

Sistemi a classificatori

Sistemi a classificatori

- **Corso di laurea in Informatica**
(anno accademico 2024/2025)

- Insegnamento: Apprendimento ed evoluzione in sistemi artificiali
- Docente: Marco Villani

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



Dipartimento di
Scienze Fisiche,
Informatiche
e Matematiche


E' vietata la copia e la riproduzione dei contenuti e immagini in qualsiasi forma. E' inoltre vietata la redistribuzione e la pubblicazione dei contenuti e immagini non autorizzata espressamente dall'autore o dall'Università di Modena e Reggio Emilia

Ispirazione biologica

- Nelle reti neurali l'ispirazione proviene dalla biologia
 - neuroni simili ai neuroni biologici
 - metodi di apprendimento ispirati anch'essi (a volte) dalla biologia, p.es. regola di Hebb, mexican hat ...
- un altro filone di sistemi capaci di apprendere è anch'esso ispirato dalla biologia: evoluzione biologica
- vi sono altre possibili fonti di ispirazione:
 - sistema immunitario
 - colonie di insetti
 - ...

Introduzione ai sistemi a classificatori

- Motivazioni e cenni storici
- Architettura dei CS
- Le componenti: messaggi e classificatori
- Apprendimento a breve termine : bucket brigade
- Apprendimento a lungo termine: algoritmi genetici

- 
- All'inizio dell'IA si era enfatizzata la capacità di risolvere problemi generali
 - ragionamento, euristiche generali, etc.
 - In seguito la ricerca si è orientata verso un uso sempre maggiore di conoscenza specifica del dominio
 - per ridurre il numero di alternative ad ogni passo



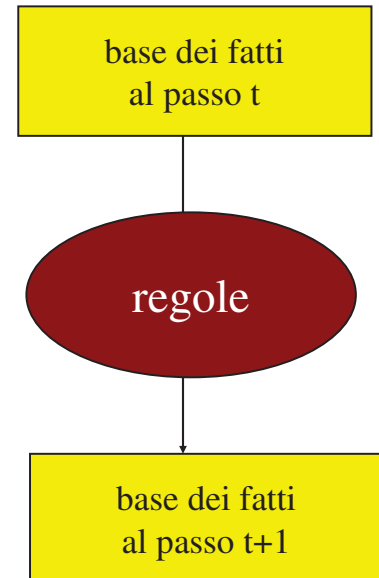
I primi sistemi esperti

- L'importanza della conoscenza specifica
 - rispetto ai metodi generali di ragionamento
- quindi l'importanza della rappresentazione della conoscenza
- la comparsa (e il successo) dei primi sistemi esperti alla fine degli anni 70
 - Mycin
 - Dendral
 - Prospector
 - R1

- **Rappresentazione della conoscenza** mediante regole del tipo

SE condizione ALLORA azione

- SE canta e vola ALLORA è un uccello
- Quando la base dei fatti contiene informazioni che soddisfano la condizione, la regola può essere applicata
canta e vola
- la parte “azione” può aggiungere nuova conoscenza alla base dei fatti
è un uccello



- La **conoscenza** è composta da
 - regole
 - fatti specifici relativi al caso in questione
- il motore inferenziale determina quale regola applicare ad ogni passo
- la base di conoscenza e il motore inferenziale costituiscono il cuore (o la mente) del sistema
- Vi è **separazione fra la conoscenza ed il suo uso**
 - l'ingegnere della conoscenza può concentrarsi sul primo aspetto
 - il concatenamento delle inferenze è lasciato alla macchina inferenziale
- è necessaria una interfaccia che acquisisca le informazioni necessarie
 - ed eventualmente le richieda

Mycin

- **Diagnosi** di malattie infettive
- la conoscenza è rappresentata mediante regole
- Esempio:
 - SE
 - L'organismo è gram-positivo e
 - la morfologia dell'organismo è cocco e
 - la conformazione di crescita dell'organismo è “a blocchi”
 - ALLORA
 - L'organismo potrebbe essere uno stafilococco

