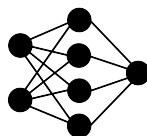


# **SISTEMI INTELLIGENTI**

Prof. Nunzio Alberto Borghesi  
Prof. Nicola Basilico  
6 CFU

**Luca Cappelletti**  
**Marco Tiraboschi**

Lecture Notes  
Year 2017/2018



Magistrale Informatica  
Università di Milano  
Italy  
8 maggio 2018

# Indice

<b>1</b>	<b>Note sulla logica fuzzy</b>	<b>4</b>
1.1	Logica fuzzy vs classica . . . . .	4
1.1.1	Le funzioni di appartenenza . . . . .	4
1.1.2	Classi di appartenenza . . . . .	4
1.2	Logica fuzzy e probabilità . . . . .	4
1.3	Gli operatori logici nella logica fuzzy . . . . .	5
1.4	Misure in un insieme fuzzy . . . . .	5
1.4.1	Norma di un vettore . . . . .	5
1.4.2	Entropia . . . . .	5
1.5	Fuzzy Associative Memory FAM . . . . .	5
1.5.1	Come opera il sistema . . . . .	5
<b>2</b>	<b>Note su statistica</b>	<b>6</b>
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
<b>3</b>	<b>Note sul clustering</b>	<b>8</b>
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
<b>4</b>	<b>Domande da temi d'esame</b>	<b>12</b>
4.1	Domande su Macchine ed Intelligenza . . . . .	12
4.1.1	Descrivere il test di Turing, l'esperimento della stanza cinese e l'esperimento della stanza di Maxwell . . . . .	12
4.1.2	Discutere la relazione tra algoritmo, macchina di Turing ed intelligenza. . . . .	12
4.1.3	Cosa si intende per ipotesi forte ed ipotesi debole dell'AI? . . . . .	13
4.1.4	Riportare il contraddittorio sulle ipotesi su cui è basata l'ipotesi debole sull'AI . . . . .	13
4.1.5	Descrivere il "Brain prosthesis thought experiment" di Moravec e commentarlo. . . . .	13
4.2	Domande sui Sistemi Fuzzy . . . . .	14
4.2.1	Definire i passi per costruire un sistema fuzzy . . . . .	14
4.2.2	Cos'è un insieme fuzzy? Cos'è una membership function? Con quali altri nomi viene anche indicata? . . . . .	14
4.2.3	Esiste una corrispondenza biunivoca tra insiemi fuzzy e valori numerici? . . . . .	14
4.2.4	Distinzione tra fuzzyness e probabilità . . . . .	14
4.2.5	La frase seguente: con la mia preparazione potrei prendere 24 all'esame, sottintende un processo fuzzy o probabilistico? . . . . .	14
4.2.6	Cosa si intende per FAM? Una FAM memorizza numeri o preposizione logiche? Come? . . . . .	14
4.2.7	Cosa è l'entropia fuzzy? . . . . .	14
4.2.8	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 . . . . .	14
4.3	Domande su Statistica . . . . .	18
4.3.1	Enunciare il teorema di Bayes . . . . .	18
4.3.2	Enunciare la formula delle probabilità totali . . . . .	18
4.3.3	Discutere l'analisi di varianza per un sistema lineare . . . . .	18
4.3.4	Che cosa è la stima della massima verosimiglianza? . . . . .	19

4.3.5	Dimostrare che la stima ai minimi quadrati è equivalente alla stima a massima verosimiglianza nel caso di errore Gaussiano sui dati. Cosa fornisce? Come? . . . . .	19
4.3.6	Cosa si intende per problema di regolarizzazione? Che tipo di funzione costo utilizza? Quali sono i suoi componenti? . . . . .	20
4.3.7	Mostrare la stima a massima verosimiglianza è equivalente a un problema di regolarizzazione. . . . .	20
4.3.8	Esercizio sui Taxi . . . . .	22
4.3.9	Soluzione esercizio sui Taxi . . . . .	22
4.3.10	Esercizio sul tumore al seno . . . . .	24
4.3.11	Soluzione esercizio sul tumore al seno . . . . .	24
4.3.12	Esercizio delle macchine . . . . .	26
4.3.13	Soluzione esercizio delle macchine . . . . .	26
4.4	Domande su Apprendimento con Rinforzo . . . . .	27
4.4.1	Cosa si intende per Apprendimento con Rinforzo? . . . . .	27
4.4.2	Quali sono gli attori? . . . . .	27
4.4.3	Cosa rappresenta la critica? . . . . .	27
4.4.4	Che tipo di architettura si può ipotizzare nell'apprendimento con rinforzo? . . . . .	27
4.4.5	Condizionamento classico e condizionamento operante . . . . .	27
4.4.6	Come potreste illustrare: Exploration vs Exploitation? . . . . .	28
4.4.7	Cos'è il problema del credit assignment? È un problema che riguarda la dimensione temporale o spaziale del task? . . . . .	28
4.4.8	Cos'è l'eligibility trace (traccia) e quale è il suo ruolo? . . . . .	28
4.4.9	Definire l'algoritmo di Q-learning, descrivendo le equazioni opportune. . . . .	28
4.4.10	Scrivere le equazioni dell'algoritmo Q-learning in cui si consideri anche la traccia. . . . .	29
4.4.11	Cosa si intende per politica epsilon-greedy? Come entra nell'algoritmo di Q-learning? . . . . .	29
4.4.12	Che differenza c'è tra Q-learning e SARSA? . . . . .	29
4.4.13	Quale criterio si sceglie per definire i Reward? A quali elementi sono associati? Allo stato? All'azione? Allo stato prossimo? Perché? . . . . .	29
4.4.14	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. . . . .	29
4.5	Domande su Apprendimento Supervisionato . . . . .	31
4.5.1	Cosa si intende per modello? . . . . .	31
4.5.2	Definire l'algoritmo di apprendimento di una rete neurale con unità arbitrarie. Definire la funzione obiettivo utilizzata. . . . .	31
4.5.3	Come si utilizza la funzione obiettivo nell'algoritmo di apprendimento. . . . .	31
4.5.4	Cosa si intende per apprendimento per epoche e per trial? Qual è il vantaggio di ciascuna delle modalità di apprendimento? . . . . .	31
4.5.5	Cosa si intende per training e test set? Perché mai vengono utilizzati? Quali problemi si vogliono evitare? . . . . .	31
4.5.6	Una rete neurale con unità a sigmoide è un modello parametrico? È lineare? Perché? . . . . .	32
4.5.7	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? . . . . .	32
4.5.8	Cosa si intende per un problema di regressione ed illustrare una possibile soluzione. . . . .	32
4.5.9	Come funziona l'approssimazione incrementale multi-scala, cosa garantisce e quali vantaggi può avere? . . . . .	32
4.6	Domande su Clustering . . . . .	33
4.6.1	Cosa si intende per clustering? In quali famiglie vengono divisi? Esplicitare la differenza tra soft e hard clustering . . . . .	33
4.6.2	Che relazione c'è tra clustering e classificazione e quali sono le criticità? . . . . .	33
4.7	Domande su Intelligenza Artificiale . . . . .	34
4.7.1	Si descriva il funzionamento della Forward Search. Perché è considerato un template e non un algoritmo? . . . . .	34
4.7.2	Si elenchino due possibili implementazioni di Forward Search elencandone proprietà, vantaggi e svantaggi. . . . .	34
4.8	Domande sul Planning . . . . .	35
4.8.1	Come è definito il problema dell'esplorazione autonoma di ambienti con robot mobili? . . . . .	35
4.8.2	Cosa è un sistema multi-agente? Se ne dia un esempio in applicazioni robotiche . . . . .	35
4.9	Domande su Biologia . . . . .	36
4.9.1	Definire il neurone biologico evidenziandone le parti più significative per la trasmissione dell'informazione ed il loro comportamento. . . . .	36
4.9.2	Descrivere il funzionamento complessivo del neurone biologico. . . . .	36
4.9.3	Dove avviene principalmente l'"apprendimento" nei neuroni biologici? . . . . .	36
4.9.4	Che differenza c'è tra neuroni sensoriali, neuroni motori ed inter-neuroni? . . . . .	36
4.9.5	Come viene trasmessa ed elaborata l'informazione da un neurone? . . . . .	36
4.9.6	Cos'è uno spike? . . . . .	36
4.9.7	Quali sono le aree corticali principali? . . . . .	37
4.9.8	Cos'è il codice di popolazione? . . . . .	37
4.9.9	Data un'area cerebrale, è univoca la funzione implementata in quell'area? . . . . .	37
4.9.10	Cosa sono i neuroni a specchio? Quali implicazioni hanno per i sistemi intelligenti e l'apprendimento? . . . . .	37

---

4.9.11 Cosa si intende per algoritmi genetici ed ottimizzazione evolutiva? . . . . .	37
4.9.12 Quali sono le differenze ed i punti forti degli algoritmi genetici ed ottimizzazione evolutiva? . . . . .	37
4.9.13 Cosa si intende per elitismo? . . . . .	38
4.9.14 Che cosa è una matrice di covarianza in generale e come viene applicata nell'ottimizzazione evolutiva? . . . . .	38

# Note sulla logica fuzzy

## 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  $\Delta x$  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 fuzziness** 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. Defuzzificazione

## 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:

**Teorema 2.1.1 (Teorema di Bayes).** Dati due eventi,  $X$  e  $Y$ , con  $P(Y) \neq 0$ , allora vale l'equazione:

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

Dove:

- $P(X|Y)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $X$  avvenga dato che è avvenuto  $Y$ .
- $P(Y|X)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $Y$  avvenga dato che è avvenuto  $X$ .
- $P(X)$  è la probabilità a priori di  $X$  ed è detta **probabilità marginale**.
- $P(Y)$  è la probabilità a priori di  $Y$  e funge da costante di normalizzazione.

