) = 1 - (21/26)^2 - (5/26)^2 = .310$
$I_G(A = F) = 1 - (8/38)^2 - (30/38)^2 = .332$
$I_G(A) = .310 \cdot 26/64 + .332 \cdot 38/64 = .323$

Per B:
$I_G(B = T) = 1 - (18/51)^2 - (33/51)^2 = .456$
$I_G(B = F) = 1 - (11/13)^2 - (2/13)^2 = .260$
$I_G(B) = .456 \cdot 51/64 + .260 \cdot 13/64 = .416$

Quindi, anche con Gini meglio A!

Aspetti positivi degli alberi di decisione

- facili da capire: semplici regole logiche, gli alberi possono essere visualizzati
- poco preprocessing è di solito necessario (attenzione però al bilanciamento delle classi)
- costo predittivo molto basso
- può essere utilizzato sia con features discrete che continue

Aspetti negativi degli alberi di decisione

- rischio elevato di overfitting
- selezione degli attibuti piuttosto instabile al variare del dataset
- facile costruire alberi profondamente sbilanciati, sepcialmente in ppresenza di una classe dominante.

Foreste (Random Forests)

Gli alberi di decisione sono abitualmente utilizzati come componenti delle cosiddette Random Forests, dove sono utilizzati con una tecnica ad ensemble.

Le tecniche ad "ensemble" sfruttano il principio che un gran numero di modelli sufficentemente scorrelati che operano come un comitato di solito forniscono risultati predittivi migliori dei singoli componenti.

Garantire la differenziazione degli alberi:

- **Bagging**: allenare i modelli su sottoinsiemi random (bags) dei dati di input
- **Feature Randomness**: costruire gli alberi a partire da sottoinsiemi random delle features

Sommario delle principali nozioni apprese

- **Problemi di approssimazione**:
  - Features X, labels (categorie) Y
  - Training set $\{\langle x^{(i)}, y^{(i)} \rangle\}$
  - Spazio delle ipotesi $H = \{h \mid h : X \to Y\}$

- **Learning = ricerca/ottimizzazione in H**
  - Differenti obiettivi sono possibili
    - minimizzare l'errore sul training set (accuratezza)
    - tra i vari possibili modelli privilegiare quelli più semplici (rasoio di occam)
    - ...

Approccio probabilistico

Idea principale

Siamo interessati ad approssimare funzioni

Invece di calcolare

$$f : X \rightarrow Y$$

possiamo calcolare

$$p : P(Y|X)$$

Ripasso di Probabilità

- **Eventi**
  - variabili aleatorie discrete e continue
- **Assiomi della Teoria della Probabilità**
  - principi fondamentali della incertezza
- **Eventi indipendenti**
- **Probabilità condizionata**
- **Regola di Bayes**
- **Distribuzione Congiunta di Probabilità**
- **Valore atteso**
- **Independenza, Independenza condizionata**

Variabile Aleatoria (Random variable)

Una variabile aleatoria X denota l'esito di un fenomeno riguardo al quale sussiste incertezza, come ad esempio il risultato di un processo stocastico.

Esempis

- $X = true$ se uno studente scelto a caso in questa aule è biondo
- $X =$ first name of the student
- $X = true$ se due studenti scelti a caso sono nati nello stesso giorno dell'anno

Definiamo $P(X)$ (probabilità di X) come la frazione di volte per cui X é vero, in esecuzioni ripetute dell'esperimento.

Più formalmente

- l'insieme Ω dei possibili esiti di un esperimento casuale è detto spazio campionario (sample space)
  - ad esempio, l'insieme degli studenti in questa stanza
- a variabile aleatoria una funzione misurabile su Ω:
  - discreta: es. genere: $\Omega \rightarrow \{m, f\}$
  - continuoua: es. altezza: $\Omega \rightarrow \mathcal{R}$
- un evento è un qualche sottoinsieme di Ω
  - $\{x \in \Omega | genere(x) = m\}$
  - $\{x \in \Omega | altezza(x) \leq 175cm\}$
- siamo interessati alle probabilità di eventi specifici
  - $P(genere = m)$
- e in probailità condizionate da altri eventi
  - $P(genere = m | altezza \leq 175cm)$

Spazio campionario

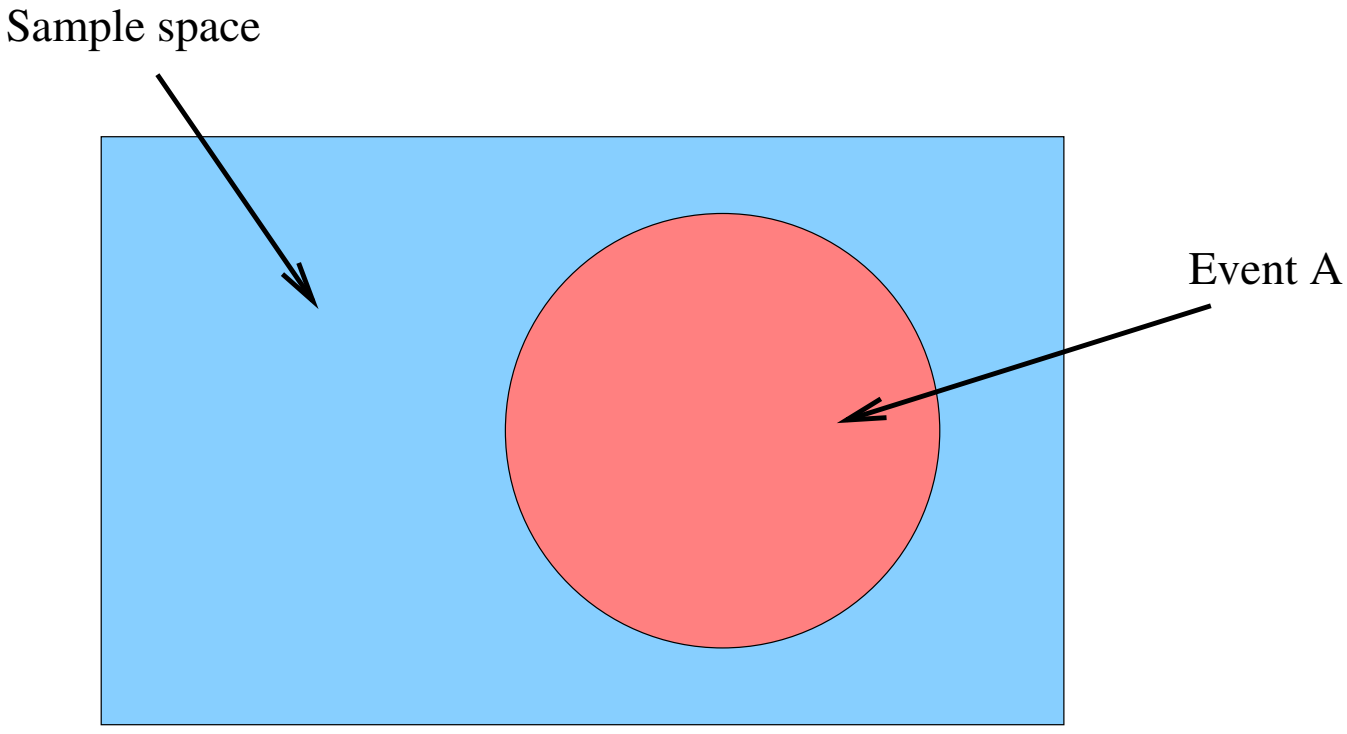
La scelta dello spazio campionario richiede un po' di cautela.

Spesso la nostra intuizione della probabilità si basa sul presupposto che gli eventi elementari nello spazio campionario abbiano la stessa probabilità (cosa non necessarimente vera).

Esempio: lancio di una coppia di dadi.
Il risultato è un numero tra 2 e 12 e potrei utilizzare questi come spazio campionario. Tuttavia, non tutti questi eventi hanno la stessa probabilità.

Meglio considerare con spazio campionario tutte le coppie dei valori dei singoli dadi, e definire quindi una variabile aleatoria X che esprime la somma dei due risultati.

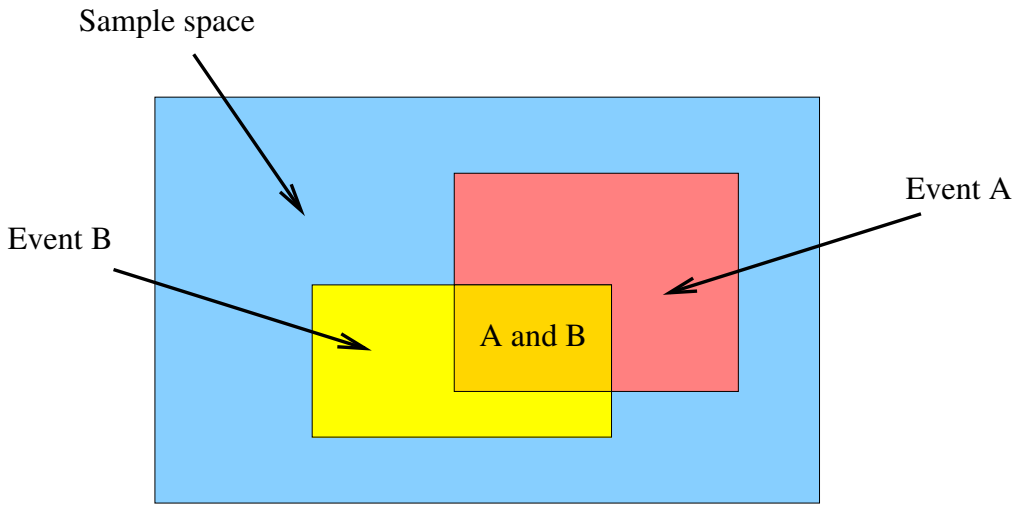
Visualizzazione grafica



La probabilità $P(A)$ dell'evento A è il rapporto tra l'area di A e l'area dell'intero spazio campionario (supposto uniforme).

Assiomi della teoria della probabilità

- $0 <= P(A) <= 1$
- $P(True) = 1$
- $P(False) = 0$
- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



Teoremi derivati

$$P(\neg A) = 1 - P(A)$$

Prova: Sappiamo che

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

e in particolare

$$P(A \vee \neg A) = P(A) + P(\neg A) - P(A \wedge \neg A)$$

ma

$$P(A \vee \neg A) = P(\mathit{True}) = 1 \;\; e \;\; P(A \wedge \neg A) = P(\mathit{False}) = 0$$

e quindi

$$1 = P(A) + P(\neg A) - 0$$

q.e.d.

