GRANULAR COMPUTING E KNOWLEDGE DISCOVERY

Antonello Rizzi

L'Intelligenza Computazionale, essenzialmente una branca di quella Artificiale, *comprende una serie di tecniche e metodologie ispirate a sistemi naturali, utilizzabili per affrontare problemi di* Machine Learning *in presenza di incertezze nei dati e/o nei modelli. Fanno parte di questa 'cassetta degli attrezzi' le reti neurali, la logica fuzzy, gli algoritmi evolutivi e il* Granular Computing. *Quest'ultimo indica un paradigma per il modellamento data driven, fondato sull'estrazione e sull'elaborazione di entità fondamentali d'informazione, chiamate* information granules. *L'articolo presenta le peculiarità di questo nuovo paradigma, evidenziandone le potenzialità come efficace approccio per la sintesi di modelli predittivi e come strumento per l'estrazione d'informazioni utili per esprimere in modo sintetico, e possibilmente in linguaggio naturale, le regolarità sottostanti al processo osservato.*

L'Intelligenza Artificiale (AI) sta vivendo una fase di forte espansione, peraltro ampiamente prevedibile, sia come tema di ricerca e sviluppo, sia come tecnologia abilitante in numerosi ambiti applicativi, dovuta alla 'riscoperta' degli algoritmi di *Machine Learning* (ML). Infatti, intorno agli inizi degli anni Novanta la AI classica (basata su sistemi esperti puramente deduttivi) subì un declino, evidenziando notevoli limiti. Nel frattempo si faceva strada l'approccio «connessionista», che si è nel tempo ampliato e sviluppato in un insieme di tecniche a cui ci si riferisce più recentemente con il temine *Intelligenza Computazionale*. Si tratta di algoritmi che traggono ispirazione da sistemi biologici e naturali, comprendendo le reti neurali artificiali, le meta-euristiche di ottimizzazione evolutiva, la logica fuzzy, finalizzati per l'analisi intelligente di dati in presenza di incertezze e basati principalmente sul ragionamento induttivo. Le tecniche di intelligenza computazionale costituiscono una branca (fondamentale) della AI e sono anche indicate con il termine *Soft Computing*.

Per comprendere appieno il ruolo del G*ranular Computing* nei moderni sistemi di elaborazione dell'informazione occorre fare un passo indietro, introducendo definizioni precise, seppur informali, dei termini AI e ML, considerando il fatto che sono a volte utilizzati a sproposito, non solo nell'ambito della divulgazione scientifica.