#### 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:

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



## 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.

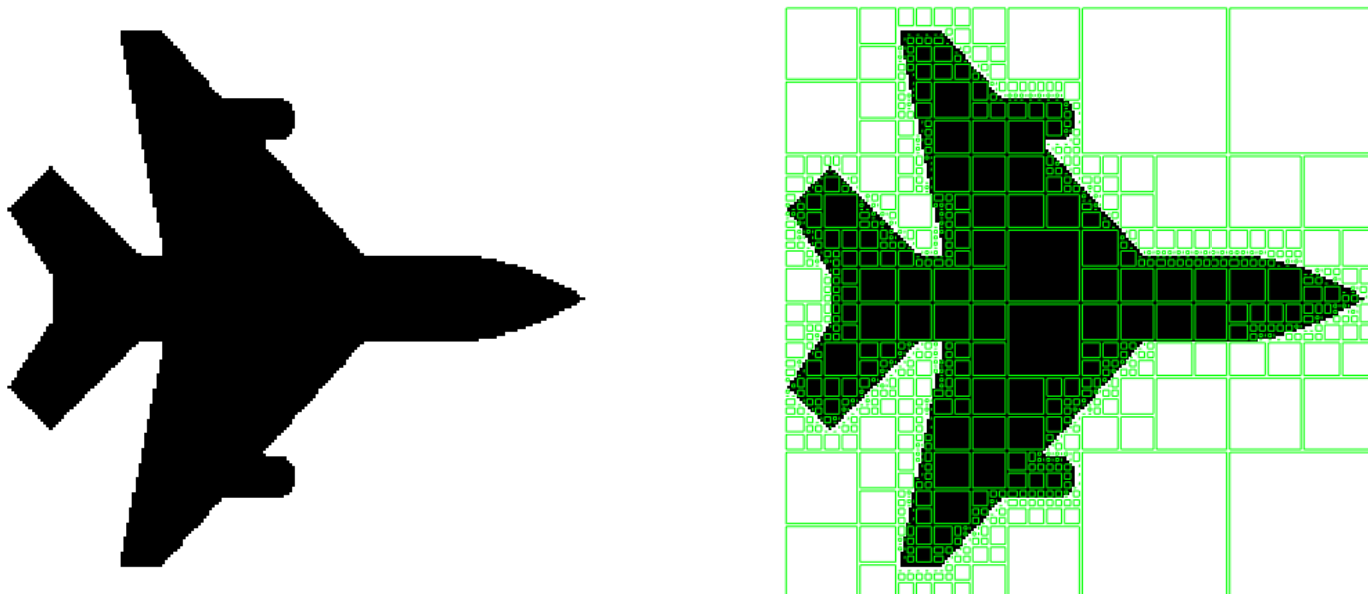
I vantaggi del clustering gerarchico sono l'indipendenza dall'inizializzazione e il fatto che non sia necessario specificare a priori il numero di cluster. Però, sono poco robusti (molto sensibili a rumore e agli outlier), non riconsiderano le scelte già fatte (l'errata classificazione di un punto non 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 degenero 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

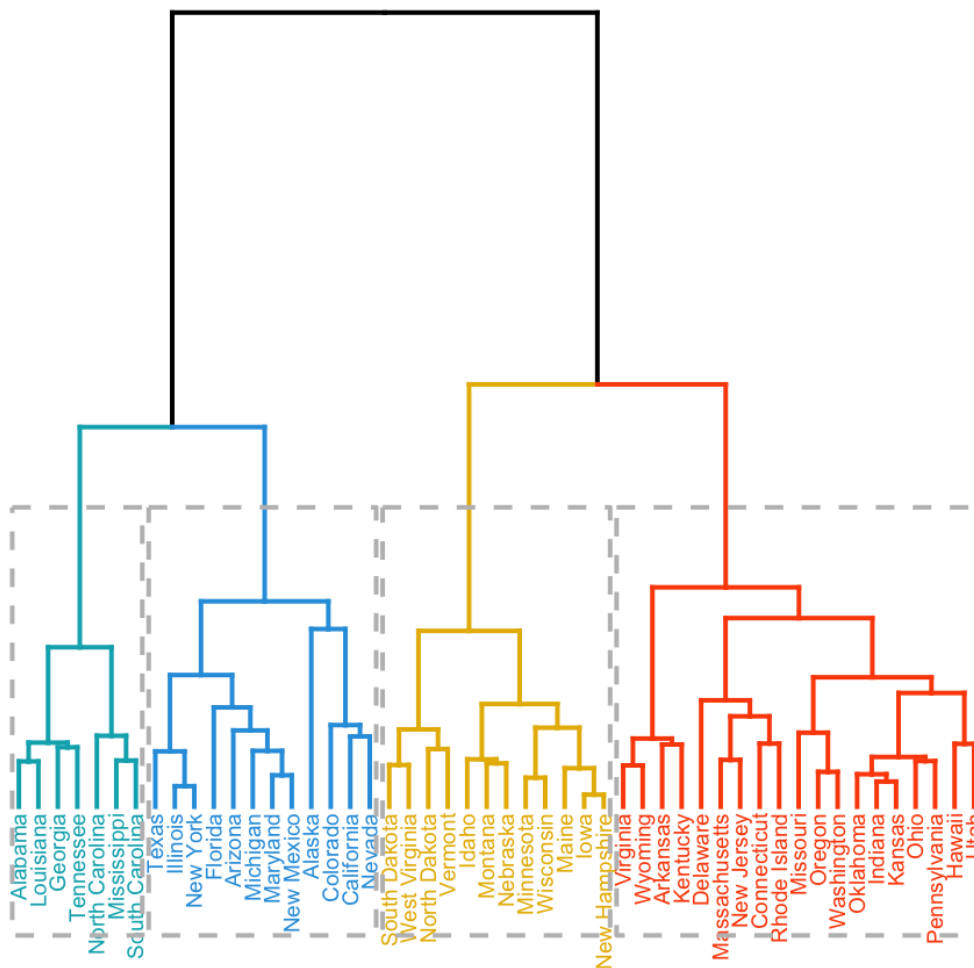
- *Metodo di Ward*: aumento di varianza *whitin-groups*

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:

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

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

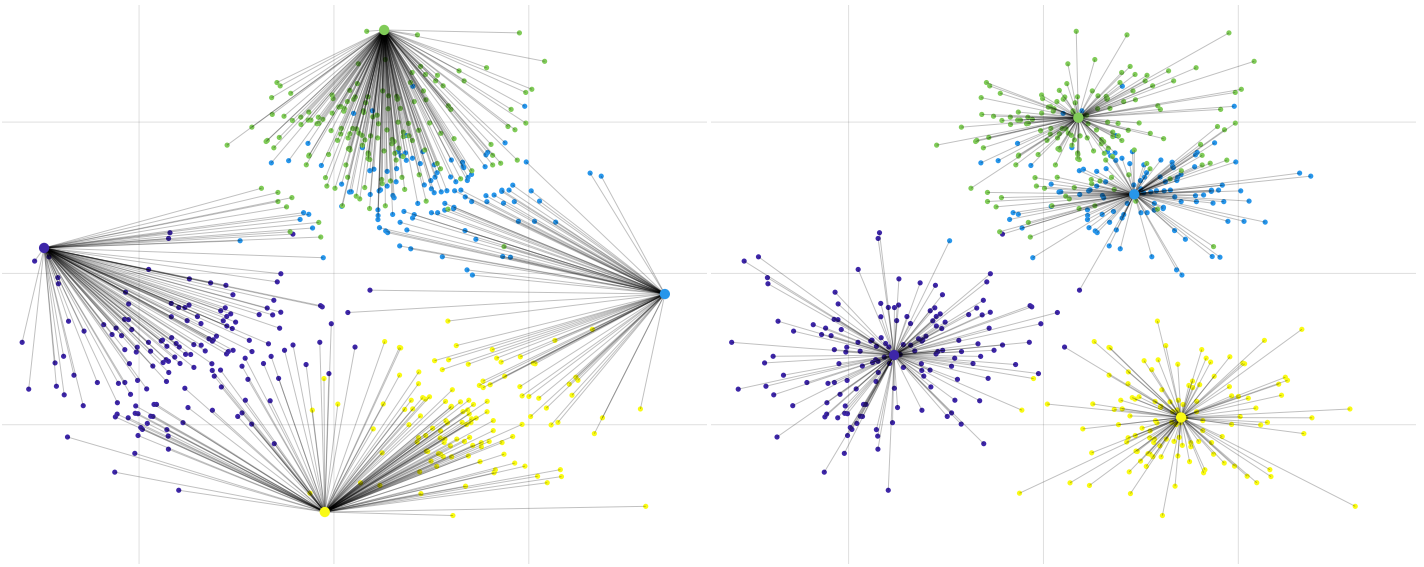
Figura 3.2: Esempio di dendrogramma risultante dal clustering agglomerativo degli stati membri degli USA per numero di arresti



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.

## Domande da temi d'esame

### 4.1 Domande su Macchine ed Intelligenza

#### 4.1.1 Descrivere il test di Turing, l'esperimento della stanza cinese e l'esperimento della stanza di Maxwell

##### Test di Turing

Il test di Turing, proposto da Alan Turing nel 1950, è un metodo per valutare l'intelligenza di un'intelligenza artificiale. Turing, preso spunto dal gioco dell'imitazione in cui una persona doveva comprendere se un interlocutore nascosto fosse uomo o donna in base ai messaggi che questo inviava, propone un gioco analogo in cui la conversazione avviene con una macchina od una persona, ed il nuovo obiettivo è determinare se si sta conversando con una macchina o una persona.

Il test è considerato passato quando la macchina è riconosciuta come umana.

##### Stanza Cinese

L'esperimento della stanza cinese è stato proposto da Jhon Searle nel 1980 in contrapposizione all'ipotesi che una macchina possa essere davvero 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

Esperimento mentale da Paul e Patricia Churchland, è una critica alla stanza cinese. Propone assiomi plausibili (elettricità e magnetismo sono forze e le forze non hanno a che fare con la luminosità), ma errati, sulla natura di elettricità, magnetismo e luce ed immagina Maxwell, intento a realizzare luce usando elettricità e magnetismo, costretto a spiegare che questi assiomi non sono validi e che non hanno nessuna giustificazione sulla natura della 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 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".

#### 4.1.2 Discutere la relazione tra algoritmo, macchina di Turing ed intelligenza.

Un algoritmo è una sequenza di passi elementari computabili, usati per risolvere un problema.