- **Mycin** si basa sui dati di culture dei microorganismi e su informazioni sullo stato del paziente (febbre, etc.)
- lavora ipotizzando una possibile diagnosi e, mediante backward chaining in profondità, cerca le informazioni a supporto della ipotesi
- utilizza un meccanismo basato su fattori di certezza per dare un peso alle varie catene di inferenze
- confronta i pesi complessivi e propone la diagnosi che ha maggiore supporto
- buoni risultati nel confronto coi medici dell'università di Stanford

I primi sistemi esperti

- Mycin
 - buoni risultati nel confronto con medici esperti
- Prospector
 - scoperta di un nuovo giacimento
- Dendral
 - analisi di composti mediante spettrometria di massa
- R1
 - usato per la configurazione dei computer DEC-Vax

Le speranze

- I programmi tradizionali sono **rigidi** e **difficili da mantenere**
- i sistemi esperti sono molto più efficaci e “naturali” da comprendere
 - forniscono prestazioni umane, non sovrumane
- Grande **interesse applicativo**: il computer avrebbe risolto il problema della gestione delle conoscenze come aveva risolto quelli del calcolo numerico, della contabilità e della gestione dei magazzini
 - standardizzazione delle conoscenze in una azienda
 - disponibilità ubiquitaria di competenze di alto livello
 - sopravvivenza delle conoscenze al trasferimento o pensionamento dell'esperto

I primi entusiasmi (prima metà anni '80)

- I primi successi crearono **grandi attese**
- Nacquero le “shell” o gusci, costituiti da interfacce veramente avanzate per l'epoca e da motori inferenziali
- Si diffuse la convinzione (erronea!) che sviluppare sistemi esperti potesse essere semplice e veloce
- Sarebbe stato sufficiente “riempire” una base di conoscenze mediante un insieme di regole ottenute da un “ingegnere della conoscenza” attraverso un dialogo con un esperto umano
 - le conoscenze avrebbero tenuto conto anche delle euristiche esperienziali

L'inverno dell'AI (fine anni '80 - metà anni 90)

- **Sottovalutazione dei problemi** legati alla acquisizione e alla rappresentazione formale delle conoscenze
 - formalismi innaturali
 - natura sfumata di alcuni tipi di conoscenze
- **Tempo** necessario per sviluppare un sistema esperto
- **Dinamicità** delle conoscenze
 - difficoltà di manutenzione
- Necessità di integrazione col resto del sistema informativo
- **Assenza di prestazioni sovrumane**
- **Improvvisazione** (dei programmatori)
- **Overselling** (dei venditori)

La ripresa

- Le aziende che hanno saputo investire su alcune **applicazioni ben meditate** hanno ottenuto risultati dimostrabili
- Si è capito che i sistemi esperti possono funzionare egregiamente, ma che **non ci sono scorciatoie**
- L'enfasi crescente sull'aumento di efficienza fornisce stimoli importanti alla adozione estesa di sistemi di questo genere
 - time to market
 - riduzione del personale
- Tendenza allo sviluppo di sistemi per la gestione delle conoscenze aziendali

I nuovi entusiasmi

- Le **reti neurali**, come vedremo in breve, rappresentano un approccio alternativo a quello simbolico
 - O complementare?
- La crescita esplosiva
 - Delle capacità computazionali (schede grafiche, calcolo parallelo)
 - Degli esempi disponibili (Internet)
- Consente oggi di ottenere prestazioni eccellenti in molti settori
- Le reti neurali **rimangono tuttavia "oscuri"**, non adatte a "spiegare" i motivi delle loro conclusioni

Apprendimento automatico

- Esistono anche metodi di apprendimento simbolico che “ragionano” esplicitamente sugli esempi
 - vulnerabili alla **presenza di difetti, rumore e contraddizioni** nell'insieme di esempi
- Può essere interessante combinare le proprietà di auto-organizzazione di metodi dinamici con le capacità di spiegazione e di concatenazione di inferenze dei metodi simbolici

■ Sistemi a classificatori

Cosa sono i sistemi a classificatori (CS)

