



Corso Luigi Einaudi, 55 - Torino

Appunti universitari

Tesi di laurea

Cartoleria e cancelleria

Stampa file e fotocopie

Print on demand

Rilegature

NUMERO: 1430A -

ANNO: 2015

A P P U N T I

STUDENTE: Tortorici

MATERIA: Classificazione e interpretazione di dati biomedici,
Prof. Balestra

Il presente lavoro nasce dall'impegno dell'autore ed è distribuito in accordo con il Centro Appunti.

Tutti i diritti sono riservati. È vietata qualsiasi riproduzione, copia totale o parziale, dei contenuti inseriti nel presente volume, ivi inclusa la memorizzazione, rielaborazione, diffusione o distribuzione dei contenuti stessi mediante qualunque supporto magnetico o cartaceo, piattaforma tecnologica o rete telematica, senza previa autorizzazione scritta dell'autore.

**ATTENZIONE: QUESTI APPUNTI SONO FATTI DA STUDENTIE NON SONO STATI VISIONATI DAL DOCENTE.
IL NOME DEL PROFESSORE, SERVE SOLO PER IDENTIFICARE IL CORSO.**

Classificazione e Interpretazione di Segnali Biomedici

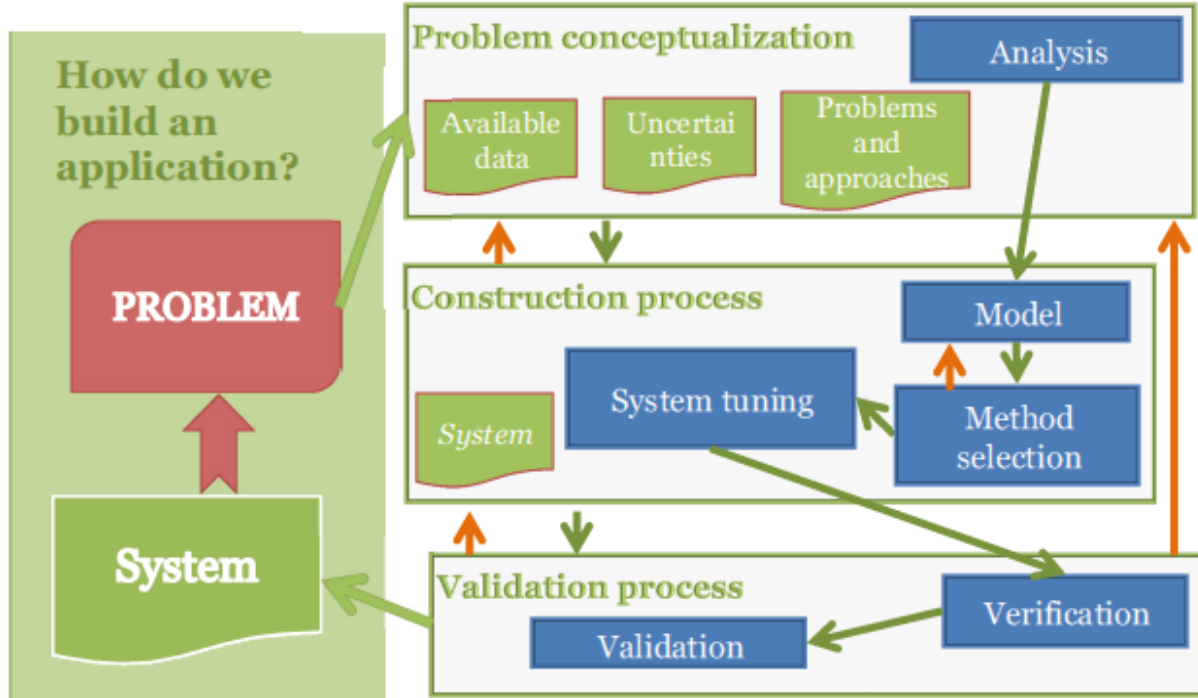
Lezioni: G. Balestra

INDICE

Introduzione.....	1
Application Development.....	1
Processo di Costruzione.....	3
Regressione Lineare.....	6
Sviluppo di un Classificatore.....	10
Ottimizzazione.....	14
Concettualizzazione.....	20
Logica Fuzzy.....	20
Validazione.....	24
Clustering.....	25
Reti Neurali.....	27

In seguito si svolge il processo di costruzione: si sviluppa un modello attraverso vari metodi (matematica, logica, regole, ecc), poi si sceglie il metodo da usare sul modello. Modello e metodo sono spesso interconnessi. Si adeguano i parametri al problema specifico. In questa fase si potrebbe dover tornare all'analisi.

Infine si ha il processo di verifica e validazione del sistema: se questa fase viene superata, il sistema può essere applicato al problema.



PROCESSO DI COSTRUZIONE

La prima cosa utile è avere strumenti per descrivere i nostri dati. Si incontra questo problema, per esempio, quando si prova ad analizzare i dati acquisiti durante un trial. Un trial è uno studio controllato in cui vengono inseriti dei soggetti: si raccoglie una gran quantità di dati e si deve usare la statistica descrittiva.

La statistica può essere di due tipi:

- *descrittiva*: si limita a descrivere e non ha problemi di numerosità del campione;
- *inferenziale*: si prende il campione e si calcolano dei parametri, ma si fanno anche inferenze su come il valore calcolato rappresenti l'intera popolazione.

La popolazione include tutti gli oggetti di interesse, mentre un campione è solo una porzione della popolazione.

Per estrarre il campione ci sono 5 possibilità:

- campionamento random;
- campionamento sistematico: si acquisisce 1 campione ogni n (come nel campionamento di segnali);
- convenience sampling: si inserisce nel campione chi si presenta. È la tecnica peggiore, ma spesso viene utilizzata per ragioni pratiche;
- cluster sampling: si divide la popolazione in gruppi (di solito su base geografica) e si sceglie un campione da ognuno;
- campionamento stratificato: anche in questo caso si divide la popolazione in gruppi, ma basati su caratteristiche.

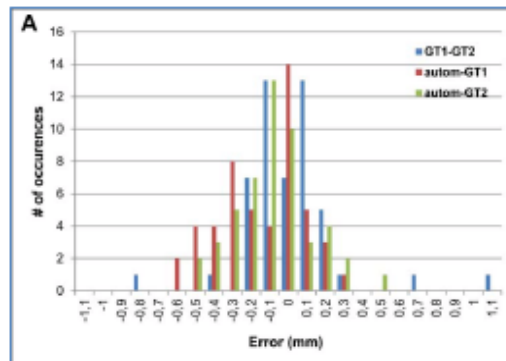
$$skewness = \frac{1}{(n-1)s^3} \sum_{i=1}^n (x_i - \bar{x})^3$$

- kurtosis: misura se i dati sono più a punta o più piatti rispetto a una distribuzione normale. I set di dati con un alto kurtosis tendono ad avere un picco distinto vicino al centro, sono ripide e hanno lunghe code. I set di dati con basso kurtosis tendono ad avere una cima piatta vicino al centro, piuttosto che un picco appuntito.

$$kurtosis = \frac{1}{(n-1)s^4} \sum_{i=1}^n (x_i - \bar{x})^4$$

Spesso c'è bisogno di rappresentare graficamente i parametri e i dati. Le rappresentazioni più usate sono:

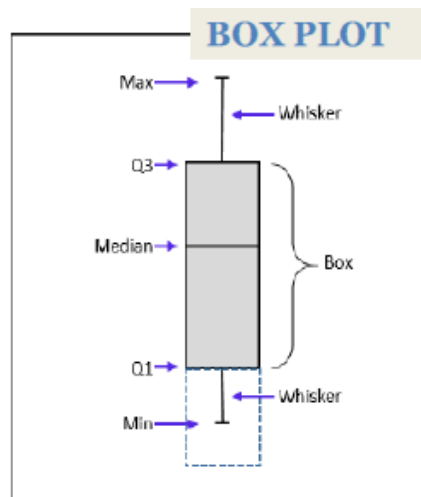
- *bar diagram*: gli elementi vengono divisi in gruppi e si calcolano le frequenze in ciascuno;



- *scatter plot*: confronta due variabili diverse;



- *box plot*: si costruisce un rettangolo che contiene il 50% dei dati. L'intero campione viene diviso in 4 percentili. Questo diagramma indica se la popolazione è distribuita equamente intorno alla mediana.



$x \rightarrow$ variabile indipendente;
 $\alpha \rightarrow$ intercetta;
 $\beta \rightarrow$ pendenza.

La dispersione è importante per stabilire la bontà della regressione: se σ^2 è troppo elevato, la regressione lineare non è un buon modello dei dati.