La macchina di Turing, proposta nel 1936 dall'omonimo matematico, è un formalismo in grado di eseguire algoritmi computabili ed arrivare in un tempo finito alla soluzione basandosi su regole definite su un alfabeto di simboli.

Una macchina di Turing è intelligente secondo l'ipotesi debole (cioè appare intelligente) ma non secondo l'ipotesi forte.

#### 4.1.3 Cosa si intende per ipotesi forte ed ipotesi debole dell'AI?

Sono due linee di pensiero nella filosofia dell'intelligenza artificiale:

**Ipotesi forte** È possibile realizzare un'intelligenza artificiale cosciente, senza mostrare necessariamente processi di pensiero umani. Questa ipotesi è spesso accompagnata da proposte di imitazione della struttura fisica del cervello (neuroni, sinapsi, etc.).

**Ipotesi debole** È possibile realizzare un'intelligenza artificiale che appare intelligente (una macchina di Turing che risolve un algoritmo molto complesso) ma non è effettivamente cosciente.

#### 4.1.4 Riportare il contraddittorio sulle ipotesi su cui è basata l'ipotesi debole sull'AI

1. Una macchina non può originare nulla di nuovo, esegue dei programmi.
2. Il comportamento intelligente non può essere completamente replicato, per esempio l'aspetto emotivo.
3. Il comportamento intelligente non può essere completamente catturato da regole formali (argument for informality), per esempio il subconscio.
4. Anche se una macchina di Turing riuscisse a superare test di Turing, mancherebbe comunque di una coscienza.

#### 4.1.5 Descrivere il "Brain prosthesis thought experiment" di Moravec e commentarlo.

Proposto da Hans Moravec nel 1988, chiede cosa succederebbe se sostituissimo uno a uno tutti i neuroni con un dispositivo elettronico equivalente. Esistono due risposte:

**Risposta funzionalistica** La mente è de-facto una scatola nera e modificare i costituenti fisici non comporta modifiche.

**Risposta strutturalista** Ad un certo punto la coscienza svanisce.

## 4.2 Domande sui Sistemi Fuzzy

### 4.2.1 Definire i passi per costruire un sistema fuzzy

1. Identifico le variabili di input e output del sistema specificando per ognuna il proprio **range**.
2. Identifico le **classi fuzzy** in cui le variabili sono da suddividere e stabilisco le **funzioni di membership**.
3. Definisco le regole logiche della FAM: per ogni combinazione di classi in input deve essere possibile definire una classe di output.
4. Definisco la **modalità di defuzzyficazione**, che riconverte le regole attivate in un valore continuo tramite media pesata, massimo o media pesata su aree.

### 4.2.2 Cos'è un insieme fuzzy? Cos'è una membership function? Con quali altri nomi viene anche indicata?

Un insieme fuzzy è un insieme caratterizzato da una **funzione di membership**  $M$  che, dato un elemento, restituisce il valore di verosimiglianza che esso appartenga al set, compreso tra 0 e 1.

### 4.2.3 Esiste una corrispondenza biunivoca tra insiemi fuzzy e valori numerici?

Gli insiemi fuzzy **non** godono di relazioni di univocità e biunivocità tra gli elementi di insiemi diversi. Pertanto gli insiemi fuzzy sono un'estensione ma non una generalizzazione degli insiemi della teoria classica.

### 4.2.4 Distinzione tra fuzzyness e probabilità

La **probabilità** riguarda un evento non ancora avvenuto e descrive l'incertezza che l'evento avvenga (Pioverà?). Una volta accaduto l'evento diviene certo, per cui non si parla più di probabilità, ma al più della vaghezza che contraddistingue l'evento (quanto l'evento è "pioggia" e quanto è "sereno").

La **fuzzyness** delinea un'incertezza, che è **deterministica**.

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

Si tratta di un evento non ancora avvenuto e la frase sottolinea un'incertezza: si tratta quindi di un processo probabilistico.

### 4.2.6 Cosa si intende per FAM? Una FAM memorizza numeri o preposizione logiche? Come?

Una **Fuzzy Associative Memory** è il "motore logico" di un sistema fuzzy: essa associa allo spazio delle classi discrete in input le classi discrete in output seguendo regole di logica booleana applicate alle classi fuzzy in ingresso. Essa memorizza queste regole e la conoscenza del sistema e le applica in cascata con struttura if-else.

### 4.2.7 Cosa è l'entropia fuzzy?

Si tratta di una misura di quanto un determinato evento è fuzzy, date due classi. Si calcola come il rapporto tra la distanza dalla classe più vicina e la distanza dalla classe più lontana.

### 4.2.8 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

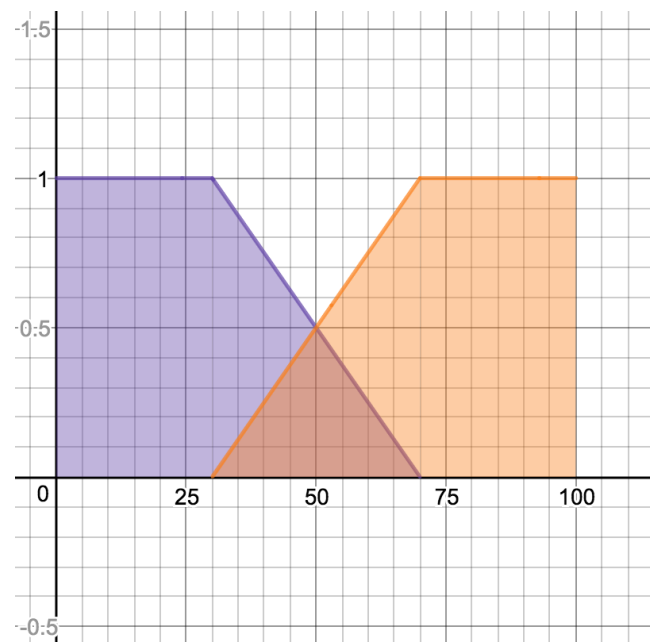
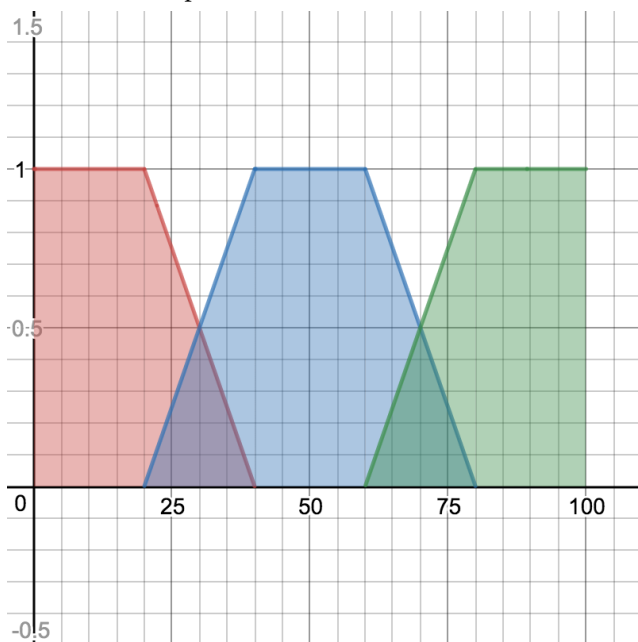
2. Carico di lavoro, che dipende dal genere di supporto che i clienti richiedono.

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.  $\text{ALTO} \wedge \text{LEGGERO} = \text{PERSONALE MEDIO}$
2.  $\text{ALTO} \wedge \text{PESANTE} = \text{PERSONALE COMPLETO}$
3.  $\text{MEDIO} \wedge \text{LEGGERO} = \text{POCO PERSONALE}$
4.  $\text{MEDIO} \wedge \text{PESANTE} = \text{PERSONALE MEDIO}$
5.  $\text{BASSO} \wedge \text{LEGGERO} = \text{POCO PERSONALE}$
6.  $\text{BASSO} \wedge \text{PESANTE} = \text{PERSONALE MEDIO}$

### **Costruiamo le regole della FAM per il tempo**

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

### **Definiamo una regola di defuzzyficazione**

Utilizzo la media pesata come regola di defuzzyficazione.

### **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.  $\text{ALTO}=0 \wedge \text{LEGGERO}=0.75 = \text{PERSONALE MEDIO}=0$
2.  $\text{ALTO}=0 \wedge \text{PESANTE}=0.25 = \text{PERSONALE COMPLETO}=0$
3.  $\text{MEDIO}=0.5 \wedge \text{LEGGERO}=0.75 = \text{POCO PERSONALE}=0.5$
4.  $\text{MEDIO}=0.5 \wedge \text{PESANTE}=0.25 = \text{PERSONALE MEDIO}=0.25$
5.  $\text{BASSO}=0.5 \wedge \text{LEGGERO}=0.75 = \text{POCO PERSONALE}=0.5$
6.  $\text{BASSO}=0.5 \wedge \text{PESANTE}=0.25 = \text{PERSONALE MEDIO}=0.25$

FAM per il tempo:

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

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.

## 4.3 Domande su Statistica

### 4.3.1 Enunciare il teorema di Bayes

**Teorema 4.3.1 (Teorema di Bayes).** Dati due eventi,  $X$  e  $Y$ , con  $P(Y) \neq 0$ , allora vale l'equazione:

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

Dove:

- $P(X|Y)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $X$  avvenga dato che è avvenuto  $Y$ .
- $P(Y|X)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $Y$  avvenga dato che è avvenuto  $X$ .
- $P(X)$  è la probabilità a priori di  $X$  ed è detta **probabilità marginale**.
- $P(Y)$  è la probabilità a priori di  $Y$  e funge da costante di normalizzazione.

