

SISTEMI INTELLIGENTI

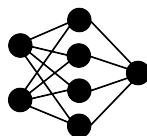
Prof. Nunzio Alberto Borghesi

Prof. Nicola Basilico

6 CFU

Luca Cappelletti

Lecture Notes
Year 2017/2018



Magistrale Informatica
Università di Milano
Italy
14 febbraio 2018

Indice

1 Logica Fuzzy	3
1.1 Logica fuzzy vs classica	3
1.1.1 Le funzioni di appartenenza	3
1.1.2 Classi di appartenenza	3
1.2 Logica fuzzy e probabilità	4
1.3 Gli operatori logici nella logica fuzzy	4
1.4 Misure in un insieme fuzzy	4
1.4.1 Norma di un vettore	4
1.4.2 Entropia	4
1.5 Fuzzy Associative Memory FAM	4
1.5.1 Come opera il sistema	5
2 Statistica	6
2.1 Probabilità o visione frequentista	6
2.1.1 Probabilità di eventi indipendenti e contemporanei	6
2.1.2 Probabilità condizionata (eventi indipendenti e successivi)	6
2.1.3 Teorema di Bayes	7
3 Clustering	8
3.1 Clustering	8
3.1.1 Ontologia	8
3.1.2 Quad-Tree Decomposition	9
3.1.3 Agglomerative Clustering	9
3.1.4 K-means	11
4 Apprendimento	12
4.1 Value Function	12
4.1.1 Rappresentazione delle azioni	12
4.1.2 Reward a lungo termine	12
A Domande da temi d'esame	13
A.1 Domande sui Macchine ed Intelligenza	13
A.1.1 Descrivere il test di Turing, l'esperimento della stanza cinese e l'esperimento della stanza di Maxwell [4]	13
A.1.2 Discutere la relazione tra algoritmo, macchina di Turing ed intelligenza.	14
A.1.3 Cosa si intende per ipotesi forte ed ipotesi debole dell'AI? [4]	14
A.1.4 Riportare almeno due elementi del contraddittorio sulle ipotesi su cui è basata l'ipotesi debole sull'AI [3]	14
A.1.5 Descrivere il "Brain prosthesis thought experiment" di Moravec e commentarlo. [3]	14
A.2 Domande sui Sistemi Fuzzy	15
A.2.1 Cose un insieme fuzzy?	15
A.2.2 Esiste una corrispondenza biunivoca tra insiemi fuzzy e valori numerici?	15
A.2.3 Cose una membership function? Con quali altri nomi viene anche indicata?	15
A.2.4 La frase seguente: con la mia preparazione potrei prendere 24 allestame, sottintende un processo fuzzy o probabilistico?	15
A.2.5 Cosa si intende per FAM? [3]	15
A.2.6 Una FAM memorizza numeri o preposizione logiche? Come? [3]	15
A.2.7 Descrizione del funzionamento di una FAM.	15
A.2.8 Definire i passi per costruire un sistema fuzzy	15
A.2.9 Definire un problema a piacere che involva almeno due variabili in ingresso e due in uscita. Definire tutti i componenti e calcolare l'uscita passo a passo per un valore di input a piacere	15
A.3 Domande su Apprendimento con Rinforzo	18
A.3.1 Cosa si intende per Apprendimento con Rinforzo?	18
A.3.2 Quali sono gli attori?	18

A.3.3	Cosa rappresenta la critica?	18
A.3.4	Che tipo di architettura si può ipotizzare nell'apprendimento con rinforzo?	18
A.3.5	Condizionamento classico e condizionamento operante	19
A.3.6	Quale relazione c'è con l'intelligenza?	19
A.3.7	Come potreste illustrare: Exploration vs Exploitation?	19
A.3.8	Cos'è il problema del credit assignment? È un problema che riguarda la dimensione temporale o spaziale del task?	19
A.3.9	Cos'è l'eligibility trace (traccia) e quale è il suo ruolo?	19
A.3.10	Definire l'algoritmo di Q-learning, descrivendo le equazioni opportune. [2]	19
A.3.11	Scrivere le equazioni dell'algoritmo Q-learning in cui si consideri anche la traccia. [2]	19
A.3.12	Cosa si intende per politica epsilon-greedy? Come entra nell'algoritmo di Q-learning?	19
A.3.13	Che differenza c'è tra Q-learning e SARSA? [2]	19
A.3.14	Dato un problema a piacere si descriva uno degli algoritmi e mostrare due passaggi di addestramento	20
A.3.15	Quale criterio si sceglie per definire i Reward? A quali elementi sono associati? Allo stato? All'azione? Allo stato prossimo? Perché? [2]	20
A.3.16	Impostare un problema su griglia (apprendimento del percorso di un agente, con partenza ed arrivo prescelti + ostacoli). La griglia fornisce un reward, diverso da zero, in ogni transizione. [2]	20
A.4	Domande su Statistica	20
A.4.1	Esercizio sui Taxi	20
A.4.2	Esercizio sul Tumore al seno	20
A.4.3	Esercizio delle Macchine [2]	20
A.4.4	Enunciare il teorema di Bayes	20
A.4.5	Discutere l'analisi di varianza per un sistema lineare [4]	21
A.4.6	Dimostrare che la stima ai minimi quadrati è equivalente alla stima a massima verosimiglianza nel caso di errore Gaussiano sui dati. Cosa fornisce? Come? [3]	21
A.5	Domande su Apprendimento Supervisionato	23
A.5.1	Definire l'algoritmo di apprendimento di una rete neurale con unità arbitrarie. Definire la funzione obiettivo utilizzata. [3]	23
A.5.2	Come si utilizza la funzione obiettivo nell'algoritmo di apprendimento. [3]	23
A.5.3	Cosa si intende per apprendimento per epoche e per trial? Qual è il vantaggio di ciascuna delle modalità di apprendimento? [3]	23
A.5.4	Cosa si intende per training e test set? Perché mai vengono utilizzati? Quali problemi si vogliono evitare? [3]	23
A.5.5	Una rete neurale con unità sigmoidali e un modello parametrico? È lineare? Perché? [3]	23
A.5.6	Se i dati sono acquisiti senza errori, è una buona scelta aumentare di molto i parametri del modello in modo da garantirsi che l'errore sul training set vada a zero? Perché? [3]	23
A.5.7	Cosa si intende per un problema di regressione ed illustrare una possibile soluzione. [3]	23
A.5.8	Come funziona l'approssimazione incrementale multi-scala, cosa garantisce e quali vantaggi può avere? [3]	23
A.5.9	Determinare la forma analitica dell'aggiornamento dei parametri nel caso di unità lineari e di reti a singolo strato. [2]	23
A.5.10	Definire l'algoritmo di apprendimento di una rete neurale con unità lineari e con unità non-lineari. Definire la funzione obiettivo utilizzata.	23
A.6	Domande su Intelligenza Artificiale	23
A.6.1	Si descriva il funzionamento della Forward Search. Perché è considerato un template e non un algoritmo? [2]	23
A.6.2	Si elenchino due possibili implementazioni di Forward Search elencandone proprietà, vantaggi e svantaggi. [2]	23
A.7	Domande su Clustering	24
A.7.1	Cosa si intende per clustering? In quali famiglie vengono divisi? [3]	24
A.7.2	Che relazione c'è tra clustering e classificazione e quali sono le criticità? [3]	24
A.8	Domande su Biologia	25
A.8.1	Definire il neurone biologico evidenziandone le parti più significative per la trasmissione dell'informazione ed il loro comportamento. [2]	25
A.8.2	Descrivere il funzionamento complessivo del neurone biologico.	25
A.8.3	Dove avviene principalmente l'"apprendimento" nei neuroni biologici?	25
A.8.4	Descrivere la modalità di trasmissione dell'informazione nel sistema nervoso e identificare le caratteristiche peculiari.	25
A.8.5	Che differenza c'è tra neuroni motori, neuroni sensoriali ed inter-neuroni? [2]	25
A.8.6	Come viene trasmessa ed elaborata l'informazione da un neurone?	25
A.8.7	Cos'è uno spike? [2]	25
A.8.8	Quali sono le aree corticali principali? [2]	25
A.8.9	Cos'è il codice di popolazione? [2]	25
A.8.10	Data un'area cerebrale è univoca la funzione implementata in quell'area? [2]	25
A.8.11	Cosa sono i mirror neurons? Quali implicazioni hanno per i sistemi intelligenti e l'apprendimento? [2]	25

1.1 Logica fuzzy vs classica

1.1.1 Le funzioni di appartenenza

In logica classica la funzione che descrive la verità di un'affermazione è rappresentabile come una funzione impulsiva, per esempio:

$$\begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Mentre la funzione di appartenenza nella logica fuzzy sono più adeguate funzioni come:

1. Una lineare che aumenta progressivamente da 0 a 1 in un certo Δx determinato.
2. Un sigmoide.
3. Funzioni probabilistiche, come una normale.