Per calcolare i coefficienti si usa il metodo dei *minimi quadrati*. Sia (x_i, y_i) una coppia di valori misurati, mentre \hat{y}_i sia il valore di Y ricavato a partire dall'i-esimo valore di X tramite la linea di regressione. Si vuole che $d_i = y_i - \hat{y}_i$ sia la minore possibile. Il metodo dei minimi quadrati sceglie due coefficienti α e β tali da minimizzare la somma del quadrato delle distanze dei campioni dalla linea.

$$S = \sum_{i=1}^n d_i^2$$

I coefficienti sono dati da:

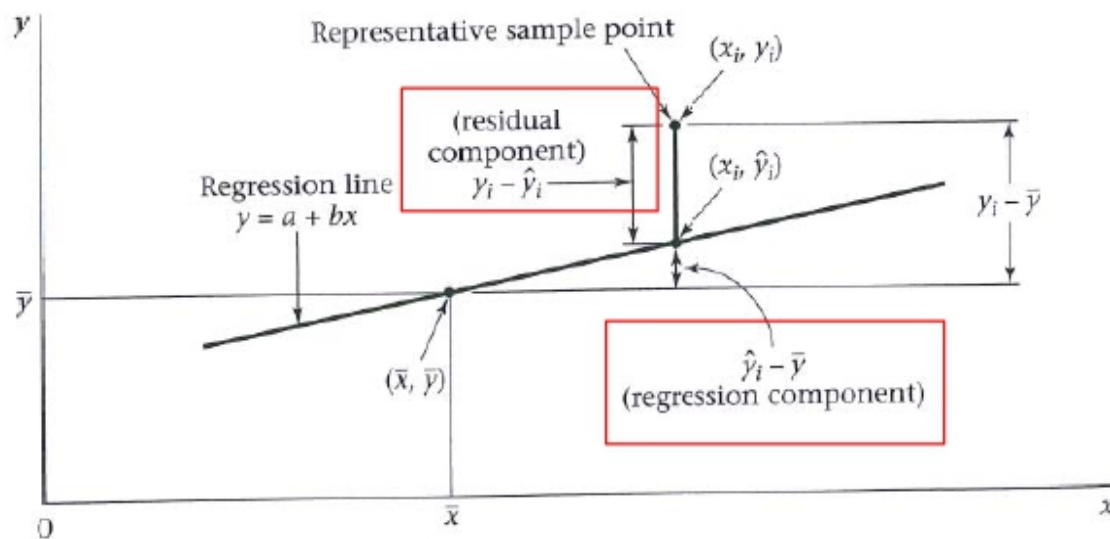
$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$a = \bar{y} - b\bar{x}$$

Si ricorda che la notazione in lettere greche (α, β) si riferisce ai valori veri, cioè a quelli della popolazione, mentre quella in lettere latine (a, b) si riferisce alla stima, cioè ai campioni.

Per valutare la bontà di una regressione si fa riferimento a due parametri:

- *residui*: differenza tra il valore vero y e il valore stimato \hat{y} in corrispondenza di un dato x ;
- *regression component*: valore stimato $\hat{y} - \bar{y}$.



Un buon fitting deve avere la somma dei regression component **molto maggiore** della somma dei residui.

La vera relazione tra X e Y di solito non è nota per i dati veri, ma si può sempre ottenere la linea di regressione tramite il metodo dei minimi quadrati. Questo significa che nelle applicazioni reali si ha accesso a una serie di osservazioni dalle quali si può calcolare la linea con i minimi quadrati; tuttavia la linea di regressione della popolazione non si conosce. Set di dati diversi estratti dalla stessa popolazione danno come risultato linee di regressione leggermente diverse, ma la linea di regressione della popolazione non varia.

La *distribuzione di probabilità* di una variabile casuale finita è una lista di probabilità associate a

$$a \pm 2 \text{ VAR}(a)$$

$$b \pm 2 \text{ VAR}(b)$$

Più l'intervallo di confidenza è piccolo più si è certi della stima fatta.

I test sui coefficienti possono essere:

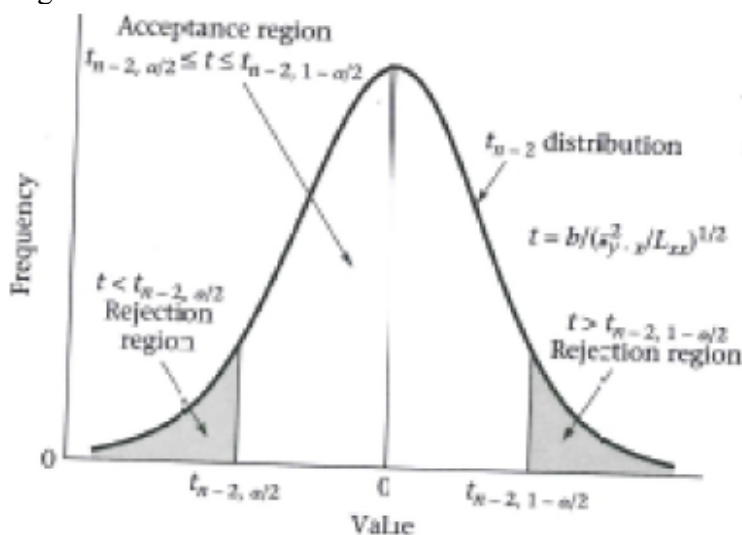
- single sided: si stabilisce se la variabile è maggiore oppure minore o uguale a un dato valore;
- two sided: si stabilisce se la variabile è uguale oppure diversa a un dato valore (nel caso fosse diversa potrebbe essere maggiore o minore: questo non viene stabilito).

Per esempio:

$$H_0 \rightarrow \text{coefficiente } \beta = 0$$

$$H_A \rightarrow \text{coefficiente } \beta \neq 0$$

A ogni test è associata una distribuzione.



Si può rigettare H_0 se il valore calcolato di t cade in una delle due aree evidenziate; se il valore calcolato è compreso tra i due valori, non si può disprovare H_0 e la regressione non ha valore.

In realtà di solito non ci sono solo due variabili, ma ciascuna variabile dipendente può dipendere da numerose variabili indipendenti. In tal caso si potrebbe fare una regressione lineare per ogni coppia di variabili, ma così si perde l'effetto congiunto delle variabili indipendenti tra loro. È meglio fare una regressione multipla. Se si hanno p distinte variabili, la regressione lineare ha forma:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

Anche in questo caso i parametri sono stimati con il metodo dei minimi quadrati.

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

La relazione tra il coefficiente della regressione del campione (b) e il suo coefficiente di correlazione (r) è:

$$b = r \frac{s_y}{s_x}$$

dove s_y e s_x sono la deviazione standard delle variabili.

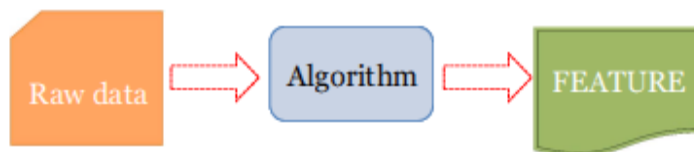
Il valore R^2 è una misura della bontà dell'interpolazione:

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Può essere ritenuto come la porzione della varianza di y che può essere spiegata da x . Se vale 1, significa che tutta la variazione di y può essere spiegata dalla variazione in x (tutti i punti cadono sulla linea di regressione); se vale 0, significa che x non dà informazioni su y .

Bisogna identificare le possibili features, selezionarle, costruire il classificatore, verificarlo e validarlo. A volte è necessario tornare indietro!

Quando si identificano le features, si possono usare i dati così come sono oppure applicarvi degli algoritmi.



Bisogna ridurre le features. Per farlo esistono due tecniche:

- *construction*: si costruiscono nuove caratteristiche, perdendo il contatto diretto con la base del problema;
- *selection*: permette di mantenere il legame con le caratteristiche di partenza. Si seleziona un sottoinsieme di caratteristiche, rimuovendo quelle inutili o ridondanti.
 - Caratteristica rilevante → è essenziale per ottenere buone prestazioni dal sistema;
 - caratteristica irrilevante → non aggiunge informazioni utili;
 - caratteristica ridondante → è altamente correlata con altre caratteristiche.

Gli scopi principali della features selection sono diminuire la complessità del modello e migliorare le performance del sistema. Ha i seguenti benefici:

- facilita la visualizzazione dei dati;
- riduce la richiesta di misure e di spazio per salvarle;
- riduce i tempi per usare il sistema e per imparare a usarlo;
- migliora la previsione della performance.

La rilevanza di una caratteristica può essere forte o debole. Una caratteristica fortemente rilevante non può essere rimossa senza perdere accuratezza nella previsione. Una caratteristica debolmente rilevante può contribuire all'accuratezza del sistema, ma dipende da quali altre features vengono considerate. Una caratteristica può essere irrilevante da sola, ma rilevante se considerata con altre.

