Tecniche di Soft Computing per riconoscimento di anomalie: una prima analisi

Antonia Azzini, Andrea G.B. Tettamanzi
Dipartimento di Tecnolige dell'Informazione,
Università degli Studi di Milano
Via Bramante, 65, 26013 Crema (CR)
Italy; Tel. +39 0373898025, Fax +39 0373898010
antonia.azzini,andrea.tettamanzi@unimi.it

Matteo De Felice
Dipartimento di Tecnolige dell'Informazione,
Università degli Studi di Milano
Via Bramante, 65, 26013 Crema (CR)
Italy; Tel. +39 0373898025, Fax +39 0373898010
antonia.azzini,andrea.tettamanzi@unimi.it

1. Introduzione

Il riconoscimento di guasti o anomalie è uno dei processi fondamentali in diversi ambiti, legato in particolare al problema del mantenimento della sicurezza all'interno di un sistema.

Il concetto di "anomalia" di un sistema spesso parte dalla definizione di condizione di "normalità". Essa rappresenta l'obiettivo dei processi di controllo che in ogni ambito cercano di mantenere un sistema all'interno di uno spazio di funzionamento normale, con un comportamento quindi prevedibile e controllabile. Il riconoscimento automatico delle anomalie di un sistema può essere eseguito con diverse metodologie; fra esse sono interessanti quelle basate su tecniche di machine learning, in grado di apprendere in maniera automatica gli stati di un sistema a partire dall'osservazione dei dati.

La mancanza di una definizione formale e precisa del concetto di malfunzionamento o di anomalia e spesso la mancanza di dati utili al loro riconoscimento ha portato nel tempo ad orientarsi anche a tecniche bio-ispirate, che imitano il funzionamento dei sistemi esistenti in natura. Tra questi, uno dei più efficienti è il sistema immunitario umano, in grado di reagire a sostanze dannose sia endogene che esogene, come virus e batteri. Proprio a questo sistema sono ispirati i Sistemi Immunitari Artificiali (AIS, Hofmeyer et al. 2000), un insieme di tecniche che imitano la capacità del sistema immunitario di reagire sia

ad anomalie che si presentano per la prima volta, sia ad altre già note, grazie alla capacità di apprendimento e memorizzazione delle soluzioni ai problemi già affrontati in passato. In letteratura sono stati implementati diversi algoritmi basati sugli AIS, e fra essi particolarmente interessante è l'algoritmo di Negative Selection (NS) (Hofmeyer et al., 2000).

Fra gli algoritmi che si ispirano a modelli naturali nell'ambito del riconoscimento di anomalie, risultano interessanti ed efficienti anche quelli basati sugli algoritmi evolutivi, reti neurali, e quelli basati su algoritmi di swarm intelligence come l'algoritmo di Particle Swarm Optimization (PSO, Kennedy et al. 1995).

A tal proposito sono state implementate e confrontate in precedenza alcune di queste tecniche nell'ambito di anomalie e guasti su reti informatiche (Azzini et al. 2009) e i risultati soddisfacenti ottenuti dalle prime analisi incoraggiano ad approfondire lo studio del comportamento degli AIS e di altre tecniche ispirate a modelli naturali nell'ambito di tali problematiche.

Il seguente lavoro si concentra sul confronto tra due delle metodologie sviluppate, cercando di apportarne dei miglioramenti anche a livello di algoritmo: un AIS basato su NS e un algoritmo che implementa un tipo di PSO.

2. Metodologie

Ipersfere modellano i riconoscitori utilizzati in questo caso, e il loro volume è in grado di classificare lo spazio come anomalo o normale. Esse sono rappresentate nella seguente forma vettoriale:

$$D = [x_1 \dots x_N \; r]$$

dove x sono le coordinate nello spazio N-dimensionale e r è il raggio. I principali vantaggi di tali riconoscitori si traducono nella loro facilità di definizione e di implementazione. Un punto z nello spazio viene considerato coperto se:

$$\mathrm{dist}(D, z) < r$$

Dove dist è una funzione di distanza o similiarità (definita in questo lavoro attraverso la funzione di distanza euclidea).

