

# Tesina - Algoritmi genetici

Alessio Marchetti

**Nota:** Le note a margine indicano quando cambiare la slide. Tuttavia alcune slides sono frammentate in più fasi, e necessitano di andare avanti anche se non specificato.

L'argomento della mia tesina è uno studio del funzionamento degli algoritmi genetici. Gli algoritmi genetici sono innanzi tutto algoritmi, ovvero un insieme di istruzioni utili a risolvere determinate classi di problemi. L'aggettivo "genetici" viene attribuito in quanto i processi che studieremo attuano meccaniche ispirate ai meccanismi della natura, in particolare quelli della genetica e della selezione naturale. Slide

La metafora si struttura in questo modo: dato un certo problema, l'obiettivo dell'algoritmo è quello di trovare la soluzione migliore. Allora si genera in modo casuale una popolazione di candidate soluzioni e di queste si scelgono le più adatte a risolvere il problema. Poi a partire da esse si costruiscono nuove candidate in modo tale da avere una popolazione mediamente migliore. Quest'ultimo processo ricalca la riproduzione sessuata, infatti da coppie di soluzioni, vengono prodotte soluzioni figlie con caratteristiche comuni ai due genitori. Slide

Per spiegare meglio il funzionamento operativo di un algoritmo genetico, seguiremo passo a passo un caso specifico. Il problema di cui tratterò è di carattere matematico, ovvero la ricerca del massimo e del punto di massimo di una funzione in un determinato intervallo chiuso. Ovvero, dato un grafico, il compito è quello di cercarne il punto più alto. La funzione che ho scelto è molto semplice ed è una parabola. In questo caso, il massimo si trova all'estremo. Vediamo come opera un algoritmo genetico. Slide

Innanzitutto occorre trovare un modo efficace di descrivere una soluzione. A tal fine definisco ciò che potrebbe essere l'analogo di un DNA, che andrà a identificare ogni individuo della popolazione di soluzioni. Dunque ho bisogno di un insieme (chiamato vocabolario) di basi azotate. Nel mio esempio lo scelgo nel modo più semplice (e naturale) possibile, ovvero composto da due elementi: zero e uno. Un DNA consiste in una stringa di cinque elementi. È anche necessaria una codifica dal genotipo al fenotipo, ovvero da ciò che il DNA indica e quale caratteristica effettivamente esprime un Slide

certo individuo. Nella pratica la stringa di zeri e uno verrà letta in codice binario e interpretata come posizione sull'asse delle ascisse.

La popolazione iniziale viene generata in modo totalmente casuale. Ciò significa che ogni possibile DNA ha la stessa probabilità di essere rappresentato. Si può pensare in tal senso che ogni gene sia il risultato di un lancio di una moneta: zero se esce testa e uno se esce croce. Nell'esempio ho scelto una popolazione molto piccola per poterci lavorare comodamente a mano. Questi sono i risultati ottenuti.

Giunti a questo punto è necessario scegliere gli individui migliori. Ovvero serve trovare un modo per identificare quali DNA sono i più adatti a risolvere il problema. Per questo motivo definisco una funzione, detta di fitness, che valuta ogni individuo. Nel nostro esempio più è alta l'immagine di un certo punto, più il candidato risulta buono, dunque prendere come funzione di fitness  $f(x)$  stessa, è una scelta sensata. La tabella di prima aggiornata risulta dunque essere la seguente.

Adesso si ha tutto l'occorrente per costruire una nuova generazione. Essa deve avere come proprietà innanzi tutto quella di essere mediamente migliore della precedenti. In secondo luogo deve avere caratteristiche in comune con essa. La prima fase del passaggio di generazione è quello di eliminare gli individui peggiori. A tal fine definisco la probabilità di sopravvivenza in questo modo. Si noti che tale probabilità è proporzionale al fitness, e la somma di essa su tutti gli individui è pari a uno. Nella pratica assumeranno i valori che si vedono nella tabella. Risulta immediato verificare che a maggiore fitness si associa una maggiore probabilità di sopravvivenza.

A cosa servono? Come passaggio intermedio per arrivare alla generazione successiva costruisco il cosiddetto *mating pool*, piscina di accoppiamento. Esso conterrà lo stesso numero di individui della popolazione originale, e ogni individuo avrà una probabilità pari alla sua probabilità di sopravvivenza di entrare nel mating pool. In pratica cosa succede? Per scegliere ogni individuo del mating pool, faccio girare una ruota simile a quella di una lotteria. Ogni tacca rappresenta un individuo della popolazione iniziale. Quindi l'individuo scelto sarà esattamente uguale a quello indicato dalla ruota.

Arrivati sin qui è lecito chiedersi se effettivamente sia avvenuto un miglioramento dalla popolazione iniziale. Per dare un'analisi quantitativa occorre definire il fitness medio, che non è null'altro che la media dei fitness di una popolazione. Se ogni individuo ha una probabilità  $p$  di passare nel mating pool, ci si aspetta che la frazione di individui nel mating pool uguali ad esso sia proprio  $p$ . [breve commento alla formula]

Si noti che con questo metodo nel mating pool potrebbe esserci più di una copia di uno stesso individuo. Per questo motivo si rendono necessari i prossimi passaggi di crossing over e di mutazione. Infatti se non ci fosse né una ricombinazione, né una modificazione dei DNA, l'algoritmo si limiterebbe a scegliere la migliore soluzione tra quelle iniziali, che ricordo

essere generate casualmente. Serve allora implementare dei meccanismi che costruiscano soluzioni nuove.

Slide

Il primo di questi meccanismi è detto, in analogia con ciò che avviene durante la meiosi, crossing over. Nel mating pool, gli individui vengono accoppiati, e ad ogni coppia succede essenzialmente quello che è mostrato in figura: i DNA si scambiano casualmente alcune sezioni.

Slide

Il secondo meccanismo è invece detto mutazione. Alcuni geni nel DNA vengono alterati con una probabilità fissata generalmente bassa, nell'esempio pari a 0.001.

Slide

Dalla tabella, in cui vengono messi a confronto gli individui delle due generazioni, risulta effettivamente evidente come si abbia avuto un incremento generale di fitness. Ciò è indice del fatto che le soluzioni della popolazione risolvono meglio il problema posto. Ripetendo più volte il percorso seguito di selezione, crossing over e mutazione, la tendenza sarà quella di una migrazione e convergenza verso la soluzione migliore. Quante volte sia necessario ripetere non è stabilito a priori e questo fatto rappresenta il grosso limite degli algoritmi genetici. Tuttavia non sempre è necessaria la soluzione migliore in assoluto, ma solo una buona approssimazione di essa. Nel video che vi mostrerò si riproduce lo stesso algoritmo seguito ora, solo eseguito dal computer, con più individui per generazione e un DNA dalle dimensioni maggiori.

Mostrare  
program  
max

Un altro esempio in cui sono riuscito ad applicare un algoritmo genetico è il problema del commesso viaggiatore. Il problema è così strutturato: dato un certo insieme di città (ovvero punti nel piano), trovare il percorso più breve che passi per ciascuno di essi. Per quanto semplice sia la consegna, gli algoritmi tradizionali sono messi in difficoltà. Sebbene la soluzione possa essere trovata, richiederebbe un tempo di elaborazione molto lungo. Un algoritmo genetico magari non troverà la soluzione migliore, ma almeno una soluzione molto buona, in un tempo minore.

Mostrare  
program  
salesman