1.1.2 Classi di appartenenza

In logica classica le classi sono nette, come nel caso della funzione istintiva si ha una condizione del tipo:

$$\begin{cases} A & x \geq 0 \wedge x < 1 \\ B & x \geq 1 \wedge x < 2 \\ C & x \geq 2 \wedge x < 3 \\ D & x \geq 3 \wedge x < 4 \end{cases}$$

Nella logica fuzzy, vengono descritte per ogni gruppo funzioni che assumono valori anche negli insiemi in cui nella logica classica esse non sono definite. Linearmente esse raggiungono lo 0 mano a mano che esse si sovrappongono con le altre funzioni. In un qualsiasi punto di ascissa, vale la formula:

$$\sum_{i=0}^n m_i = 1$$

1.2 Logica fuzzy e probabilità

Descrivono cose diverse: prendendo per esempio le previsioni meteo, la **probabilità** si occupa di prevedere i mm di pioggia che potrebbero andare a cadere, mentre la **logica fuzzy** si occuperebbe di descrivere il grado di **fuzzyness** tramite il quale andiamo a descrivere quanto è "pioggia", con una funzione che in base a quante gocce di pioggia sono cadute si descrive la *funzione di appartenenza fuzzy* tra le classi "piove" e "non piove".

Ulteriormente, una volta che un evento è avvenuto la sua **probabilità** scompare, nel senso che ora è un dato noto, mentre il valore di **fuzzyness** mantiene il suo valore descrittivo per l'evento.

1.3 Gli operatori logici nella logica fuzzy

Operatore	Logica Classica	Logica Fuzzy
\wedge	$A \wedge B$	$\min(T(A), T(B))$
\vee	$A \vee B$	$\max(T(A), T(B))$
\neg	$\neg A$	$1 - T(A)$

1.4 Misure in un insieme fuzzy

1.4.1 Norma di un vettore

$$M(A) = \sqrt[p]{\sum_{i=1}^n |m_A(x_i)|^p}$$

Figura 1.1: Norma di un vettore

1.4.2 Entropia

Dato un certo punto A , definisco due vettori \vec{a} e \vec{b} che descrivono la posizione del punto A a partire dagli estremi opposti del quadrato.

L'entropia minima risulta pari a 0.

L'entropia massima risulta pari a 1 e si trova nel punto di mezzo (Es. quando una macchina parcheggia tra un posto e l'altro e non è chiaro in quale posto andrebbe vista come parcheggiata). Questa coincide con la **massima fuzzyness** e in questo punto vale che $A \cup A_c = A \cap A_c$.

$$E(A) = \frac{a}{b} = \frac{l^1(A, A_{vicino})}{l^1(A, A_{lontano})}$$

Figura 1.2: Entropia

1.5 Fuzzy Associative Memory FAM

Una FAM trasforma uno spazio di input in uno spazio di output. Esse implementano una serie di regole su delle variabili logiche fuzzy in ingresso.

Le regole sono regole della logica classica, mentre le variabili sono fuzzy.

Una FAM va a descrivere un insieme di classi ed assegna un valore di una funzione di appartenenza ad ogni variabile su ogni classe, poi su queste classi vengono eseguite operazioni di logica classica.

1.5.1 Come opera il sistema

1. Riceve le classi attivate in input
2. Riceve il grado di fit per ogni classe
3. Identifica le regole attivate
4. Determino le classi in uscita attivate
5. Determino il grado di fitness per ogni classe in uscita (regola)
6. Defuzzyficazione

2

Statistica

2.1 Probabilità o visione frequentista

Per il teorema centrale del limite la frequenza di un evento su infinite realizzazioni è uguale alla sua probabilità.

$$P(A = a_1) = \lim_{N \rightarrow \infty} \frac{n_{A=a_1}}{N} = \lim_{N \rightarrow \infty} \frac{n_i}{N}$$

2.1.1 Probabilità di eventi indipendenti e contemporanei

Il prodotto nelle probabilità rappresenta la probabilità che entrambi gli eventi descritti dalle probabilità siano veri, premesso che gli eventi siano **INDIPENDENTI** ed essi non avvengano successivamente. Per esempio, sia $P(A)$ la probabilità che un dato A cada con la faccia esposta pari a 4 e $P(B)$ che un dato B mostri 6. La probabilità che entrambi gli eventi avvengano, cioè sia il dato A cade su 4 e il dato B su 6 è pari al prodotto, cioè $P(A)P(B) = P(A \wedge B)$.

2.1.2 Probabilità condizionata (eventi indipendenti e successivi)

Quando un evento avviene prima di un altro si parla di probabilità condizionata, cioè una tecnica che restringe lo spazio di ricerca della probabilità con cui un evento accadrà sapendo che l'altro ha avuto un determinato esito, probabilisticamente parlando. Ora, se tirassi il dato A dell'esempio precedente, leggendo il risultato prima di tirare il dato B vado a calcolare la probabilità $P(A \wedge B)$ come:

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

In cui la barra verticale nella probabilità viene letta come "La probabilità di B dato che so A".

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

Figura 2.1: Formula delle probabilità condizionate

Esempio su probabilità condizionata: gioco delle carte

Sia dato un mazzo di 40 carte con 12 figure, di cui 4 re.

P. di estrarre un re $P(E) = \frac{\text{Numero di re}}{\text{Numero di carte}} = \frac{4}{40} = \frac{1}{10}$

P. di estrarre un re, sapendo di avere estratto una figura $P(E) = \frac{\text{Numero di re}}{\text{Numero di carte che sono figure}} = \frac{4}{12} = \frac{1}{3}$

2.1.3 Teorema di Bayes

Si tratta di un teorema estremamente utilizzato in statistica e nel machine learning come strumento per l'apprendimento statistico, la cui principale caratteristica è il fatto che permette di trarre deduzioni dalle conclusioni alle cause (inverte le Y con le X), viene chiamato anche **stima a posteriori**. Si deriva dalla formula della probabilità condizionata. In generale, la statistica bayesiana si basa su una modellizzazione tramite la quale è possibile trarre deduzioni sulla realtà, utilizzando il teorema di Bayes (figura 2.2).

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

Figura 2.2: Teorema di Bayes

Esempio su teorema di bayes: i taxi

In una città abbiamo due società di taxi, ed uno di questi investe un anziano che non è particolarmente credibile. Bisogna, con i seguenti dati, andare a capire a quale società questo taxi appartenesse a una delle società.

$$Taxi = \{verde, blue\} = \{85\%, 15\%\}$$

$$Attendibilit_{anziano} = \{vero, falso\} = \{80\%, 20\%\}$$

Applico il teorema di Bayes:

$$P(\text{Taxi incidente blue} | \text{Taxi testimone blu}) = \frac{P(\text{Taxi testimone blu} | \text{Taxi incidente blu}) P(\text{Taxi incidente blu})}{P(\text{Taxi testimone blu})}$$

3

Clustering

3.1 Clustering

La classificazione non-supervisionata, più spesso chiamata *clustering*, consiste nel separare un insieme di dati non etichettati in insiemi, i *cluster*, internamente omogenei.

3.1.1 Ontologia

Obiettivi

Gli obiettivi del clustering possono essere: la ricerca di conferma di ipotesi effettuate a priori, oppure esplorare lo spazio delle *feature*, per effettuare dei ragionamenti a posteriori. Può essere impiegato per effettuare delle statistiche differenziate su più gruppi, oppure per elaborare i dati diversamente a seconda del cluster a cui appartengono.

Dati

I dati in input sono detti *pattern* e sono solitamente valori in uno spazio multidimensionale \mathbb{R}^d . Le caratteristiche dei dati significative per il clustering sono dette *feature*: i *pattern* possono essere presentati come array di *feature* oppure le *feature* possono essere proprietà calcolate a partire dai *pattern*.

Metrica

Spesso, la diversità tra due *pattern* viene espressa come distanza all'interno dello spazio delle *feature*: dovrà essere, quindi, definita la metrica di distanza da utilizzare (distanza euclidea, *Manhattan*, *Mahalanobis*, distanze di *Minkowski*, ...).

Algoritmo

Per effettuare clustering esistono molti tipi di algoritmi, che si dividono principalmente in due classi: algoritmi gerarchici e algoritmi partizionali.

Gli algoritmi partizionali impongono una suddivisione dello spazio delle *feature* in più sottoinsiemi, che sono i cluster: se ogni *pattern* può appartenere ad un solo cluster si parla di *hard clustering*, altrimenti, se ogni *pattern* può appartenere a più cluster con un grado di *membership* si parla di *soft clustering* o *fuzzy clustering*.

Gli algoritmi gerarchici organizzano il dataset in una struttura ad albero dividendo cluster troppo disomogenei (algoritmi divisivi) o unendo cluster simili tra loro (algoritmi agglomerativi). I risultati del clustering gerarchico vengono rappresentati con un albero binario (o un dendrogramma) in cui il nodo radice è il dataset e le foglie sono gli oggetti: i nodi intermedi indicano le divisioni del

dataset in cluster; il risultato finale del clustering si ottiene troncando il dendrogramma ad una certa altezza: si otterrà una foresta in cui ogni albero corrisponde a un cluster.