- Sistemi che **apprendono** a svolgere un compito **interagendo con un ambiente** parzialmente ignoto, utilizzando **meccanismi di feedback** per guidare un **processo evolutivo interno** che modifica il proprio **modello del mondo** (basato su **regole**)
- Apprendimento da esempi (come NN)
- I sistemi a classificatori sono particolarmente interessanti perché combinano aspetti simbolici e meccanismi di auto-organizzazione dinamica

Caratteristiche dei CS

- **Aspetti simbolici:** **rappresentazione esplicita della conoscenza**
 - ✓ basata su regole
- **Aspetti sub-simbolici:**
 - le **regole esistenti vengono valutate** in funzione del loro apporto al funzionamento del sistema
 - ✓ non sono oggetto di ragionamento esplicito
 - le **nuove regole vengono generate** mediante variazione e ricombinazione casuale di regole esistenti
 - ✓ mediante algoritmi genetici
- **La compresenza di questi due aspetti** rende i CS estremamente interessanti dal punto di vista teorico

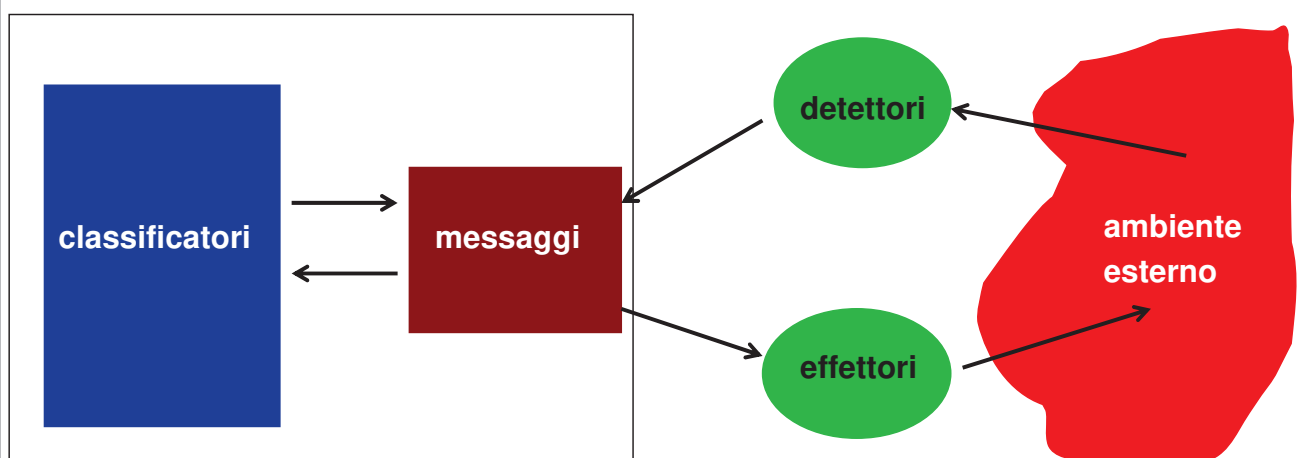
Cenni storici

- Introdotti da John Holland negli anni '80
- **Simulazione dei processi di apprendimento induttivo** mediante metodi ispirati dalle scienze biologiche (**algoritmi genetici**)
- Fanno inoltre ricorso ad un **meccanismo di valutazione** dell'apporto delle regole al funzionamento del sistema basato su una metafora economica
- Hanno destato un **notevole interesse teorico**
- Inizialmente, modesti successi applicativi
- **Alcune applicazioni di successo** di sistemi ispirati ai CS negli ultimi anni hanno riaccessato l'interesse anche dal punto di vista applicativo per questo filone di ricerca

Compiti che possono essere affrontati da un CS

- **Classificare** una serie di casi
 - diagnosi
- **Controllare** un apparato tecnologico
 - p.es. gasdotto
- Imparare a **muoversi in un ambiente** (robot autonomi)
- Inventare manovre di combattimento fra aerei
- **Sistemi multiagente**: i CS sono alla base di modelli di agenti autonomi interagenti, p.es. nella simulazione di sistemi economici
 - **capacità di sviluppare autonomamente “regole di comportamento”** che cambiano nel tempo

Schema dell'architettura di un CS



Schema dell' architettura di un CS (Michigan approach)