La selezione delle features può essere svolta in due modi:

- ordinare le caratteristiche secondo un certo criterio e selezionare le prime k;
- selezionare un sottoinsieme di features senza valutare la deteriorazione della performance.

La selezione può essere *supervised* (si verifica il corretto funzionamento del classificatore su un data set preventivamente classificato) o *unsupervised* (non si può valutare la bontà del classificatore con il confronto).

Alcuni aspetti importanti della selezione delle caratteristiche sono:

- modelli:
 - filtraggio: si adotta una regola indipendente dal classificatore per scegliere le features;
 - wrapper: la regola di scelta dipende dal classificatore;
 - embedded: la features selection è un tutt'uno con la costruzione del classificatore;
- strategie per definire come selezionare le features:
 - backward: da un insieme di caratteristiche si tolgono features, fermandosi quando le prestazioni del classificatore peggiorano;
 - forwards: si aggiungono features finché le prestazioni del classificatore non smettono di migliorare;
 - randomized: si costruiscono sottoinsiemi del training set di features, cercando di capire quale combinazione sia la più adatta;
- valutazione:
 - si comparano i casi di prima e dopo la features selection;
 - si comparano due algoritmi di selezione per vedere qual è il migliore.

Si devono usare un training set per selezionare le caratteristiche e uno per addestrare il

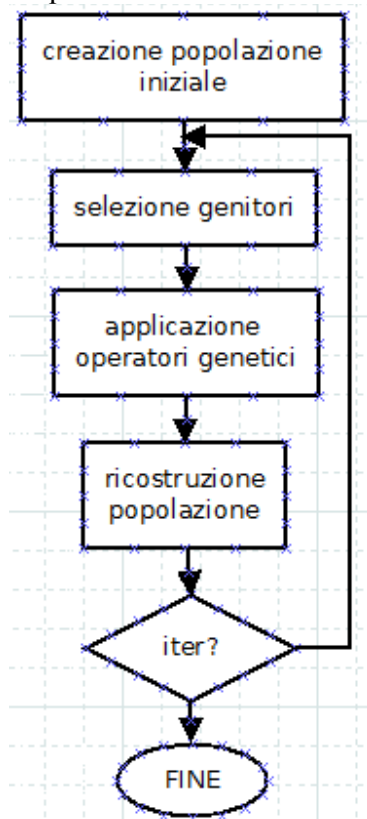
di stimare quante sono le soluzioni possibili. Se non ci sono vincoli, se k è la lunghezza della stringa si hanno 2^k-1 possibili soluzioni. Sapere quante sono le soluzioni potenzialmente esplorabili serve a basare le scelte sul numero di iterazioni e sulle dimensioni della popolazione;

2. *quanto è grande la popolazione iniziale? Quante iterazioni vanno fatte?* Si deve definire il numero di individui n_{ind} , che deve essere correlato al numero di iterazioni n_{iter} . I due valori sono funzione l'uno dell'altro e del numero massimo di soluzioni;
3. *definizione del numero di genitori e della regola di selezione.* Il numero di genitori deve essere inferiore al numero di individui (grandezza della popolazione), ma non può essere troppo piccolo. Il numero di genitori corrisponde al numero di soluzioni che vanno ricombinate con gli operatori genetici. Di solito il numero di genitori è l'**80-90%** degli individui;
4. *pm? pc?* Bisogna definire quante soluzioni genitori in percentuale vengono usate per il cross over (pc) e la mutazione (pm). Le soluzioni possono essere selezionate tramite vari metodi: tutte, random, solo una parte, ecc. Di solito pc è alta (a volte è 1), mentre pm è bassa (intorno a 0,1). Non è detto che pm e pc rimangano costanti per tutte le iterazioni. Si deve definire come l'operatore viene implementato. Dall'applicazione degli operatori si possono generare un numero di figli pari, maggiore o minore del numero di genitori;
5. *regola per la ricostruzione della popolazione.*

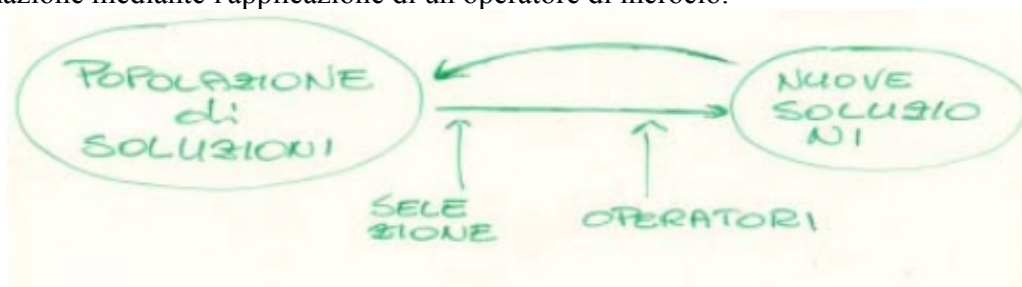
Se si esegue l'algoritmo una volta sola non si può sapere nulla sulla variabilità della soluzione (l'obiettivo è quello di ottenere sempre soluzioni simili). Bisogna ripetere il processo un certo numero di volte: la popolazione iniziale viene usata con lo stesso GA e gli stessi parametri un certo numero di volte (n_{rip}): un numero di ripetizioni ragionevole è **10** (mai di meno, mai più di 20). si confrontano le soluzioni ottenute.

Iterazioni → ricostruzione della popolazione in un singolo GA;

ripetizioni → quante volte lo stesso GA viene rifatto.



individui sono espresse usando genotipi. Le principali operazioni di un GA sono la selezione e la ricombinazione mediante l'applicazione di un operatore di incrocio.



Nel contesto dell'EC, ogni individuo rappresenta un candidato di soluzione per un problema di ottimizzazione. Le caratteristiche di un individuo sono rappresentate da cromosomi (genoma). Queste caratteristiche si riferiscono alle variabili del problema di ottimizzazione.

Ci riferisce come gene ad ogni variabile che ha bisogno di essere ottimizzata; è la più piccola unità di informazione. *An assignment of a value from the allowed domain of the corresponding variable is referred to as an allele.*

Una fase importante nella progettazione di un EA è trovare una rappresentazione appropriata delle soluzioni candidate (cromosomi). L'efficienza e la complessità dell'algoritmo di ricerca dipendono molto dallo schema di rappresentazione. La maggior parte degli EA rappresenta le soluzioni come vettori di uno specifico tipo di dati. La classica rappresentazione per i GA è una **stringa binaria**.

Deve esistere una regola che permetta di codificare e decodificare la soluzione.

Quando si lavora con GA per prima cosa si deve creare la popolazione di partenza, cosa che si fa generando ns possibili soluzioni. La dimensione della popolazione rimane costante durante l'iterazione; esistono dei casi particolari di GA che permettono di modificarla, in particolare di incrementarla. Il numero di elementi della popolazione ns e il training set non hanno nessuna correlazione: ns va scelto in base alla lunghezza della stringa (tanto più lunga è la stringa, tanto più numerosa è la popolazione).

Il modo standard di generare una popolazione iniziale consiste nell'assegnare valori casuali dal dominio permesso a ciascuno dei geni di ogni cromosoma. Lo scopo della selezione casuale è di assicurare che la popolazione sia una rappresentazione uniforme dell'intero spazio di ricerca: se delle zone dello spazio non sono coperte dalla popolazione iniziale, c'è la possibilità che tali zone siano trascurate nel processo di ricerca.

Lo scopo è quello di esplorare lo spazio delle soluzioni: quanto si riesca a esplorarlo dipende da ns e dal numero di iterazioni (cioè quante evoluzioni della popolazione si portano avanti). Una popolazione piccola richiede un numero maggiore di iterazioni per consentire l'esplorazione di un numero sufficientemente grande di soluzioni, ma non esiste nessun rapporto di proporzionalità tra ns e numero di iterazioni. Un elevato numero di individui aumenta la diversità, consentendo di migliorare la capacità di esplorazione della popolazione; tuttavia richiede un alto grado di complessità computazionale per ogni generazione.

La funzione di fitness rappresenta ciò che si vuole ottimizzare. È una funzione usata per quantificare quanto sia buona la soluzione rappresentata da un cromosoma. Tipicamente è una funzione matematica, ma potrebbe essere anche un modello di simulazione o una funzione logica.