2.1 Negative Selection

L'algoritmo di NS è uno dei principali algoritmi sviluppati nell'ambito degli AIS (Forrest 1994). Esso definisce un insieme di

riconoscitori a copertura dello spazio complementare alla “normalità”, in modo tale da classificare nuove informazioni come normali o anomale. L’algoritmo consiste nella creazione di ipersfere i cui raggi sono inizializzati casualmente nell’intervallo [0.1, 10], mentre le coordinate del centro sono inizializzate nell’intervallo [-5, 5]. Il riconoscitore appena creato viene mantenuto se non copre alcun punto definito normale. L’algoritmo termina nel momento in cui si è raggiunto il numero totale di detector richiesti.

2.2 Particle Swarm Optimization

L’algoritmo PSO utilizzato per creare l’insieme dei riconoscitori (detectors) è basato su un’implementazione standard con i parametri indicati in Tabella 1. E’ stato implementato un sistema iterativo per la creazione dell’insieme dei detector, specificato dallo pseudocodice:

```
punti_dataset ← carica_dataset()
punti_non_coperti ← dimensione(punti_dataset)
insieme_detector ← { }
while (punti_non_coperti > 0)
    best_detector ← PSO_algorithm(numero_individui, numero cicli, raggio massimo, punti_dataset)
    punti_dataset ← puntidataset \ punti_coperti(best_detector)
    punti_non_coperti ← dimensione(punti_dataset)
    insieme_detector ← insieme_detector AND best_detector
end while
```

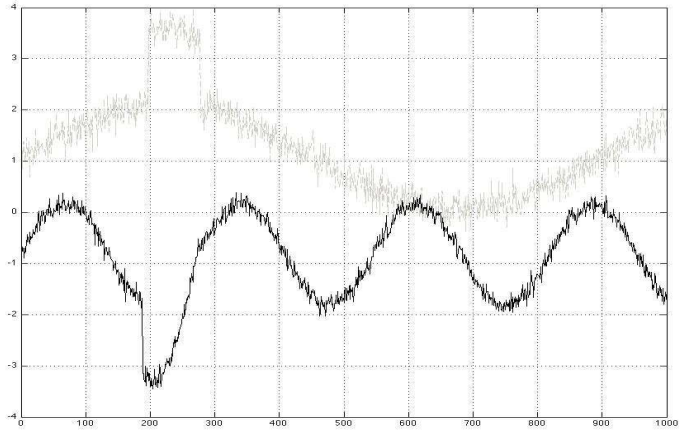

Figura 1: esempio di segnali del dataset

la funzione obiettivo dell'algoritmo di PSO implementato è basata sul numero di punti presenti all'interno del riconoscitore. Al termine di un'iterazione i punti coperti dall'individuo migliore (best_detector) vengono eliminati dal dataset e se questo non risulta vuoto viene eseguito di nuovo l'algoritmo di PSO sui restanti punti.

In questo modo però l'algoritmo tende a convergere ad un riconoscitore con raggio grande in modo da comprendere tutti i punti, mentre dovrebbe essere definito in modo da coprire il maggior numero di punti non-anomali, minimizzando la copertura di regioni di spazio non note. Tale problema è stato risolto inserendo un raggio massimo per i riconoscitori e un valore di penalizzazione nella funzione di fitness:

$$f(D) = \frac{\text{copertura}(D)}{(r+1)}$$

dove D è il detector considerato, copertura è la funzione che restituisce il numero di punti coperti dal detector D e r è il raggio dello stesso.

Tabella 1: parametri PSO e NS

Parametri	Valori
PSO	
C_1, C_2	1.49
Velocità max.	3
Inerzia	descrescente da 0.95 a 0.4
NS	
Raggi	[0.1,10]
Coordinate dei centri	[-0.5,0.5]

3. Impostazioni

I parametri utilizzati dai due modelli implementati sono definiti sulla base di una serie di esperimenti condotti al fine di individuarne il settaggio migliore. La Tabella 1 riporta le combinazioni dei valori migliori rispettivamente per PSO e NS.