Classificatori

- SE UN MESSAGGIO PRESENTE NELLA LISTA DEI MESSAGGI SODDISFA LA **condizione**
- ALLORA CERCA DI IMPOSTARE IL MESSAGGIO **azione**
 - SE **allatta i piccoli** ALLORA **è un mammifero**
 - **allatta i piccoli**
- La parte “azione” può consistere
 - in **nuova conoscenza** che viene aggiunta alla lista dei messaggi
 - oppure in un **messaggio** per gli effettori
- Più classificatori possono agire simultaneamente, impostando diversi messaggi
 - è **ammessa la presenza di messaggi contraddittori** nella ML, ma nel caso di messaggi contraddittori per gli effettori uno solo viene scelto

Evoluzione dei sistemi a classificatori

- **Dinamica rapida**: un sistema di **apprendimento con rinforzo**, che assegna **un valore** alle diverse regole, sulla base del loro contributo al buon funzionamento del sistema
- **Dinamica lenta**: un sistema di **generazione di nuove regole**, basato su **algoritmi genetici**, che elimina le regole meno utili e le sostituisce con combinazioni e variazioni di quelle più utili

Il ciclo dei CS (a regole fisse) (1)

Numero_di_iterazioni = 1

WHILE (Numero_di_iterazioni <= Massimo)

{ alla lista dei messaggi (impostati al passo precedente) si
aggiungono eventuali messaggi di input dai **detettori**;

si confrontano i messaggi con le condizioni di tutti i classificatori;

i classificatori le cui condizioni sono soddisfatte competono per
poter impostare i loro messaggi;

i classificatori vittoriosi impostano e “ripagano” quelli che hanno
impostato i messaggi che hanno consentito loro di attivarsi;

Il ciclo dei CS (a regole fisse) (2)

la vecchia lista dei messaggi viene cancellata e sostituita dai
messaggi impostati dai classificatori vincenti;

gli effettori verificano se vi sono messaggi di output sulla lista;
in caso affermativo, effettuano l'azione corrispondente (dopo
aver risolto eventuali conflitti);

i classificatori che hanno impostato i messaggi di output
vengono ripagati dall'esterno (premio/punizione);

Numero_di_iterazioni = Numero_di_iterazioni + 1;

}

Messaggi e classificatori

- i messaggi sono stringhe binarie, composte da un numero fisso L di elementi appartenenti all'alfabeto $\{0,1\}$

|1|0|0|1|1|

- i classificatori sono definiti da
- una parte “condizione”: una stringa di L simboli $\in \{0,1,\#\}$:

|1|0|**#**|**#**|1|

- una parte “azione”: una stringa di L simboli $\in \{0,1,*\}$:

|0|0|0|*|1|

|1|0|**#**|**#**|1| | |0|0|0|*|1|

- una variabile reale s (la “forza”)

Condizioni di match e nuovi messaggi

- Un classificatore fa match con un messaggio se la sua parte condizione coincide col messaggio in tutte le posizioni in cui non c'è il simbolo $\#$ (don't care)

- p.es. unico messaggio presente nella lista dei messaggi : |1|0|0|1|1|

- |1|0|0|1|1| | |0|0|0|1|1| → |0|0|0|1|1|

- |1|0|1|1|1| | |0|0|0|1|1| → no match, nessun nuovo messaggio

- |1|0|**#**|1|1| | |0|0|0|1|1| → |0|0|0|1|1|

- |1|0|**#**|**#**|**#**| | |0|1|*|*|*| → |0|1|0|1|1|

- il simbolo $*$ significa “pass through” (lascia filtrare il valore)

Competizione fra classificatori attivabili

- La **specificità di un classificatore** è definita come la frazione di elementi della condizione diversi da “#”
 - sia λ il numero di # nella parte di condizione
 - $\gamma = (L - \lambda) / L$
- La **competizione** è basata su una **funzione (bid)** della specificità e della forza del classificatore: $b_i = b_i(s_i)$

