

Agenti e apprendimento automatico

Date	@June 8, 2022 → June 10, 2022
------	-------------------------------

Agenti

- [Caratteristiche dell'ambiente](#)
- [Funzione agente ≠ programma agente](#)
- [Diversi tipi di agente](#)
- [Agenti reattivi](#)
- [Agenti reattivi basati su modello](#)
- [Agenti guidati dagli obiettivi e dall'utilità](#)
- [Agenti che apprendono](#)

Classificazione

- [Cosa farci coi modelli appresi](#)
- [Il modello risultante è buono?](#)
 - [Accuratezza](#)
 - [Error rate](#)
- [Alberi di decisione](#)
- [Algoritmo di Hunt](#)
 - [Varie tipologie di split di un DT](#)
- [Quale split scegliere prima per un DT](#)
 - [Entropia](#)
- [Come gestire l'overfitting di un modello](#)
- [Valutare la bontà dei modelli](#)
 - [Metodi per valutare i test test](#)
 - [Intervallo e livello di confidenza](#)

Classificatori a regole

- [Due misure per stabilire la qualità di una regola](#)
 - Copertura**
 - Accuratezza**
- [Produrre le regole](#)
- [Learn one rule: un esempio di sequential covering](#)
- [Strategie di raffinamento](#)
 - [General to specific](#)
 - [Specific to general](#)
- [Eager vs lazy learners](#)
 - [Eager](#)
 - [Lazy](#)
- [K-nearest neighbour](#)

Reti neurali

- [A cosa serve un perceptron?](#)

Definizioni

Agenti



agente = software + architettura

RAZIONALITÀ

ONNISCIENZA/CHIAROVEGGENZA


Ottimizza il risultato atteso

Ottimizza il risultato reale

Un agente cerca di agire razionalmente

La razionalità dipende dalla **sequenza percettiva** e può influenzare la sequenza percettiva.

Caratteristiche dell'ambiente

Completamente osservabile in ogni istante i sensori danno accesso a tutti gli aspetti dell'ambiente rilevanti per la scelta dell'azione	Parzialmente osservabile i sensori danno accesso solo a parte dell'informazione rilevante (cause: sensori imprecisi oppure non in grado di rilevare alcuni dati) 
Deterministico lo stato successivo è determinato dallo stato corrente e dall'azione applicata	Stocastico applicando più volte una stessa azione in uno stesso stato si possono raggiungere stati diversi. Si dice strategico quando è stocastico solo per quanto riguarda le azioni degli altri agenti.
Episodico l'esperienza degli agenti è divisa in episodi atomici: un episodio è dato da una percezione seguita da una singola azione (esempio: classificazione)	Sequenziale Attività composta da più passi ognuno dei quali in generale influenzerà i successivi

Statico l'ambiente non cambia mentre l'agente "pensa" (cioè mentre decide quale azione eseguire)	Dinamico l'ambiente può cambiare mentre l'agente "pensa"
Discreto possono essere discreti stato, tempo, percezioni, azioni (esempio: gli scacchi hanno stati, percezioni, azioni discreti)	Continuo possono essere continui stato, tempo, percezioni, azioni (esempio: gli scacchi hanno tempo continuo)
Singolo Agente Viene modellata come agente una sola entità	Multiagente Vengono modellate come agenti più entità

Funzione agente ≠ programma agente

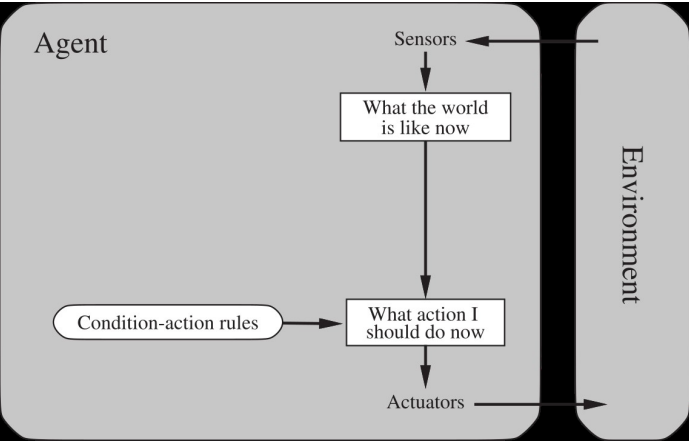
Funzione agente: ha come input la sequenza percettiva