Esistono diversi tipi di problemi di ottimizzazione, che hanno un'influenza sulla formulazione della funzione di fitness:

- ottimizzazione senza vincoli: la funzione di fitness è la funzione obiettivo;
- ottimizzazione con vincoli: non tutte le soluzioni sono possibili, ci sono delle condizioni da rispettare (per es l'università deve organizzare le lezioni tenendo conto di un numero massimo di ore fattibili);
- ottimizzazione con più obiettivi: un tipico esempio è quello di massimizzare la qualità e minimizzare i costi;

Gli operatori di evoluzione negli EA sono applicati iterativamente finché non viene soddisfatta una condizione di stop. La condizione di stop più semplice è quella di limitare il numero di generazioni che l'EA può eseguire; oppure si possono porre condizioni di stop relative ai miglioramenti della funzione di fitness. Si possono usare i seguenti criteri di convergenza:

- termina quando non si osservano miglioramenti dopo un certo numero di iterazioni consecutive;
- termina quando non ci sono cambi nella popolazione;
- termina quando è stata trovata una soluzione accettabile;
- termina quando la pendenza della funzione obiettivo è circa zero.

Oltre che dalle dimensioni della popolazione, la performance di un GA è influenzata dalla velocità di mutazione (pm) e di cross over (pc). Normalmente i valori di pm e pc sono tenuti fissi. Poiché si ritiene che questi parametri abbiano una grande influenza sulla performance, è importante che pc e pm abbiano i valori ottimali per migliorarla. Ottenere i valori ottimali tramite la variazione di parametri empirici è un processo che consuma molto tempo: in alternativa si possono usare parametri che cambino dinamicamente.

I **metodi euristici** si usano quando le soluzioni utilizzabili sono molto numerose (almeno qualche centinaio di soluzioni possibili). Si cerca un algoritmo che permette di ottenere una soluzione in un tempo accettabile: quanto sia lungo tale tempo dipende dalla situazione. Se si volesse ottenere solo un ottimo globale, gli algoritmi dovrebbero esplorare tutto lo spazio delle soluzioni; invece si preferisce avere una buona soluzione del problema in un tempo conveniente; non c'è certezza di arrivare a un ottimo globale e nemmeno a un ottimo locale.

Una soluzione s' è un *ottimo globale* se ha una funzione obiettivo migliore di tutte le soluzioni nello spazio di ricerca: $\forall s \in S f(s') \leq f(s)$. L'ottimo globale può non essere unico (se la funzione assume più volte lo stesso valore).

Un algoritmo ha bisogno di due risorse per risolvere un problema: tempo e spazio. La complessità temporale di un algoritmo è il numero di step richiesto per risolvere un problema di dimensione n . La complessità viene generalmente definita in termini di analisi del caso peggiore.

Gli algoritmi possono concentrarsi sull'arrivare a un ottimo (anche locale) oppure avere l'obiettivo di esplorare un largo numero di soluzioni. Combinando i due scopi si ottengono delle buone soluzioni.

I metodi euristici possono essere classificati come:

- *ispirati alla natura* oppure *non ispirati alla natura*;
- *con uso di memoria* oppure *senza uso di memoria*: per esempio nei GA non si usa la memoria, in quanto le nuove soluzioni sono indipendenti da quelle precedenti. Esistono anche metodi che tengono traccia di ciò che è capitato nelle iterazioni precedenti (per es Tabù search);
- *deterministici* oppure *stocastici*: se la decisione viene presa in base alla probabilità, partendo dalle stesse condizioni iniziali si può avere una soluzione diversa (cosa da tenere in conto durante la validazione dell'algoritmo);
- *basati su popolazione* oppure *basati su singola soluzione*: i primi evolvono l'intera popolazione durante la ricerca della soluzione, mentre i secondi evolvono solo una soluzione;
- *iterativi* oppure *greedy*: i primi partono da una soluzione completa e la trasformano a ogni iterazione usando degli operatori di ricerca. I secondi partono da una soluzione vuota e ad ogni step le aggiungono un pezzo, finché non si ottiene una soluzione completa.

Le euristiche basate sulla ricerca locale cercano di trovare una soluzione migliore di quella corrente attraverso una ricerca nel vicinato della soluzione corrente.

Il vicinato è l'insieme delle soluzioni raggiungibili dalla soluzione corrente attraverso un'operazione

- diversificazione (memoria a lungo termine): la memoria a lungo termine immagazzina informazioni sulle soluzioni visitate durante la ricerca. Il processo di diversificazione può essere usato periodicamente o dopo un certo numero di iterazioni senza miglioramenti. Ve ne sono di 3 tipi:
 - restart diversification: si introducono le componenti meno visitate nella soluzione corrente, poi si fa partire una nuova ricerca a partire da questa nuova soluzione;
 - continuous diversification: durante la ricerca si introducono errori per incoraggiare la diversificazione;
 - strategic oscillation: permette di considerare e penalizzare soluzioni intermedie che non sono fattibili.

Il Tabù search è poco utilizzato. In genere viene usato per problemi di assegnazione e di localizzazione.

La **swarm intelligence** consiste in algoritmi ispirati al comportamento collettivo di alcune specie. Le principali caratteristiche sono:

- le particelle sono agenti semplici e non sofisticati;
- le particelle cooperano mediante un mezzo di comunicazione indiretto;
- le particelle si muovono nello spazio di decisione.

L'*ant colony optimization* (ACO) si basa sull'osservazione del comportamento delle formiche. Le formiche riescono a trovare il percorso più breve dal formicaio al cibo. L'algoritmo è basato sui seguenti step:

1. all'inizio c'è un movimento caotico e casuale per la ricerca del cibo;
2. quando la fonte di cibo è individuata, i percorsi di attività diventano più organizzati, con sempre più formiche che seguono lo stesso percorso per arrivare al cibo;
3. alla fine tutte le formiche seguono lo stesso percorso, il più breve.

Il comportamento delle formiche è guidato da un ferormone che viene lasciato sul terreno. È una sostanza volatile. Maggiore è la quantità di ferormone su un sentiero, maggiore è la probabilità che le formiche scelgano quel sentiero. La sostanza chimica ha un'azione che diminuisce nel tempo (processo di evaporazione) e la quantità lasciata da una formica dipende dalla quantità di cibo (processo di rinforzo).

L'ACO comincia con l'inizializzazione del ferormone ed è composto dall'iterazione di due step principali: costruzione della soluzione e aggiornamento del ferormone.

La costruzione della soluzione viene effettuata secondo una regola probabilistica. Le formiche artificiali possono essere considerate come procedure greedy stocastiche che costruiscono una soluzione in maniera probabilistica tramite l'addizione di componenti a una soluzione parziale, finché non si deriva una soluzione completa. Questo processo tiene conto di: percorsi di ferormone, informazioni euristiche dipendenti dal problema.

L'aggiornamento del ferormone è svolto usando le soluzioni generate e viene applicato in due fasi: evaporazione e rinforzo. L'evaporazione consiste nella diminuzione automatica del ferormone di una proporzione definita; serve a favorire maggiori movimenti casuali. Il rinforzo avviene quando la fitness è buona.

Il *simple ant colony optimisation* (SACO) risolve il problema generale di trovare il percorso più breve tra due nodi di un grafico.

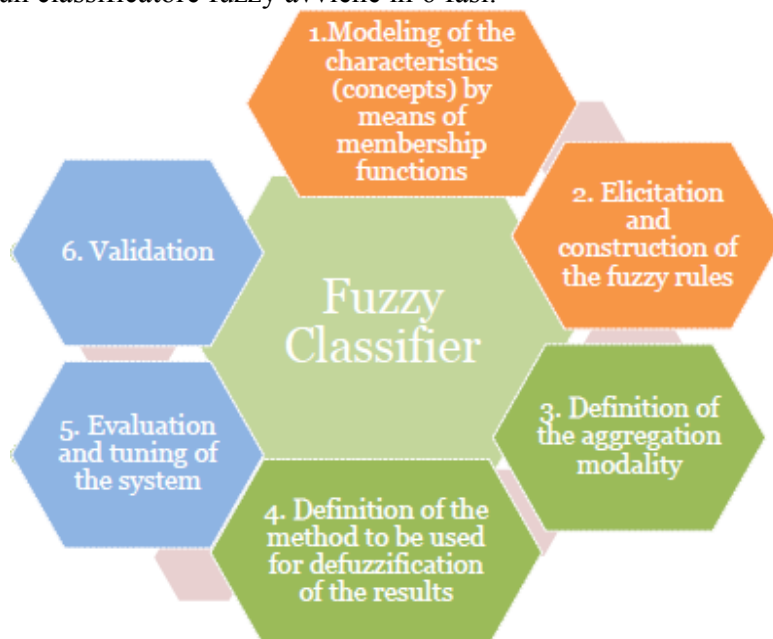
[fine metodi di ottimizzazione]

Molti problemi di ottimizzazione sono **vincolati**. I vincoli possono essere di qualsiasi tipo (lineari o non lineari, di uguaglianza o di disuguaglianza). La gestione dei vincoli può essere correlata a due approcci diversi:

- strategie che agiscono sulla rappresentazione delle soluzioni o della funzione obiettivo;
- strategie non direttamente correlate alla rappresentazione delle soluzioni o della funzione

Il classificatore riceve dall'esterno degli input crisp, che vengono fuzzificati, cioè si identificano i valori della loro funzione di appartenenza alle varie classi. L'inferenza applica le regole attive. Si ottiene l'output, che viene trasformato in un valore crisp (passaggio non necessario, ma svolto quasi sempre).