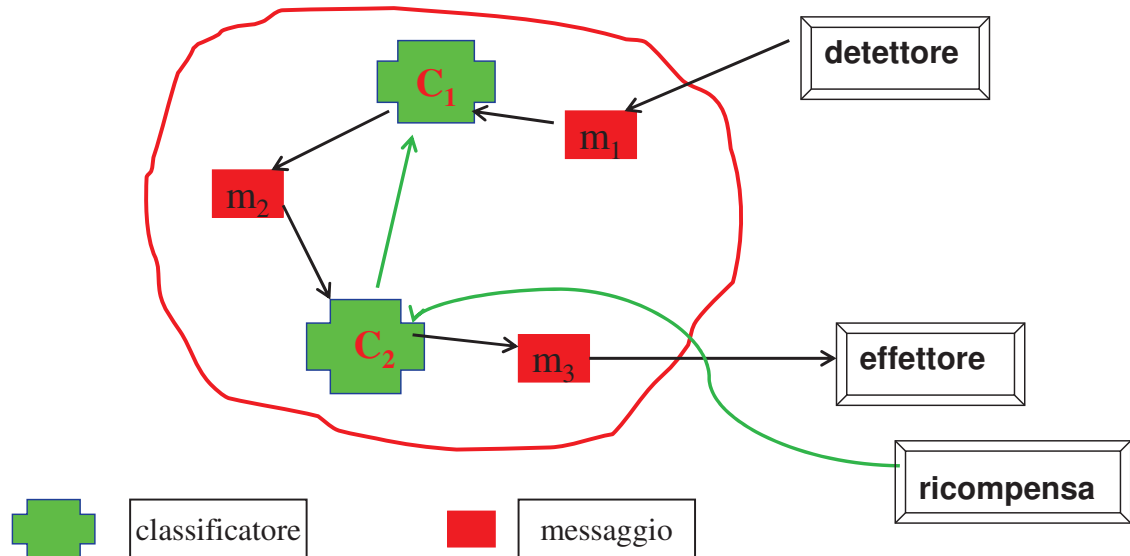
$$b_i = \gamma_i s_i$$

- dove γ_i è la specificità del classificatore i-esimo ed s_i la sua forza
- La **scelta dei vincitori** viene effettuata in maniera probabilistica
 - la probabilità di selezione è una funzione crescente del bid (tipicamente, direttamente proporzionale al bid)
 - nel caso di **messaggi contraddittori** per gli effettori, la scelta è anch'essa dipendente dal bid

Pagamenti

- La **forza di un classificatore** è una misura della sua utilità dimostrata per il funzionamento del sistema
 - forza elevata -> bid elevato -> elevata possibilità di impostare
- La forza dei classificatori che impostano messaggi di output viene modificata direttamente dall'esterno
 - ricompensa (aumento della forza) o punizione
- è necessario ricompensare anche le regole a monte di quella che ha fornito un output corretto
 - mantenere memoria di tutte le concatenazioni di regole sarebbe proibitivo
 - si introduce un meccanismo “locale”, che coinvolge solo i classificatori che si attivano in due istanti successivi
- L'algoritmo di “bucket brigade” definisce l'evoluzione delle forze

Dinamica delle ricompense



Bucket brigade

Aggiornamento dei valori della forza $s[i]$ {

$p[i] = \alpha s[i]$;

$new_s[i] = s[i] * (1 - \beta)$;

IF (i ha impostato Q messaggi)

$new_s[i] = new_s[i] - Qp[i]$;

IF (C_i ha impostato un messaggio all'istante precedente AND i classificatori $i_1, i_2 \dots$ hanno impostato usando quel messaggio)

$new_s[i] = new_s[i] + p[i_1] + p[i_2] + \dots$;

IF (C_i ha impostato un messaggio di output)

$new_s[i] = new_s[i] + reward$;

$s[i] = new_s[i]$

}

Gerarchie di default

- E' utile far coesistere, nello stesso sistema, regole più generali e regole più specifiche (“**gerarchie di default**”)
- In assenza di informazioni specifiche le regole più generali possono fornire utili indicazioni di default, mentre in presenza di informazioni ulteriori si applicheranno le regole più specifiche
 - SE **nuota nel mare** ALLORA **è un pesce**
 - SE **nuota nel mare** e **viene a galla per respirare** ALLORA **è un mammifero**
- La dipendenza del bid dalla specificità consente appunto di privilegiare, a parità di forza, i classificatori più specifici, conservando regole generali anche se contraddette in casi particolari
 - la regola generale tende a soccombere nei casi in cui fornirebbe indicazioni erronee, e non viene quindi penalizzata