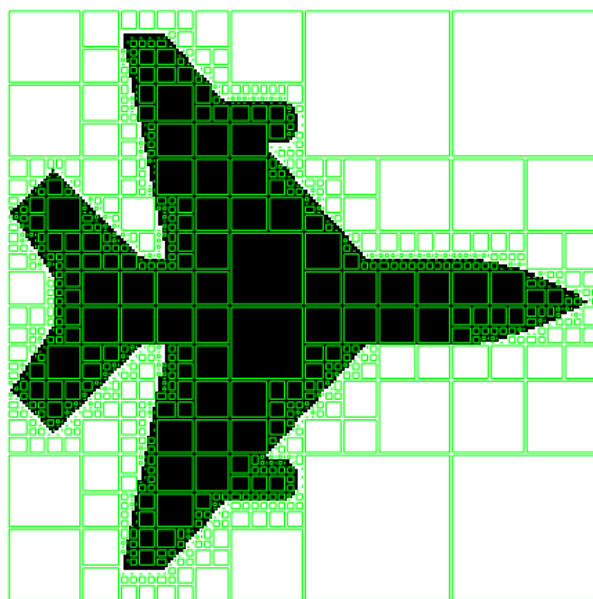
I vantaggi del clustering gerarchico sono l'indipendenza dall'inizializzazione e il fatto che non sia necessario specificare a priori il numero di cluster. Però, sono poco robusti (molto sensibili a rumore e agli outlier), non riconsiderano le scelte già fatte (l'errata classificazione di un punto non viene mai corretta), hanno costo computazionale almeno quadratico ($O(N^2)$) e tendono a creare cluster sferici e fenomeni di inversione.

Validazione

Qualsiasi sia la strategia utilizzata, devono essere applicate delle procedure consolidate di validazione del clustering, per confermare la fondatezza del risultato ottenuto.

3.1.2 Quad-Tree Decomposition

Figura 3.1: Esempio di utilizzo di *quad-tree decomposition* per la compressione di un'immagine raster binaria



Con il termine *quadtree* si indica una classe di strutture dati basate sulla decomposizione ricorsiva dello spazio: essi possono variare per il tipo di dati rappresentati, per il criterio che guida la decomposizione e la risoluzione (che può essere variabile o fissa). È un algoritmo gerarchico divisivo.

Il più comune approccio di tipo *quad-tree decomposition* si basa sulla successiva divisione di un'immagine in quattro quadranti di uguali dimensioni (come nell'esempio in figura 3.1): un quadrante viene suddiviso solo se i pixel al suo interno sono disomogenei. Il nodo radice è l'intera immagine e il caso degenerare della ricorsione è costituito dal singolo pixel (indivisibile). Occorre definire cosa si intende per disomogenei: per immagini binarie si può dire che ogni quadrante è omogeneo solo se tutti i pixel hanno lo stesso valore, per immagini RGB si può imporre una soglia di varianza.

3.1.3 Agglomerative Clustering

Il clustering agglomerativo si basa sulla seguente procedura

1. Ogni oggetto del dataset costituisce un singoletto (un cluster di un solo elemento). Si calcola la matrice di prossimità (la matrice che calcola per ogni coppia di cluster la distanza)
2. Vengono combinati i cluster a distanza minima
3. La matrice di prossimità viene aggiornata con le distanze tra il nuovo cluster e gli altri
4. Se i cluster sono più di uno, ritornare al passo 2

Gli algoritmi di clustering agglomerativo differiscono soprattutto per la politica di calcolo della distanza (oltre che per la funzione di distanza scelta) fra cluster, detta *linkage*. Le più comuni sono

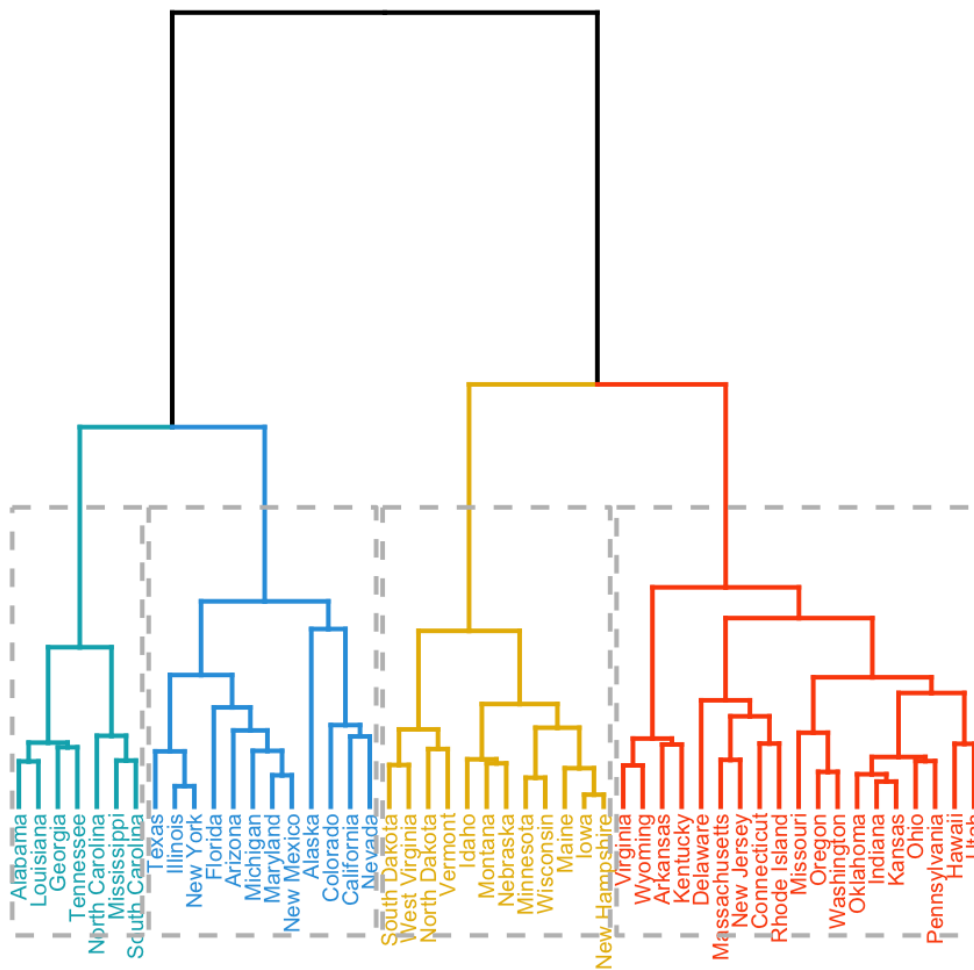
- *Single linkage*: distanza minima tra oggetti dei due cluster
- *Complete linkage*: distanza massima tra oggetti dei due cluster
- *Group average linkage*: distanza media tra oggetti dei due cluster
- *Median linkage*: distanza mediana tra oggetti dei due cluster
- *Centroid linkage*: distanza tra i centroidi dei due cluster
- *Metodo di Ward*: aumento di varianza *within-groups*

Per ognuna di queste definizioni di *linkage*, esistono dei pesi $\{\alpha_i, \alpha_j, \beta, \gamma\}$, tali per cui il valore della distanza del nuovo cluster da uno degli altri cluster C_l può essere ottenuto dai valori di distanza fra C_l e i due cluster C_i e C_j che stiamo unendo, tramite la formula di ricorrenza di Lance e Williams:

$$D(C_l, (C_i, C_j)) =$$

$$\alpha_i D(C_l, C_i) + \alpha_j D(C_l, C_j) + \beta D(C_i, C_j) + \gamma |D(C_l, C_i) - D(C_l, C_j)|$$