### 4.3.2 Enunciare la formula delle probabilità totali

**Definizione 4.3.2 (Formula delle probabilità totali).** Sia  $(\Omega, \mathcal{F}, P)$  uno spazio di probabilità e  $F_1, F_2, \dots, F_n \in \mathcal{F}$  una partizione finita di  $\omega$ ,  $\bigcup_{k=1}^n F_k = \Omega$  e  $F_h \cap F_k = \emptyset$  se  $h \neq k$ , tale che  $P(F_k) > 0$  per  $k = 1, 2, \dots, n$ . Allora ogni evento  $E \in \mathcal{F}$  si ha:

$$P(E) = \sum_{k=1}^n P(E|F_k)P(F_k)$$

### 4.3.3 Discutere l'analisi di varianza per un sistema lineare

Nei casi reali, tutte le misurazioni hanno errori, indicati con  $v$ , i quali sono supposti provenienti da una distribuzione gaussiana a media nulla  $N(0, \sigma^2)$ . Essendo la media nulla, l'unico parametro rimasto per valutare la bontà di una stima è la varianza  $\sigma^2$ .

Nei sistemi lineari  $\underline{A}\underline{x} = \underline{b} + \underline{v}$  la varianza stimata è pari alla somma degli errori di misura al quadrato:

$$\sigma_0^2 = \sum v_i^2$$

Risolvendo il sistema per il vettore  $\underline{x}$  ed aggiungendo un certo errore nella stima  $\underline{u} \approx N(0, \sigma^2)$ , andiamo a calcolare l'errore sui parametri che siamo andati a stimare:

$$\underline{x} + \underline{u} = (\underline{A}^T \underline{A})^{-1} \underline{A}^T (\underline{b} + \underline{v})$$

Chiamo  $\underline{C} = (\underline{A}^T \underline{A})^{-1}$ .

Considerando  $\underline{x}$  i valori reali e  $\underline{u}$  gli errori ricavo:

$$\underline{x} = \underline{C} \underline{A}^T \underline{b} \quad \underline{u} = \underline{C} \underline{A}^T \underline{v}$$

Costruiamo la matrice di covarianza

$$\underline{u} \underline{u}^T = (\underline{C} \underline{A}^T \underline{v}) (\underline{v}^T \underline{A} \underline{C}^T) = \sigma_0^2 (\underline{C} \underline{A}^T \underline{A} \underline{C}^T) = \sigma_0^2 \underline{C}^T$$

La matrice di covarianza descrive la varianza dell'errore sui vari parametri:

$$\sigma^2(u_{ij}) = c_{ij} \sigma_0^2$$

Questa ci fornisce un'idea sulla mutua influenza dei vari parametri tra loro. Se due parametri hanno un'alta covarianza significa che è difficile distinguerli:

$$-1 \leq \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}} \leq 1$$

Tanto più si avvicina agli estremi i parametri covariano, e in quel caso significa che c'è qualche parametro di troppo nella stima. Empiricamente si scartano parametri quando l'indice di correlazione è superiore del 95%.

#### 4.3.4 Che cosa è la stima della massima verosimiglianza?

Si tratta di un metodo che realizza una stima dei parametri di una distribuzione partendo da un vettore di misurazioni  $\underline{y}$  che corrisponde a delle realizzazioni di una variabile che si suppone appartenga alla distribuzione data. Per esempio nel caso di una distribuzione gaussiana si procede come segue:

$$L(\underline{y} | \mu, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2\right\}$$

A questo punto si procede calcolando il logaritmo negativo di  $L$ ,  $-\ln L$  e si calcolano le derivate parziali in funzione dei parametri. Queste vengono quindi poste pari a 0 per massimizzarle e si risolve in funzione del parametro di interesse.

#### 4.3.5 Dimostrare che la stima ai minimi quadrati è equivalente alla stima a massima verosimiglianza nel caso di errore Gaussiano sui dati. Cosa fornisce? Come?

Considerando il caso in cui vogliamo stimare una retta  $y = mx + q$ , stimando il coefficiente angolare  $m$  ed il parametro  $q$ , con  $n$  misurazioni  $\underline{x}$  e  $\underline{y}$  e considerando l'errore di misura gaussiano, procediamo con la **stima a massima verosimiglianza**.

Costruisco la funzione di verosimiglianza:

$$L(\underline{y}, \underline{x} | m, q) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{y_i - mx_i - q}{\sigma}\right)^2\right\}$$

Calcolo le derivate parziali del logaritmo negativo della verosimiglianza:

Per il coefficiente angolare  $m$ :

$$\frac{\partial(-\ln L)}{\partial m} = 0 \Rightarrow \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - mx_i - q)(-2x_i) = 0 \Rightarrow m \sum_{i=1}^n x_i^2 + q \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i$$

Per il parametro  $q$ :

$$\frac{\partial(-\ln L)}{\partial q} = 0 \Rightarrow \sum_{i=1}^n (y_i - mx_i - q)x_i = 0 \Rightarrow m \sum_{i=1}^n x_i + q \cdot n = \sum_{i=1}^n y_i$$

In forma matriciale le equazioni ottenute sono le seguenti:

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

Procediamo ora con la **stima ai minimi quadrati**. Lo stesso problema nei minimi quadrati è impostato come segue:

$$m\underline{x} + b\underline{1} = \underline{y}$$

Si tratta quindi di risolvere l'equazione:

$$\begin{bmatrix} \underline{x}^T \\ \underline{1}^T \end{bmatrix} \begin{bmatrix} \underline{x} & \underline{1} \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \underline{x}^T \\ \underline{1}^T \end{bmatrix} \underline{y} \Rightarrow \begin{bmatrix} \underline{x}\underline{x}^T & \underline{x}^T \underline{1} \\ \underline{1}^T \underline{x} & \underline{1}^T \underline{1} \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \underline{x}^T \underline{y} \\ \underline{1}^T \underline{y} \end{bmatrix} \Rightarrow \begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{bmatrix}$$

I risultati ottenuti dai due metodi coincidono.

Entrambi i metodi offrono una stima dei parametri, ma coincidono solo quando il rumore segue una distribuzione normale a media nulla.

La stima alla massima verosimiglianza massimizza la probabilità condizionata che i singoli punti assumano tali valori per i parametri, quella ai minimi quadrati minimizza i quadrati dei residui:  $\min_x (\underline{A}\underline{x} - \underline{b})^2$ .

### 4.3.6 Cosa si intende per problema di regolarizzazione? Che tipo di funzione costo utilizza? Quali sono i suoi componenti?

La **regolarizzazione** è un processo di introduzione di informazioni aggiuntive per risolvere problemi di ottimizzazione mal posti o prevenire over-fitting sui dati o ancora per estendere la generalità del modello.

#### A priori di Gibbs

Un tipo di distribuzione molto usato è la probabilità a priori di Gibbs:

$$p(x) = \frac{1}{Z} \exp \left\{ -\frac{1}{\beta} U(x) \right\}, \quad Z = \int_{-\infty}^{+\infty} \exp \left\{ -\frac{1}{\beta} U(x) \right\} dx$$

$$J_R(x) = \ln \left[ \frac{1}{Z} \exp \left\{ -\frac{1}{\beta} U(x) \right\} \right] = \ln \left[ \frac{1}{Z} \right] - \frac{1}{\beta} U(x)$$

I cui **componenti** sono:

$U(x)$ : potenziale.

$J_R(x) = \ln [p(x)]$ : funzione lineare (con coefficiente negativo) del potenziale.

$\beta$ : parametro di regolarizzazione.

#### Ridge Regression o regolarizzazione di Tikhonov

Nel caso  $U(x) = x^2$ , la stima per massima probabilità a posteriori viene detta *ridge regression* (o regolarizzazione Tikhonov).

$$J_R(x) = \ln \left( \frac{1}{Z} \right) - \frac{1}{\beta} x^2$$

#### Regolarizzazione di Tikhonov del gradiente locale

Quando in un'immagine vengono identificate discontinuità forti potrebbe essere presente un rumore di tipo additivo.

In questi casi è possibile usare una regolarizzazione di Tikhonov che ha come potenziale la norma del gradiente dell'immagine:

$$U(x) = \|\nabla x\|^2, \Rightarrow J_R(x) = \ln \left( \frac{1}{Z} \right) - \frac{1}{\beta} \|\nabla x\|^2$$

### 4.3.7 Mostrare la stima a massima verosimiglianza è equivalente a un problema di regolarizzazione.

Consideriamo il caso di un'immagine  $\underline{y}$  (vettore di  $n$  pixel), corrotta da un errore gaussiano a media nulla  $\underline{\epsilon} \sim N(0, \sigma^2)$ .

Cerchiamo di ripristinare l'immagine originaria  $\underline{x}$ .

$$\underline{y} = \underline{x} + \underline{\epsilon}$$

Cerchiamo di stimare l'immagine originaria  $\underline{x}$  per massima verosimiglianza.

$$\begin{aligned} \underline{x}^* &= \underset{\underline{x}}{\operatorname{argmax}} L(\underline{y} | \underline{x}, \mu = 0, \sigma^2) \\ &= \underset{\underline{x}}{\operatorname{argmax}} \left[ \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - x_i)^2}{2\sigma^2} \right\} \right] \\ &= \underset{\underline{x}}{\operatorname{argmin}} \left[ \sum_{i=1}^n (y_i - x_i)^2 \right] \\ &= \underline{y} \end{aligned}$$

Ne otteniamo che l'immagine più verosimile è l'immagine di partenza: non è un risultato utile.

Allora, utilizziamo come stima la massima probabilità a posteriori, o MAP (*Maximum A Posteriori*):

$$\begin{aligned}
 \underline{\mathbf{x}}^* &= \operatorname{argmax}_{\underline{\mathbf{x}}} p(\underline{\mathbf{x}} | \underline{\mathbf{y}}) \\
 &= \operatorname{argmax}_{\underline{\mathbf{x}}} \frac{p(\underline{\mathbf{y}} | \underline{\mathbf{x}}) p(\underline{\mathbf{x}})}{p(\underline{\mathbf{y}})} && \text{Applico il teorema di Bayes.} \\
 &= \operatorname{argmax}_{\underline{\mathbf{x}}} \frac{L(\underline{\mathbf{y}} | \underline{\mathbf{x}}) p(\underline{\mathbf{x}})}{p(\underline{\mathbf{y}})} && \text{Sostituisco la densità } p(\underline{\mathbf{y}} | \underline{\mathbf{x}}) \text{ con la verosimiglianza } L(\underline{\mathbf{y}} | \underline{\mathbf{x}}) \\
 &= \operatorname{argmin}_{\underline{\mathbf{x}}} -\ln \left[ \frac{L(\underline{\mathbf{y}} | \underline{\mathbf{x}}) p(\underline{\mathbf{x}})}{p(\underline{\mathbf{y}})} \right] && \text{Applico il logaritmo negativo} \\
 &= \operatorname{argmin}_{\underline{\mathbf{x}}} -\ln [L(\underline{\mathbf{y}} | \underline{\mathbf{x}}) p(\underline{\mathbf{x}})] && \text{Elimino il termine } p(\underline{\mathbf{y}}) \text{ che è indipendente da } \underline{\mathbf{x}}
 \end{aligned}$$

La massima probabilità a posteriori integra la massimizzazione della verosimiglianza con delle conoscenze a priori.