La costruzione di un classificatore fuzzy avviene in 6 fasi.



Costruzione della membership function (MF). Per ogni variabile c'è un insieme di membership function: quante di preciso lo decidiamo noi. Per esempio la variabile “età” può avere membership function relative alle classi “giovane”, “adulto”, “anziano”, oppure “adolescente”, “giovane”, “adulto”, “anziano”, “molto anziano”. Le membership function che si riferiscono allo stesso oggetto vengono rappresentate sullo stesso grafico. Normalmente le MF si intersecano con una intersezione simmetrica, anche se non è una condizione obbligatoria e può capitare che uno stesso punto abbia grado di appartenenza 1 a più di una classe contemporaneamente. La costruzione delle MF può avvenire mediante processi diversi:

- likelihood view: si intervistano delle persone chiedendo di assegnare una classe a un dato (per es: 35 anni è giovane, adulto o vecchio?)
- random set view: si intervistano delle persone sui confini che secondo loro dovrebbero avere le classi (per es: quali sono gli estremi per dire che uno è giovane, adulto o vecchio?);
- similarity view: si confronta il dato con un esempio di riferimento.

La MF dipende dal contesto! Per esempio nel contesto dei pensionati ci saranno classi di giovani, vecchi e anziani con estremi diversi rispetto al contesto degli studenti.

Si ha un set totalmente ordinato quando si dividono gli elementi di un insieme in classi che contengono solo elementi con la stessa MF e il gruppo delle classi viene ordinato dalla completa appartenenza alla non appartenenza.

La MF di un set fuzzy discreto è:
$$\mu = \left\{ \frac{\mu_1}{x_1}, \frac{\mu_2}{x_2}, \dots, \frac{\mu_n}{x_n} \right\}$$

ATTENZIONE! La logica fuzzy e la probabilità sono concetti diversi. Entrambe si riferiscono a gradi di certezza relativi a un evento. Per la probabilità, però, il grado di certezza è significativo solo prima che l'evento associato ad essa accada: dopo, la probabilità non si applica più perché il risultato dell'evento è noto. Per la logica fuzzy, invece, il grado di certezza è rilevante anche dopo che un evento si è verificato.

Per l'unione, negli insiemi crisp ci si chiede quale elemento appartenga a uno degli insiemi; negli insiemi fuzzy ci si chiede quanto di un elemento è in uno degli insiemi. Le funzioni usate per l'unione di set fuzzy vengono dette t-conorms.

A t-conorm C is a function

$$z = C(a, b), 0 \leq a, b, z \leq 1,$$

having the following four properties:

1. $C(a, 0) = a$;
2. $C(a, b) = C(b, a)$;
3. if $b_1 \leq b_2$, then $C(a, b_1) \leq C(a, b_2)$;
4. $C(a, C(b, c)) = C(C(a, b), c)$.

The basic t-conorms are

standard union	$C_m(a, b) = \max(a, b)$
bounded sum	$C_b(a, b) = \min(1, a + b)$
algebraic sum	$C_p(a, b) = a + b - ab$
drastic union	$C^*(a, b) = \begin{cases} a & \text{if } b = 0 \\ b & \text{if } a = 0 \\ 1 & \text{otherwise} \end{cases}$

Nella pratica si usano coppie di T e C che sono duali. Si dice che T e C sono duali se:

$$T(a, b) = 1 - C(1 - a, 1 - b)$$

$$C(a, b) = 1 - T(1 - a, 1 - b)$$

Tra le funzioni descritte, quelle con lo stesso pedice sono duali.

Per il complementare, negli insiemi crisp ci si chiede quale elemento non appartenga al set; negli insiemi fuzzy ci si chiede quanto dell'elemento non appartenga al set. Il complementare standard fuzzy è dato da: $\mu_{Ac}(x) = 1 - \mu_A(x) \forall x \in X$

In general a fuzzy complement may be defined by a functional mapping c of the form:

1. $c(0) = 1$ and $c(1) = 0$;
2. $\mu_1 \leq \mu_2 \Rightarrow c(\mu_1) \geq c(\mu_2)$;

In most practical cases two additional requirements are considered:

3. c is a continuous function
4. $c[c(\mu)] = \mu \quad \forall \mu \in [0, 1]$

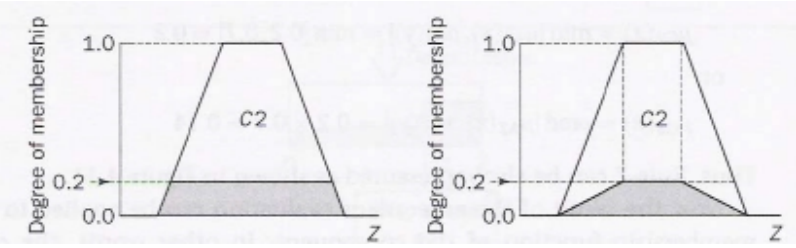
Sugeno class of complements

$$c_{Sug}(\mu) = \frac{1 - \mu}{1 + \lambda \mu} \quad \text{with } \lambda \in]-1, \infty[$$

Yager class of complements

$$c_{Yag}(\mu) = (1 - \mu^w)^{1/w} \quad \text{with } w \in]0, \infty[$$

Il metodo più comune per correlare la conseguenza di una regola al valore vero dell'antecedente è quello di tagliare semplicemente la M.F. della conseguenza. Questo metodo è chiamato *clipping* o minima correlazione. Dal momento che la parte superiore della M.F è tagliata, il set fuzzy perde un po' di informazione. Tuttavia questo metodo coinvolge matematica meno complessa e più veloce e genera un output che è più facile da defuzzificare. Lo *scaling* o prodotto di correlazione, invece, è un approccio migliore per mantenere la forma originale del set fuzzy.



La M.F originale della conseguenza viene aggiustata moltiplicando tutti i suoi gradi di membership per il vero valore dell'antecedente della regola. Con questo metodo si perde meno informazione.

In genere i sistemi fuzzy incorporano più di una regola. I risultati di ciascuna regola sono aggregati in un unico set fuzzy di output e nella maggior parte dei casi il fuzzy set risultante è defuzzificato in un singolo numero.

Esistono numerosi metodi di defuzzificazione, ma il più usato è la centroid technique, che trova il punto in cui una linea verticale dividerebbe il set in due masse uguali.

Il processo che parte da un input e ottiene un output usando la teoria degli insiemi fuzzy è detta *fuzzy interference*. Esistono diverse tecniche per farlo: una delle più usate è la *Mamdani inference*, il

gli elementi in modo casuale.

Se il training set e il test set coincidono, si ottengono le migliori prestazioni del classificatore.

Si può dividere un data set in due parti separate, una per il training e una per il test. Si può fare se le parti sono numericamente grandi, altrimenti il test perde significatività. Le classi devono essere rappresentate in modo equo. Le parti possono essere suddivise a metà, oppure 70% per il training e 30% per il test.

Se il numero di elementi del data set è troppo esiguo per essere diviso, si usa il metodo del *leave one out*. In questo caso tutti gli elementi eccetto uno vengono usati per il training set: si ripete l'operazione per ogni elemento. Infine si costruisce un'unica confusion matrix, legata a tutti i classificatori che si sono costruiti. Nei casi precedenti il classificatore rimane sempre lo stesso, mentre in questo varia: tuttavia si fa l'ipotesi che un singolo elemento non vada a modificare significativamente il classificatore.

Se il classificatore è costruito in supporto a un test di screening si usano le seguenti definizioni:

- sensitività → probabilità che un test sia positivo se la persona ha una malattia;
- specificità → probabilità che un test sia negativo se la persona non ha una malattia;
- falso negativo → una persona il cui test risulta negativo ma che in realtà è positiva;
- falso positivo → una persona il cui test risulta positivo ma che in realtà è negativa.

CLUSTERING

Il clustering spesso precede la classificazione. Per la classificazione, infatti, servono degli esempi già classificati: un elemento i del training set è caratterizzato dal vettore $\{x_i\}$ di variabili e dalla classe corretta C_i . Si vuole minimizzare l'errore tra C_i e C_{li} (che è il risultato del classificatore).