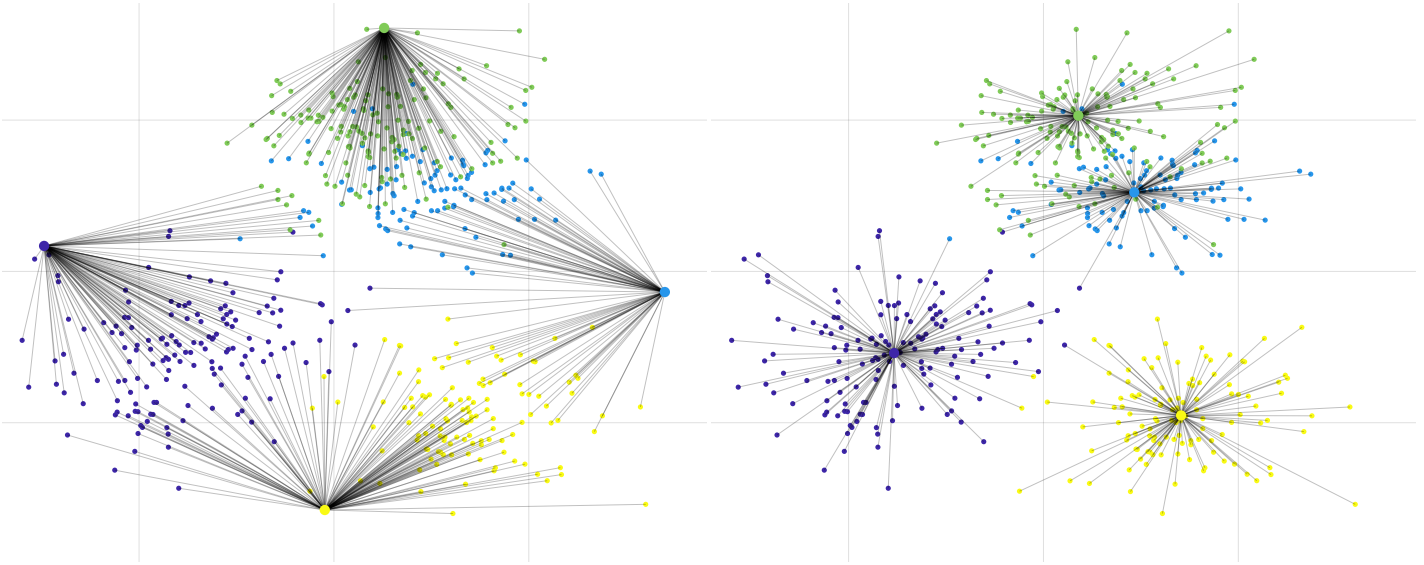
Figura 3.2: Esempio di dendrogramma risultante dal clustering agglomerativo degli stati membri degli USA per numero di arresti



I risultati del clustering agglomerativo vengono solitamente visualizzati con un dendrogramma (figura 3.2).

3.1.4 K-means

Figura 3.3: Esempio di applicazione di K-means: inizializzazione e risultato



Uno dei più famosi algoritmi di clustering è *K-means*: è un algoritmo partitivo *squared error-based*.

1. Viene inizializzata una partizione in K parti del dataset casualmente o in base a informazioni a priori, imponendo un prototipo per il clustering che è un vettore di medie
2. Ogni oggetto viene assegnato al cluster più vicino
3. Il vettore di medie viene aggiornato con le medie dei nuovi cluster
4. Se il vettore è cambiato, tornare al punto 2

Il cluster di appartenenza viene determinato come la media più vicina

$$x_j \in C_w \Leftrightarrow \|x_j - m_w\| < \|x_j - m_i\| \quad \forall i \neq w$$

K-means è molto semplice e si può implementare per risolvere problemi che coinvolgono grandi moli di dati (la sua complessità è lineare nel numero di pattern di input), inoltre, è estremamente parallelizzabile e lavora molto bene su cluster ipersferici.

Presenta vari svantaggi: non esiste un metodo universale ed efficiente per determinare le partizioni iniziali e il numero di cluster. Una strategia generale è di utilizzarlo più volte con inizializzazioni casuali, un'altra consigliata è il metodo di Kaufman. L'algoritmo iterativo non garantisce la convergenza in un ottimo globale: per ottenere ciò in modo efficiente, sono state sviluppate modifiche basate su tecniche stocasticamente ottime e algoritmi genetici.

Inoltre K-means è molto sensibile a rumore e agli outliers: algoritmi di tipo *K-medoids* sono stati proposti, in cui i prototipi dei cluster sono scelti come i medoid dei cluster, ovvero i punti di ogni cluster che minimizzano la somma delle distanze dei punti del cluster dal medoid.

K-means ha anche il problema di basarsi sulla definizione di *media*: non può essere applicato per tipi di dato sul quale non è possibile calcolare la media (o non avrebbe senso): in questi casi si può comunque usare K-medoids.

4

Apprendimento

Il sistema considerato è caratterizzato da **due attori**, l'ambiente e l'agente. L'agente modifica le proprie azioni in base alle reazioni dell'ambiente e questo comportamento adottato è diretto alla massimizzazione di una certa fitness. L'agente cerca di trovare una **policy**, cioè l'insieme delle azioni che in ogni istante massimizzano la **reward**. Lo stato dell'ambiente non cambia sino a che non viene effettuata un'azione (che in questo caso consideriamo le azioni come unicamente prodotte dall'agente). Le azioni esterne possono essere modellizzate o come ulteriori agente o come **interferenze esterne** o **rumore**.

4.1 Value Function

Si tratta del **reward a lungo termine** (Figura 4.1) legato ad una determinata strategia di interazioni con l'ambiente, ed è legata ad una determinata policy π .

$$V^{\pi}(S) = \sum_t^{\infty} R_t$$

Figura 4.1: Value Function

4.1.1 Rappresentazione delle azioni

Il set delle azioni può essere rappresentato tramite un grafo a stati finiti (STG, state transition graph) che considera solitamente lo stato ad alta energia. Un automa solitamente o si muove verso lo stato a energia più bassa con una determinata probabilità o verso lo stato a energia più alta verso lo stato a energia più alta.

Una volta raggiunto lo stato a low energy, solitamente o si va a ricaricare o sta fermo.

4.1.2 Reward a lungo termine

Questo valore è pari al **valore atteso** della somma di tutti i reward da 0 a ∞ per un determinato valore γ .

$$E^{\pi} \left[\sum_t^{\infty} \gamma^t R_t \right]$$

Figura 4.2: Reward a lungo termine



Domande da temi d'esame

A.1 Domande sui Macchine ed Intelligenza

A.1.1 Descrivere il test di Turing, l'esperimento della stanza cinese e l'esperimento della stanza di Maxwell [4]

Test di Turing

Il test di Turing si svolge con 3 agenti. A e B sono rispettivamente una macchina ed un uomo mentre C è un esaminatore. A e B sono in una stanza separata rispetto a C. C pone delle domande sia ad A che a B e questi risponderanno. C dovrà capire quale delle 2 risposte proviene dall'uomo. Se C non riesce a riconoscere chi sia l'uomo rispetto alla macchina, allora la macchina si può definire intelligente. Se si volesse fare un paragone tra questo esperimento ed un altro, si potrebbe introdurre al posto di una macchina una donna. Il funzionamento del "gioco" sarebbe il medesimo e se l'esaminatore riesce a riconoscere chi è l'uomo contro la donna con la stessa percentuale di probabilità dell'uomo contro la macchina, allora la macchina si può definire intelligente.

Stanza Cinese

L'esperimento della stanza cinese è invece stato proposto da John Searle in contrapposizione all'ipotesi che una macchina possa essere intelligente. Secondo Searle una macchina non può essere intelligente in nessun modo in quanto manca di quella che possiamo definire "coscienza". L'esperimento consiste nel mettere una persona in una stanza con un traduttore di simboli cinesi in un alfabeto conosciuto alla persona e un foglio con delle domande scritte in cinese. L'uomo riuscirà a rispondere alle domande pur non avendo coscienza di quel che sta facendo in quanto sta semplicemente traducendo i simboli.

Stanza di Maxwell

Prende il nome dall'esperimento di Maxwell:

L'esperimento di Maxwell consiste nel creare luce utilizzando le onde elettromagnetiche. Questo non è possibile in quanto le sole forze non sono in grado di creare luce.

Sebbene la stanza cinese di Searle possa apparire "semanticamente buia", non vi è nessunissima giustificazione alla sua pretesa, fondata su quest'apparenza, che la manipolazione di simboli secondo certe regole non potrà mai dar luogo a fenomeni semantici, specie se i lettori hanno soltanto una concezione vaga e basata sul buon senso dei fenomeni semantici e cognitivi di cui si cerca una spiegazione. Invece di sfruttare la comprensione che i lettori hanno di queste cose, l'argomento di Searle sfrutta senza troppi scrupoli la loro ignoranza in proposito.

Come mai son stati proposti? Cosa volevano dimostrare?

I 2 esempi sono stati proposti perché Turing sosteneva che una macchina possa essere definita intelligente nel momento in cui una macchina riesce a far credere ad un osservatore di essere una persona, mentre Searle sostiene che una macchina non potrà mai essere definita intelligente in quanto assente di "coscienza".

A.1.2 Discutere la relazione tra algoritmo, macchina di Turing ed intelligenza.

Un algoritmo è una serie di operazioni semplici che eseguite in sequenza porta a un risultato, la macchina di Turing esegue algoritmi sulla base di regole che vengono definite da una funzione chiamata funzione di transizione e sull'inizializzazione della macchina stessa, la macchina termina quando questa arriva in uno stato terminale. Una macchina come la macchina di Turing non è intelligente perché non conosce cosa sta eseguendo ma si limita a compiere le istruzioni che le sono state dettate, al più la si può istruire in modo che sia capace di adattarsi ed eseguire le operazioni che le sono state dettate in base alle situazioni, senza però formulare niente di nuovo.