Cosa si debba intendere con «intelligenza» è ancora un tema dibattuto, a cui contribuiscono varie sfere del sapere (filosofia, psicologia, biologia, ingegneria). Nel 1950 Alan Turing fornì un criterio per stabilire se una macchina potesse considerarsi 'intelligente', senza entrare nel merito delle capacità fondamentali che caratterizzano un essere intelligente (naturale o artificiale che sia). Si tratta di una *caratterizzazione esterna*, ossia di un criterio di indistinguibilità nei confronti di un osservatore terzo, in linea con i dettami del comportamentismo di fine Ottocento. Il test di Turing ha avuto il merito di rinvigorire una discussione, antica quanto la filosofia greca e tuttora in corso, sul significato stesso di «mente» e di «coscienza». Sono argomenti di straordinario interesse, ma dal punto di vista ingegneristico – di chi come mestiere ha il compito di «far funzionare le cose» – è più utile limitare lo sviluppo scientifico e tecnologico a obiettivi più modesti: la AI tratta algoritmi e sistemi che consentono di imitare alcune funzioni cognitive di base del cervello biologico. Non si tratta di «abbassare l'asticella» perché, anche riferendosi a funzioni basilari, le sfide tecnologiche rimangono notevoli. Certamente nessuno di noi è disposto ad attribuire una forma di intelligenza al proprio portatile o al proprio smartphone; questi dispositivi sono macchine di calcolo progettate e realizzate secondo l'architettura di John von Neumann, perfettamente equivalenti a una «macchina di Turing». Sono 'macchine' perché si limitano a eseguire meccanicamente e molto velocemente algoritmi, manipolando simboli secondo precise sequenze di istruzioni. Paradossalmente, quando invece vediamo in azione un assistente virtuale, come Alexa o Siri, allora riconosciamo (non a caso) una qualche forma di comportamento intelligente. Siamo concordi nell'attribuire al cervello biologico capacità superiori, specialmente (e non solo) quando ci si riferisce alla classe dei mammiferi (uomo compreso). Il cervello biologico si è evoluto per modellare la realtà che ci circonda, specializzandosi nella sintesi di modelli predittivi. Avere la capacità di prevedere il comportamento di un predatore o di una preda è una questione di vita o di morte. Semplici sistemi di controllo (percezione-azione) sono presenti anche nei batteri flagellati e nei protisti, ma certamente non siamo autorizzati a parlare di comportamento intelligente. Invece, già a partire da alcuni invertebrati (si pensi al polpo), reti di neuroni consentono di 'imparare', fornendo un vantaggio evolutivo notevole al singolo individuo, e dunque alla specie. Siamo arrivati alla questione essenziale: possiamo ancora non essere in grado di definire l'intelligenza in modo preciso e rigoroso, ma possiamo concordare che un elemento di base (necessario e non sufficiente) per riconoscere un comportamento intelligente sia la capacità di sintetizzare modelli predittivi. I primati hanno eccellenti capacità di apprendere un compito (una relazione anche complessa tra percezione e azione, che in seguito chiameremo più genericamente «processo orientato») tramite

esempi, dimostrando capacità di generalizzazione, ossia di poter utilizzare il modello sintetizzato in contesti nuovi (carattere predittivo del modello). Il tema essenziale del ML è quello di progettare e realizzare sistemi in grado di imitare questa capacità, in contesti rumorosi e in presenza di incertezza nei dati. Più precisamente, per ML si intende un insieme di tecniche finalizzate alla sintesi di modelli di processi (orientati o non orientati) basandosi su un campionamento finito (ma statisticamente significativo) del processo stesso. La sintesi (*training*) ha per obiettivo la massimizzazione di una misura di prestazione del modello su un insieme di dati completamente distinto da quello utilizzato durante l'apprendimento. Considerati i notevoli progressi in questo campo, si direbbe trattarsi di un 'giochino semplice' su cui rimane poco da aggiungere, ma non è così. Ci sono ancora sfide aperte, scientifiche e tecnologiche, su cui occorre lavorare, a partire dalle stesse architetture di calcolo (hardware) sui cui tali sistemi possono essere efficacemente implementati. I sistemi di modellamento *data driven* sono anche detti «sistemi di modellamento induttivi» per la natura delle inferenze logiche che entrano in gioco. Senza perdita di generalità, si consideri un particolare problema di modellamento supervisionato, noto come classificazione. Del resto, alcuni princìpi generali si applicano anche ad altri problemi notevoli, come la regressione, la predizione di serie storiche, il modellamento non supervisionato (*clustering*). In un problema di classificazione si intende sintetizzare il modello di un processo orientato in cui lo spazio di uscita è un insieme di possibili etichette, ossia l'uscita è un dato discreto nominale. Ad esempio, per un problema di manutenzione predittiva su infrastrutture critiche, le possibili etichette possono essere «funzionamento normale» e «funzionamento anomalo». La capacità di generalizzazione dipende essenzialmente dal tipo di inferenza induttiva adottata dal modello. Per meglio chiarire il punto, è essenziale comprendere la distinzione tra inferenze ampliative e non ampliative, e in particolare tra deduzione e induzione. Il ragionamento principe della deduzione è il seguente: se supponiamo che tutti gli elementi di un certo insieme A soddisfino un determinato predicato P, allora ne deduciamo che tutti gli elementi di un qualsiasi suo sottoinsieme B soddisfino sicuramente anch'essi P (se supponiamo che gli italiani siano amanti della musica, allora i toscani sono amanti della musica). La deduzione, attività logica per eccellenza, è viziata da un'insuperabile sterilità, giacché si limita a trarre tutte le possibili conseguenze da alcune premesse, e dunque a esplicitare quanto implicitamente già contenuto in esse. L'irrilevanza della deduzione nella ricerca e nella co-