Una visione astratta della evoluzione di una specie

- Una **popolazione** è composta da individui differenti
 - ad ogni individuo è associato un genotipo ereditabile
- Esiste un **meccanismo per generare nuovi genomi** a partire da quelli di uno o di alcuni individui (riproduzione)
 - consente l'introduzione di novità, in larga misura casuali (ricombinazione di materiale genetico, mutazioni)
 - i figli assomigliano ai genitori più di quanto non assomiglino in media ad altri individui
- Una **competizione fra individui** simili per riprodursi
 - è l'ambiente stesso, composto da fattori naturali e umani, altre specie, membri della stessa specie, a privilegiare alcuni individui

Algoritmi genetici

- Una popolazione composta da “individui” differenti
 - ogni individuo può rappresentare una possibile soluzione al problema in esame
 - ad ogni individuo è associata una sua descrizione che può essere trasmessa
- Esistono meccanismi per generare nuovi individui a partire da individui esistenti (operatori genetici)
 - consentono di introdurre novità, in larga misura casuali
 - i figli assomigliano ai genitori più di quanto non assomiglino in media ad altri individui
- Esiste un sistema di valutazione della fitness di ogni individuo, e la selezione dei genitori è fatta in maniera probabilistica, privilegiando quelli a fitness elevata
- Ad ogni passo si genera una nuova popolazione, finché non si raggiunge una opportuna condizione di terminazione

Apprendimento a lungo termine: algoritmi genetici

Con una certa frequenza si modificano le regole

è necessario “lasciare tempo” al bucket brigade per fornire una stima attendibile dell'utilità delle regole

- 1 si scelgono i “genitori” delle nuove regole
 - selezione probabilistica basata sulla forza
 - sulla popolazione completa di C classificatori
- 2 si applicano “operatori genetici” ai genitori per ottenere un numero $G < C$ di nuove regole
 - Elitismo (si lasciano nella popolazione le regole migliori)
- 3 si eliminano dalla popolazione G classificatori
 - eliminazione probabilistica basata sulla forza
 - vengono preservati i classificatori con forza maggiore
- 4 i nuovi classificatori vengono aggiunti alla base di regole

Mutazione e crossover

- Mutazione (singolo genitore)

| 1 | 0 | 0 | 1 | 1 | | | 0 | 0 | 0 | 1 | 1 |

- figlio

| 1 | 0 | 0 | 1 | 1 | | | 0 | 1 | 0 | 1 | 1 |

- oppure

| 1 | 0 | 0 | 1 | # | | | 0 | 0 | 0 | 1 | 1 |

- esplorazione “locale” di varianti

- Crossover a punto unico (due genitori)

| 1 | 0 | 0 | 1 | 1 | | | 0 | 0 | 0 | 1 | 1 |

| 1 | 1 | 0 | 0 | 0 | | | 1 | 1 | 0 | 1 | 0 |

- incrocio (due figli)

| 1 | 0 | 0 | 1 | 0 | | | 1 | 1 | 0 | 1 | 0 |

| 1 | 1 | 0 | 0 | 1 | | | 0 | 0 | 0 | 1 | 1 |

- varianti: crossover a due punti, crossover omogeneo

- esplorazione “a distanza”

Il ciclo della genetica

Numero_generazioni = 1

WHILE (Numero_generazioni <= Max_gen){

- Valuta la forza di tutti i classificatori della popolazione attuale P , col ciclo a regole fisse;
- valutane la fitness;
- $P' = \emptyset$;
- Scegli fra i classificatori G/2 coppie di genitori, in maniera proporzionale alla fitness;
- Incrociali con crossover a un punto, ottenendo G figli, e inseriscili in P' ;
- Ad ognuno degli elementi di P' applica l'operatore di mutazione puntuale (con una piccola probabilità);
- Aggiungi a P' le C-G regole con la fitness più alta ;
- $P = P'$;
- Numero_generazioni = Numero_generazioni + 1;