Programma agente: ha come input la percezione corrente

Diversi tipi di agente

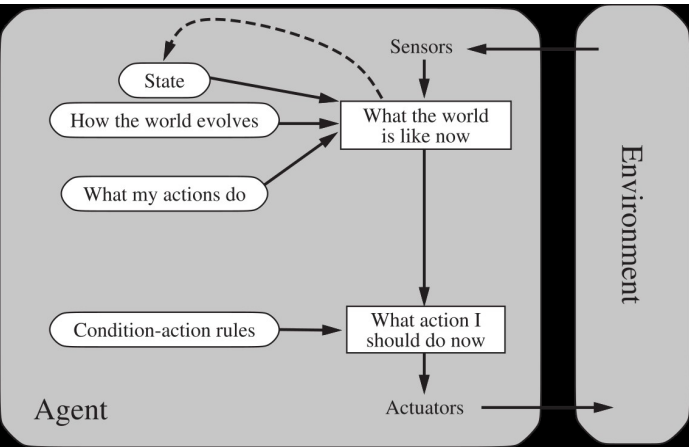
- agenti reattivi
- agenti reattivi basati su modello
- agenti guidati dagli obiettivi
- agenti guidati dall'utilità
- agenti capaci di apprendere

Agenti reattivi



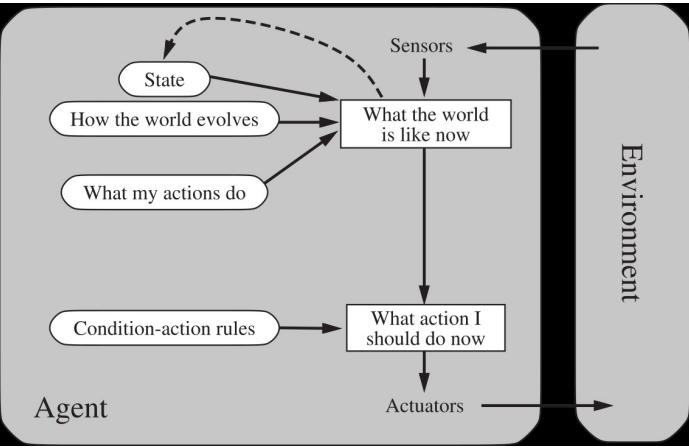
basano la scelta dell'azione sulla sola percezione

Agenti reattivi basati su modello



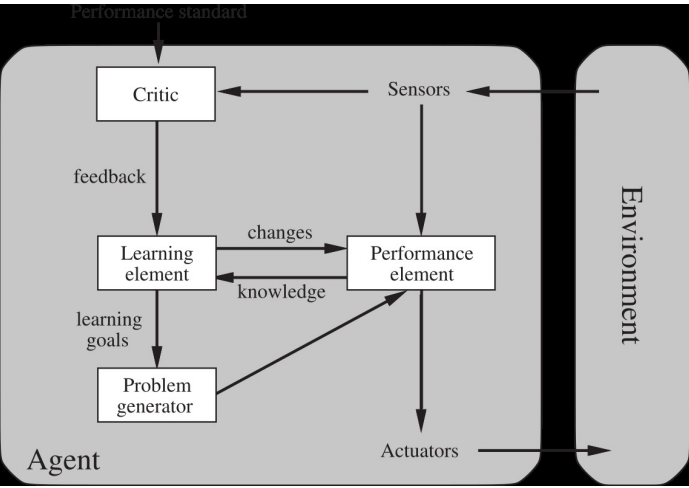
- hanno conoscenza su come il mondo evolve e sugli effetti delle azioni
- avere un modello permette di **prevedere gli effetti delle azioni** e di scegliere l'azione da eseguire anche sulla base di questa informazione