A.1.3 Cosa si intende per ipotesi forte ed ipotesi debole dell'AI? [4]

Nella filosofia dell'intelligenza artificiale, l'intelligenza artificiale forte è l'idea che opportune forme di intelligenza artificiale possano veramente ragionare e risolvere problemi; l'intelligenza artificiale forte sostiene che è possibile per le macchine diventare sapienti o coscienti di sé, senza necessariamente mostrare processi di pensiero simili a quelli umani. In contrasto con l'intelligenza artificiale forte, l'intelligenza artificiale debole si riferisce all'uso di programmi per studiare o risolvere specifici problemi o ragionamenti che non possono essere compresi pienamente nei limiti delle capacità cognitive umane. Diversamente dall'intelligenza artificiale forte, quella debole non realizza un'auto-consapevolezza e non dimostra il largo intervallo di livelli di abilità cognitive proprio dell'uomo, ma è esclusivamente un problem-solver specifico e, parzialmente, intelligente.

A.1.4 Riportare almeno due elementi del contraddittorio sulle ipotesi su cui è basata l'ipotesi debole sull'AI [3]

1. Una macchina non può originare nulla di nuovo, esegue dei programmi. Una macchina però può imparare dall'esperienza e quindi costruirsi una cultura in grado di migliorare.
2. Il comportamento intelligente non può essere completamente replicato.
3. Il comportamento intelligente non può essere completamente catturato da regole formali (argument for informality).
4. Anche se un computer si comportasse in modo da superare il test di Turing, non sarebbe comunque classificato come intelligente.

A.1.5 Descrivere il "Brain prosthesis thought experiment" di Moravec e commentarlo. [3]

Ipotizzando di disporre di neuroni artificiali equivalenti a quelli biologici, l'esperimento consisterebbe nel sostituire uno ad uno ogni neurone, fino ad ottenere un cervello completamente artificiale. Secondo Moravec, questo cervello conterrebbe la stessa mente del cervello di partenza, risposta funzionalista. Secondo Searle, la coscienza sparirebbe ma il comportamento visibile all'esterno sarebbe indistinguibile dall'originale, risposta strutturalista.

Vanno quindi prese in esame le varie possibili conclusioni: 1. i meccanismi causali della coscienza stanno ancora operando nel cervello elettronico, 2. gli eventi mentali non sono presenti nel cervello elettronico, quindi non 'è cosciente, 3. l'esperimento 'è inattuabile, pertanto proporre ipotesi al riguardo non ha alcun senso. Escludiamo la terza possibilità perché siamo interessati alla questione filosofica e non alla realizzabilità dell'esperimento. La seconda opzione proposta ci porta a ritenere la coscienza come un qualcosa che è influente ai fini della determinazione dell'output del soggetto. Avendo, però, riprodotto il funzionamento di un cervello reale, dovremmo anche asserire che gli eventi mentali consci nel cervello umano non hanno un collegamento casuale con il comportamento. Se quindi accettiamo il fatto che l'esperimento della sostituzione del cervello dimostra che quello elettronico è cosciente, dobbiamo concordare che la coscienza è conservata anche quando l'intero cervello è sostituito da un insieme di chip elettronici; ovvero la prima opzione proposta.

A.2 Domande sui Sistemi Fuzzy

A.2.1 Cose un insieme fuzzy?

A.2.2 Esiste una corrispondenza biunivoca tra insiemi fuzzy e valori numerici?

A.2.3 Cose una membership function? Con quali altri nomi viene anche indicata?

A.2.4 La frase seguente: con la mia preparazione potrei prendere 24 all'esame, sottintende un processo fuzzy o probabilistico?

A.2.5 Cosa si intende per FAM? [3]

A.2.6 Una FAM memorizza numeri o preposizione logiche? Come? [3]

A.2.7 Descrizione del funzionamento di una FAM.

A.2.8 Definire i passi per costruire un sistema fuzzy

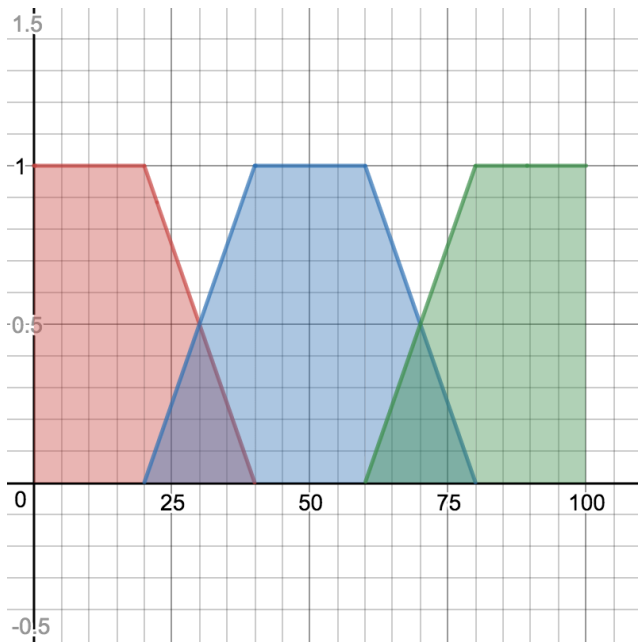
1. Identificazione delle variabili di I/O del sistema e del loro range (A,B).
2. Identificazione delle classi fuzzy in cui le variabili sono da suddividere e dei loro boundaries.
3. Definizione della trasformazione I/O come insieme di regole fuzzy: per ogni combinazione di classi fuzzy (con OR e/o AND) di input è possibile definire una classe di output (FAM).
4. Modalità di de-fuzzyfication.

A.2.9 Definire un problema a piacere che involva almeno due variabili in ingresso e due in uscita. Definire tutti i componenti e calcolare l'uscita passo a passo per un valore di input a piacere

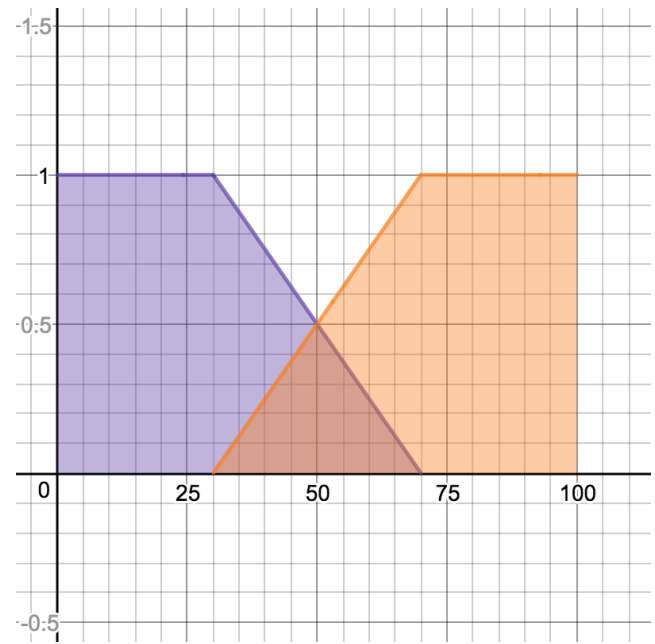
Ipotizziamo di dover gestire la coda di un callcenter di supporto clienti.

Variabili in ingresso

1. Numero di persone che telefonano all'ora.
Può variare tra 0 e 100.
Viene classificato in:
Alto numero
Medio numero
Basso numero
2. Carico di lavoro, che dipende dal genere di supporto che i clienti richiedono.
Viene trattato come una percentuale.
Viene classificato in:
Carico leggero
Carico pesante



(a) Rappresentazioni delle classi fuzzy del numero di persone (Basso, medio e alto)



(b) Rappresentazioni delle classi fuzzy del carico di lavoro (leggero e pesante)

Variabili in uscita

1. Numero di centralinisti da attivare

L'idea è di attivare più centralinisti quando il carico è elevato.

Può variare da 0 a 50.

Classificato in:

Poco personale: 10 persone

Personale medio: 30 persone

Personale completo: 50 persone

2. Tempo dedicabile ad ogni chiamata

L'idea è di dedicare meno tempo se il carico di lavoro ed il numero di clienti è elevato.

Può variare dai 5 ai 30 minuti

Classificato in:

Poco tempo: 5 minuti

Medio tempo: 15 minuti

Molto tempo: 30 minuti

Costruiamo le regole della FAM per il personale

1. ALTO \wedge LEGGERO = PERSONALE MEDIO
2. ALTO \wedge PESANTE = PERSONALE COMPLETO
3. MEDIO \wedge LEGGERO = POCO PERSONALE
4. MEDIO \wedge PESANTE = PERSONALE MEDIO
5. BASSO \wedge LEGGERO = POCO PERSONALE
6. BASSO \wedge PESANTE = PERSONALE MEDIO

Costruiamo le regole della FAM per il tempo

1. ALTO \wedge LEGGERO = POCO TEMPO
2. ALTO \wedge PESANTE = POCO TEMPO
3. MEDIO \wedge LEGGERO = MEDIO TEMPO
4. MEDIO \wedge PESANTE = MEDIO TEMPO
5. BASSO \wedge LEGGERO = MOLTO TEMPO
6. BASSO \wedge PESANTE = MOLTO TEMPO

Definiamo una regola di defuzzificazione

Utilizzo la media pesata come regola di defuzzificazione.