Per ciascuno dei due metodi, la campagna di esperimenti è stata condotta considerando un aumento lineare della dimensionalità del problema, con valori definiti nell'intervallo [2,15].

Per ogni dimensione è stato definito un nuovo dataset. E' stato creato un segnale sinusoidale con ampiezza fissa, frequenza e bias

generati attraverso distribuzioni casuali uniformi. Il cambio di ampiezza presentato in un intervallo casuale del segnale indica

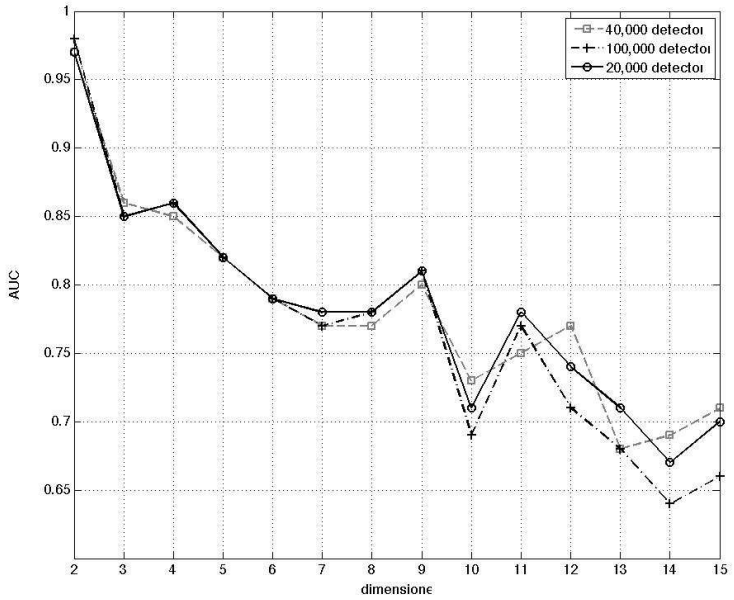

Figura 2: confronto NS con diverso numero di detector

un'anomalia. L'indice dell'anomalia di riferimento (il target) di un dataset a N-dim è generato dalla sovrapposizione degli intervalli di anomalia di tutti gli N segnali. Un esempio di dataset 2D è rappresentato in Figura 1.

4. Esperimenti e Risultati

Diversi esperimenti sono stati condotti al variare di dim., e per ognuna di esse sono state eseguite 10 valutazioni per PSO e NS. I risultati mostrano le prestazioni ottenute all'aumentare della complessità del segnale. In particolare sono stati definiti due tipi di confronti, basati rispettivamente sul calcolo della AUC, l'area sottesa dalla curva ROC (Bradley, 1997), e sul numero di richieste della funzione di distanza utilizzata nel calcolo della fitness. La Figura 2 mostra il valore di AUC delle tecniche di NS con numero diverso di

detector. Da una prima analisi si osserva come l'aumento del numero di riconoscitori applicato ai segnali porta, in generale, ad una saturazione delle prestazioni degli algoritmi, dopo soddisfacenti livelli raggiunti inizialmente. Con NS si osserva infatti come un aumento dei riconoscitori (da 40000 a 100000) non migliori in modo significativo le prestazioni di alte dimensioni.

Contemporaneamente però in entrambi i metodi si osserva anche un calo di prestazioni all'aumentare della dimensione del problema.

I migliori risultati ottenuti con PSO prevedono una popolazione formata da 100 particelle con un numero di iterazioni massimo di 200 e con un raggio definito in [1,3]. La Figura 3 mostra i risultati medi ottenuti per il PSO. Si osserva che a dimensioni elevate raggi troppo bassi non riescono a far fronte alla bassa densità dei punti nello spazio.