Agenti guidati dagli obiettivi e dall'utilità



- l'agente *simula* nella propria *mente* l'effetto delle azioni per capire se e **quanto l'avvicinano all'obiettivo**
- si può aggiungere una **misura della prestazione**, cioè della bontà delle possibili scelte calcolata da una funzione metrica

Agenti che apprendono



La parte di sinistra (che si aggiunge ai modelli visti sopra) è quella deputata al **learning** ed è costituita da:

- un **critico** ⇒ decide se è il caso di attivare l'apprendimento
- un **modulo di apprendimento** ⇒ che modifica la conoscenza dell'agente

- un **generatore di problemi** \Rightarrow che espone l'agente a nuove esperienze

Classificazione

L'idea è che si vuole costruire una rappresentazione astratta (**modello**) che permetta di associare in modo corretto nuove istanze alla **classe** (o alle classi) **di appartenenza**.

nome	temperat ura	Copertur a pelle	viviparo	Creatura acquatica	Creatura volatile	zampe	letargo	CLASSE
uomo	Sangue caldo	peluria	sì	no	no	sì	no	mammifero
pitone	Sangue freddo	squame	no	no	no	no	sì	rettile
salmone	Sangue freddo	squame	no	sì	no	no	no	pesce
balena	Sangue caldo	peluria(?)	sì	sì	no	no	no	mammifero
rana	Sangue freddo	nessuna	no	semi	no	sì	sì	anfibia
pinguino	Sangue caldo	piumaggio	no	semi	no	sì	no	uccello
piccione	Sangue caldo	piumaggio	no	no	sì	sì	no	uccello

Esempio di **learning set** formato da un insieme di tuple (x,y) , dove x è a sua volta un'altra tupla contenente i valori descrittivi dell'istanza e y è la classe di appartenenza dell'istanza

Cosa farci coi modelli appresi

Modello predittivo

Viene usato per **predire la classe di appartenenza di istanze ignote in fase di apprendimento**

Es. data la descrizione di una salamandra (sangue freddo, nessuna, no, semi, no, sì, sì) si usano le regole apprese per decidere a quale classe appartiene

Modello descrittivo

Viene usato come **strumento esplicativo che permette di evidenziare quali caratteristiche distinguono le diverse categorie**

Esprime in maniera sintetica delle descrizioni evitando di ragionare direttamente sugli esempi

Es. i mammiferi hanno il sangue caldo e solitamente non sono esseri acquatici

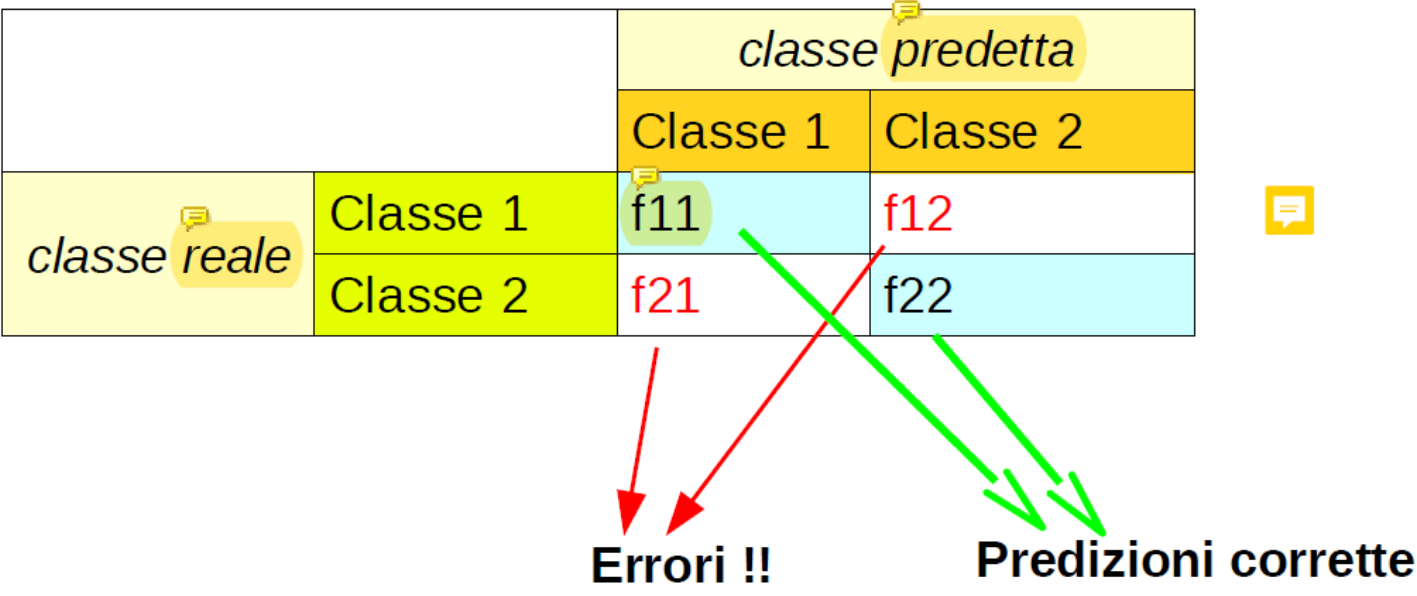
Due possibili usi differenti di un modello estratto da un learning set