Esempio: 25 persone chiamano ed il carico è al 40%

25 persone in chiamata significa attivare al 50% la classe BASSO, al 50% la classe MEDIO e non attivare la classe ALTO.

Il carico di lavoro al 40% significa attivare al 75% la classe LEGGERO ed al 25% la classe PESANTE.

FAM per il personale:

1. ALTO=0 \wedge LEGGERO=0.75 = PERSONALE MEDIO=0
2. ALTO=0 \wedge PESANTE=0.25 = PERSONALE COMPLETO=0
3. MEDIO=0.5 \wedge LEGGERO=0.75 = POCO PERSONALE=0.5
4. MEDIO=0.5 \wedge PESANTE=0.25 = PERSONALE MEDIO=0.25
5. BASSO=0.5 \wedge LEGGERO=0.75 = POCO PERSONALE=0.5
6. BASSO=0.5 \wedge PESANTE=0.25 = PERSONALE MEDIO=0.25

FAM per il tempo:

1. ALTO=0 \wedge LEGGERO=0.75 = POCO TEMPO=0
2. ALTO=0 \wedge PESANTE=0.25 = POCO TEMPO=0
3. MEDIO=0.5 \wedge LEGGERO=0.75 = MEDIO TEMPO=0.5
4. MEDIO=0.5 \wedge PESANTE=0.25 = MEDIO TEMPO=0.25
5. BASSO=0.5 \wedge LEGGERO=0.75 = MOLTO TEMPO=0.5
6. BASSO=0.5 \wedge PESANTE=0.25 = MOLTO TEMPO=0.25

Eseguo la media pesante per defuzzificare i risultati:

$$n_{\text{centralinisti}} = \frac{(0.5 + 0.5) * \text{POCO} + (0.25 + 0.25) * \text{MEDIO}}{0.5 + 0.5 + 0.25 + 0.25} = \frac{(0.5 + 0.5) * 10 + (0.25 + 0.25) * 30}{0.5 + 0.5 + 0.25 + 0.25} = \frac{10 + 15}{1.5} \approx 17$$

$$n_{\text{minuti}} = \frac{(0.5 + 0.25) * \text{MEDIO} + (0.5 + 0.25) * \text{MOLTO}}{0.5 + 0.5 + 0.25 + 0.25} = \frac{(0.5 + 0.25) * 15 + (0.5 + 0.25) * 30}{0.5 + 0.5 + 0.25 + 0.25} = \frac{11.25 + 22.5}{1.5} \approx 23$$

Le FAM così costruite suggeriscono quindi 17 centralinisti che dedicano 23 minuti per cliente.

A.3 Domande su Apprendimento con Rinforzo

A.3.1 Cosa si intende per Apprendimento con Rinforzo?

L'apprendimento per rinforzo è una tecnica di apprendimento automatico che viene utilizzata per ideare sistemi capaci di apprendere ed adattarsi all'ambiente che li circonda grazie ad una "ricompensa", detta rinforzo, che consiste nella valutazione delle loro prestazioni. L'apprendimento avviene mediante l'interazione con l'ambiente ed è funzione del raggiungimento di uno o più obiettivi.

A.3.2 Quali sono gli attori?

I principali attori quando si parla di reinforcement learning sono:

Agente software che svolge servizi per conto di un altro programma, solitamente in modo automatico ed invisibile. Formulato un problema l'agente cerca una soluzione e la implementa. Valuta inoltre la soluzione implementata e ragiona se ha avuto successo o no, se e quanto è adeguata e cerca di ottimizzare le prestazioni per creare agenti migliori. L'agente sente l'input, modifica lo stato e genera un'azione che massimizza la ricompensa a lungo termine.

Stato variabile interna all'agente che contiene una rappresentazione interna dell'ambiente. I possibili stati sono un insieme finito.

Azione le possibili azioni applicabili sull'ambiente. Sono un insieme finito contenuto nell'agente. Viene selezionata l'azione da effettuare in base alla policy.

Policy descrive l'azione scelta dall'agente. Mapping tra stato (input dall'ambiente) e azione. Funzione di controllo. Le policy possono avere una componente stocastica.

Ambiente descrive tutto quello su cui agisce la policy. È tutto quanto quello che non è modificabile direttamente dall'agente. Si può rappresentare come una funzione che preso uno stato e una azione come input restituisce un altro stato come output, ma è una funzione non conosciuta a priori. L'agente deve costruirsi una rappresentazione implicita dell'ambiente attraverso la value function e deve selezionare i comportamenti che ripetutamente risultano favorevoli a lungo termine.

Segnale di rinforzo è un'informazione qualitativa (a volte binaria: giusto/sbagliato, successo/fallimento), puntuale. Due tipi: istantaneo, cioè azione per azione (condizionamento classico), o una tantum, (condizionamento operante) cioè viene rinforzata una catena di azioni, un comportamento. Viene generato all'esterno dell'agente e per esso rappresenta un input.

Reward/Quality function è la ricompensa immediata. Associata all'azione intrapresa in un certo stato. Può essere data al raggiungimento di un goal. È uno scalare (può essere associato allo stato e/o input e/o stato prossimo).

Value function ricompensa a lungo termine. Somma dei reward: costi associati alle azioni scelte istante per istante più costo associato allo stato finale. Orizzonte temporale ampio. Rinforzo secondario. Ricompensa attesa. Viene stimata all'interno dell'agente.

Goal obiettivo che deve raggiungere l'agente. Si può aggiungere che l'agente deve raggiungere l'obiettivo con una policy ottima.

A.3.3 Cosa rappresenta la critica?

L'ambiente o l'interazione con esso può essere complessa e/o variabile, inoltre il rinforzo può avvenire solo dopo una più o meno lunga sequenza di azioni (delayed reward). Questo genera difficoltà nell'analisi e nella valutazione di ambiente e azioni, questi problemi prendono il nome di: temporal credit assignment e structural credit assignment. L'apprendimento non è più determinato da esempi, ma dall'osservazione del proprio comportamento nell'ambiente. È fondamentale definire come vengono assegnati i reward, che possono essere molto direzionati nel tempo (temporal CA problem) o distribuiti dall'interno di una struttura complessa (structural CA problem). Il problema consiste nell'identificare quali siano le azioni che hanno inferito maggiormente nel raggiungimento di uno stato di goal ad alto reward. La critica è la value function stimata, ovvero una componente che definisce per ogni stato un'altra value function, che si unisce a quella dell'agente e in questo modo si rinforza. È una value function alternativa.

A.3.4 Che tipo di architettura si può ipotizzare nell'apprendimento con rinforzo?

L'architettura dei sistemi di RL si basa sulla retroazione. L'agente si muove nell'ambiente utilizzando una Policy, l'ambiente fornisce un reward e fa cambiare lo stato dell'agente il quale utilizza i risultati ottenuti per aggiornare la propria Policy così da raggiungere il goal. Tramite dei sensori l'agente monitora continuamente l'ambiente (input), tramite degli attuatori compie azioni che modificano lo stato dell'ambiente.

A.3.5 Condizionamento classico e condizionamento operante

Nel condizionamento classico il rinforzo viene eseguito istante per istante o azione per azione e permette di ottenere un riscontro ad ogni azione eseguita o ad ogni variazione dell'ambiente o dello stato. Nel condizionamento operante invece il rinforzo avviene "una-tantum", viene quindi valutata una catena di azioni, un comportamento nel suo insieme e non nella singola azione.

A.3.6 Quale relazione c'è con l'intelligenza?

Se consideriamo l'intelligenza come una funzione attiva, che permetta di prendere decisioni e compiere azioni nell'interazione con l'ambiente, il Reinforcement Learning consente di far comportare un agente in modo intelligente.

A.3.7 Come potreste illustrare: Exploration vs Exploitation?

L'esplorazione (exploration) consiste nel provare varie azioni per scoprire nuove possibili azioni (promettenti). Esplorazione dello spazio delle azioni per scoprire quelle migliori. Un agente che esplora solamente raramente troverà una buona soluzione. Exploitation consiste nello scegliere sempre la soluzione che garantisca il miglior reward tra quelle conosciute (le informazioni vengono raccolte tramite exploration). Se un agente non esplora nuove soluzioni potrebbe essere surclassato da nuovi agenti più dinamici. Inoltre l'ambiente potrebbe essere mutevole e una soluzione potrebbe non restare valida. Occorre quindi non interrompere del tutto l'esplorazione e usare un approccio statistico per valutare le bontà delle azioni bilanciando Exploration ed exploitation.

A.3.8 Cos'è il problema del credit assignment? È un problema che riguarda la dimensione temporale o spaziale del task?

I sistemi time-extended single-agent hanno il problema di valutare il contributo di un'azione rispetto alle altre (temporal credit assignment). Questo problema viene accentuato se le azioni rilevanti sono temporalmente molto distanti tra di loro perché è come se fossero spalmate su azioni non rilevanti e questo può portare a cattive valutazioni. Un algoritmo dovrebbe riuscire a valutare l'influenza di azioni precedenti rispetto all'azione finale. Sistemi multi-agente hanno inoltre il problema di determinare il contributo di un particolare agente ad un compito comune (structural credit assignment).