Un altro teorema interessante

$$P(A) = P(A \wedge B) + P(A \wedge \neg B)$$

Prova: Sappiamo che

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

Siccome

$$A = A \wedge (B \vee \neg B) = (A \wedge B) \vee (A \wedge \neg B)$$

abbiamo:

$$\begin{aligned} P(A) = & \quad P(A \wedge B) + P(A \wedge \neg B) - P((A \wedge B) \wedge (A \wedge \neg B)) \\ = & \quad P(A \wedge B) + P(A \wedge \neg B) - P(\mathit{False}) \\ = & \quad P(A \wedge B) + P(A \wedge \neg B) \end{aligned}$$

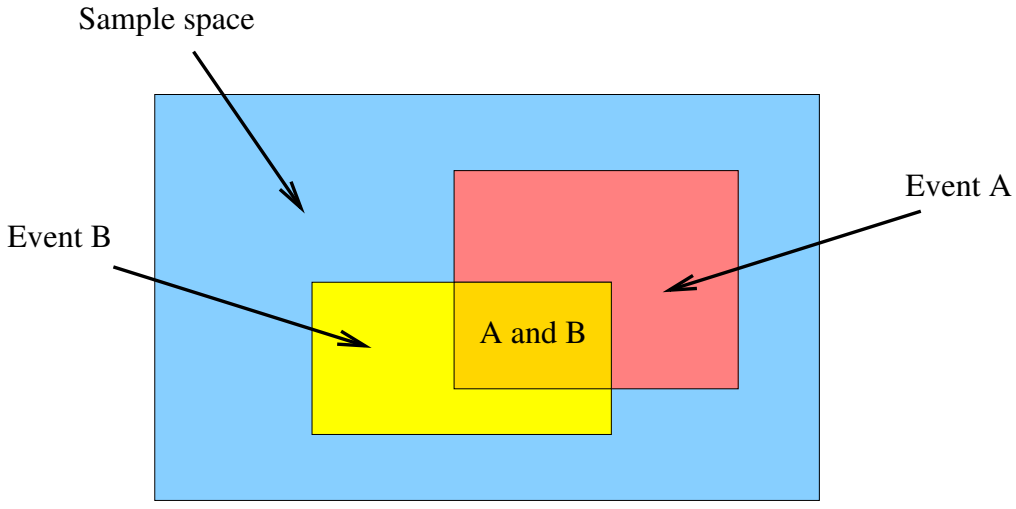
Variabili aleatorie multivalore

A è una variabile aleatoria discreta a k-valori se può assumere esattamente uno dei valori $\{\nu_1, \nu_2, \ldots \nu_k\}$.

$P(A = \nu_i)$ è la probabilità che un elemento dello spazio campionario abbia valore $A = \nu_i$.

$P(A = \nu_i \wedge A = \nu_j) = 0$ se $i \neq j$
$P(A = \nu_1 \vee A = \nu_2 \vee \ldots A = \nu_k) = 1$

Probabilità condizionata



Definizione La probabilità condizionata dell'evento A dato l'evento B è definita come la quanità

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

Corollario: Chain rule

$$P(A \wedge B) = P(B) \cdot P(A|B) = P(A) \cdot P(B|A)$$

Eventi indipendenti

Due eventi A e B sono detti independenti se

$$P(A|B) = P(A)$$

Ovvero, l'evento B non ha influenza sull'evento A.
Come corollario,

$$P(A \wedge B) = P(A) \cdot P(B)$$

Inoltre, siccome

$$P(A \wedge B) = P(B|A) \cdot P(A)$$

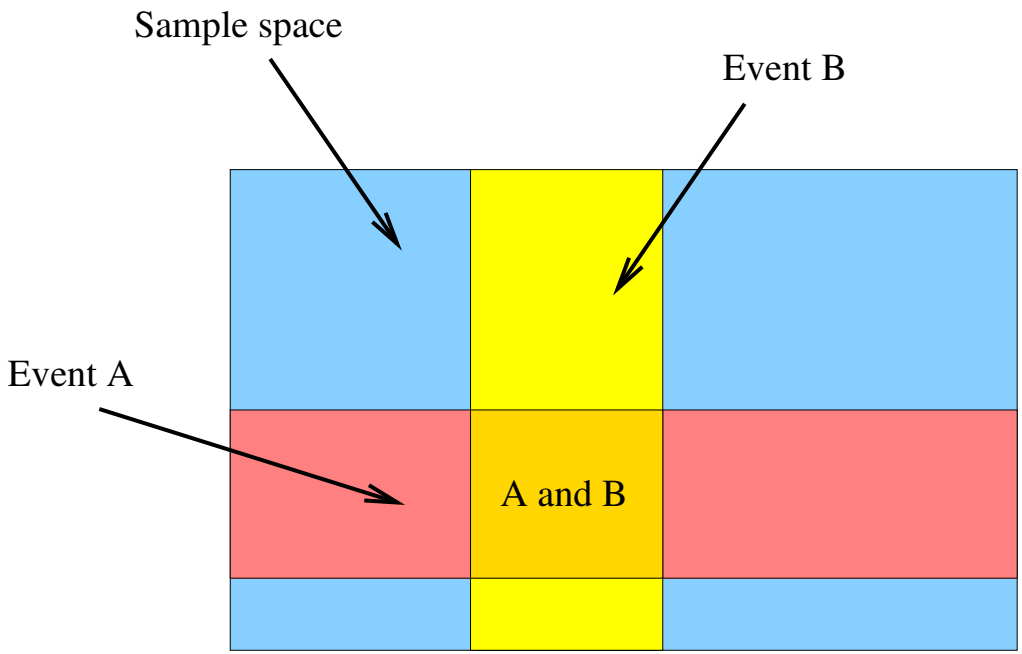
abbiamo anche che

$$P(B|A) = P(B)$$

cioè anche B è indipendente da A.

Intuizione grafica

Due eventi "ortogonali" tra loro:



Bayes' rule

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

Esempio

- in una scuola 60% degli studenti sono ragazzi e il 40% sono ragazze.
- le ragazze indossano in ugual numero gonne e pantaloni
- i ragazzi indossano solo pantaloni

Se vediamo uno studente che indossa pantaloni, che probabilità c'è che sia una ragazza?

Soluzione

La probabilità a priori che lo studente sia una ragazza è

$$P(G) = 2/5$$

La probabilità che uno studente indossi i pantaloni è

$$P(T) = 1/5 + 3/5 = 4/5$$

La probabilità che uno studente indossi i pantaloni sapendo che è una ragazza è

$$P(T|G) = 1/2$$

Quindi, la probabilità che uno studente che indossa pantaloni sia una ragazza è

$$P(G|T) = \frac{P(G) \cdot P(T|G)}{P(T)} = \frac{2/5 \cdot 1/2}{4/5} = 1/4$$

Un'altra formulazione della regola di Bayes

$$P(Y|X) = \frac{P(Y) \cdot P(X|Y)}{P(X)}$$

ed in particolare, per ogni i, j,

$$P(Y = y_i|X = x_j) = \frac{P(Y = y_i) \cdot P(X = x_j|Y = y_i)}{P(X = x_j)}$$

Ma sappiamo che

$$P(X = x_j) = \sum_i P(X = x_j, Y = y_i) = \sum_i P(Y = y_i) \cdot P(X = x_j|Y = y_i)$$

e quindi

$$P(Y = y_i|X = x_j) = \frac{P(Y = y_i) \cdot P(X = x_j|Y = y_i)}{\sum_i P(Y = y_i) \cdot P(X = x_j|Y = y_i)}$$

Posterior, likelihood, prior e marginal

$$\underbrace{P(Y|X)}_{posterior} = \frac{\overbrace{P(X|Y)}^{likelihood} \cdot \overbrace{P(Y)}^{prior}}{\underbrace{P(X)}_{marginal\ likelihood}} = \frac{\overbrace{P(X|Y)}^{likelihood} \cdot \overbrace{P(Y)}^{prior}}{\underbrace{\sum_{Y} P(X|Y) \cdot P(Y)}_{marginal\ likelihood}}$$

"Marginale" in quanto stiamo marginalizzando (integrando) su Y

prossimo argomento

- La distribuzione congiunta
- Classificatori Bayesiani
- Naïve Bayes

La distribuzione congiunta

1. costruire una tabella con tutte le possibili combinazioni dei valori delle features

2. stimare la probabilità per ogni combianzione di valori

gender	working hours	health	prob (params)
F	≤ 40	poor	0.25
F	≤ 40	rich	0.03
F	> 40	poor	0.04
F	> 40	rich	0.01
M	≤ 40	poor	0.33
M	≤ 40	rich	0.10
M	> 40	poor	0.13
M	> 40	rich	0.11

Date n features booleane, dobbiamo stimare $2^n - 1$ parametri.

Uso della distribuzione congiunta

Disponendo della distribuzione congiunta possiamo stimare la probailità di qualunque evento espriminile come combinazione logica delle features

$$P(E) = \sum_{row \in E} P(row)$$

Esempio

Calcoliamo la probabilità $P(M, poor)$

gender	w. hours	wealth	prob.
F	≤ 40	poor	0.25
F	≤ 40	rich	0.03
F	> 40	poor	0.04
F	> 40	rich	0.01
M	≤ 40	poor	0.33
M	≤ 40	rich	0.10
M	> 40	poor	0.13
M	> 40	rich	0.11

$P(M, poor) = 0.33 + 0.13 = 0.46$

Inferenza con la distribuzione congiunta

È altrettanto semplice calcolare la probabilità condizionata di un evento E_1 dato un altro eveento E_2

$$P(E_1|E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{row \in E_1 \wedge E_2} P(row)}{\sum_{row \in E_2} P(row)}$$

Esempio

$$P(M|poor) = \frac{P(M, poor)}{P(poor)}$$

Sappiamo che $P(M, poor) = 0.46$. Calcoliamo $P(poor)$:

gender	w. hours.	wealth	prob.
F	≤ 40	poor	0.25
F	≤ 40	rich	0.03
F	> 40	poor	0.04
F	> 40	rich	0.01
M	≤ 40	poor	0.33
M	≤ 40	rich	0.10
M	> 40	poor	0.13
M	> 40	rich	0.11

$P(poor) = .75$ and $P(M|poor) = 0.46/0.75 = 0.61$

Probabilità condizionata vs. learning

Il nostro punto di partenza era che invece di calcolare

$$f : X \rightarrow Y$$

possiamo calcolare la probabilità condizionata

$$p : P(Y|X)$$

Abbiamo appena visto che la conoscenza della distribuzione congiunta permette di calcolare la proabilità condizionata.