Se, però, non si conosce C_i , un elemento i è caratterizzato solo da $\{x_i\}$; non si può minimizzare l'errore perché manca il termine di confronto. In questo caso si usa il clustering per suddividere gli elementi in sottogruppi (detti cluster), che poi possono essere usati per estrarre conoscenza, per costruire un classificatore, ecc.

Quanti cluster servono? Non si conosce il prototipo di una classe, né il numero totale di classi. Bisogna capire qual è la suddivisione che dà il risultato migliore. Si parte ipotizzando un certo numero di cluster; in seguito con una regola si suddividono gli elementi del data set nei diversi cluster e iterativamente si costruisce un prototipo di ogni cluster e si riesegue la suddivisione. Si continua finché gli elementi non si spostano più da un cluster all'altro.

Per valutare la bontà ci sono 2 criteri: si deve avere una suddivisione in gruppi che siano il più possibile diversi tra loro e che al loro interno abbiano elementi il più uguali possibile.

Min(variabilità intra cluster)

MAX(distanza tra cluster)

Un metodo di clustering è caratterizzato da:

- misura di similarità: come si confrontano gli elementi?
- Regola di classificazione.

Per costruire i prototipi da un'iterazione all'altra si comincia con l'inizializzazione (spesso di tipo random), poi c'è bisogno di un metodo per l'aggiornamento del prototipo.

L'approccio è iterativo e si può riassumere in 2 fasi:

1. trovare una ragionevole partizione iniziale;
2. spostare i campioni da un cluster all'altro.

L'algoritmo termina quando non ci sono più spostamenti. Non si ha la certezza di trovare la miglior soluzione in assoluto perché l'approccio è di tipo euristico.

Gli algoritmi possono essere:

- flat: lavorano andando a suddividere i campioni in gruppi disgiunti tra loro;
- gerarchici: i cluster hanno una dipendenza gerarchica.

RETI NEURALI

Le reti neurali fanno parte della CI e delle metaeuristiche. Prendono spunto dai meccanismi di ragionamento del cervello, al quale assomigliano per due aspetti:

- la conoscenza è acquisita dalla rete attraverso un processo di apprendimento;
- la conoscenza viene immagazzinata mediante pesi sinaptici nelle connessioni tra neuroni.

Vengono usate prevalentemente per classificazione e clustering, ma anche per la stima o il riconoscimento di un'immagine modificata rispetto a un template originale. Sono lo strumento legato all'apprendimento per eccellenza. Subiscono una fase in cui apprendono dati (fase di training: c'è bisogno di un training set) e poi una fase di uso. Tra le due fasi c'è la validazione (c'è bisogno di un test set).

Una rete neurale si comporta come una “scatola nera” perché non si possono vedere le regole sulle quali si basa, a differenza dei classificatori fuzzy. Per questo motivo in passato venivano criticate, ma adesso hanno migliaia di applicazioni e hanno prestazioni molto buone. Tuttavia possono sbagliare e noi non abbiamo elementi per capire quando questo accada. Per ridurre al minimo la probabilità di errore bisogna usare più reti neurali.

I principali elementi di una rete neurale sono i *pesi*.

Una rete neurale è un *grafo* formato da nodi detti neuroni artificiali, che rappresentano la parte della rete in cui avviene il calcolo (elaborazione dati). L'elaborazione e la conformazione del grafo sono diverse a seconda del tipo di rete. A ciascuno degli archi (collegamenti tra neuroni) è associato un peso. L'apprendimento consiste nell'andare a modificare i pesi iterativamente finché non variano più. Ciò che fa variare i pesi dipende dal tipo di rete.

Le reti neurali possono essere:

- *supervisionate*: usate per problemi di classificazione, hanno bisogno di un training set con elementi con classificazione associata. I pesi vengono modificati calcolando l'errore tra l'organizzazione della rete e la classificazione corretta. Le loro prestazioni vengono valutate mediante una confusion matrix;
- *non supervisionate*: usate per il clustering, hanno un training set costituito da elementi senza classe associata. I pesi vengono modificati in base a distanza e variabilità, come nel clustering. La validazione varia a seconda del tipo di rete: in certi casi i pesi sono i prototipi delle classi, quindi si ragiona su quelli.

Il calcolo avviene in parallelo. Le reti supervisionate sono divise in layer. Ciascun neurone appartenente a un dato layer è collegato solo a neuroni del layer successivo. Il layer di input ha tanti neuroni quante sono le variabili. Quando si classifica un elemento, si danno in input i valori delle variabili a ciascun neurone, che poi lavora in modo totalmente indipendente dagli altri. Ogni neurone avrà un output che diventerà l'input dei neuroni dello strato successivo.

Per costruire una rete neurale bisogna definire:

- architettura (grafo);
- neurone (come elabora le informazioni?);
- algoritmo di apprendimento.

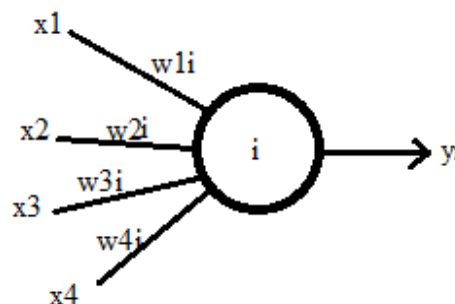
I 3 elementi non sono indipendenti e sono ugualmente importanti nel definire la rete neurale.


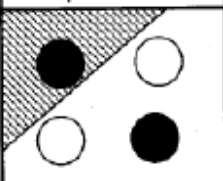

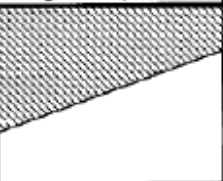
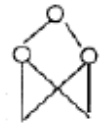
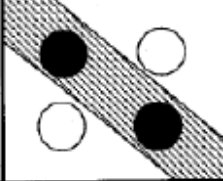



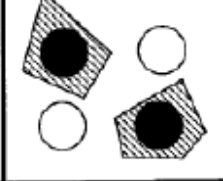


Nel neurone usato più frequentemente per le reti supervisionate, l'output è funzione di pesi e input:

$$y = f(w, x)$$

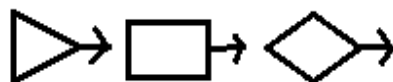
$$x = \{x_1, x_2, x_3, x_4\}$$

$$w = \{w_{1i}, w_{2i}, w_{3i}, w_{4i}\}$$



Structure	Description of decision regions	Exclusive-OR problem	Classes with meshed regions	General region shapes
 Single layer	Half plane bounded by hyperplane			
 Two layer	Arbitrary (complexity limited by number of hidden units)			
 Three layer	Arbitrary (complexity limited by number of hidden units)			

Le possibili forme dei layer nascosti sono numerose e non è facile scegliere a priori.



Le reti non supervisionate hanno una struttura a grafo con neuroni non collegati tra loro.

Competitive learning è un algoritmo di apprendimento in cui ciascun neurone compete con gli altri per vincere. Per ciascun neurone si calcola la differenza tra i pesi e gli input:

$$d = \sum |w_j - x_j|$$

In seguito si cerca la differenza minore. Il neurone con la differenza minore è il vincitore: i suoi pesi vengono aggiornati, sommando o sottraendo una quantità in funzione di d . I pesi degli altri neuroni non vengono cambiati. In questo modo, se arriva un input simile sarà sempre lo stesso neurone a vincere. I pesi vengono usati come centroidi (prototipi) di tanti cluster quanti sono i neuroni. All'inizio i pesi sono assegnati in modo random.

Una variante del competitive learning sono le **mappe di Kohonen**. Si tratta di reti neurali basate sul meccanismo del competitive learning, ma con in più un rapporto di vicinanza tra i neuroni. Le mappe sono quadrate o esagonali. Quando un neurone vince, vengono aggiornati anche i suoi vicini.

APPROFONDIMENTI

Box plot

In statistica il **box-plot**, detto anche **box and whiskers plot** (*diagramma a scatola e baffi*) o semplicemente **boxplot**, è una rappresentazione grafica utilizzata per descrivere la distribuzione di un campione tramite semplici indici di dispersione e di posizione.

Viene rappresentato (orientato orizzontalmente o verticalmente) tramite un rettangolo diviso in due parti, da cui escono due segmenti. Il rettangolo (la "scatola") è delimitato dal primo e dal terzo quartile, $q_{1/4}$ e $q_{3/4}$, e diviso al suo interno dalla mediana, $q_{1/2}$. I segmenti (i "baffi") sono delimitati dal minimo e dal massimo dei valori.

In questo modo vengono rappresentati graficamente i quattro intervalli ugualmente popolati delimitati dai quartili.