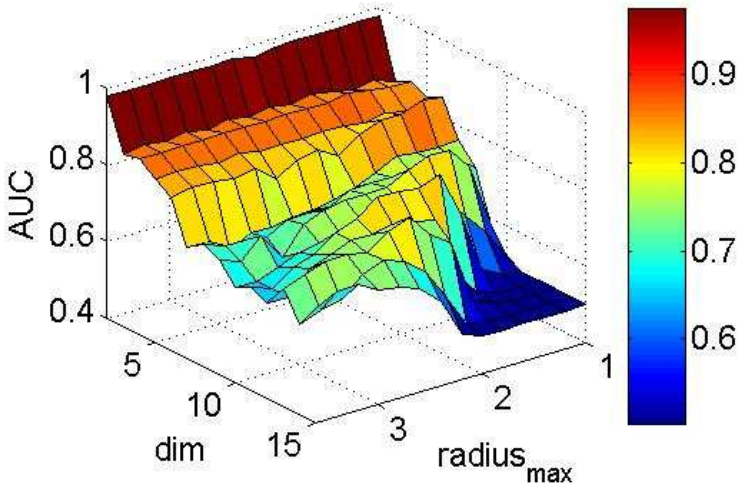

Figura 3: valore di AUC per l'algoritmo PSO al variare di dimensionalità e raggio massimo

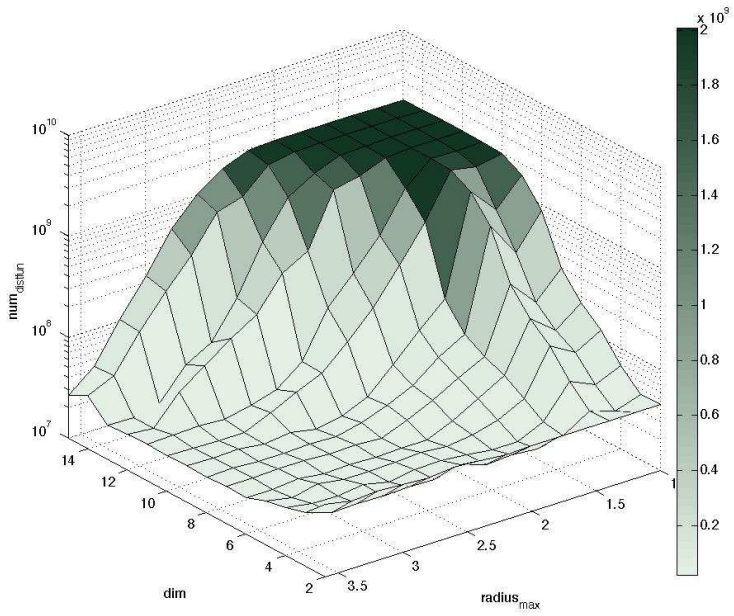

Figura 4: chiamate alla funzione di distanza del PSO

Un ulteriore ed utile strumento di confronto è indicato dallo sforzo computazionale calcolato nei vari casi, basato sul numero di chiamate alla funzione di distanza (Figura 4). Interessante notarne la crescita.

La Tabella 2 riporta un confronto fra i due metodi. Per ogni dimensione viene riportato il raggio con le migliori prestazioni.

5. Conclusioni

Il riconoscimento delle anomalie è un problema poco generalizzabile e dipendente dal caso specifico considerato. Questo lavoro ha sviluppato una prima analisi, usando dataset sintetici, tra due diversi metodi, basati su tecniche di selezione opposte: una negativa, attraverso un AIS, e una positiva, con un algoritmo di swarm intelligence.

Le prestazioni riassunte in Tabella 2 mostrano come il PSO ottenga in generale dei risultati più soddisfacenti, ma con un aumento dello sforzo computazionale. Le Figure 3 e 4 mostrano come nel PSO

prestazioni e numero di valutazioni della funzione di distanza siano correlati: un raggio minore può portare in alcuni casi ad una copertura più precisa dello spazio al costo però di uno sforzo maggiore di calcolo.