struzione di nuova conoscenza dipende dal fatto che il processo deduttivo è un passaggio dal generale al particolare. È chiaro che un processo di costruzione di nuove conoscenze deve essere fondato proprio su un percorso in senso inverso (dal particolare al generale): se supponiamo che tutti gli elementi di un certo sottoinsieme B di un insieme A soddisfino un determinato predicato P, allora azzardiamo la tesi che tutti gli elementi dell'intero insieme A soddisfino anch'essi P. Questo procedimento spericolato, questo temerario salto nel buio, si chiama induzione. L'induzione, sia pure a rischio di commettere errori, ci consente di acquisire nuove informazioni. Le definizioni di deduzione e induzione su esposte sono valide in un qualunque universo di riferimento. Ma se, in particolare, operiamo su un universo del discorso X in cui è possibile definire una misura di dissimilarità o similarità tra coppie di oggetti in X, e dunque sia possibile definire una misura di (dis-)similarità tra un oggetto in X e un insieme di oggetti S incluso in X, la fastidiosa attitudine delle inferenze ampliative (come l'induzione o l'analogia) a produrre grossolani errori può essere mitigata, enunciando un semplice principio di inferenza induttiva su spazi normati.
Sia A un qualunque insieme in uno spazio normato X e B un sottoinsieme di A; supponiamo che ogni elemento di B soddisfi un predicato P; indichiamo con $s_B(x)$ una misura di similarità tra un elemento x e l'insieme B. Formuliamo quindi l'ipotesi che per un dato elemento a in A, il predicato P sia vero in misura proporzionale a $s_B(a)$.
La corretta definizione di un'inferenza induttiva, e dunque la scelta di una metrica, è in sostanza la questione essenziale di ogni algoritmo di ML. La progettazione di un sistema di ML dipende, peraltro, da altre scelte critiche del progettista. Anzitutto gli oggetti da classificare (gli elementi dell'universo del discorso) sono entità reali, e occorre in primo luogo decidere come rappresentarle, ossia quali caratteristiche considerare e come misurarle. Ad esempio, in un sistema di controllo accessi occorre stabilire come gli individui verranno rappresentati nella memoria di un calcolatore. Possiamo acquisire l'immagine del volto, oppure una impronta digitale, o una impronta vocale, come anche una combinazione di tali informazioni. Queste scelte sono espresse da una funzione di rappresentazione, che mappa oggetti del dominio (persone fisiche, reali) del processo considerato in un opportuno spazio di strutture dati. E infine, secondo l'approccio classico, una funzione di pre-processamento ha il compito di generare un insieme di caratteristiche, ciascuna codificata in un numero, se ad esempio queste caratteristiche esprimano qualità misurabili. La corretta combinazione di tutte queste scelte (rappresentazione, pre-processamento e metrica alla base dell'inferenza induttiva) fanno la differenza tra un sistema che funziona e uno che fallisce miseramente.