Probabilità condizionata vs. learning

Il nostro punto di partenza era che invece di calcolare

$$f : X \rightarrow Y$$

possiamo calcolare la probabilità condizionata

$$p : P(Y|X)$$

Abbiamo appena visto che la conoscenza della distribuzione congiunta permette di calcolare la proabilità condizionata.

Fine della storia?

Problemi di complessità

Costruiamo la tabella relativa a

$$P(Y = wealth|X_1 = gender, X_2 = orelav.)$$

X_1=gender	X_2=ore lav.	$P(rich\|X_1, X_2)$	$P(poor\|X_1, X_2)$
F	≤ 40	.09	.91
F	> 40	.21	.79
M	≤ 40	.23	.77
M	> 40	.38	.62

Per riempire la tabella dobbiamo stimare $4 = 2^2$ parameters.

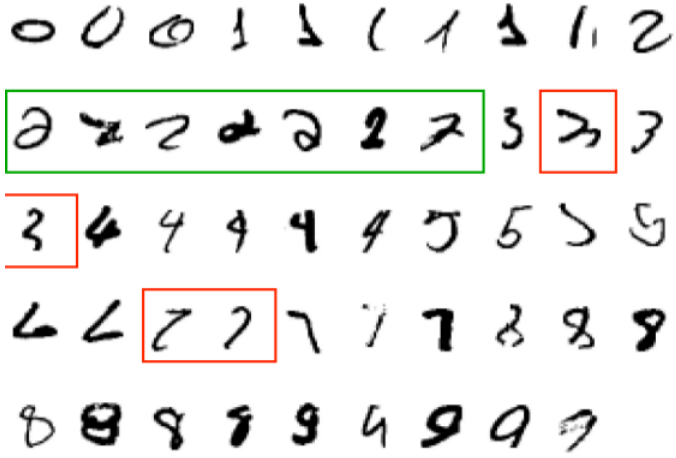
Se avessimo n features $X = X_1 \times X_2, \ldots X_n$ dove ogni X_i è booleana, dovremmo calcolare 2^n parametri.

Questi parametri sono probabilities: per ottenere valori statisticamente significativi dovremmo avere molti esempi per ogni configurazione.

Esempio: il caso di mnist

Modified National Institute of Standards and Technology database

- immagini di 28×28 pixels in scale di grigi che rappresentano esempi di cifre da 0 a 9 scritte a mano
- 60,000 immagini di training più altre 10,000 immagini di testing



Nel caso delle immagini ogni pixel è una feature
Una piccola immagine 28×28 ha già 784 features.

Anche supponendo di avere binarizzato le immagini, la distribuzione congiunta ha 2^{784} righe (più 10 colonne corrispondenti alle classi).

Abbiamo molte più righe che esempi di training. Un sacco di configurazioni non sono coperte dal training set.

Possiamo sfruttare la regola di Bayes?

Sappiamo che

$$P(Y = y_i|X = x_j) = \frac{P(Y = y_i) \cdot P(X = x_j|Y = y_i)}{\sum_i P(Y = y_i) \cdot P(X = x_j|Y = y_i)}$$

Quindi per calcolare $P(Y = y_i|X = x_j)$ è sufficiente calcolare

$$P(Y) \quad \text{e} \quad P(X_1, X_2, \ldots X_n|Y)$$

- quanti parametri per $P(Y)$?
- quanti parametri per $P(X_1, X_2, \ldots X_n|Y)$?

Naïve Bayes

Naïve Bayes suppone che

$$P(X_1, X_2, \ldots X_n|Y) = \prod_i P(X_i|Y)$$

cioè che, dato Y, X_i siano X_j indipendenti tra loro.

Indipendenza condizionale

Due eventi X_i e X_j sono independenti dato Y se

$$P(X_i|X_j, Y) = P(X_i|Y)$$

Esempio 1 Una scatola contiene due monete: una regolare e una con due teste coin (P(H)=1). Scegliamo una moneta, la lanciamo due volte e consideriamo i seguenti eventi:

A = il primo lancio restituisce H
B = il secondo lancio restituisce H
C = la prima moneta è regolare

A and B are NOT independent (prove it), but they are conditionally independent given C.

Esempio 2 Per gli uomini, altezza e vocabolario non sono indipendenti (crescendo sviluppiamo il linguaggio), ma lo sono se ad esempio è nota l'età.

Indipendenza condizionale in Naïve Bayes

$$\begin{aligned} P(X_1, X_2|Y) &= P(X_1|X_2, Y) \cdot P(X_2|Y) && \text{per la chain rule} \\ &= P(X_1|Y) \cdot P(X_2|Y) && \text{per ind. cond.} \end{aligned}$$

In generale,

$$P(X_1, X_2, \ldots X_n|Y) = \prod_i P(X_i|Y)$$

Quanti parametri per $P(X_1, X_2, \ldots X_n|Y)$ (nel caso boolean)?

- senza indipednenza condizionale
- per Naïve Bayes

Naïve Bayes riassunto

Bayes rule

$$P(Y = y_i|X_1, \ldots X_n) = \frac{P(Y = y_i) \cdot P(X_1, \ldots X_n|Y = y_i)}{P(X_1, \ldots X_n)}$$

Naïve Bayes

$$P(Y = y_i|X_1, \ldots X_n) = \frac{P(Y = y_i) \cdot \prod_j P(X_j|Y = y_i)}{P(X_1, \ldots X_n)}$$

Classificazione di un nuovo dato $x^{new} = \langle x_1, \ldots x_n \rangle$

$$Y^{new} = arg\ max_{y_i} P(Y = y_i) \cdot \prod_j P(X_j = x_j|Y = y_i)$$

Tecnica di apprendimento - stima dei parametri

Variabili aleatorie discrete X_i, Y

- **Training**
  - per tutti i valori y_k di Y, stimare

$$\pi_k = P(Y = y_k)$$

  - per ogni possibile valore x_{ij} of X_i, stimare

$$\theta_{ijk} = P(X_i = x_{ij} | Y = y_k)$$

- **Classificazione di** $a^{new} = \langle a_1, \ldots a_n \rangle$

$$\begin{aligned} Y^{new} &= arg\ max_{y_k}\ P(Y = y_k) \cdot \prod_i P(X_i = a_i | Y = y_k) \\ &= arg\ max_k\ \pi_k \cdot \prod_i \theta_{ijk} \end{aligned}$$

supponendo che $a_i = x_{ij}$, cioè che a_i sia il j-th tra i possibili valori discreti dell'attributo X_i)

Stima dei parametri

Maximum likelihood estimates (MLE's):

$$\pi_k = P(Y = y_k) = \frac{\#\mathcal{D}\{Y = y_k\}}{|\mathcal{D}|}$$

$$\theta_{ijk} = P(X = x_{ij}|Y = y_k) = \frac{\#\mathcal{D}\{X_i = x_{i,j} \wedge Y = y_k\}}{\#\mathcal{D}\{Y = y_k\}}$$

Esempio: è un buon giorno per giocare a tennis?

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Priors

$\pi_{Yes} = 9/14 = .64$

$\pi_{No} = 5/14 = .36$

Calcolo di θ (Yes)

Outlook	Temp	Humidity	Wind	Play
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes

Outlook

$\theta_{Sunny,Yes} = 2/9$

$\theta_{Overc.,Yes} = 4/9$

$\theta_{Rain,Yes} = 3/9$

Temp

$\theta_{Hot,Yes} = 2/9$

$\theta_{Mild,Yes} = 4/9$

$\theta_{Cool,Yes} = 3/9$

Humidity

$\theta_{High,Yes} = 3/9$

$\theta_{Normal,Yes} = 6/9$

Wind

$\theta_{Weak,Yes} = 6/9$

$\theta_{Strong,Yes} = 3/9$

Calcolo di θ (No)

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Rain	Cool	Normal	Strong	No
Sunny	Mild	High	Weak	No
Rain	Mild	High	Strong	No

Outlook

$\theta_{Sunny,No} = 3/5$

$\theta_{Overc.,No} = 0$

$\theta_{Rain,No} = 2/5$

Temp

$\theta_{Hot,No} = 2/5$

$\theta_{Mild,No} = 2/5$

$\theta_{Cool,No} = 1/5$

Humidity

$\theta_{High,No} = 4/5$

$\theta_{Normal,No} = 1/5$

Wind

$\theta_{Weak,No} = 2/5$

$\theta_{Strong,No} = 3/5$

Predizione

Nuova istanza (esempio):

Outlook=Sunny,Temp.=Cool,Humidity=High,Wind=Strong

Dobbiamo calcolare

$$arg\ max_{y \in yes,no} p(y) \cdot p(sunny|y) \cdot p(cool|y) \cdot p(high|y) \cdot (strong|y)$$

ovvero, dobbiamo confrontare

$$\pi_{yes} \cdot \theta_{sunny,yes} \cdot \theta_{cool,yes} \cdot \theta_{high,yes} \cdot \theta_{strong,yes}$$
$$= 9/14 \cdot 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = .0053$$

$$\pi_{no} \cdot \theta_{sunny,no} \cdot \theta_{cool,no} \cdot \theta_{high,no} \cdot \theta_{strong,no}$$
$$= 5/14 \cdot 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = .0205$$

NO!