Il modello risultante è buono?

Per valutare la bontà di un modello ottenuto si prova a far classificare al modello un insieme di dati chiamato **test set**.

Da questa classificazione si produce la **matrice di confusione**:

Matrice di confusione:



Matrice quadrata che ha numero di righe/colonne pari a quello delle classi che stiamo considerando.

I valori all'interno della matrice rappresentano il numero delle istanze che appartengono alla classe reale c_i e che il modello ha detto appartenere a c_j . Si cerca dunque di **massimizzare i valori nella diagonale**.
Da questa matrice si estraggono due formule complementari:

Accuratezza

$$acc = \frac{\text{valoriDiagonale}}{\text{predizioniTotali}}$$

dove i *valoriDiagonale* rappresentano il numero di **predizioni corrette**.

Error rate

$$err = \frac{\text{predizioniSbagliate}}{\text{predizioniTotali}}$$

dove *predizioniSbagliate* indica il numero di valori **non in diagonale** nella matrice.

	classe predetta	
	lattina	altro ogg.
lattina	18	2
altro ogg.	1	19

Accuratezza: $(18 + 19) / 40 = 92.5\%$

Error rate: $3 / 40 = 7.5\%$

Esempio di modello che riconosce varie istanze di lattine

da notare che *err* si può ottenere anche come $1 - acc$.

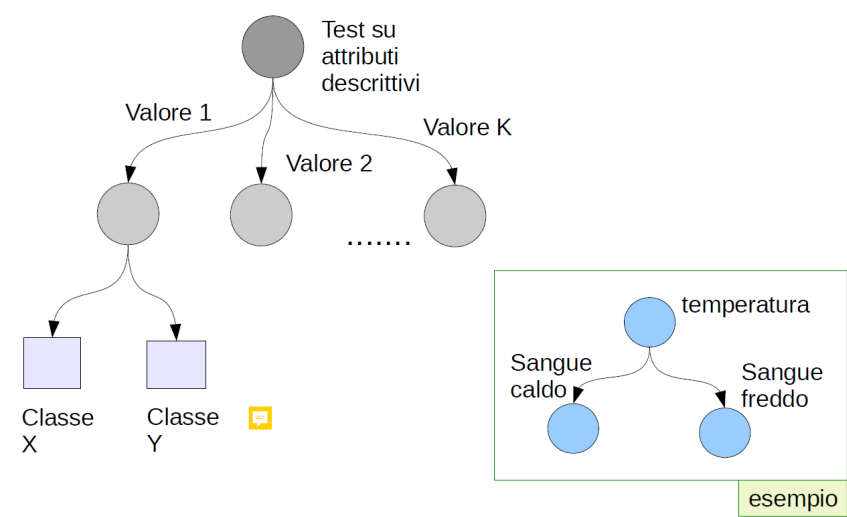
Per differenziare i tipi di errori che un modello può compiere si associa alla matrice di confusione una matrice dei costi.