}

Alcuni vantaggi degli algoritmi genetici

- Il **sistema è stocastico**, quindi non è vincolato a cadere in un estremo locale
 - il crossover consente di esplorare regioni distanti nello spazio degli stati
 - mentre la mutazione consente esplorazioni “a corto raggio”
- Non vi sono **richieste di alcun genere** da imporre a priori alla funzione da ottimizzare
 - funzionano particolarmente bene quando vi è una certa struttura nell'insieme dei valori estremi, per cui massimi locali forniscono indicazioni sulla localizzazione del massimo assoluto
- Rappresentano una buona alternativa quando non si hanno informazioni a priori sulla funzione di fitness

Fitness dipendente dall'interazione

- Nelle “classiche” applicazioni di GA a problemi di ottimizzazione, esiste una regola per determinare la fitness di ogni individuo
 - l'unica interazione fra individui diversi è il confronto delle rispettive fitness, ed eventualmente la riproduzione
- Nei sistemi a classificatori “alla Michigan” non è possibile - in generale - assegnare una fitness direttamente all’“individuo” su cui agisce l'algoritmo genetico
- La **fitness (forza) dipende dalle interazioni**
 - è una proprietà collettiva, non individuale
- I **classificatori co-evolvono**
 - i cambiamenti di uno di essi possono influenzare la fitness degli altri

Operatori genetici specifici

- Partendo da una popolazione limitata di classificatori completamente random, può accadere con elevata probabilità che
 - i messaggi provenienti dai detettori non facciano match con alcun classificatore
 - non vi sia nessun messaggio di output
 - ad esempio, i messaggi interpretati dagli effettori come segnali di output possono essere quelli che iniziano con 5 “0”
- **Cover detector**: se nessun classificatore fa match con un messaggio di input, crea un classificatore che faccia match, con una parte di azione casuale
- **Cover effector**: se non ci sono messaggi di output (in una situazione in cui si richiede una azione o un segnale) genera un classificatore che fa match nella situazione attuale e che imposta un messaggio di output (casuale)

Aspetti suggestivi dei sistemi a classificatori

- Il sistema elabora in parallelo molte informazioni e conoscenze
- L'apprendimento combina una **metafora economica** (per l'apprendimento a breve termine) e una **metafora biologica** (per la scoperta di nuove regole)
- Si basano sulla interazione fra metodi genetici e sistemi dinamici
 - che determinano l'evoluzione delle forze

Aspetti suggestivi dei sistemi a classificatori

- Le **regole** sono “**agenti in un mercato**”
 - con regole definite dal bucket brigade
- Gli agenti sono semplici
 - stringhe di simboli tratti da un alfabeto molto semplice
- Nascita spontanea di **catene di regole** che si affermano
 - ricca dinamica di interazione
- Gerarchie di default

Elementi importanti dei sistemi a classificatori

- **Co-evoluzione**: il valore di una **porzione di stringa** (classificatore) dipende dal resto della stringa (alleli coadattati)
- Ancora co-evoluzione: **il valore di una regola** dipende dalla presenza di altre regole
 - il significato di una regola nel sistema è definito dalle interazioni con le altre regole
- Le **relazioni fra regole** (mediate dai messaggi) si creano nel corso dell'evoluzione, non sono prescritte
- Base per la modellistica di agenti economici

Alcuni aspetti problematici

- Possibile “**convergenza prematura**” verso soluzioni poco soddisfacenti
- Individui di uguale **lunghezza**
 - introdurre lunghezze variabili
 - numero variabile di regioni codificanti
- Mancanza di **previsioni**
 - introdurre le previsioni
 - misurare la fitness sulla accuratezza delle previsioni
- Limitata efficacia nella **formazione di catene lunghe**
- **Struttura piatta**
 - le gerarchie dovrebbero formarsi spontaneamente

Applicazioni

- Modelli ad agenti di sistemi socio-economici
- Animazione, figure in movimento
- Simulazione e scoperta di manovre aeree
- Knowledge discovery in medicina: stime del rischio
- Robot autonomi