prossimo argomento

- Tecniche generative
- Limitazioni e cautele per Naïve Bayes

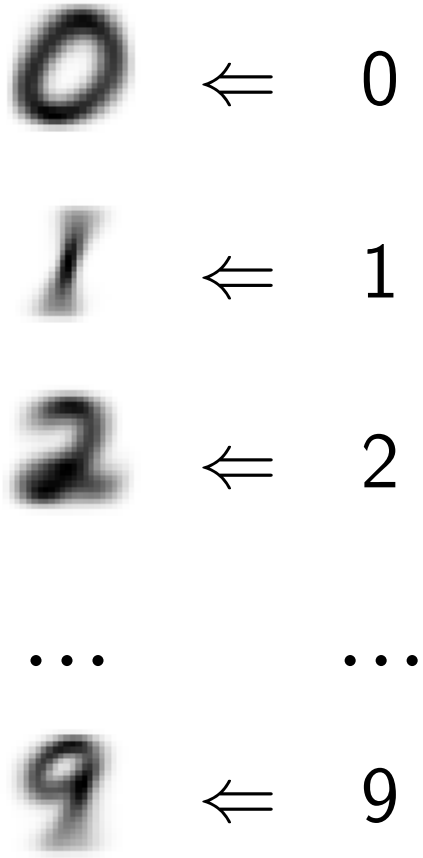
La natura generativa di Naïve Bayes

Siamo interessati a calcolare

$$P(Y = y_i | X_1, \ldots X_n)$$

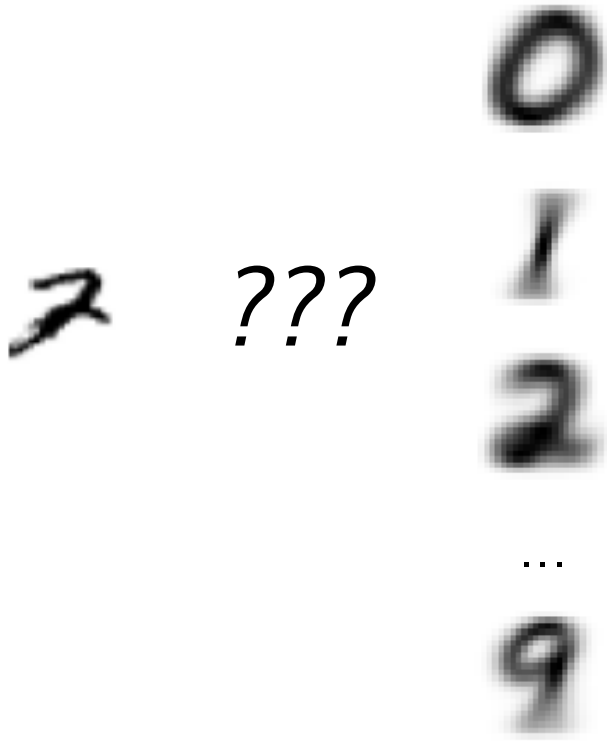
La regola di Bayes ci permette di rovesciare il problema, cercando di stimare la **distribuzione dei dati, data la categoria**

$$P(X_1, \ldots X_n | Y = y_i)$$



Classificazione di un nuovo dato

Classifichiamo una nuova istanza chiedendoci a quale delle varie distribuzioni dei dati che abbiamo stimato è più probabile che appartenga



Congiunta vs. Naïve

Saremmo interessati alle distribuzioni congiunte

$$P(X_1, \ldots X_n | Y = y_i)$$

Tuttavia abbiamo visto che questo in generale non è fattibile per questioni di complessità.
Quindi adottiamo un approccio naïve assumendo che le features siano indipendenti tra loro (data la classe), riducendo il problema alla stima di

$$P(X_j | Y = y_i)$$

per tutte le features $j = 1, \ldots, n$.

Cautele (1)

In alcuni case, MLE for $P(X_i|Y)$ può essere 0
e.g. $P(Play = No|Outlook = Overcast)$

- Perchè ci dovremmo preoccupare?

- Come possiamo evitare il problema?

Cautele (2)

Naïve Bayes suppone che gli eventi siano indipendenti tra loro (dato Y).

Ma se non fosse così?

Limitazioni

Suponiamo di avere un insieme di immagini random, con pixels a valori 0 o 1.

Sciegliamo due pixel p_1 e p_2. Vogliamo classificare l'immagine nella categoria A se $p_1 == p_2$ e nella categoria B altrimenti .

Sembra un problema di classificazione abbastanza semplice. Proviamo a usare Naïve Bayes:

Quanto vale $P(p_1 = 1|A)$?

Quanto vale $P(p_1 = 1|B)$?

Quanto vale $P(p_2 = 1|A)$?

Quanto vale $P(p_2 = 1|A)$?

Quindi? ...

Riguardo alla Maximum Likelihood

Stima dei parametri

Maximum likelihood estimates (MLE's):

$$\pi_k = P(Y = y_k) = \frac{\#\mathcal{D}\{Y = y_k\}}{|\mathcal{D}|}$$

$$\theta_{ijk} = P(X_i = x_{ij}|Y = y_k) = \frac{\#\mathcal{D}\{X_i = x_{i,j} \wedge Y = y_k\}}{\#\mathcal{D}\{Y = y_k\}}$$

Stima dei parametri

Maximum likelihood estimates (MLE's):

$$\pi_k = P(Y = y_k) = \frac{\#\mathcal{D}\{Y = y_k\}}{|\mathcal{D}|}$$

$$\theta_{ijk} = P(X_i = x_{ij}|Y = y_k) = \frac{\#\mathcal{D}\{X_i = x_{i,j} \wedge Y = y_k\}}{\#\mathcal{D}\{Y = y_k\}}$$

Ma perchè ?

Esempio

Supponiamo di lanciare una moneta (forse falsa) per 10 volte.
Osserviamo 6 Heads (H) e 4 Tails (T).

Si potrebbe "naturalmente" concludere che $P(H) = .6$ e $P(T) = .4$.

Esempio

Supponiamo di lanciare una moneta (forse falsa) per 10 volte.
Osserviamo 6 Heads (H) e 4 Tails (T).

Si potrebbe "naturalmente" concludere che $P(H) = .6$ e $P(T) = .4$.

Supponiamo tuttavia che esistano solo due possibilità:

- la moneta è regolare, dunque $P(H) = .5$ e $P(T) = .5$
- la moneta è falasa, con probabilità $P(H) = .7$ and $P(T) = .3$

Quale delle due ipotesi ha maggiore probabilità (likelihood) in base alla sequenza osservata, e perchè?

Eventi indipendenti e equiprobabili

Distribuzione di Bernoulli (es. lancio di una moneta)

due possibili esiti 0 e 1 con probabilità θ e $1 - \theta$.

Sia X^n il numero di 0 in una sequenza di n lanci
X^n segue una distribuzione binomiale

$$P(X^n = \alpha_0|\theta) = \binom{n}{\alpha_0} \cdot \theta^{\alpha_0} \cdot (1-\theta)^{\alpha_1}$$

dove $\alpha_0 = n - \alpha_1$ è il numero di 0 nella sequenza $(\alpha_0 + \alpha_1 = n)$

Tornando all'esempio

$n = 10$, $\alpha_0 = 6$ e $\alpha_1 = 4$.

Se $\Theta = .5$ e $1 - \Theta = .5$

$$P(X^n = 6|\theta) = \binom{n}{\alpha_0} \cdot \theta^{\alpha_0} \cdot (1-\theta)^{\alpha_1} = \binom{10}{6} \cdot \left(\frac{1}{2}\right)^6 \cdot \left(\frac{1}{2}\right)^4 = .205$$

Se $\Theta = .7$ e $1 - \Theta = .3$

$$P(X^n = 6|\theta) = \binom{n}{\alpha_0} \cdot \theta^{\alpha_0} \cdot (1-\theta)^{\alpha_1} = \binom{10}{6} \cdot \left(\frac{7}{10}\right)^6 \cdot \left(\frac{3}{10}\right)^4 = .200$$

Quindi, $\Theta = .5$ è leggermente più probabile che $\Theta = .7$.

Quale è il valore più probabile per Θ?

Calcolo di MLE

$$\hat{\theta} = arg\ max_\theta\ P(X^n = \alpha_0|\theta) =$$
$$= arg\ max_\theta\ \theta^{\alpha_0} \cdot (1-\theta)^{\alpha_1}$$

In modo equivalente possiamo cercare θ in modo che massimizzi il logaritmo della espressione precedente

$$ln(\theta^{\alpha_0} \cdot (1-\theta)^{\alpha_1}) = \alpha_0 ln(\theta) + \alpha_1 ln(1-\theta)$$

Derivando rispetto a θ otteniamo

$$\frac{\alpha_0}{\theta} - \frac{\alpha_1}{1-\theta} = \frac{\alpha_0 - \alpha_0\theta - \alpha_1\theta}{\theta \cdot (1-\theta)}$$

Che si annula per

$$\theta = \frac{\alpha_0}{\alpha_0 + \alpha_1} = \frac{\alpha_0}{n}$$

Caso multivalore

Distribuzione discreta (es. lancio di un dado)
k possibili esiti $\{1, 2, \ldots, k\}$ con
probabilità θ_i dove $\sum_i \theta_i = 1$.

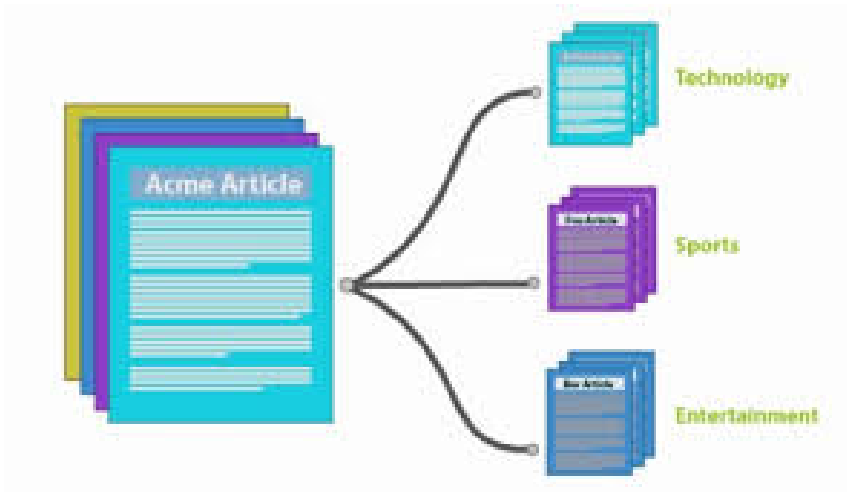
Le sequenze di n lanci seguono una distribuzione multinomiale

$$P(X^n = \alpha_i|\theta) = c_{\alpha_i} \prod_i \theta_i^{\alpha_i}$$

dove α_i é il numero di i nella sequenza e $c_{\overline{\alpha_i}}$ è una costante combinatoria che non dipende da θ

$$\textbf{MLE}: \quad \theta_i = \frac{\alpha_i}{\sum_i \alpha_i} = \frac{\alpha_i}{n}$$

Classificazione di documenti (approccio “bag of words”)