Nel caso gaussiano a media nulla, il calcolo diventa:

$$\begin{aligned}
 \underline{\mathbf{x}}^* &= \operatorname{argmin}_{\underline{\mathbf{x}}} -\left\{ \ln [L(\underline{\mathbf{y}} | \underline{\mathbf{x}})] + \ln [p(\underline{\mathbf{x}})] \right\} \\
 &= \operatorname{argmin}_{\underline{\mathbf{x}}} -\left\{ \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - x_i)^2 + \ln [p(\underline{\mathbf{x}})] \right\} \\
 &= \operatorname{argmin}_{\underline{\mathbf{x}}} -\left\{ \frac{\|\underline{\mathbf{y}} - \underline{\mathbf{x}}\|^2}{2\sigma^2} + J_R(\underline{\mathbf{x}}) \right\}
 \end{aligned}$$

### 4.3.8 Esercizio sui Taxi

In una città lavorano due compagnie di taxi: blue e verde. La maggior parte dei tassisti 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%.

1. Qual è la probabilità che il taxi fosse effettivamente blu?
2. Quale deve essere l'affidabilità del testimone perché la probabilità che il taxi fosse effettivamente blu sia del 99%?

### 4.3.9 Soluzione esercizio sui Taxi

Definiamo una variabile aleatoria  $X$  che descrive la probabilità che un taxi appartenga ad una determinata compagnia:

$$X: \begin{cases} \text{blu} : 0.15 \\ \text{verde} : 0.85 \end{cases}$$

Il valore dell'affidabilità del testimone può essere modellata tramite la seguente probabilità condizionata, rappresentante quanto sono sicuro che il testimone sia in grado di dire che il taxi è blu quando è effettivamente blu:

$$\text{Affidabilità del testimone} = P(\text{Testimone vede blu} \mid \text{Il taxi è blu}) = 0.7$$

A noi interessa ottenere la probabilità  $P(\text{Il taxi è blu} \mid \text{Testimone vede blu})$ , per cui procederemo con il **teorema di Bayes**:

**Teorema 4.3.3 (Teorema di Bayes).** Dati due eventi,  $X$  e  $Y$ , con  $P(Y) \neq 0$ , allora vale l'equazione:

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

Dove:

- $P(X \mid Y)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $X$  avvenga dato che è avvenuto  $Y$ .
- $P(Y \mid X)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $Y$  avvenga dato che è avvenuto  $X$ .
- $P(X)$  è la probabilità a priori di  $X$  ed è detta **probabilità marginale**.
- $P(Y)$  è la probabilità a priori di  $Y$  e funge da costante di normalizzazione.

Per determinare la probabilità  $P(\text{Testimone vede blu})$  utilizziamo la **formula delle probabilità totali**:

**Definizione 4.3.4 (Formula delle probabilità totali).** Sia  $(\Omega, \mathcal{F}, P)$  uno spazio di probabilità e  $F_1, F_2, \dots, F_n \in \mathcal{F}$  una partizione finita di  $\omega$ ,  $\bigcup_{k=1}^n F_k = \Omega$  e  $F_h \cap F_k = \emptyset$  se  $h \neq k$ , tale che  $P(F_k) > 0$  per  $k = 1, 2, \dots, n$ . Allora ogni evento  $E \in \mathcal{F}$  si ha:

$$P(E) = \sum_{k=1}^n P(E \mid F_k) P(F_k)$$

Per cui abbiamo che:

$$\begin{aligned} P(\text{Testimone vede blu}) &= P(\text{Testimone vede blu} \mid \text{Il taxi è blu}) P(\text{Il taxi è blu}) + P(\text{Testimone vede blu} \mid \text{Il taxi è verde}) P(\text{Il taxi è verde}) \\ &= 0.7 \cdot 0.15 + 0.3 \cdot 0.85 \\ &= 0.36 \end{aligned}$$

$$P(\text{Il taxi è blu} \mid \text{Testimone vede blu}) = \frac{P(\text{Testimone vede blu} \mid \text{Il taxi è blu}) P(\text{Il taxi è blu})}{P(\text{Testimone vede blu})} = \frac{0.7 \cdot 0.15}{0.36} = 0.291\bar{6}$$

L'affidabilità del testimone per garantire una probabilità del 99% deve essere tale che:

$$\frac{P(\text{Testimone vede blu} \mid \text{Il taxi è blu}) P(\text{Il taxi è blu})}{P(\text{Testimone vede blu})} = 0.99$$
$$P(\text{Testimone vede blu} \mid \text{Il taxi è blu}) = \frac{0.99 \cdot P(\text{Testimone vede blu})}{P(\text{Il taxi è blu})} = \frac{0.99 \cdot 0.36}{0.7} = 0.509$$



### 4.3.10 Esercizio sul tumore al seno

Lo strumento principe per lo screening per il tumore al seno è la radiografia (mammografia).

Sappiamo che la **sensitività** della mammografia è intorno al 90% e che la **specificità** sia anch'essa del 90%.

1. Qual è la probabilità che l'esame dia risultato positivo, sapendo che le donne malate sono lo 0,01%?
2. Qual è la percentuale di donne che hanno uno screening positivo, di essere effettivamente malate?

### 4.3.11 Soluzione esercizio sul tumore al seno

Sensitività e specificità sono associate rispettivamente ad errore del primo e secondo tipo:

**Errore di primo tipo** È quando si rifiuta come falsa l'ipotesi vera.

**Errore di secondo tipo** È quando si accetta come vera l'ipotesi falsa.

**Sensitività** Probabilità che lo strumento non compia un errore di primo tipo.

**Specificità** Probabilità che lo strumento non compia un errore di secondo tipo.

La **sensitività** di uno strumento è la probabilità che esso dia esito positivo in un caso positivo:

$$P(\text{Esito positivo} \mid \text{Donna malata}) = 0.9$$

La **specificità** di uno strumento è la probabilità che esso dia esito negativo in un caso negativo:

$$P(\text{Esito negativo} \mid \text{Donna sana}) = 0.9$$

Viene chiesta la probabilità  $P(\text{Esito positivo})$  sapendo che  $P(\text{Donna malata}) = 0.01$ , che possiamo ottenere tramite la formula delle probabilità totali:

**Definizione 4.3.5 (Formula delle probabilità totali).** Sia  $(\Omega, \mathcal{F}, P)$  uno spazio di probabilità e  $F_1, F_2, \dots, F_n \in \mathcal{F}$  una partizione finita di  $\omega$ ,  $\bigcup_{k=1}^n F_k = \Omega$  e  $F_h \cap F_k = \emptyset$  se  $h \neq k$ , tale che  $P(F_k) > 0$  per  $k = 1, 2, \dots, n$ . Allora ogni evento  $E \in \mathcal{F}$  si ha:

$$P(E) = \sum_{k=1}^n P(E \mid F_k) P(F_k)$$

$$\begin{aligned} P(\text{Esito positivo}) &= P(\text{Esito positivo} \mid \text{Donna malata}) P(\text{Donna malata}) + P(\text{Esito positivo} \mid \text{Donna sana}) P(\text{Donna sana}) \\ &= 0.9 \cdot 0.01 + (1 - 0.9) \cdot (1 - 0.01) \\ &= 0.9 \cdot 0.01 + 0.1 \cdot 0.99 \\ &= 0.108 \end{aligned}$$

Viene chiesta la probabilità  $P(\text{Donna malata} \mid \text{Esito positivo})$ , ottenibile tramite il teorema di Bayes:

**Teorema 4.3.6 (Teorema di Bayes).** Dati due eventi,  $X$  e  $Y$ , con  $P(Y) \neq 0$ , allora vale l'equazione:

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

Dove:

- $P(X \mid Y)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $X$  avvenga dato che è avvenuto  $Y$ .
- $P(Y \mid X)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $Y$  avvenga dato che è avvenuto  $X$ .
- $P(X)$  è la probabilità a priori di  $X$  ed è detta **probabilità marginale**.
- $P(Y)$  è la probabilità a priori di  $Y$  e funge da costante di normalizzazione.

$$\begin{aligned}P(\text{Donna malata} \mid \text{Esito positivo}) &= \frac{P(\text{Esito positivo} \mid \text{Donna malata}) P(\text{Donna malata})}{P(\text{Esito positivo})} \\&= \frac{0.9 \cdot 0.01}{0.108} \\&= 0.083\end{aligned}$$

Una donna è malata quando l'esito è positivo nell'8.3% dei casi. Da questi valori di evince che una sensitività del 90% risulta essere un valore troppo basso quando la probabilità dell'evento positivo (essere malata) è basso.

### 4.3.12 Esercizio delle macchine

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%.

1. Determinare la probabilità di estrarre un pezzo difettoso.
2. Viene estratto a caso un pezzo che risulta difettoso. Determinare la probabilità che quel pezzo sia stato prodotto da  $C$ .