		classe predetta	
		Classe 1	Classe 2
classe reale	Classe 1	-1	50
	Classe 2	10	0

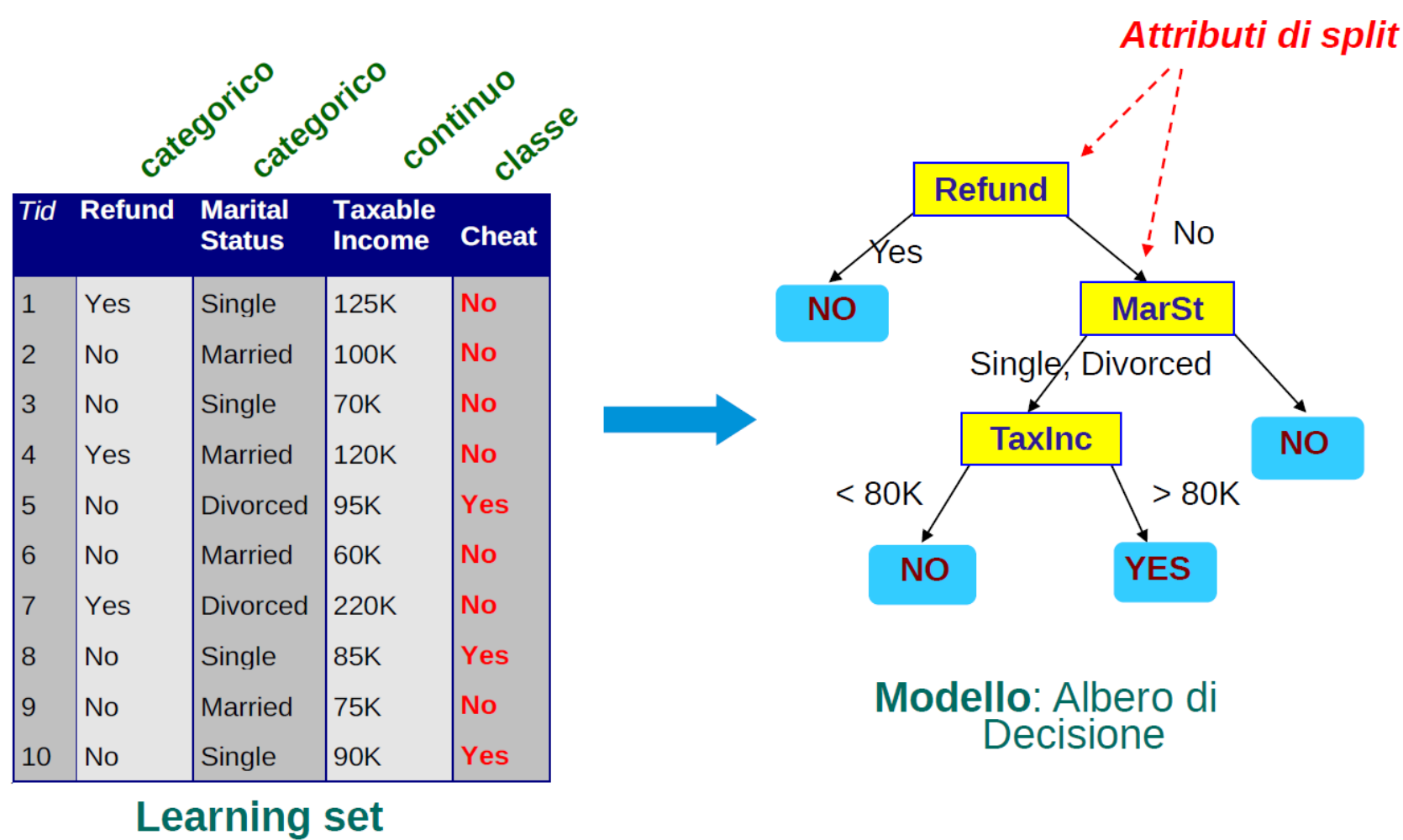
Esempio di matrice dei costi, dove ogni istanza di classe 1 correttamente classificata riduce il costo complessivo e dove sbagliare a classificare gli oggetti di classe 1 è molto più costoso che sbagliare quelli di classe 2.