Calssificazione di documenti

Gli eventi elementari sono le parole che occorrono in posizione i nel documento. Su ciascuna definiziamo una variabile aleatoria X_i che assume tanti possibili valori quante sono le parole nel vocabolario

$$\theta_{i,word,\ell} = P(X_i = word | Y = \ell)$$

indica la probability che nel documento della categoria "ℓ" la parola "word" appare in posizione "i"

assumiamo per semplicità che tutti gli eventi siano indipendenti (discutibile) e che abbaino una probabilità indipendente dalla loro poisizione (discutibile)

$$\theta_{i,word,\ell} = \theta_{j,word,\ell} = \theta_{word,\ell}$$

Training e classificazione

Variabili aleatorie discrete X_i, Y

- **Training**
  - per ogni possibile valore y_k di Y, stimiamo (probabilità a priori di appartenere alla categoria y_k

$$\pi_k = P(Y = y_k)$$

  - per ogni possibile valore x_{ij} dell'attributo X_i stimiamo

$$\theta_{ijk} = P(X_i = x_{ij} | Y = y_k)$$

- **Classificazione di** $a^{new} = \langle a_1, \ldots a_n \rangle$ (sequenza di n parole)

$$\begin{aligned} Y^{new} &= arg\ max_{y_k}\ P(Y = y_k) \cdot \prod_i P(X_i = a_i | Y = y_k) \\ &= arg\ max_k\ \pi_k \cdot \prod_i \theta_{ijk} \end{aligned}$$

dove $x_{ij} = a_i$

Stima dei parametri

Maximum likelihood estimates (MLE's):

- $\pi_k = P(Y = y_k)$
  frazione dei documenti nella categoria y_k
- $\theta_{i,word,k} = \theta_{word,k} = P(X = word | Y = y_k)$
  frequenza della parola "word" nei documenti della categoria y_k

Log likelihood

Invece di

$$\begin{aligned} Y^{new} &= arg\ max_{y_k}\ P(Y = y_k) \cdot \prod_i P(X_i = w_j | Y = y_k) \\ &= arg\ max_k\ \pi_i \cdot \prod_i \theta_{ijk} \end{aligned}$$

possiamo calcolare

$$\begin{aligned} Y^{new} &= arg\ max_{y_k}\ log(P(Y = y_k) \cdot \prod_i P(X_i = w_j | Y = y_k)) \\ &= arg\ max_k\ log(\pi_k) + \sum_i log(\theta_{ijk}) \end{aligned}$$

Inoltre, se $\theta_{ijk} = \theta_{i'jk} = \theta_{jk}$

$$\sum_i log(\theta_{ijk}) = \sum_j n_j \cdot log(\theta_{jk})$$

dove n_j è il numero di occorrenze della parola w_j nel documento da classificare.

Prodotto scalare, correlazione, similutidine del coseno

Consideriamo dei vettori della stessa dimensione del vocabolario.

Training

Per ogni categoria k dei documenti costruiamo un vettore "spettrale"

$$s_k = \langle log(\theta_{jk}) \rangle_{j \in words}$$

θ_{jk} = frequenza della parola j nei documenti della categoria k.

Classificazione

Dato un nuovo documento, calcoliamo un vettore

$$d = \langle n_j \rangle_{j \in words}$$

e classifichiamo il documento in base alla categoria il cui spettro è maggiormente correlato a questo vettore

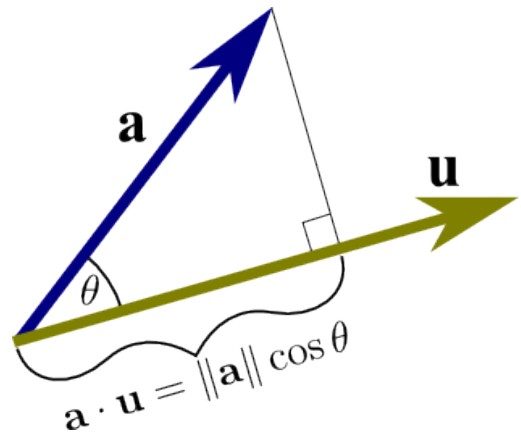
$$arg\ max_k \quad \underbrace{d \cdot s_k}_{correlazione} = \sum_j d_j \cdot s_{jk}$$

Prodotto scalare, geometricamente e analiticamente

Definizione geometrica

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}| cos(\theta)$$

dove θ è l'angolo tra i due vettori



Definizione analitica

dati $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ e $\mathbf{b} = (b_1, b_2, \ldots, b_n)$

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$$

La similitudine del coseno (cosine similarity) tra $\mathbf{a}$ e $\mathbf{b}$ è il prodotto scalare normalizzato rispetto alla lunghezza dei vettori:

$$S_C(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} = cos(\theta)$$

Equivalenza della definizione geometrica e analitica

Per la regola del coseno (una generalizzaione del teorema di Pitagora),

$$|\mathbf{a} - \mathbf{b}|^2 = |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2|\mathbf{a}||\mathbf{b}|cos(\theta)$$

quindi

$$\mathbf{a} \cdot \mathbf{b} = \frac{|\mathbf{a}|^2 + |\mathbf{b}|^2 - |\mathbf{a} - \mathbf{b}|^2}{2}$$

Analiticamene (caso planare):

sia $\mathbf{a} = (a_1, a_2)$ e $\mathbf{b} = (b_1, b_2)$; abbiamo

$$\mathbf{a} \cdot \mathbf{b} = \frac{a_1^2 + a_2^2 + b_1^2 + b_2^2 - (a_1 - b_1)^2 - (a_2 - b_2)^2}{2} = a_1 b_1 + a_2 b_2$$

- La natura lineare di Naïve Bayes

  (caso booleano)

La natura lineare di Naïve Bayes (1)

X_i, Y booleani

Classificazione di $\vec{x} = \langle x_1, \ldots, x_n \rangle$:

$$\frac{P(Y=1|X_1 \ldots X_n = \vec{x})}{P(Y=0|X_1 \ldots X_n = \vec{x})} = \frac{P(Y=1)\prod_i P(X_i = x_i|Y=1)}{P(Y=0)\prod_i P(X_i = x_i|Y=0)} \geq 1$$

Passando ai logaritmi

$$log\frac{P(Y=1)}{P(Y=0)} + \sum_i log\frac{P(X_i = x_i|Y=1)}{P(X_i = x_i|Y=0)} \geq 0$$

La natura lineare di Naïve Bayes (2)

$$log \frac{P(Y=1)}{P(Y=0)} + \sum_i log \frac{P(X_i = x_i|Y=1)}{P(X_i = x_i|Y=0)} \geq 0$$

Utilizziamo il fatto che per una variabile boolena x,

$$f(x) = x * f(1) + (1-x) * f(0)$$

Posto $\theta_{ik} = P(X_i = 1|y = k)$ (da cui $P(X_i = 0|y = k) = 1 - \theta_{ik}$) la disequazione precedene diventa:

$$log \frac{P(Y=1)}{P(Y=0)} + \sum_i x_i \cdot log \frac{\theta_{i1}}{\theta_{i0}} + \sum_i (1-x_i) \cdot log \frac{1-\theta_{i1}}{1-\theta_{i0}} \geq 0$$

lineare nelle features x_i.

Tecniche di Classificazione lineari

Algoritimi di classsificazione basati su una combinazione lineare delle features.

Ogni caratteristica del dato è valutata independentemente dalle altre e contribuisce al risultato in modo lineare, con un peso opportuno. Questo peso è un parametro del modello che deve essere stimato.

Naïve Bayes Gaussiano

Che fare quando le features X_i sono continue?

Per esempio, X_i potrebbe essere l'altezza, o l'età o la retibuzione annuale di un individuo, oppure l'intensità di un pixel in una immagine, o la coordinata spaziale di un particella, etc.

Per utilizzare Naïve Bayes dobbiamo calcolare $P(X_i|Y)$, ma quando X_i è continua, le probabilità puntuali sono nulle, pper cui si parla di **densità** delle distribuzioni

Un approccio tradizionale consiste nel supporre che $P(X_i|Y)$ abbia una distribuzione Gaussiana (detta anche Normale).

Distribuzione Gaussiana

Densità di probabilità (con integrale = 1)