### 4.3.13 Soluzione esercizio delle macchine

I valori sono modellati come segue:

$$\begin{aligned}P(A) &= 0.5 \\P(B) &= 0.4 \\P(C) &= 0.1 \\P(\text{Difettoso} | A) &= 0.02 \\P(\text{Difettoso} | B) &= 0.01 \\P(\text{Difettoso} | C) &= 0.04\end{aligned}$$

Viene chiesto di ottenere la probabilità  $P(\text{Difettoso})$ . Procediamo con la **formula delle probabilità totali**:

**Definizione 4.3.7 (Formula delle probabilità totali).** Sia  $(\Omega, \mathcal{F}, P)$  uno spazio di probabilità e  $F_1, F_2, \dots, F_n \in \mathcal{F}$  una partizione finita di  $\omega$ ,  $\bigcup_{k=1}^n F_k = \Omega$  e  $F_h \cap F_k = \emptyset$  se  $h \neq k$ , tale che  $P(F_k) > 0$  per  $k = 1, 2, \dots, n$ . Allora ogni evento  $E \in \mathcal{F}$  si ha:

$$P(E) = \sum_{k=1}^n P(E | F_k) P(F_k)$$

$$\begin{aligned}P(\text{Difettoso}) &= P(\text{Difettoso} | A) P(A) + P(\text{Difettoso} | B) P(B) + P(\text{Difettoso} | C) P(C) \\&= 0.2 \cdot 0.5 + 0.1 \cdot 0.4 + 0.4 \cdot 0.1 \\&= 0.018\end{aligned}$$

Viene chiesto di ottenere la probabilità  $P(C | \text{Difettoso})$ , che otteniamo tramite il **teorema di Bayes**:

**Teorema 4.3.8 (Teorema di Bayes).** Dati due eventi,  $X$  e  $Y$ , con  $P(Y) \neq 0$ , allora vale l'equazione:

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

Dove:

- $P(X | Y)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $X$  avvenga dato che è avvenuto  $Y$ .
- $P(Y | X)$  è la probabilità condizionata rappresentante la verosimiglianza che l'evento  $Y$  avvenga dato che è avvenuto  $X$ .
- $P(X)$  è la probabilità a priori di  $X$  ed è detta **probabilità marginale**.
- $P(Y)$  è la probabilità a priori di  $Y$  e funge da costante di normalizzazione.

$$\begin{aligned}P(C | \text{Difettoso}) &= \frac{P(\text{Difettoso} | C) P(C)}{P(\text{Difettoso})} \\&= \frac{0.04 \cdot 0.1}{0.018} = 0.22\bar{2}\end{aligned}$$

## 4.4 Domande su Apprendimento con Rinforzo

### 4.4.1 Cosa si intende per Apprendimento con Rinforzo?

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

### 4.4.2 Quali sono gli attori?

In un problema di *Reinforcement Learning* si ha alla base un agente che, interagendo con l'ambiente, va a costruire una policy per massimizzare una **reward a lungo termine** ottenuta eseguendo delle azioni.

**Policy** Descrive lo schema di comportamento dell'agente, mappando stati ad azioni.

**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.

**Reward 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.

### 4.4.3 Cosa rappresenta la critica?

La **critica** nell'apprendimento è rappresentata da tutti quei processi che valutano a posteriori le azioni prese dall'agente. Essa fornisce un rinforzo secondario, interno ed a lungo termine interpretato come *cost-to-go* da ogni stato fino al goal che consiste nella **value function** e permette di valutare se la *policy* stia dando buoni risultati.

### 4.4.4 Che tipo di architettura si può ipotizzare nell'apprendimento con rinforzo?

La tipica architettura di un agente è costituita da 4 elementi:

**Memoria interna** Tiene traccia dello stato dell'ambiente.

**Processore** Un componente che, dato uno stato, compie una serie di elaborazioni che hanno lo scopo di trovare la migliore azione in quello stato.

**Sensori** Servono per consentire all'agente di percepire le caratteristiche del mondo esterno.

**Attuatori** Servono all'agente per agire nell'ambiente e provocare quindi dei cambiamenti.

### 4.4.5 Condizionamento classico e condizionamento operante

**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.

**Condizionamento operante** Il rinforzo avviene "una-tantum", viene quindi valutata **una catena di azioni**, un comportamento nel suo insieme e non nella singola azione.

#### Quale relazione c'è con l'intelligenza?

L'apprendimento con rinforzo imita, con il processo di esecuzione delle azioni e valutazione successiva, il meccanismo di apprendimento dell'intelligenza umana che, tramite un processo di trial and error, migliora il proprio comportamento.

#### 4.4.6 Come potreste illustrare: Exploration vs Exploitation?

Sono due strategie che consistono nel:

**Exploration** Provare varie azioni per scoprire nuove possibili azioni ed esplora lo spazio delle azioni per scoprire quelle migliori.

**Exploitation** Scegliere sempre la soluzione che garantisca il miglior reward tra quelle conosciute (politica greedy).

La soluzione ottima viene identificata bilanciando opportunamente queste due strategie.

Il parametro utilizzato tipicamente per indicare la probabilità di scegliere una strategia piuttosto che l'altra è  $\epsilon$ .

#### 4.4.7 Cos'è il problema del credit assignment? È un problema che riguarda la dimensione temporale o spaziale del task?

I sistemi a singolo agente estesi nel tempo hanno il problema di valutare il contributo di un'azione rispetto alle altre (**temporal credit assignment**), che viene accentuato se le azioni rilevanti sono temporalmente distanti tra di loro con un intermezzo di azioni poco rilevanti: questo può portare a cattive valutazioni.

I sistemi multi-agente hanno inoltre il problema di determinare il contributo di ogni agente ad un compito comune (**structural credit assignment**).

#### 4.4.8 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 ed è utilizzata per gestire i reward ritardati nel tempo.

Gli eventi passati gradualmente perdono di importanza seguendo una legge esponenziale.

Quando si deve valutare il peso di uno stato su un aggiornamento di più step, la traccia dice se esso sia eleggibile.

Essa viene modellata come segue:

$$e_t(s, a) = \begin{cases} \gamma \lambda e_{t-1}(s, a) + 1 & \text{se } s = s_t \text{ e } a = a_t \\ \gamma \lambda e_{t-1}(s, a) & \text{altrimenti} \end{cases}$$

#### 4.4.9 Definire l'algoritmo di Q-learning, descrivendo le equazioni opportune.

L'algoritmo di Q-learning appartiene alla famiglia degli algoritmi di apprendimento da differenze temporali: esso basa l'aggiornamento della **value function** sulle informazioni relative allo step di esecuzione precedente, utile quando la conoscenza dell'ambiente è parziale.

Lo scopo dell'algoritmo è l'ottimizzazione della funzione valore  $Q$  per ogni azione  $a$  e stato  $s$ :

$$Q(s_t, a) = (1 - \alpha) Q(s_t, a) + \alpha \left( r + \gamma \max_{a'} Q(s_{t+1}, a') \right)$$

I parametri coinvolti sono:

$s$  Lo stato attuale.

$a$  L'azione scelta.

$s_{t+1}$  Lo stato raggiunto eseguendo l'azione  $a$  nello stato attuale  $s_t$ .

$\alpha$  Lo step-size, il peso dell'aggiornamento, compreso in  $[0, 1]$ , il parametro che indica quanto il vecchio ed il nuovo passaggio contribuiscono a  $Q$ .

$r$  Reward osservato dopo aver eseguito l'azione  $a$  nello stato attuale  $s_t$ .

$\gamma$  Il fattore di sconto, compreso in  $[0, 1]$ , che modella l'importanza di reward futuri.

$\max_{a'} Q$  è la migliore value function dello stato successivo.

#### 4.4.10 Scrivere le equazioni dell'algoritmo Q-learning in cui si consideri anche la traccia.

Considerare anche la traccia va modificare lo step-size, cioè il modo in cui passaggi vecchi e nuovi contribuiscono al nuovo valore, che diviene:

$$Q(s_t, a) = (1 - \alpha e(s_t, a))Q(s_t, a) + \alpha e(s_t, a) \left( r + \gamma \max_{a'} Q(s_{t+1}, a') \right)$$

#### 4.4.11 Cosa si intende per politica epsilon-greedy? Come entra nell'algoritmo di Q-learning?

Si tratta di un particolare tipo di policy che si basa su di un ulteriore parametro  $\epsilon \in [0, 1]$ . Ad ogni step, un agente che applica una politica  $\epsilon$ -greedy sceglie l'azione greedy (quella a massimo reward) con probabilità  $1 - \epsilon$ , altrimenti sceglie un'altra azione in modo casuale. Si tratta quindi del parametro che indica la probabilità di scegliere una strategia esplorativa piuttosto che una exploitative.

#### 4.4.12 Che differenza c'è tra Q-learning e SARSA?

La differenza è nel modo in cui viene aggiornato il valore di Q: SARSA è on-policy mentre Q-Learning è off-policy, ovvero:

**Off-policy** Q-Learning aggiorna il valore di Q in funzione della migliore scelta futura possibile.

**On-policy** SARSA aggiorna il valore di Q 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.

#### 4.4.13 Quale criterio si sceglie per definire i Reward? A quali elementi sono associati? Allo stato? All'azione? Allo stato prossimo? Perché?

I comportamenti vengono scelti tenendo a mente quale comportamento si desidera ottenere dall'agente che deve essere ricompensato se non fallisce nel compimento di un'azione e se tale azione lo porta un stato più vicino al goal, penalizzato altrimenti.

Il reward è associato a stato corrente, azione e stato prossimo: nella maggior parte dei problemi di apprendimento reali le transizioni sono non deterministiche e bisogna quindi considerare anche l'eventualità che l'azione scelta non produca l'effetto sperato.