Alberi di decisione

Usati in molti ambiti, noi li vediamo come strumenti **indotti da dei learning set** al fine di classificare in maniera più chiara nuove istanze ignote. Svolgono quindi il ruolo di **modelli**.



Struttura di un albero di decisione, dove le foglie indicano la classe dell'istanza



Esempio di come da un learning set si possa estrarre un DT (decision tree), sulla base delle istanze coinvolte nel learning set (un'istanza è rappresentata da una tupla). In rosso la classificazione che ci interessa conoscere delle nuove istanze.

📌 Da un learning set si possono ricavare molti DT

Algoritmo di Hunt

- è un esempio di **algoritmo di apprendimento**

L'algoritmo di Hunt è un algoritmo ricorsivo che permette di **costruire un albero di decisione** partizionando i record di allenamento in sottoinsiemi sempre più omogenei, con il fine di far sì che ognuno di essi contenga record aventi la medesima classe.

Varie tipologie di split di un DT

- split su **attributi binari**
- split su **attributi nominali** (il nodo avrà tanti figli quanti sono i possibili valori dell'attributo)
- split su **attributi ordinali** (il raggruppamento dei valori deve rispettare l'ordinamento)
- split su **attributi continui**

Quale split scegliere prima per un DT

Alberi compatti sono preferiti ad alberi che consentono di raggiungere lo stesso grado di accuratezza (e di error rate) usando un maggior numero di test.



Sono preferiti gli split che producono nodi figli la cui estensione prevede **minore confusione** (il cui grado di purezza è maggiore).

Entropia

$$entropia(t) = - \sum_{i=0}^{c-1} p(i|t) * \log_2 p(i|t)$$

dove si assume $0 * \log_2 0 = 0$ e dove $p(i|t)$ indica la probabilità che l'elemento appartenga alla classe i

Le distribuzioni (0, 1) e (1, 0) sono le migliori, **purezza massima** dell'insieme, nessuna confusione.

La distribuzione (0.5, 0.5) è la peggiore, **massimo grado di confusione** (entropia pari a 1).

Il calcolo dell'entropia ci serve per poter ottenere un valore confrontabile per ciascuno split, l'**information gain** (usata come misura per valutare il guadagno).

$$\Delta = entropia(parent) - \sum_{j=1}^k \frac{N(v_j)}{N} entropia(v_j)$$

dove N indica il numero di record del nodo genitore, $N(v_j)$ indica il numero di record del figlio j -esimo

Come gestire l'overfitting di un modello

Facendo **pre-pruning** o **post-pruning** ossia tagliando certi rami di un DT, così da rendere il DT più generico.

Valutare la bontà dei modelli

La cardinalità del learning set influenza l'accuratezza del modello appreso: se il campione è piccolo il modello risulta focalizzato e poco affidabile.

Metodi per valutare i test test

- **holdout**: partiziono i dati che ho in *test set* (che mi serviranno a valutare i risultati del modello) e *training set* (che mi serviranno a costruire il modello)
- **random subsampling**: le prestazioni potrebbero dipendere dalla partizione effettuata, se ne eseguono diverse e si fa la media
- **cross-validation**: si fa una partizione ma si cerca di usare i dati in modo omogeneo
- **bootstrap**: se ci sono pochi dati partizionare può produrre insiemi troppo piccoli per essere significativi

Intervallo e livello di confidenza

$$P(a \leq \theta \leq b) = 1 - \alpha$$

dove $[a, b]$ è l'**intervallo di confidenza** e $1 - \alpha$ è il **livello di confidenza**