$$p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

valore medio

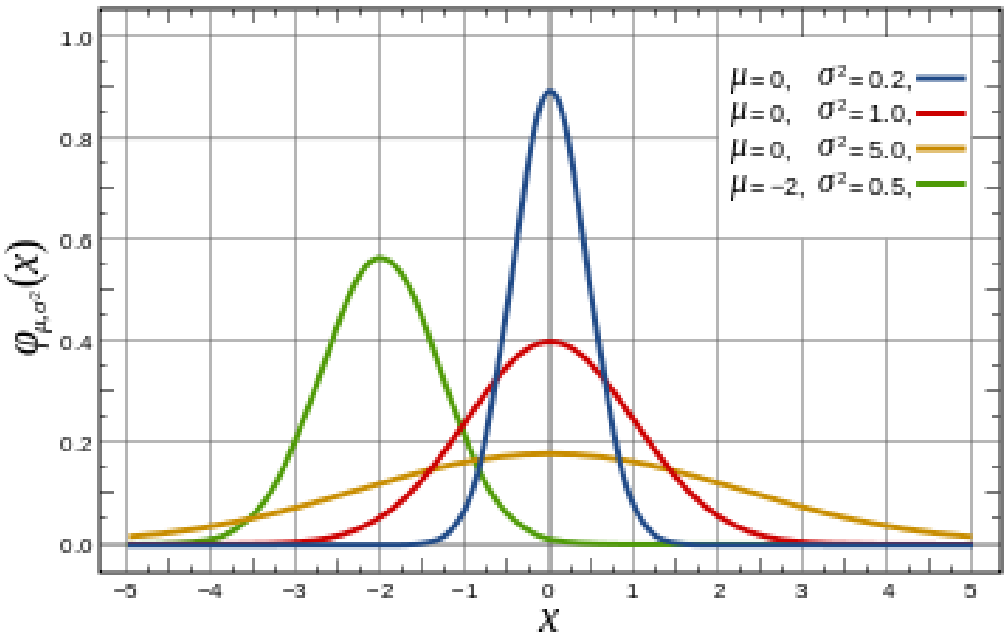
$E[X] = \mu$

varianza

$Var[X] = \sigma^2$

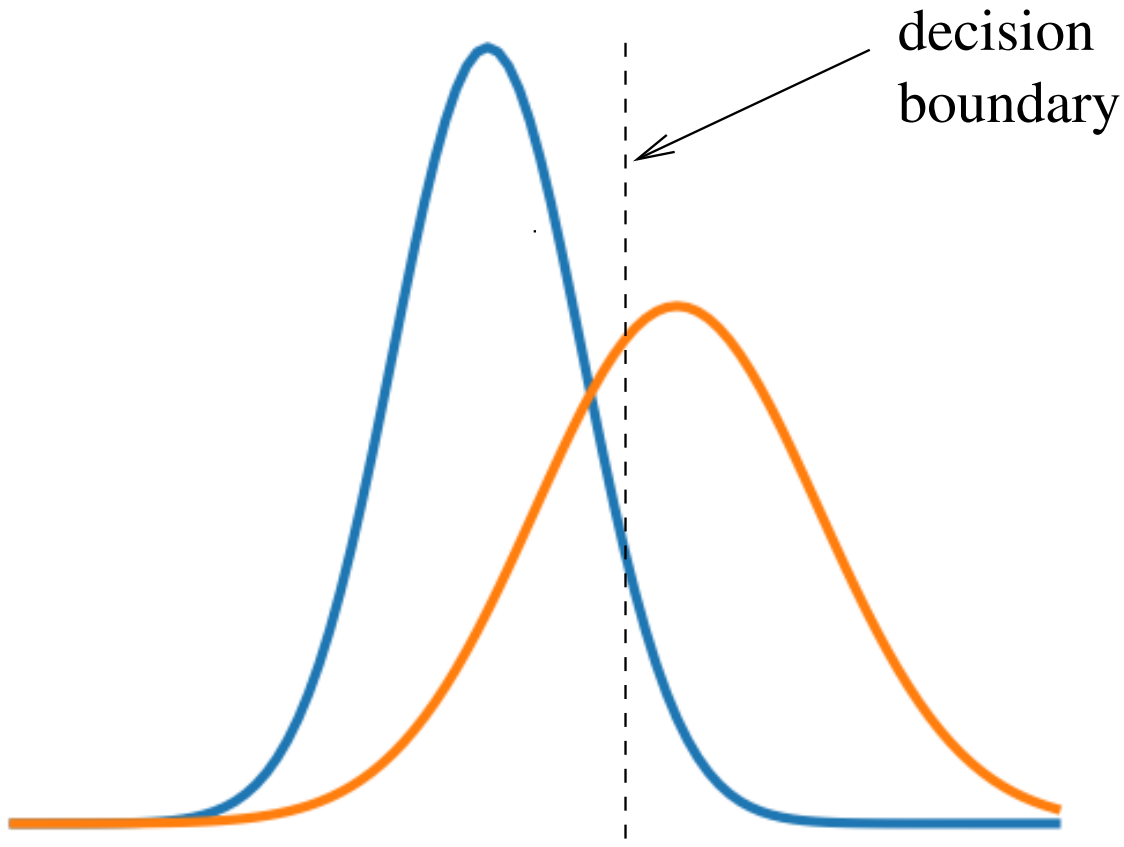
deviazione standard

$\sigma_X = \sigma$

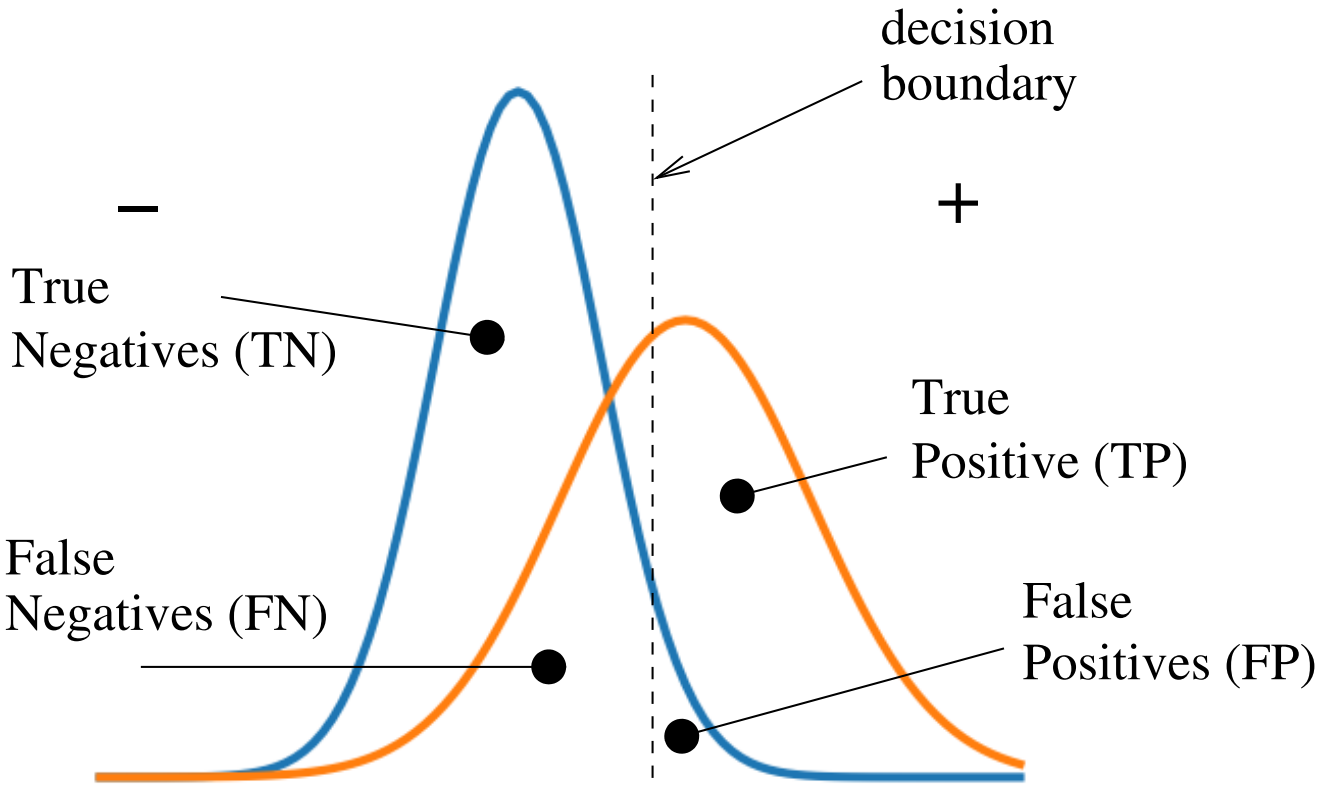


Esempio

Classificare il genere (m/f) data l'altezza.



TP, FP, TN, FN


decision
boundary
–
+
True
Negatives (TN)
True
Positive (TP)
False
Negatives (FN)
False
Positives (FP)

Accuratezza, Precisione e Richiamo

$$Accuratezza = \frac{TP + TN}{All}$$

Quante istanze sono classificate correttamente?

$$Precisione = \frac{TP}{TP + FP}$$

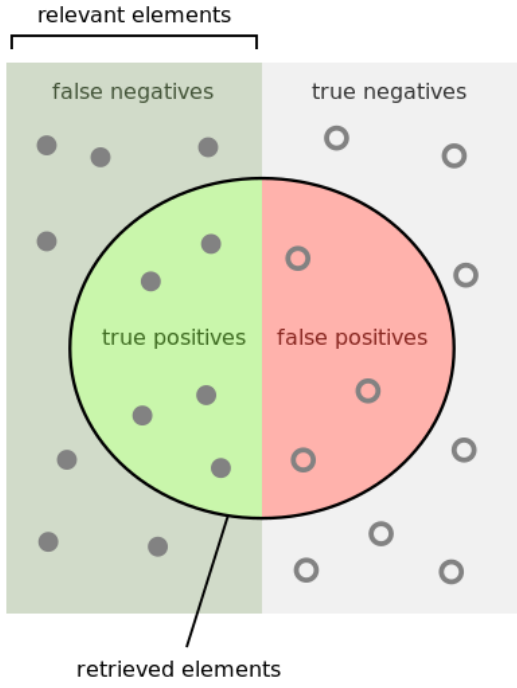
Quanto è precisa la classificazione (sui positivi)

$$Richiamo = \frac{TP}{TP + FN}$$

Che percentuale dei positivi è recuperata?

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

media armoonica tra precsione e richiamo



Parametri descrittivi del modello

Assumiamo che per ogni valore y_k of Y la variabile aleatoria $P(X_i|Y = y_k)$ abbia una distribuzione Gaussiana

$$N(x|\mu_{ik}, \sigma_{ik}) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Apprendimento: stimare i valori dei parametri μ_{ik}, σ_{ik} e $\pi_k = P(Y = y_k)$.

Classificazione of $x^{new} = \langle a_1, \ldots a_n \rangle$

$$\begin{aligned} Y^{new} &= arg\ max_{y_k}\ P(Y = y_k) \cdot \prod_i P(X_i = a_i|Y = y_k) \\ &= arg\ max_k\ \pi_k \cdot \prod_i N(a_i|\mu_{ik}, \sigma_{ik}) \end{aligned}$$

Maximum Likelihood Estimates

μ_{ik} = **valore medio di** X_i **per i dati con etichetta** $Y = y_k$

Formalmente:

$$\mu_{ik} = \frac{\sum_j X_i^j \delta(Y^j = y_k)}{\sum_j \delta(Y^j = y_k)}$$

dove j varia sulle istanze del training set e

$$\delta(Y^j = y_k) = \begin{cases} 1 & \textit{se } Y^j = y_k \\ 0 & \textit{altrimenti} \end{cases}$$

σ_{ik}^2 = varianza di X_i per le istanze con etichetta $Y = y_k$

$$\sigma_{ik}^2 = \frac{\sum_j (X_i^j - \mu_{ij})^2 \delta(Y^j = y_k)}{\sum_j \delta(Y^j = y_k)}$$

prossimo argomento

- Regressione Logistica
- La funzione logistica

Regressione Logistica

Idea:

- Naïve Bayes ci permette di calcolare $P(Y|X)$ dopo aver appreso $P(Y)$ e $P(X|Y)$
- Perchè non cercare di apprendere direttamente $P(Y|X)$?

Che forma adottare per $P(Y|X)$

Che forma abbiamo nel caso di Naive Bayes?