#### 4.4.14 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.

		3	4
5	6		8
	10	11	12
	14	15	16

$$R_{s_t \rightarrow s_{t+1}} \begin{cases} 100 & \text{se è goal: } s_{t+1} = 16 \\ -5 & \text{se l'agente è bloccato: } s_t = s_{t+1} \\ 1 & \text{se non è goal} \end{cases} \quad Q(s, a) = 0 \quad \forall s, a$$

(c) Value function iniziale

(a) Griglia degli stati, in nero gli ostacoli, agente nello stato  $s = 1$

(b) Reward function

$$Q(s_t, a) = (1 - \alpha)Q(s_t, a) + \alpha \left( r + \gamma \max_{a'} Q(s_{t+1}, a') \right)$$

$$\epsilon = 0.1 \quad \alpha = 0.7 \quad \gamma = 0.4$$

(d) Algoritmo di aggiornamento: Q-learning

(e) Policy:  $\epsilon$ -greedy

Figura 4.2: Problema su griglia

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

**Step 1** Dato che tutti i valori di Q sono a 0 la policy  $\epsilon$ -greedy a questo step coincide con una policy casuale. Supponiamo l'azione scelta sia  $a = right$ : l'agente incontra un ostacolo e rimane quindi nello stato  $s_t = s_{t+1} = 1$ :

$$R_{s_t=s_{t+1}} = -5 \Rightarrow Q(s, right) = (1 - 0.7) \cdot 0 + 0.7 \cdot (-5 + 0.4 \cdot 0) = -3.5$$

**Step 2** Supponiamo che la policy  $\epsilon$ -greedy scelga di svolgere un'azione greedy (exploitative strategy) e quindi vada a compiere l'azione  $a = down$ :

$$R_{s_t \neq s_{t+1}} = 1 \Rightarrow Q(s, down) = (1 - 0.7) \cdot 0 + 0.7 \cdot (1 + 0.4 \cdot 0) = 0.7$$

## 4.5 Domande su Apprendimento Supervisionato

### 4.5.1 Cosa si intende per modello?

Un modello è una struttura costruita utilizzando il linguaggio e gli strumenti matematici con lo scopo di rappresentare al meglio oggetti o fenomeni.

### 4.5.2 Definire l'algoritmo di apprendimento di una rete neurale con unità arbitrarie. Definire la funzione obbiettivo utilizzata.

Apprendere in una rete neurale significa modificare i pesi dei singoli nodi in modo da approssimare sempre meglio una funzione che mappa dei valori di ingresso a dei valori di uscita dati, mediante l'ottimizzazione dell'errore quadratico medio.

I passi usualmente sono:

1. Inizializzazione dei pesi.
2. Calcolo dell'output stimato dal vettore pattern di input.
3. Calcolo dell'errore tra il valore ottenuto e quello reale.
4. Retro-propago il gradiente dell'errore dal layer di uscita a quello di ingresso.
5. Ripeto dallo step 2, fino all'ottenimento di una buona stima o convergenza della rete.

Nell'algoritmo di apprendimento si minimizza l'errore quadratico medio, per cui la funzione obbiettivo risulta essere:

$$\min \text{MSE} = \min \frac{1}{n} \sum_{k=1}^n (y_{\text{stimato}_k} - y_{\text{atteso}_k})^2$$

### 4.5.3 Come si utilizza la funzione obbiettivo nell'algoritmo di apprendimento.

Essa si utilizza con la tecnica del gradiente estesa a più variabili, calcolando l'incremento da dare a ciascun peso come:

$$\Delta \omega_{ij} = \eta \frac{\partial \text{MSE}}{\partial \omega_{ij}}$$

Nel caso lineare essa viene detta  $\delta$  rule:

$$\Delta \omega_{ij} = \eta \frac{1}{n} \sum_j^n (y_{\text{stimato}_k} - y_{\text{atteso}_k}) u_i$$

### 4.5.4 Cosa si intende per apprendimento per epoche e per trial? Qual è il vantaggio di ciascuna delle modalità di apprendimento?

**Apprendimento per trial** I pesi vengono aggiornati per ogni pattern di input.

**Apprendimento per epoca** I pesi vengono aggiornati ad ogni insieme di pattern.

L'aggiornamento per trial risulta essere più veloce ma è meno preciso, viceversa per quello per epoche.

### 4.5.5 Cosa si intende per training e test set? Perché mai vengono utilizzati? Quali problemi si vogliono evitare?

**Training set** Si tratta dell'insieme dei dati utilizzati dalla rete per calibrare i pesi delle singole unità.

**Test set** Viene utilizzato una volta che il training set è stato completato per testare la bontà dei parametri.



Vengono utilizzati per effettuare una *cross-validation*, ovvero verificare che l'errore sul training set sia simile a quello sul test set. È importante che i set siano omogenei perché siano utili (per esempio, se volessimo realizzare un classificatore e avessimo solo una classe per set non riusciremmo mai ad avere una generalizzazione).

Si vuole evitare l'under-fitting (parametri errati dovuti al poco addestramento) e l'over-fitting, ovvero quando i parametri approssimano specificatamente il training set.

#### 4.5.6 Una rete neurale con unità a sigmoide è un modello parametrico? È lineare? Perché?

Si tratta di un modello parametrico in quanto a prescindere dalla unità utilizzate gli input sono pesati con dei parametri. Il sigmoide non è una funzione lineare:

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

Figura 4.3: Esempio di funzione sigmoide

#### 4.5.7 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?

Purché tecnicamente non errato, l'aggiunta di troppi parametri sarebbe semplicemente superflua e porterebbe ad un inutile aumento dei costi di computazione. Se i dati non hanno errore ci sarà un numero finito di parametri che descrivono completamente la legge.

#### 4.5.8 Cosa si intende per un problema di regressione ed illustrare una possibile soluzione.

Il problema della regressione, o *predictive learning*, consiste nel trovare un modello a partire da un insieme di dati, in modo da poterlo utilizzare per fare previsioni future o dare risposte corrette all'input di nuovi dati.

##### Una possibile soluzione

Un approccio molto usato è la combinazione lineare di funzioni di base, stile *black-box*.

$$y(x) = \sum_{i=1}^n \omega_i G(x - x_i, \sigma)$$

dove i pesi  $\omega_i$  sono i parametri da stimare mentre  $G$  possono essere gaussiane equi-spaziate.

#### 4.5.9 Come funziona l'approssimazione incrementale multi-scala, cosa garantisce e quali vantaggi può avere?

L'approssimazione incrementale multi scala è un metodo molto utilizzato per la ricostruzione digitale di superfici a partire da campionamenti. Il problema consiste nel ricostruire una superficie sottoposta a scansione, riproducendo nel modo più fedele possibile forme e profondità.

L'approccio tipico consiste nell'applicazione incrementale di layer di filtraggio, utilizzando una scala che decresce per ogni strato (da qui multi-scala).

L'approssimazione incrementale multi-scala consente una ricostruzione abbastanza veloce dei dati con precisione regolabile modificando la soglia.

Oltre alla velocità del metodo, la tecnica consente in modo simile al clustering Quadtree decomposition (QTD) di avere più risoluzione dove necessario.

## 4.6 Domande su Clustering

### 4.6.1 Cosa si intende per clustering? In quali famiglie vengono divisi? Esplicitare la differenza tra soft e hard clustering

Il *clustering* è una tecnica di apprendimento non supervisionato che consiste nel separare un insieme di dati non etichettati in insiemi, i *cluster*, internamente omogenei.

#### In quali famiglie sono divisi?

Gli algoritmi di clustering si dividono principalmente in due categorie: 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 e si dividono tra *hard* e *soft*.

Infine alcuni algoritmi possono essere basati su teorie probabilistiche: si parla di **algoritmi statistici**.

#### Differenza tra soft e hard clustering

Negli algoritmi partizionali, 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*.

### 4.6.2 Che relazione c'è tra clustering e classificazione e quali sono le criticità?

Clustering e classificazione si pongono di risolvere lo stesso problema, il raggruppamento dei dati in classi, ma agiscono in modo profondamente diverso:

Il clustering procede in modo non supervisionato, agendo unicamente sui dati, mentre la classificazione procede in modo supervisionato, usando informazioni estratte dal training set.

#### Quali sono le criticità?

Il clustering **non** è un problema ben posto: i risultati vanno validati ed interpretati. Esistono vari gradi di libertà nella modellazione di un algoritmo di clustering, rappresentazione dei pattern, calcolo delle feature, numero dei cluster attesi, scelta del tipo di distanza da usare tra cluster etc...

#### È possibile usare una mappa di Kohonen per rappresentare una sfera? Perché?

Sì, poiché le SOM distribuiscono una griglia su una superficie man mano che i dati vengono analizzati. Il dato sposta i punti della griglia con i loro vicini in modo da ottenere la migliore distribuzione possibile, nel caso di una sfera in cui i dati rappresentano la superficie la mappa, assunto che vi siano abbastanza dati, ne assumerà la forma.

## 4.7 Domande su Intelligenza Artificiale

### 4.7.1 Si descriva il funzionamento della Forward Search. Perché è considerato un template e non un algoritmo?

La **FORward Search** è un template per algoritmi legate all'esplorazione di un grafo, partendo da uno stato iniziale e cercando di arrivare ad uno stato di goal.

Alla prima iterazione si marca il nodo di start come visitato e lo si inserisce nella coda, quindi si procede iterativamente: a ogni ciclo si estrae (pop) il primo elemento della coda e se è un nodo di **goal** l'algoritmo termina con successo, altrimenti marca i nodi vicini non visitati come **alive** e li inserisce in coda. Se un nodo è stato visitato e sono stati visitati tutti i nodi raggiungibili da esso, viene marcato come **dead**. Se la coda risulta vuota senza avere identificato un nodo di goal, l'algoritmo fallisce.

Si tratta di un **template** e non un algoritmo perché non è specificato il criterio con cui ordinare la coda di priorità  $Q$ .

---

#### Algorithm 1: Algoritmo Forward Search

---

**Data:** Un grafo  $G$ , un nodo iniziale  $s \in G$  ed i nodi di goal  $X_t \subset G$ .

**Result:** Determina se un nodo  $t \in X_t$  è raggiungibile dal nodo  $s$ .

---

```

1 begin
2    $Q \leftarrow \{s\};$ 
3   Marca  $s$  come visitato;
5   while  $Q \neq \emptyset$  do
6      $x \leftarrow Q.pop();$ 
7     if  $x \in X_t$  then
8       return successo;
9     end
10    for  $x' \in \text{Intorno}(x)$  do
11      if  $x'$  non è stato visitato then
12        Marca  $x'$  come visitato;
13         $Q \leftarrow Q \cup \{x'\}$ 
14      end
15    end
16  end
17  return fallimento;
18 end
```

---

### 4.7.2 Si elenchino due possibili implementazioni di Forward Search elencandone proprietà, vantaggi e svantaggi.

#### Breadth First Search

L'algoritmo Breadth First Search, o *visita in ampiezza* implementa la coda di priorità  $Q$  come una coda FIFO.