Classificatori a regole



Una regola si attiva quando la descrizione di un'istanza soddisfa la condizione posta ad antecedente

Due misure per stabilire la qualità di una regola

Copertura

indica il numero di istanze **coperte** dalla regola

$$\frac{|A|}{|D|}$$

dove $|D|$ indica la cardinalità del set, $|A|$ il numero di istanze che soddisfano l'antecedente della regola

Accuratezza

indica il numero di istanze **correttamente classificate** dalla regola

$$\frac{|A \cap y|}{|A|}$$

dove indica $|A|$ il numero di istanze che soddisfano l'antecedente della regola e $|A \cap y|$ il numero delle istanze che soddisfano A e che sono di classe y

Produrre le regole

Le regole possono essere **estratte dai dati** oppure **estratte da altri classificatori** (quali i DT ad esempio)

Learn one rule: un esempio di sequential covering

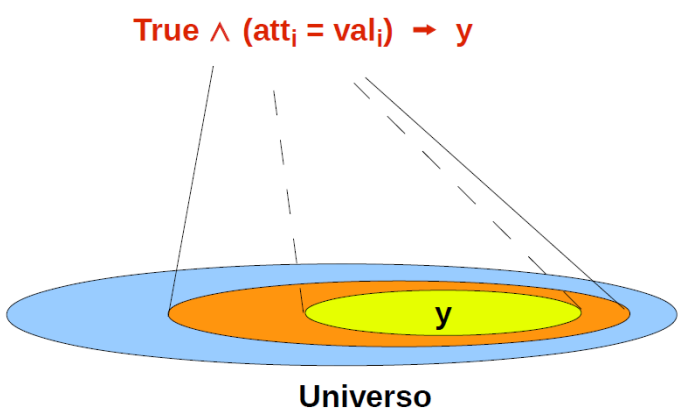
1. si decide un **ordinamento delle classi** con un qualche criterio
2. seguendo l'ordinamento ci si focalizza su **una classe alla volta** e si creano le regole di classificazione



Learn one rule è una funzione che permette di produrre una nuova regola per induzione da esempi. Si produce una regola iniziale e poi la si raffina nei passi successivi

Strategie di raffinamento

General to specific



Specific to general

Congiunti: i valori corrispondono a quelli dell'esempio scelto

Numero dei congiunti: numero degli attributi con cui sono descritti gli esempi nel training set

Problema: troppo specifica!

Meccanismo di generalizzazione:

rimozione di congiunti finalizzata a coprire un maggior numero di istanze positive



If (viviparo) \wedge (sangue caldo) \wedge (~~volante~~) \rightarrow mammifero

Eager vs lazy learners

Eager



Dati gli esempi si mettono subito all'opera per costruire un modello che in futuro useranno per classificare nuove istanze

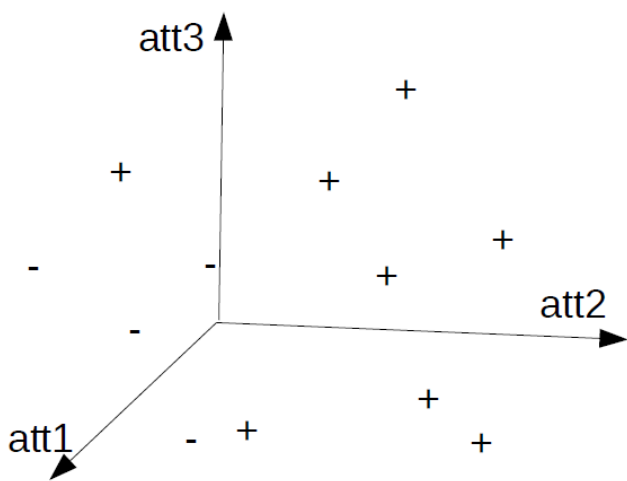
Lazy



Dati gli esempi non se ne fanno nulla finché non viene il momento di classificare nuove istanze

K-nearest neighbour

"if it walks like a duck, quacks like a duck, and looks like a duck, then it is probably a duck"