Se:

- Y variabile aleatoria booleana
- X_i variabili aleatoria continue
- X_i indipendenti l'una dall'altra data Y
- $P(X_i|Y = k)$ hnno distribuzioni Gaussiane $N(\mu_{ik}, \sigma_i)$ (**attenzione** non σ_{ik}!)
- Y ha una distribuzione di Benoulli (π)

Allora:

$$P(Y = 1|X = \langle x_1 \dots x_n \rangle) = \frac{1}{1 + exp(w_0 + \sum_i w_i x_i)}$$

Dimostrazione

Per ipotesi

$$P(X_i|Y=k) = N(X_i, \mu_{ik}, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_i^2}}$$

Quindi:

$$\begin{aligned}
P(Y=1|X) &= \frac{P(Y=1)\cdot P(X|Y=1)}{P(Y=1)\cdot P(X|Y=1) + P(Y=0)\cdot P(X|Y=0)} \\
&= \frac{1}{1+\frac{P(Y=0)\cdot P(X|Y=0)}{P(Y=1)\cdot P(X|Y=1)}} \\
&= \frac{1}{1+exp(ln(\frac{P(Y=0)\cdot P(X|Y=0)}{P(Y=1)\cdot P(X|Y=1)}))} \\
&= \frac{1}{1+exp(ln\frac{1-\pi}{\pi} + \sum_i ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})} \\
&= \frac{1}{1+exp(ln\frac{1-\pi}{\pi} + \sum_i (\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}X_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2})}
\end{aligned}$$

Molto interessante!

Se

$$P(Y = 1|X = \langle x_1 \dots x_n \rangle) = \frac{1}{1 + exp(w_0 + \sum_i w_i x_i)}$$

allora

$$P(Y = 0|X = \langle x_1 \dots x_n \rangle) = \frac{exp(w_0 + \sum_i w_i x_i)}{1 + exp(w_0 + \sum_i w_i x_i)}$$

Quindi

$$\frac{P(Y = 0|X = \langle x_1 \dots x_n \rangle)}{P(Y = 1|X = \langle x_1 \dots x_n \rangle)} = exp(w_0 + \sum_i w_i x_i)$$

e in particolare

$$ln \frac{P(Y = 0|X = \langle x_1 \dots x_n \rangle)}{P(Y = 1|X = \langle x_1 \dots x_n \rangle)} = w_0 + \sum_i w_i x_i$$

Per classificare $X = \langle x_1 \dots x_n \rangle$ è sufficiente vedere se $w_0 + \sum_i w_i x_i > 0$

Regressione Logistica

La Regressione Logistica *assume* che

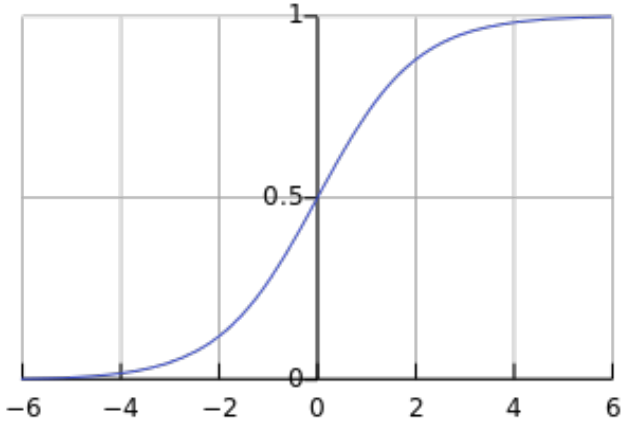
$$P(Y = 1|X = \langle x_1 \dots x_n \rangle) = \frac{1}{1 + exp(-w_0 - \sum_i w_i x_i)}$$

e cerca direttamente di stimare i parametri w_i.

La funzione

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

è detta funzione logistica



Training per regressione logistica (caso binario)

Sappiamo che

$$P(y = 1|x, w) = \sigma(w_0 + \sum_i w_i x_i)$$

Given the (independent) samples $\langle x^\ell, y^\ell \rangle$, their probability is

$$\prod_\ell P(y^\ell|x^\ell, w) = \prod_\ell P(y^\ell = 1|x^\ell, w)^{y^\ell} \cdot P(y^\ell = 0|x_i^\ell w_i)^{(1-y^\ell)}$$

Vogliamo trovare i valori dei parametri w che massimizzano questa probabiilità (MLE).

Analogamente, possiamo passare ai logaritmi e massimizzare

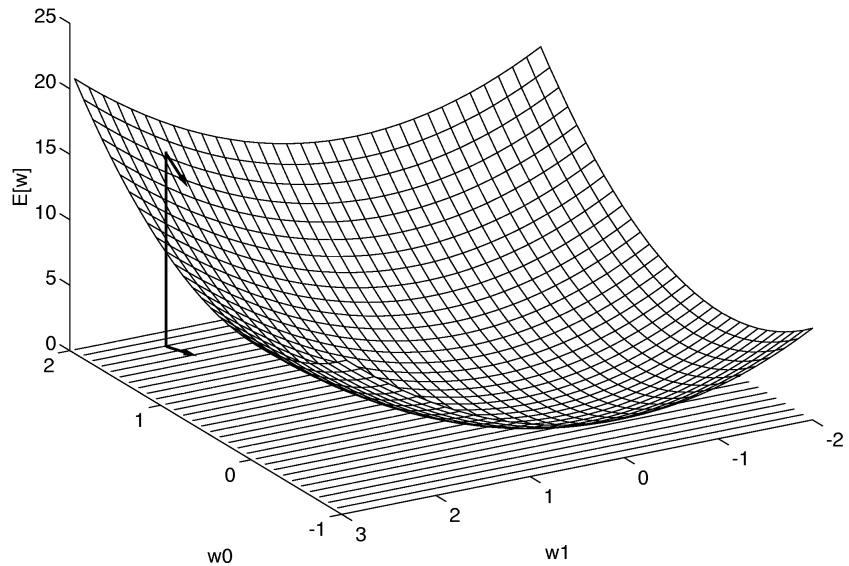
$$\sum_\ell log\, P(y^\ell|x^\ell, w) =$$

$$\sum_\ell (y^\ell \cdot log\, P(Y = 1|x^\ell, w) + (1 - y^\ell) \cdot log\, P(Y = 0|x^\ell, w))$$

Gradiente

Sfortunatamente non esiste una soluzione analitica per il precedente problema di ottimizzazione.

Quindi, utilizziamo dei metodi iterativi di ottimizzazione, come la tecnica del gradiente:



$$\nabla_w[E] = [\frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n}]$$

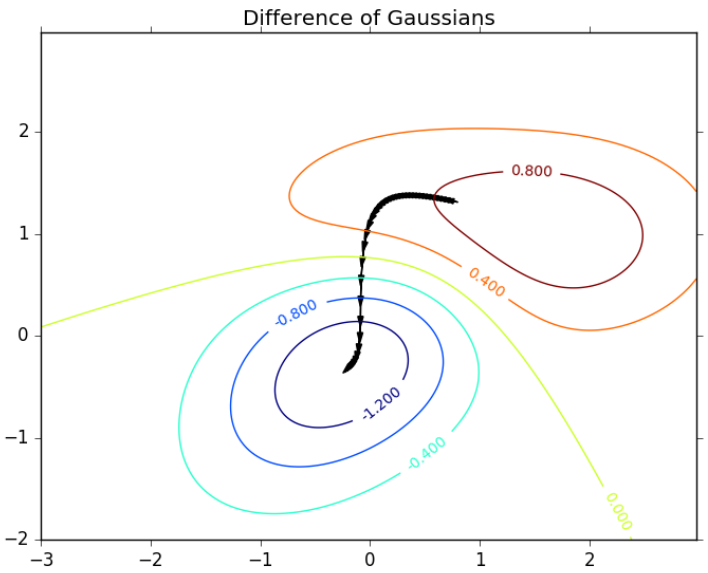
$$\text{training: } \Delta w_i = \mu \cdot \frac{\partial E}{\partial w_i}$$

$$w_i = w_i + \Delta w_i$$

La tecnica del gradiente

Approccio Iterativo

L'obiettivo consiste nel minimizzare una qualche funzione di errore $\Theta(w)$ misurata sul training set, aggiustando opportunamente i parametri del modello.

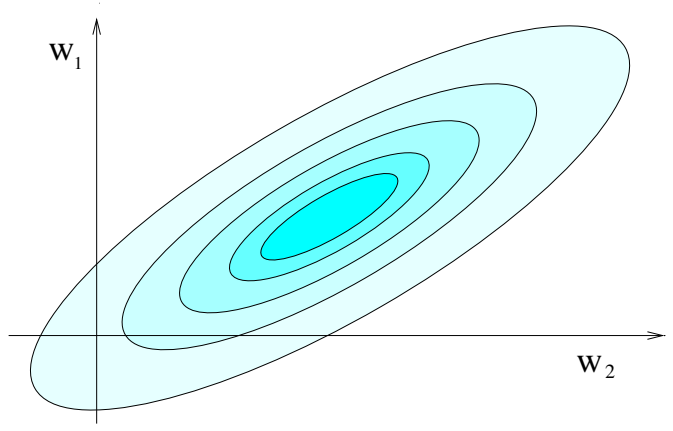


Possiamo raggiungere un minimo per $\Theta(w)$ compiendo iterativamente dei piccoli passi nella direzione opposta al gradiente (discesa del gradiente).

Si tratta di una **tecnica generale**, tuttavia non garantisce di raggiungere minimi globali.

La superficie di errore

Immaginate di avere un asse "orizzontale" per ogni parametro e un asse verticale per l'errore.



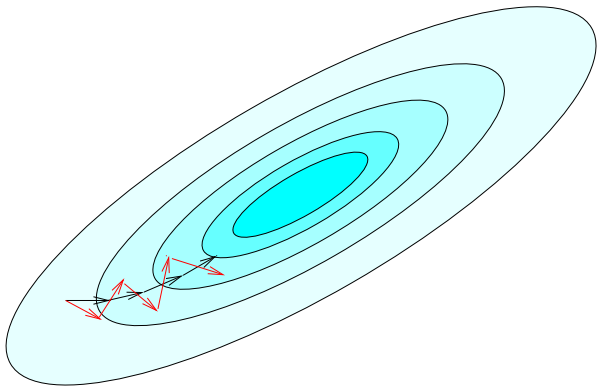
Il caso ideale si ha quando la superficie è convessa, cosa che garantisce l'esistenza di un unico minimo globale.

La superficie di errore nel caso della regressione logistica (con log-likelihood) è convessa.

Per deep neural networks la superficie di errore può essere molto più complicata (molti minimi locali).

Online vs Batch learning

Su quanti punti misurare l'errore.



Full batch: si calcola l'errore e il suo gradiente sull'intero training set. Il gradiente punta nella direzione di massima ascesa, perpendicolare alle linee di livello.

Online: si calcola l'errore e il suo gradiente su una istanza del training set alla volta. Veloce ma impreciso: il gradiente induce un cammino a zig-zag attorno alla direzione corretta.

Mini-batch: Calcolo errore e suo gradiente su un sottoinsieme dei dati: un buon compromesso.