A.3.9 Cos'è l'eligibility trace (traccia) e quale è il suo ruolo?

L'eligibility trace è un buffer di memoria contenente tracce di eventi passati. Quando viene calcolato un errore usando metodi basati su TD (Temporal Difference), l'eligibility trace suggerisce quali variabili aggiornare. Amplia l'orizzonte temporale sul quale fare l'aggiornamento a più di 1 passo definendo così se uno stato è eleggibile e "quanto", cioè che percentuale di aggiornamento meriti. L'eligibility trace è la base per risolvere i problemi di credit assignment. Gli eventi passati che vengono chiamati tracce e rappresentano stati visitati, azioni compiute, ecc. evaporano nel tempo, cioè diventano sempre meno rilevanti fino a scomparire.

A.3.10 Definire l'algoritmo di Q-learning, descrivendo le equazioni opportune. [2]

L'algoritmo Q-Learning è un algoritmo di apprendimento per rinforzo il cui obiettivo è permettere al sistema di adattarsi all'ambiente circostante migliorando la scelta delle azioni da eseguire allo scopo di massimizzare la ricompensa totale.

A.3.11 Scrivere le equazioni dell'algoritmo Q-learning in cui si consideri anche la traccia. [2]

A.3.12 Cosa si intende per politica epsilon-greedy? Come entra nell'algoritmo di Q-learning?

A.3.13 Che differenza c'è tra Q-learning e SARSA? [2]

Le differenze risiedono nel modo in cui viene aggiornato il valore di Q ad ogni azione verificata, SARSA è on-policy mentre Q-Learning è off-policy. Ovvero Q-learning aggiorna il valore in funzione della migliore possibile scelta futura, mentre SARSA aggiorna il valore in funzione della scelta che verrà applicata dalla policy. Nella pratica Q-learning converge più lentamente di SARSA, ma è in grado di imparare più rapidamente nel caso l'ambiente si modifichi in quanto SARSA continuerebbe a seguire la propria policy.

A.3.14 Dato un problema a piacere si descriva uno degli algoritmi e mostrare due passaggi di addestramento

A.3.15 Quale criterio si sceglie per definire i Reward? A quali elementi sono associati? Allo stato? All'azione? Allo stato prossimo? Perché? [2]

L'ambiente fornisce un'informazione qualitativa, ad esempio success or fail. L'informazione disponibile si chiama segnale di rinforzo e non dà alcuna informazione su come aggiornare il comportamento dell'agente (e.g. i pesi). È per questo che il reward deve essere definito dall'agente in base alla politica adottata, e quindi si necessita di sistemi intelligenti che riescano a valutare al meglio le azioni da intraprendere. In base alla politica adottata il reward sarà legato allo stato, all'azione, allo stato prossimo o a un mix di essi.

A.3.16 Impostare un problema su griglia (apprendimento del percorso di un agente, con partenza ed arrivo prescelti + ostacoli). La griglia fornisce un reward, diverso da zero, in ogni transizione. [2]

Definire chiaramente il problema, farne un modello definendo le variabili e le funzioni che le legano. [2]

Scrivere un risultato possibile dei primi 2 passi di apprendimento del problema definito al punto precedente. [2]

A.4 Domande su Statistica

A.4.1 Esercizio sui Taxi

In una città lavorano due compagnie di taxi: blue e verde, la maggior parte dei taxisti lavorano per la compagnia verde per cui si ha la seguente distribuzione di taxi in città: 85% di taxi verdi e 15% di taxi blu. Succede un incidente in cui è coinvolto un taxi. Un testimone dichiara che il taxi era blu. Era sera e buio, c'era anche un po' di nebbia ma il testimone ha una vista acuta, la sua affidabilità è stata valutata del 70%. Qual è la probabilità che il taxi fosse effettivamente blu? Quale deve essere l'affidabilità del testimone perché la probabilità che il taxi fosse effettivamente blu sia del 99%?

A.4.2 Esercizio sul Tumore al seno

Lo strumento principe per lo screening per il tumore al seno è la radiografia (mammografia). Definiamo X la situazione della donna: X=sana, malata, che non conosciamo. Definiamo Y l'esito della mammografia: Y=positiva, negativa, che viene misurato. Sappiamo che la sensibilità della mammografia è intorno al 90% ($P(Y=\text{positiva} | X=\text{malata})$) e che la specificità sia anch'essa del 90% ($P(Y=\text{negativa} | X=\text{sana})$). Qual è la probabilità che l'esame dia risultato positivo ($P(Y = \text{positivo})$), sapendo che le donne malate sono lo 0,01% ($P(X=\text{malata}) = 0,01\%$)? Qual è la percentuale di donne che hanno uno screening positivo, di essere effettivamente malate?

A.4.3 Esercizio delle Macchine [2]

Tre macchine, A B, e C, producono rispettivamente il 50%, il 40%, e il 10% del numero totale dei pezzi prodotti da una fabbrica. Le percentuali di produzione difettosa di queste macchine sono rispettivamente del 2%, 1% e 4%. Determinare la probabilità di estrarre un pezzo difettoso. Viene estratto a caso un pezzo che risulta difettoso. Determinare la probabilità che quel pezzo sia stato prodotto dalla macchina C.

A.4.4 Enunciare il teorema di Bayes

Data una partizione dello spazio degli eventi $A_1 \dots A_n$, vale che:

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{\sum_{j=1}^n P(E|A_j)P(A_j)}$$

A.4.5 Discutere l'analisi di varianza per un sistema lineare [4]

Svolgere un'analisi di varianza per un sistema lineare significa analizzare quanto la stima di un parametro possa variare nelle diverse misure dei dati relativi al problema. L'analisi consente di esaminare la matrice dei covarianti, misurare quanto varia una misura di una variabile al variare del rumore e misurare quanto covariano due misure di due variabili. L'indice di correlazione di due variabili viene calcolato proprio per misurare quanto le variabili si trovino lungo una funzione.

A.4.6 Dimostrare che la stima ai minimi quadrati è equivalente alla stima a massima verosimiglianza nel caso di errore Gaussiano sui dati. Cosa fornisce? Come? [3]

Scriviamo il logaritmo negativo della verosimiglianza:

$$P(y_1, \dots, y_n; n, b; x_1, \dots, x_n) = - \sum_{i=1}^n \ln \left\{ \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{1}{2} \left(\frac{y_i - mx_i - b}{\sigma} \right)^2 \right] \right\} \quad (\text{A.1})$$

$$= - \sum_{i=1}^n \ln \left(\frac{1}{\sqrt{2\pi}\sigma} \right) - \sum_{i=1}^n \left[-\frac{1}{2} \left(\frac{y_i - mx_i - b}{\sigma} \right)^2 \right] \quad (\text{A.2})$$

$$= - \sum_{i=1}^n \ln \left(\frac{1}{\sqrt{2\pi}\sigma} \right) + \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 \quad (\text{A.3})$$

Massimizziamo la **likelihood** ponendo a zero le derivate prime rispetto a m :

$$\frac{\partial P(y_1, \dots, y_n; n, b; x_1, \dots, x_n)}{\partial m} = \frac{\partial}{\partial m} \left[- \sum_{i=1}^n \ln \left(\frac{1}{\sqrt{2\pi}\sigma} \right) + \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 \right] \quad (\text{A.4})$$

$$= 0 + \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 2(-x_i) \quad (\text{A.5})$$

$$= -\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 x_i \quad (\text{A.6})$$

$$-\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 x_i = 0$$

$$\sum_{i=1}^n (y_i - mx_i - b)^2 x_i = 0$$

$$m \left[\sum_{i=1}^n (x_i^2) \right] + q \left[\sum_{i=1}^n (x_i) \right] = \left[\sum_{i=1}^n (y_i x_i) \right]$$

Figura A.2: Prima equazione

Massimizziamo la **likelihood** ponendo a zero le derivate prime rispetto a q :

$$\frac{\partial P(y_1, \dots, y_n; n, b; x_1, \dots, x_n)}{\partial q} = \frac{\partial}{\partial q} \left[- \sum_{i=1}^n \ln \left(\frac{1}{\sqrt{2\pi}\sigma} \right) + \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 \right] \quad (\text{A.7})$$

$$= 0 + \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 2(-1) \quad (\text{A.8})$$

$$= -\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 \quad (\text{A.9})$$

$$\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - mx_i - b)^2 = 0$$

$$\sum_{i=1}^n (y_i - mx_i - b)^2 = 0$$

$$m \left[\sum_{i=1}^n (x_i) \right] + q \left[\sum_{i=1}^n (1) \right] = \left[\sum_{i=1}^n (y_i) \right]$$

Figura A.3: Seconda equazione

Ponendo a sistema le equazioni così ottenute ottengo:

$$\begin{bmatrix} [\sum_{i=1}^n (x_i^2)] & [\sum_{i=1}^n (x_i)] \\ [\sum_{i=1}^n (x_i)] & n \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} [\sum_{i=1}^n (y_i x_i)] \\ [\sum_{i=1}^n (y_i)] \end{bmatrix}$$

Lo stesso problema visto dal punto di vista dei **minimi quadrati** è impostato nel seguente modo.