Tutti i nodi alla stessa profondità sono visitati prima di procedere al livello successivo. Se viene trovato il percorso, è garantito che questo avrà la lunghezza minima.

La ricerca risulta di tipo sistematico.

#### Deapth First Search

L'algoritmo Deapth First Search, o *visita in profondità*, implementa la coda  $Q$  come una pila LIFO, visitando quindi sempre l'ultimo nodo non visitato identificato, dando quindi priorità ai nodi più lunghi.

Nel caso di grafi infiniti la ricerca risulta non sistematica.

## 4.8 Domande sul Planning

### 4.8.1 Come è definito il problema dell'esplorazione autonoma di ambienti con robot mobili?

Il problema dell'esplorazione autonoma di ambienti con robot mobili consiste nel mettere un robot dotato di capacità di movimento in un ambiente inizialmente ignoto.

Il robot, muovendosi all'interno dell'ambiente deve costruire una rappresentazione interna della mappa dell'ambiente in modo da facilitare la navigazione per dei task futuri.

Un robot che ha esplorato efficacemente un ambiente, ad esempio, sarà in grado di muoversi da un punto A a un punto B dell'ambiente nel più breve tempo possibile.

Esistono 3 problemi tipici nell'esplorazione:

**Problema della ricarica** È importante anche che il robot si muova ottimizzando gli spostamenti, in quanto è dotato di un tempo di carica limitato. Il robot deve essere in grado di pianificare delle soste regolari presso le stazioni di ricarica.

**Problema della posizione** Dato che un robot non può mai conoscere la sua posizione assoluta nell'ambiente, deve essere in grado di ricavare la sua posizione relativa, operazione molto complessa e facilmente soggetta a errori di stima.

**Problema dell'incertezza dell'esito delle azioni** I robot si muovono in un ambiente totalmente stocastico: l'esito di ogni azione non è certo, sia per la presenza di ostacoli non previsti nell'ambiente, sia perché i robot sono molto fragili e tendono a rompersi/cadere/bloccarsi. Un robot mobile deve essere munito di una policy efficace che lo renda in grado di reagire agli ostacoli imprevisti.

### 4.8.2 Cosa è un sistema multi-agente? Se ne dia un esempio in applicazioni robotiche

Un sistema multi-agente (MAS) è definito come una collezione di agenti che interagiscono tra loro e con l'ambiente per raggiungere un obiettivo comune.

Un esempio di applicazione è la **sorveglianza multi-agente** che coinvolge un gruppo di agenti incaricato di effettuare un'operazione di "clear" di un dato ambiente, vale a dire una visita "strategica" dell'ambiente che assicura l'identificazione e la cattura delle entità ostili che lo contaminano.

I robot devono affrontare il problema della pianificazione di una strategia da seguire e il problema della comunicazione tra di essi, che spesso prendono vie molto diverse e si "perdono di vista".

## 4.9 Domande su Biologia

### 4.9.1 Definire il neurone biologico evidenziandone le parti più significative per la trasmissione dell'informazione ed il loro comportamento.

Il neurone è l'unità base del sistema nervoso e si occupa di elaborare e trasmettere segnali.

Risulta composto da 4 regioni:

**Dendriti:** Sono la componente di input del neurone, dove giungono i segnali degli altri neuroni.

**Nucleo:** Contiene l'informazione genetica della cellula.

**Soma:** Si tratta del corpo della cellula.

**Assone:** Rappresenta la componente di output del neurone e si dirama dal soma da un punto particolare detto *colle dell'assone*. Possiede ramificazioni su diversi punti di esso, compresa la fine e su di esso scorre il **potenziale d'azione**.

### 4.9.2 Descrivere il funzionamento complessivo del neurone biologico.

Il neurone a riposo ha un comportamento stereotipato legato al gradiente di concentrazione, che tende a fare uscire ioni di potassio  $K^+$  e a portarsi alla differenza di potenziale ( $-70mV$ ) che tende a farli entrare. Il sistema è così mantenuto in equilibrio, grazie alla scarsa permeabilità della membrana cellulare agli ioni di sodio  $Na^+$  e cloro  $Cl^-$ .

Questo stato, detto **sotto-soglia**, si interrompe all'arrivo di uno stimolo, in cui i dendriti inviano un segnale elettrico al soma, aumentando la differenza di potenziale. Superati i  $-50mV$  viene generato un potenziale detto di **spike** grazie all'apertura dei canali *voltage-gated* che fanno entrare ioni di sodio  $Na^+$ . Raggiunta la carica di  $30mV$  i canali iniziano a chiudersi e si aprono quelli del potassio, che facendo uscire gli ioni  $K^+$  portano la carica allo stato di equilibrio. Grazie al funzionamento della pompa sodio-potassio inoltre, la cellula torna allo stato originale.

### 4.9.3 Dove avviene principalmente l'"apprendimento" nei neuroni biologici?

L'apprendimento nei neuroni avviene ad un livello più alto, in quanto i neuroni non sono entità di per sé intelligenti. Esso è dovuto piuttosto alla plasticità del cervello, in grado di effettuare di *rewiring* di neuroni per formare il percorso migliore per l'informazione. Inoltre ricerche recenti hanno individuato categorie particolari di neuroni, detti **neuroni specchio**, i quali potrebbero essere alla base dell'apprendimento.

### 4.9.4 Che differenza c'è tra neuroni sensoriali, neuroni motori ed inter-neuroni?

**Neuroni sensoriali** Sono neuroni di input che ricevono l'informazione dagli organi sensoriali e la inviano al sistema nervoso.

**Neuroni motori** Sono neuroni di output che ricevono l'informazione dal sistema nervoso e la ritrasmettono all'apparato muscolare.

**Inter-neuroni** Sono neuroni che compiono l'elaborazione intermedia.

### 4.9.5 Come viene trasmessa ed elaborata l'informazione da un neurone?

I dendriti ricevono il segnale dai neurotrasmettitori attraverso le sinapsi, la corrente così generata passa al soma, il quale genera una spike che viaggia per l'assone fino alla sinapsi di uscita, che passa l'informazione al neurone post-sinaptico.

### 4.9.6 Cos'è uno spike?

Si tratta di un **impulso elettrico** generato dal potenziale d'azione il quale viaggia nell'assone per trasmettere l'informazione da un neurone ad un successivo.

### 4.9.7 Quali sono le aree corticali principali?

La corteccia può essere divisa in aree con criterio spaziale e funzionale.

Secondo il criterio spaziale viene divisa in 4 lobi:

**Limbico:** il più interno, legato all'apprendimento, alla memoria e le emozioni.

**Frontale:** legato al pensiero.

**Parietale:** legato al movimento e la percezione.

**Occipitale:** legato alla visione.

**Temporale:** legato all'udito, al riconoscimento e alla memoria.

Secondo il criterio funzionale viene divisa in 3 aree:

**Aree primaria** recettive ed attuative.

**Aree secondaria** elaborazione.

**Aree terziarie** integrazione sensoriale.

### 4.9.8 Cos'è il codice di popolazione?

Il codice di popolazione è il metodo che il cervello usa per discernere tra stimoli simili: ogni neurone ha una sua direzione preferenziale. L'attività di un neurone è massima quando la direzione dello stimolo coincide con quella preferenziale: in questo modo la rappresentazione interna dello stimolo è la somma pesata dell'attività di tutti i neuroni che rispondono. Così facendo il codice di popolazione trasmette lo stimolo attraverso i neuroni attivati e la frequenza di scarico di ciascuno.

### 4.9.9 Data un'area cerebrale, è univoca la funzione implementata in quell'area?

No, vi sono aree che condividono funzioni. Ad esempio la memoria è elaborata sia dal lobo limbico che da quello temporale.

### 4.9.10 Cosa sono i neuroni a specchio? Quali implicazioni hanno per i sistemi intelligenti e l'apprendimento?

I neuroni a specchio sono una particolare classe di neuroni che si attivano sia quando si compie un'azione sia quando la stessa azione viene osservata. Questo ha implicazioni notevoli, in quanto si suppone che questi neuroni siano alla base di meccanismi come l'empatia e l'apprendimento per imitazione, entrambi aspetti estremamente complessi da modellare nei sistemi intelligenti classici.

### 4.9.11 Cosa si intende per algoritmi genetici ed ottimizzazione evolutiva?

Sulla base dell'evoluzione naturale, la computazione evolutiva, termine generico per una gamma di sistemi, cerca di sfruttare questi principi per giungere alla soluzione ottima di un problema.

Gli algoritmi genetici (dei quali l'ottimizzazione evolutiva è una tipologia) codificano le soluzioni come cromosomi composti da geni e cercano di portare una popolazione (gruppo di cromosomi) all'evoluzione con processi di selezione, ricombinazione e mutazione.

### 4.9.12 Quali sono le differenze ed i punti forti degli algoritmi genetici ed ottimizzazione evolutiva?

#### Differenze nella codifica

Negli algoritmi genetici la codifica è implicita mentre nell'ottimizzazione evolutiva è esplicita.

#### Differenze nell'ordine delle operazioni

Negli algoritmi genetici avviene prima la selezione, nell'ottimizzazione evolutiva si procede prima con la mutazione.

**Ulteriori differenze**

Nell'ottimizzazione evolutiva la ricombinazione è generica e la selezione è tra padri e figli.

**Punti forti**

L'ottimizzazione evolutiva tende a convergere più rapidamente ma gli algoritmi genetici sono più leggeri e semplici da implementare.

**4.9.13 Cosa si intende per elitismo?**

L'elitismo è l'identificazione dei geni migliori attraverso una funzione di fitness che è tanto più alta quanto la soluzione risulta buona. L'introduzione di blocchi o resistenze alla modifica dei geni migliori aiuta così a preservare le buone caratteristiche per tramandarle a generazioni future.

**4.9.14 Che cosa è una matrice di covarianza in generale e come viene applicata nell'ottimizzazione evolutiva?**

La matrice di covarianza rappresenta come vari una variabile aleatoria in confronto alle altre, compresa sé stessa. Nell'ottimizzazione evolutiva si utilizza per regolare la mutazione in modo che sia calibrata alle diverse soluzioni.