RIASSUNTI CLASSIFICAZIONE

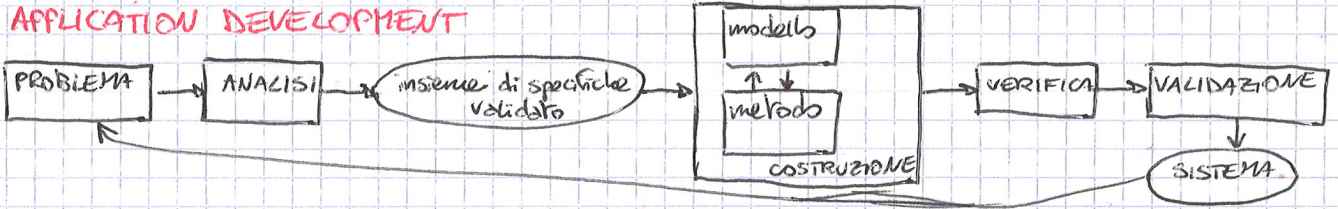
①

Medicina basata sulle evidenze → i professionisti devono avere accesso e comprendere info mediche.
 → bisogna migliorare performance degli algoritmi che processano i segnali

Soluzione di un problema → approcci diversi
 ↳ ogni approccio corrisponde a un set di metodi
 ↳ ogni metodo corrisponde a campi diversi

Per una buona analisi dei dati: non serve complessità matematica, ma ragionevolezza e conoscenza di cosa rappresentino le formule.

APPLICATION DEVELOPMENT



STATISTICA

↳ descrittiva: non ha problemi di numerosità del campione
 ↳ inferenziale: si fanno inferenze su come il campione rappresenta l'intera popolazione.

POPOLAZIONE → include tutti gli oggetti di interesse. Parametri indicati con lettere greche (valori veri)
CAMPIONE → porzione di popolazione. Parametri indicati con lettere latine (stima)

Metodi per estrarre il campione:

- campionamento random
- campionamento sistematico (1 campione ogni n)
- convenience sampling (si inserisce nel campione chi è disponibile)
- cluster sampling (gruppi in base a geografia, e campionamento gruppo)
- campionamento stratificato (gruppi in base a caratteristiche, 1 campione v gruppo)

Parametri calcolati su campione: • valor medio: $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
 • mediana
 • moda (valore assunto più frequentemente)

Misure di variabilità: • varianza: indica quanto è dispersa la popolazione. Dipende da u.d.m.!

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

- coeff. di variazione: adimensionale. $\frac{s}{\bar{x}} \cdot 100$
- range: [min, max].

Suddivisione valori: - percentili: ogni classe contiene lo stesso numero di valori. Calcolo del p-esimo percentile:
 • se $n^{\circ} \text{ pop} / 100$ è un intero, è la media tra le osservazioni più grandi di $n^{\circ} \text{ pop} / 100$ e cercare su internet → $n^{\circ} \text{ pop} / (100 + 1)$
 • se $n^{\circ} \text{ pop} / 100$ non è intero, è il (k+1)esimo punto del campione (dove k è il più grande intero minore di $n^{\circ} \text{ pop} / 100$).
 • distribuzione delle frequenze: mostra ogni valore del data set insieme alla sua frequenza (quante volte si ripete).

Per distribuzioni gaussiane: • skewness: misura di simmetria. $\text{skewness} = \frac{1}{(n-1)s^3} \sum_{i=1}^n (x_i - \bar{x})^3$

• Kurtosis: basso kurtosis ⇒ cima piatta.

$$\text{Kurtosis} = \frac{1}{(n-1)s^4} \sum_{i=1}^n (x_i - \bar{x})^4$$

Rappresentazione dati: ↳ bar diagram (frequenza)
 ↳ scatter plot (x-y)
 ↳ box plot (rettangolo con 50% dei dati.)

Probabilità: $Pr(A)$ = numero di eventi favorevoli / numero totale di eventi possibili

Probabilità condizionata: $Pr(A/B)$ = probabilità che, dato B, accada A

Variabili casuali: i possibili valori sono risultati numerici di un fenomeno casuale.

Parametri che valutano il comportamento congiunto di variabili:

- covarianza: $Cov(X,Y)$ misura il grado in cui i cambiamenti di una variabile sono associati a quelli di un'altra.

- coeff. di correlazione:

variabili inversamente
 le proporzionali

$$-1 \leq \rho = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \leq 1$$

→ variabili direttamente
 proporzionali

$\rho = 0 \Rightarrow$ variabili
 indipendenti.

FEATURES SELECTION: si fa perché spesso si hanno molte variabili a disposizione per descrivere un campione, ma non tutte sono utili. (2)

CLASSIFICAZIONE

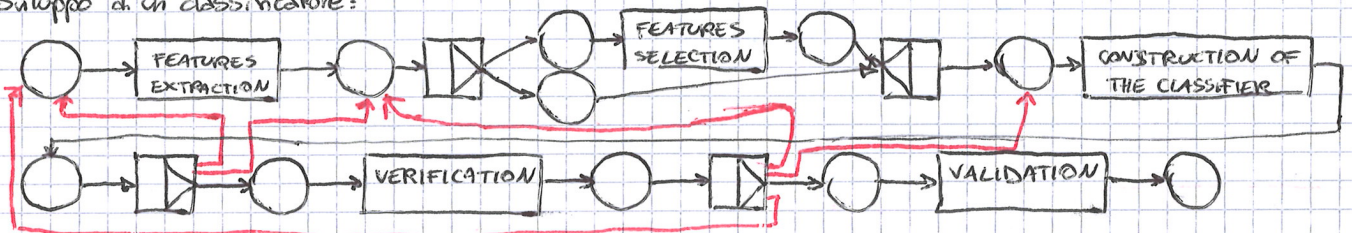
Dato un certo numero di classi e un elemento, la classificazione consiste nell'assegnare tale elemento a una delle classi. Spesso l'elemento è associato univocamente a una classe, ma può anche appartenere a più classi con un certo grado di appartenenza (logica fuzzy).

Ogni metodo di classificazione è caratterizzato da: - prototipi della classi (SET DI FEATURES CHE RAPPRESENTA LA CLASSE) - misura di similarità - regola di classificazione

Prototipo → elemento rappresentativo degli altri elementi appartenenti alla classe. Ha un valore associato. Le regole di classificazione si basano sul confronto fra il prototipo e l'elemento in base alla misura di similarità, che può essere di diversi tipi:

- distanza euclidea: $d(x,y) = \|x-y\| = \sqrt{\sum_i (x_i - y_i)^2}$
- norma L2: $d(x,y) = \|x-y\| = \sum_i |x_i - y_i|$
- norma Loo: $d(x,y) = \max_k |x_i - y_i|$
- stringhe
- insiemi

Sviluppo di un classificatore:



Quando si identificano le features si possono usare i dati così come sono oppure applicarvi algoritmi per ridurre il numero di features:

- construction: si costruiscono nuove caratteristiche. Si perde contatto diretto con la fase del problema.
- selection: si seleziona un sottoinsieme di caratteristiche, rimuovendo quelle inutili o ridondanti.

scopi della features selection → diminuire complessità del modello
 → migliorare performance del sistema

- Benefici features selection:
- facilita l'utilizzazione dati
 - riduce la richiesta di misure e lo spazio per salvarle
 - riduce i tempi per usare il sistema e per imparare a usarlo
 - migliora la previsione della performance

Una caratteristica fortemente relevante non può essere rimossa senza perdere accuratezza nella previsione. Una caratteristica debolmente rilevante può contribuire all'accuratezza del sistema, ma dipende da quali altre features vengono considerate. Una caratteristica può essere irrilevante da sola, ma rilevante con altre.

la selezione può essere supervised o unsupervised e si può fare in 2 modi:

- si ordinano le caratteristiche secondo un certo criterio e si selezionano le prime k
- si seleziona un sottoinsieme di features senza valutare la determinazione della performance

Aspetti della features selection:

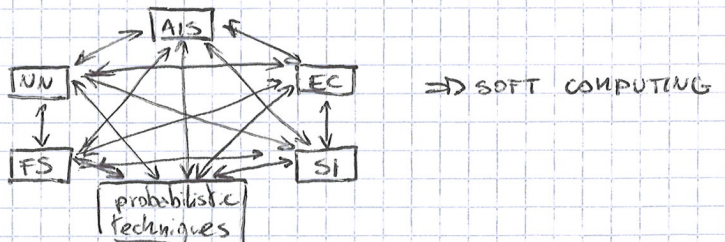
- modelli → filtraggio (scelta con regole indipendenti da classificatore)
 ↳ wrapper (scelta con regole dipendenti da classificatore)
 ↳ embedded (la features selection è fittino con la costruzione del classificatore)
- strategie per selezione → back word (si tolgono features finché prestazioni classificatore peggiorano)
 ↳ forward (si aggiungono features finché prestazioni smettono di migliorare)
 ↳ randomized (si costruiscono sottoinsiemi di features e si cerca di trovare combinazione migliore)
- valutazione → si compara prima e dopo features selection
 ↳ si comparano 2 algoritmi di selezione

Deve essere: training set per selezione caratteristiche ≠ training set per addestrare classificatore. Se no correlazione!