Inoltre, anche qualora si riuscisse con successo a sintetizzare un modello che dimostri in fase di test prestazioni eccellenti, in taluni ambiti applicativi sono richiesti altri requisiti ritenuti essenziali. Spesso, infatti, si richiede che la funzione di decisione del modello sintetizzato possa esprimersi in modo chiaro e comprensibile, fornendo alcune regole espresse in linguaggio naturale. Purtroppo, questo non è possibile per un nutrito insieme di sistemi di modellamento data driven. Non è possibile, ad esempio, esprimere in modo semplice le regole di decisione di una rete neurale a partire dall'analisi della sua topologia e dall'insieme dei pesi sinaptici. Quando questo accade si dice che il modello considerato (tipo una rete neurale di tipo *feed-forward*) è una *black box*, una scatola nera, opaca e insondabile. Quando invece un sistema di ML consente di interpretare in linguaggio naturale la struttura stessa del modello in modo semplice e diretto, si dice che il sistema di modellamento induttivo fornisce una *white box*. Situazioni intermedie vengono denominate *grey box*. È questo il punto centrale di ciò che si intende per *Knowledge Discovery*, ossia l'esigenza di estrarre nuova conoscenza dai dati, descrivendo in modo semplice il fenomeno sottostante, cioè il processo che li ha generati. Un modo diretto per definire sistemi di tipo white box, caratterizzato da modelli di classificazione facilmente interpretabili e descrivibili in linguaggio naturale, consiste nell'adottare tecniche di clustering, ossia nell'impiego di algoritmi in grado di determinare gruppi (clusters) di dati simili. Anche in questo caso è determinante il modo in cui gli oggetti del dominio sono rappresentati, pre-processati e confrontati (determinazione della metrica).
Se un processo è predicibile (in qualche misura) allora devono esistere necessariamente delle regolarità, e un'analisi di clustering consente di determinarle; il problema è definire in quale spazio e con quale metrica esse sono più evidenti e meglio descrivibili in linguaggio naturale. A tal fine, un approccio promettente consiste nel progettare sistemi di modellamento granulare dell'informazione. Il Granular Computing è un paradigma emergente nell'ambito del ML, in cui granuli di informazione (*information granules*) sono estratti dai dati disponibili e utilizzati per costruire opportuni modelli. Un granulo di informazione – concetto introdotto nel 1997 da Lotfi Aliasker Zadeh, un matematico iraniano naturalizzato statunitense, padre della logica fuzzy, come generalizzazione degli insiemi sfumati – è definito come una collezione di dati raggruppati per la loro somiglianza, adiacenza funzionale o fisica, indistinguibilità o coerenza, descritti tramite un unico comune rappresentante, ossia una nuova

struttura dati che ne sintetizza le caratteristiche. In effetti, il concetto di granulo di informazione è una generalizzazione del concetto di cluster e un modo per sintetizzare automaticamente granuli di informazione consiste proprio nell'adottare algoritmi di clustering. Granuli di informazione possono poi essere combinati tra loro, come accade nell'ambito della logica fuzzy, in cui ciascun cluster può essere efficacemente rappresentato da una funzione di appartenenza, adottando le regole del caso per trasformare, confrontare e combinare tra loro questi nuovi insiemi. Inoltre, stadi successivi di granulazione possono operare ricorsivamente su collezioni di granuli di informazione di livello inferiore, costruendo rappresentazioni gerarchiche del processo che ha generato i dati. In tale struttura gerarchica ciascun livello di granulazione corrisponde a un preciso livello semantico, e nell'ambito di tale livello ciascun granulo rappresenta un concetto ricorrente e significativo per il problema in esame. La questione essenziale è che regolarità che non sono visibili al raw data level (il livello dei dati atomici, rappresentati dai record che costituiscono il campionamento del processo) possono essere invece meglio catturate a livelli semantici superiori, organizzando i corrispondenti granuli d'informazione per costruire un modello sufficientemente adeguato. Questi sistemi sono caratterizzati da complessità computazionali elevate e richiedono dunque l'impiego di un hardware appropriato.

Sistemi di modellamento granulare delle informazioni possono essere efficacemente scelti per affrontare problemi di B*ig* D*ata* A*nalytics*, neologismo entrato nel linguaggio comune perché l'aumento delle dimensioni dei data set da elaborare hanno posto in rilevo le proprietà di scalabilità nell'elaborare grandi moli di dati (dell'ordine dei terabyte e oltre), con vincoli stringenti sugli stessi tempi di calcolo. Si tratta di un tema di assoluto rilievo strategico, peraltro con notevoli impatti multidisciplinari. Infatti, non a caso le metodologie di analisi per i cosiddetti Big Data, insieme alle tecniche di ML, costituiscono una tecnologia abilitante fondamentale per il Piano Nazionale Industria 4.0. Ma la dimensione del data set non è l'unico fattore a richiedere crescenti potenze di calcolo. Vi sono altri due fattori, spesso presenti nelle applicazioni reali, che concorrono notevolmente nel determinare la complessità computazionale di una procedura di granulazione dell'informazione:

1. ciascun record può essere definito da una struttura dati ben più complicata di un semplice insieme di caratteristiche misurate da numeri reali, come una sequenza o un grafo. Ad esempio, un Network Intrusion Detection System può essere addestrato su un data base in cui ciascun record è una sequenza di pacchetti Tcp/IP. Evidentemente, al crescere della complessità della struttura dati del record aumenta anche la complessità computazionale della misura di dissimilarità adottata per confrontare coppie di record e, di conseguenza, il tempo totale necessario alla soluzione del problema di modellamento;

2. le variabili rilevanti di solito non sono note a priori. Si consideri, ad esempio, un data base contenente le registrazioni delle attività di ciascun utente agganciato a una rete di telefonia mobile (Call Data Records). In base alla finalità del sistema di modellamento (si pensi alla profilazione degli utenti per segnalare possibili comportamenti anomali o all'individuazione di classi di utenti accomunati da comportamenti condivisi), non tutti i dati presenti nel record sono necessariamente correlati col quesito in esame. Ne deriva il problema di identificare non solo i cluster, ma anche il sottoinsieme di dati (ossia il sottospazio) in cui gli eventuali cluster emergono ben formati (compatti e popolati) con maggior chiarezza. Ovviamente, questa capacità di un algoritmo di saper identificare in modo automatico sottoinsiemi di caratteristiche semanticamente correlate alla questione in esame implica un ulteriore aumento del costo computazionale, comportando l'implementazione di algoritmi di ottimizzazione stocastica (ad esempio, di tipo evolutivo).

Più in generale, se la misura di dissimilarità adottata è definita a meno di un set di parametri (funzioni parametriche) la determinazione automatica della miglior istanza di tali parametri prende il nome di M*etric* L*earning*, ed è attualmente un campo di ricerca che ha guadagnato molta attenzione nella comunità scientifica, fondamentale nell'ambito del Knowledge Discovery. La notevole esplosione nella richiesta di risorse computazionali all'aumentare della dimensione (in termini di numero di record) del data base e della complessità strutturale dello spazio dei record, congiuntamente alla necessità di determinare in automatico la stessa metrica, comporta l'adozione di sistemi di calcolo distribuiti, realizzati su un cluster di workstation, ciascuna delle quali dotata di uno o più processori multicore (*sistemi manycore*), eventualmente supportata da hardware specifico per l'accelerazione di componenti critiche. A tal fine, oltre alle Graphics Processing Unit (Gpu), un'interessante alternativa è offerta da schede Field Programmable Gate Array (Fpga) programmate per eseguire calcoli massivamente ripetuti (come misure di dissimilarità tra sequenze), seguendo un approccio su cui recentemente Intel sta puntando in modo deciso. Recenti attività di ricerca in ambito internazionale si stanno concentrando proprio sullo sviluppo di algoritmi di modellamento gerarchico delle informazioni, basati su popolazioni di agenti intelligenti che evolvono per massimizzare la capacità collettiva di individuare significativi granuli di informazione, interagendo tra loro secondo schemi competitivi / cooperativi. Ciascun agente esegue una granulazione dell'informazione (alla ricerca di opportune regolarità) in un suo specifico sottospazio, implementando una interessantissima funzionalità dal punto di vista del Knowledge Discovery nota come L*ocal Metric Learning*. È evidente che tali sistemi devono essere necessariamente implementati in modo massivamente parallelo, eventualmente sfruttando nuovi paradigmi di calcolo su hardware dedicato. Con buona pace di tutti quelli che sono convinti che il Machine Learning sia solo una questione di librerie software precompilate e di qualche riga di codice in Python