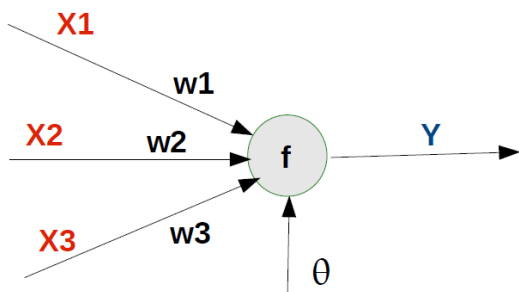


Gli esempi sono visti come **punti** in uno spazio con N dimensioni, dove N è il numero degli attributi descrittivi. Le nuove istanze sono classificate sulla base di un **criterio di vicinanza** a quelle memorizzate dal sistema.

Se k è **piccolo** si corre il rischio di **overfitting**, se k è **grande** si corre il rischio di **underfitting**.

Reti neurali

Un perceptron è in grado di calcolare una **funzione di attivazione** così formulata:



$$Y = f(net) = \frac{1}{1 + e^{-\alpha * (net - \theta)}}$$

dove $net = \sum_{i=1}^n w_i X_i$ indica la somma di tutti i pesi degli input del perceptron, θ indica la **soglia** o **bias**, α indica la pendenza della sigmoide

A cosa serve un perceptron?



Un perceptron (che apprende da un learning set supervisionato) codifica un test lineare

Ciò che cade **al di sopra** dell'iperpiano codificato dai suoi pesi fa **attivare il neurone**, ciò che è sotto non fa attivare il neurone.

I **pesi** sulle connessioni in ingresso definiscono la **posizione** e la **pendenza** dell'iperpiano nello spazio in cui sono definiti gli input. I pesi caratterizzano i neuroni e costituiscono la conoscenza del neurone.

Definizioni

▼ Agente

astrazione che rappresenta un qualsiasi sistema che percepisce il proprio ambiente tramite dei sensori ed agisce su di esso tramite degli attuatori

▼ Sequenza percettiva

storia completa delle percezioni di un agente

▼ Task environment

è il contesto in cui l'agente è inserito

▼ Protocollo di interazione

è uno schema che permette di coordinare due o più agenti, tipicamente sfruttando uno scambio di messaggi come strumento di coordinazione

▼ Apprendimento supervisionato

gli esempi dal quale astrarre le definizioni delle classi **hanno associata la classe** a cui appartengono

▼ Learning set

collezione di dati usati per svolgere il compito di apprendimento

- ▼ Imparare per induzione
generalizzare gli esempi contenuti in un learning set
- ▼ Rasoio di Occam
a parità di assunzioni, la spiegazione più semplice è da preferire
- ▼ Overfitting
si ha **overfitting** quando il modello appreso è troppo specifico
- ▼ Underfitting
contrario di overfitting, ossia quando il modello è troppo generale e non dà nessuna informazione
- ▼ Pre-pruning
la costruzione del DT si interrompe prima che l'albero sia completo. Si impone una **regola di terminazione** più restrittiva
- ▼ Post-pruning
prima si costruisce l'albero poi si potano alcuni rami, trasformando alcuni nodi interni in foglie
- ▼ Classificatori
è una **mappatura** da uno spazio (discreto o continuo) di *feature* X a un insieme di etichette Y . Esempi di classificatori basati sull'apprendimento automatico sono le *reti neurali*, i *classificatori bayesiani* e gli *alberi di decisione*.
- ▼ Intervallo di confidenza
intervallo a cui si pensa un certo valore reale e ignoto appartenga
- ▼ Livello di confidenza
probabilità che il valore reale e ignoto sia **effettivamente compreso** nell'intervallo dato
- ▼ Regole mutuamente esclusive
ogni regola è attivata da un insieme di esempi disgiunto dagli insiemi di esempi che attivano le altre regole
- ▼ Regole esaustive
tutte le istanze saranno catturate da qualche regola
- ▼ Perceptron
Un perceptron è un elemento computazionale, dotato di una piccola memoria in grado di calcolare una **funzione di attivazione**