COMPUTATIONAL INTELLIGENCE

Riunisce metodi che prendano spunto da meccanismi adattivi che regolano l'ambiente. Molti metodi non sono deterministici. Sono 5:

- evolutionary computing
- neural networks
- swarm intelligence
- fuzzy systems
- artificial immune systems



performance GA influenzata da $\begin{cases} \rightarrow \text{dimensioni popolazione} \\ \rightarrow P_m \\ \rightarrow P_c \end{cases}$

METODI EURISTICI: si usano quando le soluzioni utilizzabili sono molto numerose. Non c'è certezza di arrivare a un ottimo.

OTTIMO GLOBALE: ha funzione obiettivo migliore di tutte le soluzioni nello spazio di ricerca
 $\forall s \in S, f(s) \leq f(s')$

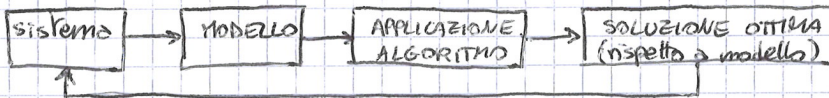
La complessità temporale di un algoritmo è il numero di step richiesti per risolvere un problema di dimensione n .
 Gli algoritmi possono concentrarsi sull'arrivare a un ottimo (anche locale) oppure avere l'obiettivo di esplorare un largo numero di soluzioni. Combinando i 2 scopi si ottengono buone soluzioni.

I metodi euristici possono essere:

- ispirati alla natura / non ispirati alla natura
- con uso di memoria / senza uso di memoria
- deterministici / stocastici
- basati su popolazione / basati su singola soluzione
- iterativi / greedy

Le euristiche basate sulla ricerca locale cercano una soluzione migliore nel vicinato della soluzione corrente. Vicinato: insieme delle soluzioni raggiungibili dalla soluzione corrente attraverso un'operazione elementare. Nei GA sono le soluzioni che si ottengono applicando gli operatori genetici.

U.B: si sta ottimizzando un modello del sistema reale!



Passi di una generica ricerca locale:

- 1 **inizializzazione** → si sceglie una soluzione ammissibile x_1 e se ne calcola la funzione obiettivo;
- 2 **generazione del vicinato** → si seleziona un vicinato $N(x_i)$ e si sceglie x_{i+1} appartenente ad esso;
- 3 **test di accettazione** → si verifica che x_{i+1} possa essere accettata; se sì si assume come soluzione corrente;
- 4 **test di terminazione** → se è positivo la procedura è terminata, se no si torna a 2

Metodi:

- steepest descent: deterministico
- simulated annealing: stocastico. Scollinamento
- Tabu search: deterministico a singola soluzione. (memoria a breve termine: tabu list) pag 18 **ASPIRACION CRITERIA**
 - ↳ intensificazione (memoria a medio termine): ricorda soluzioni migliori
 - ↳ diversificazione (memoria a lungo termine): ricorda soluzioni visitate
 - ↳ restart diversification: si introducano componenti meno visitate
 - ↳ continuous diversification: si introducano errori
 - ↳ strategic oscillation: considera e penalizza soluzioni non fattibili.

Altri metodi di ottimizzazione sfruttano la **SWARM INTELLIGENCE**, algoritmi ispirati al comportamento collettivo di alcune specie. Caratteristici due particolari:

- sono agenti semplici e non sofisticati
- cooperano con un mezzo di comunicazione indiretto
- si muovono nello spazio di decisione

Ant Colony Optimization (ACO): parte con l'inizializzazione del ferormone, poi si iterano 2 step principali: costruzione della soluzione e aggiornamento del ferormone.

Le formiche artificiali sono procedure greedy stocastiche che costruiscono la soluzione tenendo conto di percorsi di ferormone e informazioni euristiche dipendenti dal problema.

Aggiornamento ferormone → evaporazione (favorisce maggiori movimenti casuali)
 ↳ rinforzo (quando la fitness è buona)

Molti problemi di ottimizzazione sono **vincolati**. Gestione vincoli:

- strategie che agiscono sulla rappresentazione delle soluzioni o della funzione obiettivo
 - ↳ reject strategies: si tengono solo soluzioni fattibili
 - ↳ penalizing strategies: si tiene conto di soluzioni infattibili con penalità
 - ↳ repairing strategies: algoritmi euristici trasformano soluzione infattibile in soluzione fattibile
 - ↳ preserving strategies: ci si assicura la generazione di una soluzione fattibile
- strategie non direttamente correlate alla rappresentazione delle soluzioni o della funzione obiettivo

CONCETTUALIZZAZIONE

A partire dal problema, bisogna trovare il numero minimo di caratteristiche che danno dei buoni risultati.

- Quali dati si hanno a disposizione?
- Qual è il peso delle incertezze?
- Che approccio si può usare?

CLUSTERING

serve a suddividere gli elementi in sottogruppi. Non si conoscono i prototipi delle classi né il numero totale di classi. ④

Si ipotizza un n° di cluster, poi si suddividono gli elementi nei cluster con una regola e iterativamente si costruisce il prototipo di ogni cluster. Si continua finché gli elementi non si spostano più.

Per valutare bontà: - **min** variabilità intra cluster
- **max** distanza tra cluster

metodo di clustering caratterizzato da $\left\{ \begin{array}{l} \text{misura di similarità} \\ \text{regola di classificazione} \end{array} \right.$

Non si ha la certezza di trovare la miglior soluzione perché l'approccio è euristico. Gli algoritmi possono essere:
- flat (K-means, ISODATA)
- gerarchici (dendrogramma)

K-means: per ogni cluster si costruisce un elemento che ne rappresenta il valor medio. Si assegna ogni prototipo al cluster con la media più vicina.

Processo: ① definizione n° cluster
② applicazione K-means
③ if (varianza) \geq soglia \Rightarrow n+1
④ if (distanza) \leq soglia \Rightarrow n-1
⑤ if n invariato \Rightarrow stop; else, si torna a ②

Processo: ① definizione n° clusters
② inizializzazione clusters $\left\{ \begin{array}{l} \text{assegnazione arbitraria} \\ \text{di esempi ai clusters} \end{array} \right.$
③ calcolo media cluster (alcuni esempi usati come centri)
④ si assegna ogni es al cluster con media più vicina

Problema: si deve definire a priori il numero di cluster. Per risolvere si può: ⑤ STOP o ③

- riclassificare con un numero diverso di cluster
- post-elaborare il risultato unendo i cluster simili
- usare ISODATA

ISODATA: (Iterative Self-Organizing Data Analysis Technique Algorithm) seleziona automaticamente n° cluster.

L'utente definisce: - n° min elementi per cluster
- n° n° cluster desiderato
- **max** valore di varianza per divisione
- **min** distanza per unione
- **max** n° cluster che possono essere uniti.

Dendrogramma: si uniscono elementi a 2 a 2. Si taglia quando distanza tra generazione n e n-1 è molto maggiore rispetto alle altre.

La clusterizzazione va validata!

RETI NEURALI

Fanno parte di CI e metaeuristiche e prendono spunto dal cervello:

- conoscenza acquisita mediante un processo di apprendimento
- conoscenza immagazzinata mediante pesi sinaptici nelle connessioni tra neuroni.

Sono usate soprattutto per classificazione e clustering.

Non si possono vedere le regole su cui si basa la rete! \rightarrow uso di più reti per ridurre probabilità errori.

Una rete neurale è un **grafo** formato da nodi (**neuroni artificiali**), in cui avviene il calcolo. A ciascuno degli archi (collegamenti tra neuroni) è associato un **peso**. L'apprendimento avviene andando a modificare iterativamente i pesi.

costruzione rete neurale $\left\{ \begin{array}{l} \text{architettura (grafo)} \\ \text{neurone (come elabora informazioni)} \\ \text{algoritmo di apprendimento} \end{array} \right.$

	RETI NEURALI SUPERVISIONATE	RETI NEURALI NON SUPERVISIONATE
CARATTERISTICHE	training set i cui elementi hanno associata la classificazione	training set i cui elementi non hanno associata la classificazione
USO	classificazione	clustering
MODIFICA PESI	in base a errore tra organizzazione della rete e classificazione vera	in base a distanza e variabilità
STRUTTURA	a più layers	a singolo layer
VALIDAZIONE	tramite confusion matrix	vane a seconda del tipo di rete