Gradiente per regressione logistica

- probabilità che una istanza ℓ abbia etichetta $Y = 1$

$$P(Y = 1|x^\ell, w) = \sigma(w_0 + \sum_i w_i x_i^\ell) = \alpha^\ell$$

- Log-likelihood $l(w)$

$$\sum_\ell log\, P(Y = y^\ell|x^\ell, w) = \sum_\ell y^\ell log(\alpha^\ell) + (1 - y^\ell) log(1 - \alpha^\ell)$$

- gradiente (dimostrazione nel prossimo slide)

$$\frac{\partial l(w)}{\partial w_i} = \sum_\ell x_i^\ell \cdot (y^\ell - \alpha^\ell)$$

Calcolo del gradiente

$$\alpha^\ell = \sigma(z) = \frac{1}{1 + exp(-z))} \text{ where } z = w_0 + \sum_i w_i x_i^\ell$$

$$\frac{\partial log(\alpha^\ell)}{\partial z} = \frac{1}{\alpha^\ell}\frac{\partial \alpha^\ell}{\partial z} = \frac{exp(-z)}{1 + exp(-z)} = 1 - \alpha^\ell$$

$$log(1 - \alpha^\ell) = log\frac{exp(-z)}{1 + exp(-z)} = -z + log(\alpha^\ell)$$

$$\frac{\partial log(1 - \alpha^\ell)}{\partial z} = -1 + 1 - \alpha^\ell = -\alpha^\ell$$

ricordiamo che $l(w) = \sum_\ell y^\ell log(\alpha^\ell) + (1 - y^\ell) log(1 - \alpha^\ell)$, quindi

$$\frac{\partial l(w)}{\partial w_i} = \frac{\partial l(w)}{\partial z}\frac{\partial z}{\partial w_i} \quad = (\sum_\ell y^\ell(1 - \alpha^\ell) - (1 - y^\ell)\alpha^\ell)x_i^\ell$$
$$= (\sum_\ell (y^\ell - \alpha^\ell))x_i^\ell$$

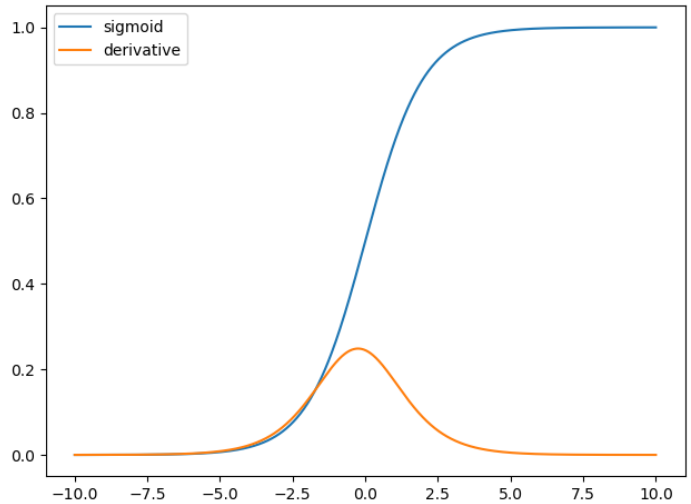
Osservazione sulla derivata della sigmoid

Nello slides prrecedente abbiamo visto che

$$\frac{\partial log(\alpha^{\ell})}{\partial z} = \frac{\partial \sigma(z)}{\partial z} = \frac{1}{\sigma(z)} \frac{\partial \sigma(z)}{\partial z} = 1 - \sigma(z)$$

Dunque:

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z))$$



La derivata della funzione logistica è molto piatta.

Il processo di apprendimento

Iterare la seguente operazione di update finchè non si raggiunge l'approssimazione desiderata (ad esempio fino a quando l'accuratezza sul set di validazione é soddisfacente o non sembra aumentare).

$$w_i \leftarrow w_i + \mu \sum_{\ell} x_i^{\ell} \cdot (y^{\ell} - P(Y = y^{\ell} | x_i w_i))$$

Frequentemente si aggiunger una qualche regolarizzatore:

$$w_i \leftarrow w_i - \mu\lambda|w_i| + \mu \sum_{\ell} x_i^{\ell} \cdot (P(Y = y^{\ell} | x_i w_i) - y^{\ell})$$

- cerca di tenere i parametri w_i vicini a 0
- tende a ridurre l'overfitting

Modelli Generativi e Discriminativi

Generative vs discriminative models

Discriminativi I modelli discriminativi sono utilizzati prevalentemente per problemi di classificazione, dove l'obiettivo consiste nell'apprendere la **frontiera** (decision boundary) che separa le differenti classi.

Generativi I modelli generativi cercano di apprendere la **distribution dei dati**. Oltre che ai fini della classificazione, possono essere usati per data generation, data augmentation (with cautele), anomaly detection, trasferimento di contenuto/stile e molte altre applicazioni.

Modelli Generativi e discriminativi (esempio)

Problema: stimare $f : X \rightarrow Y$ o $P(Y|X)$.

Classificatore Generativo (e.g., Naïve Bayes)

- si assume una certa distribuzione per $P(X|Y), P(X)$
- si stimano i parametri per $P(X|Y)$ e $P(X)$ sui dati di training
- si utilizza la regola di Bayes' per inferire $P(Y|X)$

Classificatore Discriminative (e.g., Regressione Logistica)

- si assume una certa distribuzione per $P(Y|X)$
- si stimano i parametri per $P(Y|X)$ sui dati di training

Esempio: classificazione di cifre mnist con Regressione Logistica

I metodi lineari sono potenti

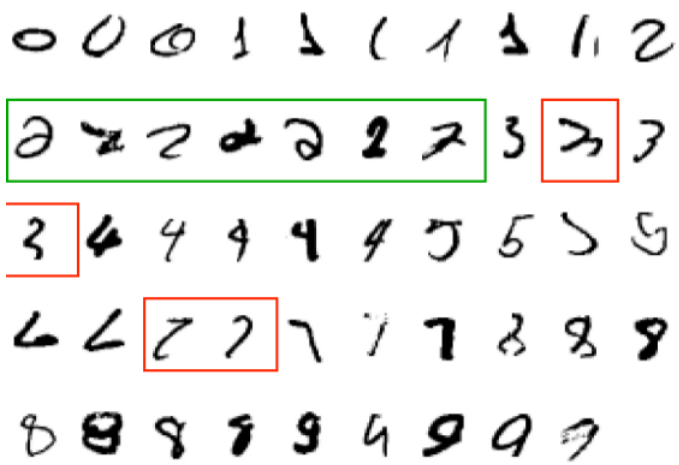
Si potrebbe avere l'impressione errata che i metodi lineari non siano di interesse effettivo in quanto troppo deboli.

Quando si hanno molte features a disposizione e i dati sono stati adeguatamente pre-processati anche i metodi lineari funzionano bene.
In uno spazio con molte dimensioni i dati assumono una distribuzione molto sparsa, quindi diventa plaausibile discriminare le varie classi con dei semplici iperpiani.

Il dataset MNSIT

Modified National Institute of Standards and Technology database

- immagini in scale di grigio di cifre numeriche scritte a mano. Bassa risoluzione; 28×28 pixels
- 60,000 immagini di training e 10,000 immagini di testing



Nel caso delle immagini ogni pixel è una feature.
Un piccola immagine 28×28 contiene già 784 features.

Classificazione con regressione multinomiale

DEMO!

Classificazione con regressione multinomiale

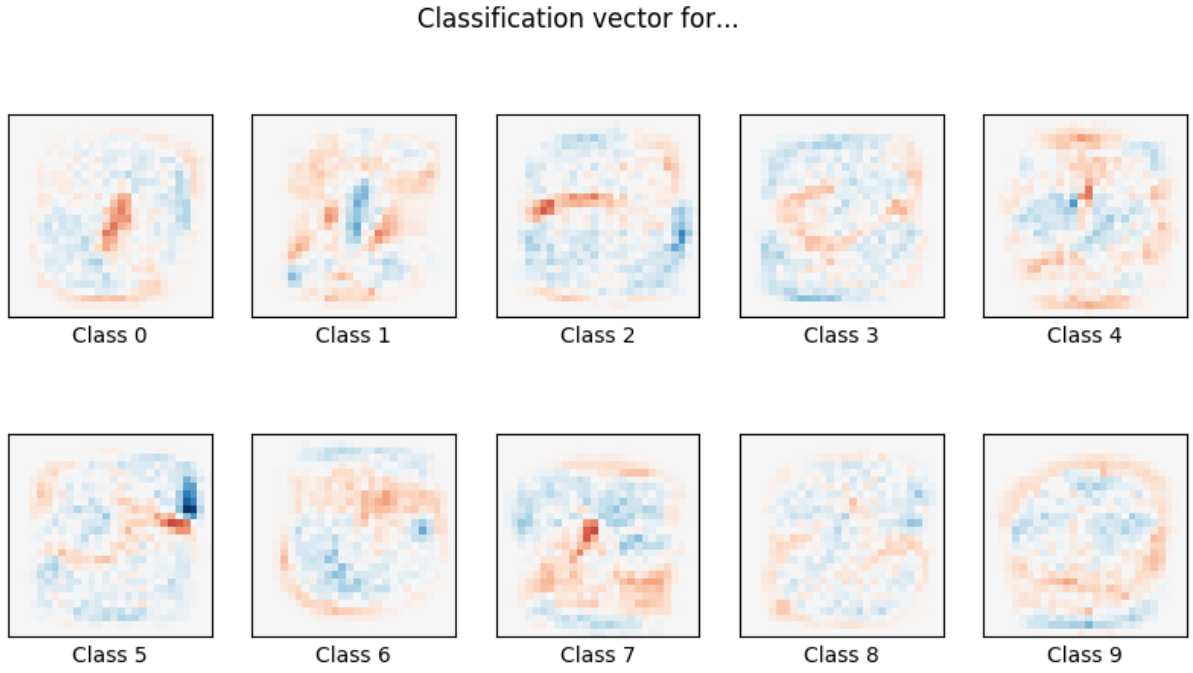
DEMO!

```
Sparsity with L1 penalty: 25.18%
Test score with L1 penalty: 0.9257
Example run in 22.404 s
```

Visualizzazione dei pesi

Per ogni classe, i pesi corrispondenti hanno la stessa dimensionalità della immagine di input: abbiamo un peso per ogni feature.

Possiamo visualizzare i pesi!



Per ogni classe, le zone rosse sono quelle dove mi aspetto di avere della scrittura, quelle blu dove non mi aspetto di averne.