Tabella 2: confronto tra NS e PSO

	NS – 20,000		NS – 40,000		PSO		
Dim	AUC	N.Dist.	AUC	N.Dist.	AUC	N.Dist.	r
2	0.97	$2.10\ 10^7$	0.97	$4.19\ 10^7$	0.98	$4.46\ 10^7$	1
3	0.86	$2.16\ 10^7$	0.85	$4.32\ 10^7$	0.85	$3.66\ 10^7$	1.2
4	0.86	$2.15\ 10^7$	0.86	$4.29\ 10^7$	0.86	$6.76\ 10^7$	1.4
5	0.82	$2.09\ 10^7$	0.82	$4.19\ 10^7$	0.82	$4.30\ 10^7$	1.6
6	0.79	$2.09\ 10^7$	0.79	$4.19\ 10^7$	0.84	$5.73\ 10^7$	1.6
7	0.77	$2.10\ 10^7$	0.78	$4.20\ 10^7$	0.77	$6.92\ 10^7$	1.6
8	0.78	$2.10\ 10^7$	0.78	$4.20\ 10^7$	0.74	$9.12\ 10^7$	1.6
9	0.81	$2.07\ 10^7$	0.81	$4.15\ 10^7$	0.81	$1.60\ 10^8$	1.6
10	0.69	$2.06\ 10^7$	0.71	$4.12\ 10^7$	0.74	$8.25\ 10^7$	2.2
11	0.77	$2.06\ 10^7$	0.78	$4.11\ 10^7$	0.81	$4.93\ 10^7$	2.4
12	0.71	$2.03\ 10^7$	0.74	$4.05\ 10^7$	0.76	$2.94\ 10^8$	2.4
13	0.68	$2.04\ 10^7$	0.71	$4.08\ 10^7$	0.80	$4.67\ 10^8$	2.4
14	0.64	$2.02\ 10^7$	0.67	$4.05\ 10^7$	0.74	$3.95\ 10^8$	2.8
15	0.66	$2.03\ 10^7$	0.70	$4.06\ 10^7$	0.73	$4.29\ 10^7$	3.4
Media	0.77	$2.07\ 10^7$	0.78	$4.15\ 10^7$	0.80	$1.35\ 10^8$	

Il PSO può quindi essere migliorato in due modi: rendendo il raggio adattivo oppure con un approccio multi-obiettivo. Il primo potrebbe essere utile per garantire una buona copertura dello spazio al crescere della dimensionalità, mentre un approccio multi-obiettivo porterebbe a massimizzare la copertura dei riconoscitori, cercando di minimizzarne il raggio. Di contro nella NS un algoritmo multi-obiettivo potrebbe cercare di trovare la copertura ottimale dei casi anomali, limitando la sovrapposizione dei riconoscitori e dei buchi che si formano nelle coperture, causa principale del calo delle prestazioni di un modello.

Bibliografia

1. Hofmeyer, S.A., Forrest, S.: Architecture for An Artificial Immune System, Evolutionary Computation 8(4), pp. 443-473, 2000
2. Azzini, A., De Felice, M., Meloni, S., Tettamanzi, A.G.B.: Soft Computing Techniques for Internet Backbone Traffic Anomaly Detection, Workshop on Evolutionary Applications for Telecommunications and Networks. (2009).
3. Dasgupta, D., Ji, Z.: Real-valued negative selection algorithm with variable-sized detectors. In GECCO'04. pp. 287-298, 2004.
4. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. Proc. of IEEE Symp. On Research in Security and Privacy. Pp. 202-212, Los Alamitos, CA. 1994.
5. Kennedy, J. Eberhart, R.: Particle swarm optimization. In: IEEE Interna tional Conference on Neural Networks, pp. 1942-1948 vol.4 (1995).
6. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition, Volume 30, Is. 7, pp. 1145-1159, 1997.