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & n \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

L'obbiettivo è trovare una x tale che $(Ax - b)^T (Ax - b)$ è minima (minimizzazione di residui). La soluzione si ottiene calcolando $A^T Ax = A^T b$.

$$A^T A = \begin{bmatrix} x_1 & \dots & x_n \\ 1 & \dots & n \end{bmatrix} \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & n \end{bmatrix} = \begin{bmatrix} [\sum_{i=1}^n (x_i^2)] & [\sum_{i=1}^n (x_i)] \\ [\sum_{i=1}^n (x_i)] & n \end{bmatrix}$$

$$A^T b = \begin{bmatrix} x_1 & \dots & x_n \\ 1 & \dots & n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} [\sum_{i=1}^n (y_i x_i)] \\ [\sum_{i=1}^n (y_i)] \end{bmatrix}$$

Ovvero:

$$\begin{bmatrix} [\sum_{i=1}^n (x_i^2)] & [\sum_{i=1}^n (x_i)] \\ [\sum_{i=1}^n (x_i)] & n \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} [\sum_{i=1}^n (y_i x_i)] \\ [\sum_{i=1}^n (y_i)] \end{bmatrix}$$

Che è la stessa soluzione ottenuta per la stima a massima verosimiglianza. Queste metodologie offrono una stima dei parametri di una funzione tramite la minimizzazione dei residui. La soluzione è quella che minimizza lo scarto quadratico medio dei residui, ovvero è a minima varianza.

A.5 Domande su Apprendimento Supervisionato

- A.5.1 Definire l'algoritmo di apprendimento di una rete neurale con unità arbitrarie. Definire la funzione obbiettivo utilizzata. [3]**
- A.5.2 Come si utilizza la funzione obbiettivo nell'algoritmo di apprendimento. [3]**
- A.5.3 Cosa si intende per apprendimento per epoche e per trial? Qual è il vantaggio di ciascuna delle modalità di apprendimento? [3]**
- A.5.4 Cosa si intende per training e test set? Perché mai vengono utilizzati? Quali problemi si vogliono evitare? [3]**
- A.5.5 Una rete neurale con unità sigmoidali e un modello parametrico? È lineare? Perché? [3]**
- A.5.6 Se i dati sono acquisiti senza errori, è una buona scelta aumentare di molto i parametri del modello in modo da garantirsi che l'errore sul training set vada a zero? Perché? [3]**
- A.5.7 Cosa si intende per un problema di regressione ed illustrare una possibile soluzione. [3]**
- A.5.8 Come funziona l'approssimazione incrementale multi-scala, cosa garantisce e quali vantaggi può avere? [3]**
- A.5.9 Determinare la forma analitica dell'aggiornamento dei parametri nel caso di unità lineari e di reti a singolo strato. [2]**
- A.5.10 Definire l'algoritmo di apprendimento di una rete neurale con unità lineari e con unità non-lineari. Definire la funzione obbiettivo utilizzata.**

A.6 Domande su Intelligenza Artificiale

- A.6.1 Si descriva il funzionamento della Forward Search. Perché è considerato un template e non un algoritmo? [2]**

L'idea è quella di esplorare il grafo partendo dal nodo iniziale, provando a trovare la strada per arrivare ad uno stato di goal. Ad ogni step della ricerca un nodo può essere etichettato in 3 modi:

1. **Unvisited:** deve essere ancora visitato dall'algoritmo.
2. **Alive:** visitato, ma l'algoritmo deve ancora visitare i nodi direttamente raggiungibili da esso. I nodi alive sono raccolti in una coda di priorità a .
3. **Dead:** visitato, ed anche ogni nodo vicino è stato visitato.

È considerato un template e non un algoritmo perché non è specificato il criterio con cui ordinare a .

- A.6.2 Si elenchino due possibili implementazioni di Forward Search elencandone proprietà, vantaggi e svantaggi. [2]**

Breadth first search

La coda a è gestita in modo FIFO (First-in First-out). I percorsi con $k + 1$ azioni vengono valutati dopo che ogni percorso con k azioni è stato esplorato. Se viene trovato il percorso, è garantito che questo avrà il minor numero di azioni (percorso più breve). Funziona in un tempo $O(|V| + |E|)$. Per costruire l'albero di ricerca individua in tutti i figli con la stessa profondità e garantisce il percorso più breve. Inoltre è simmetrico.

Depth first search

La coda a è uno stack gestito con una politica LIFO (Last-in First-out). È un algoritmo più "aggressivo" e cerca prima soluzioni nei percorsi più lunghi. Funziona con un tempo $O(|V| + |E|)$. Ha il vantaggio di ignorare cammini sbagliati nel caso in cui si trovi immediatamente una soluzione. Non è sistematico su spazi infiniti.

A.7 Domande su Clustering

A.7.1 Cosa si intende per clustering? In quali famiglie vengono divisi? [3]

La classificazione non-supervisionata, più spesso chiamata *clustering*, consiste nel separare un insieme di dati non etichettati in insiemi, i *cluster*, internamente omogenei. Per effettuare clustering esistono molti tipi di algoritmi, che si dividono principalmente in due classi: algoritmi gerarchici e algoritmi partizionali.

Gli algoritmi gerarchici organizzano il dataset in una struttura ad albero dividendo cluster troppo disomogenei (algoritmi divisivi) o unendo cluster simili tra loro (algoritmi agglomerativi).

Gli algoritmi partizionali impongono una suddivisione dello spazio delle *feature* in più sottoinsiemi, che sono i cluster: se ogni *pattern* può appartenere ad un solo cluster si parla di *hard clustering*, altrimenti, se ogni pattern può appartenere a più cluster con un grado di *membership* si parla di *soft clustering* o *fuzzy clustering*.

Alcuni algoritmi possono essere basati su teorie probabilistiche: si parla di algoritmi statistici.

A.7.2 Che relazione c'è tra clustering e classificazione e quali sono le criticità? [3]

Il clustering consiste nel separare un insieme di dati non etichettati in sottoinsiemi, mentre la classificazione separa un dataset in insiemi di dati etichettati. Per la loro somiglianza, il clustering viene anche chiamato *classificazione non supervisionata*.

Il clustering è un problema di apprendimento non supervisionato, in cui il sistema non riceve alcun riscontro sulla correttezza della propria soluzione, al contrario, la classificazione è un problema di apprendimento supervisionato, in cui il sistema viene allenato con un *training set*: oltre al pattern di ingresso, viene fornita al sistema quale è la soluzione desiderata (la classe di appartenenza).

Per validare le performance di un classificatore è possibile utilizzare il sistema già addestrato su un insieme di dati nuovi, un *test set*: questo procedimento testa la capacità del sistema di generalizzare e può rilevare il verificarsi di *overfitting*.

Per validare un sistema di clustering, invece, bisogna validare l'algoritmo stesso: inoltre, la scelta dell'algoritmo può variare notevolmente la soluzione. Infatti, il clustering viene considerato un problema mal posto.

Altri fattori che possono influenzare sulla performance di un sistema di clustering sono: lo spazio di rappresentazione dei pattern (*feature space*), la metrica di distanza implementata (distanza euclidea, *Manhattan*, *Mahalanobis*, distanze di *Minkowski*, ...).

Per gli algoritmi gerarchici agglomerativi, i risultati sono influenzati dalla strategia di *linkage* utilizzata, mentre per gli algoritmi divisivi bisogna quantificare l'omogeneità dei cluster. Algoritmi come K-means sono molto soggetti all'inizializzazione: gli algoritmi gerarchici no, ma sono comunque sensibili agli *outlier*; inoltre, gli algoritmi gerarchici non riconsiderano in nessun passo le decisioni effettuate nei passi precedenti per cercare di correggere eventuali misclassificazioni.

A.8 Domande su Biologia

- A.8.1 Definire il neurone biologico evidenziandone le parti più significative per la trasmissione dell'informazione ed il loro comportamento. [2]**
- A.8.2 Descrivere il funzionamento complessivo del neurone biologico.**
- A.8.3 Dove avviene principalmente l'"apprendimento" nei neuroni biologici?**
- A.8.4 Descrivere la modalità di trasmissione dell'informazione nel sistema nervoso e identificare le caratteristiche peculiari.**
- A.8.5 Che differenza c'è tra neuroni motori, neuroni sensoriali ed inter-neuroni? [2]**
- A.8.6 Come viene trasmessa ed elaborata l'informazione da un neurone?**
- A.8.7 Cos'è uno spike? [2]**
- A.8.8 Quali sono le aree corticali principali? [2]**
- A.8.9 Cos'è il codice di popolazione? [2]**
- A.8.10 Data un'area cerebrale è univoca la funzione implementata in quell'area? [2]**
- A.8.11 Cosa sono i mirror neurons? Quali implicazioni hanno per i sistemi intelligenti e l'apprendimento? [2]**