



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

Fondamenti di Intelligenza Artificiale

Lezione 20 - Riassunto e conclusioni



Riassunto e conclusioni

Un breve riassunto di quanto fatto, algoritmi non informati

Algoritmi che non dispongono di informazioni aggiuntive sugli stati oltre a quella fornita dal problema. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione in **ambienti osservabili, discreti, noti e deterministici**.

<u>Algoritmi non informati</u>	Completezza	Ottimalità	Complessità temporale	Complessità spaziale
Ricerca in ampiezza	Si	Solo se il costo di cammino è una funzione monotona non decrescente.	$O(b^d)$	$O(b^d)$
Ricerca a costo uniforme	Si	Si	$O(b^{1+C^*/\epsilon})$.	$O(b^{1+C^*/\epsilon})$.
Ricerca in profondità	No	No	$O(b^d)$	$O(bd)$.
Ricerca a profondità limitata	No	No	$O(b^l)$	$O(bd)$.
Ricerca con approfondimento iterativo	Si	Solo se il costo di cammino è una funzione monotona non decrescente.	$O(b^d)$	$O(bd)$.
Ricerca bidirezionale	Si	Si	$O(b^{d/2}) + O(b^{d/2})$	$O(b^{d/2}) + O(b^{d/2})$

Riassunto e conclusioni

Un breve riassunto di quanto fatto, algoritmi non informati

Algoritmi che non dispongono di informazioni aggiuntive sugli stati oltre a quella fornita dal problema. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione in **ambienti osservabili, discreti, noti e deterministici**.

Algoritmi non informati	Completezza	Ottimalità	Complessità temporale	Complessità spaziale
Ricerca in ampiezza	Si	Solo se il costo di cammino è una funzione monotona non decrescente.	$O(b^d)$	$O(b^d)$
Ricerca a costo uniforme	Si	Si	$O(b^{1+C^*/\epsilon})$.	$O(b^{1+C^*/\epsilon})$.
Ricerca in profondità	No	No	$O(b^d)$	$O(bd)$.
Ricerca a profondità limitata	No	No	$O(b^l)$	$O(bd)$.
Ricerca con approfondimento iterativo	Si	monotona non decrescente.		$O(bd)$.
Ricerca bidirezionale	Si	Si	$O(b^{d/2}) + O(b^{d/2})$	$O(b^{d/2}) + O(b^{d/2})$

Il più efficiente, ma anche il più difficile da realizzare per via della definizione di un obiettivo nella ricerca all'indietro!

Riassunto e conclusioni

Un breve riassunto di quanto fatto, algoritmi non informati

Algoritmi che non dispongono di informazioni aggiuntive sugli stati oltre a quella fornita dal problema. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione in **ambienti osservabili, discreti, noti e deterministici**.

Algoritmi non informati	Completezza	Ottimalità	Complessità temporale	Complessità spaziale
Ricerca in ampiezza	Si	Solo se il costo di cammino è una funzione monotona non decrescente.	$O(b^d)$	$O(b^d)$
Ricerca a costo uniforme	Si	Si	$O(b^{1+C*/\epsilon})$	$O(b^{1+C*/\epsilon})$
Ricerca in profondità	No	No	$O(b^d)$	$O(b^d)$
Ricerca a profondità limitata	No	No	$O(b^l)$	$O(b^d)$
Ricerca con approfondimento iterativo	Si	Solo se il costo di cammino è una funzione monotona non decrescente.	$O(b^d)$	$O(b^d)$
Ricerca bidirezionale	Si	Si	$O(b^{d/2}) + O(b^{d/2})$	$O(b^{d/2}) + O(b^{d/2})$

Nel corso abbiamo visto più volte come questi algoritmi siano “flessibili” e possano essere adottati in altri contesti.

Riassunto e conclusioni

Un breve riassunto di quanto fatto, algoritmi non informati

Algoritmi che non dispongono di informazioni aggiuntive sugli stati oltre a quella fornita dal problema. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione in **ambienti osservabili, discreti, noti e deterministici**.

<u>Algoritmi non informati</u>	Completezza	Ottimalità	Complessità temporale	Complessità spaziale
Ricerca in ampiezza	Si	Solo se il costo di cammino è una funzione monotona non decrescente.	$O(b^d)$	$O(b^d)$
Ricerca a costo uniforme	Si	Si	$O(b^{1+C^*/\epsilon})$.	$O(b^{1+C^*/\epsilon})$.
Ricerca in profondità	No	No	$O(b^d)$	$O(bd)$.
Ricerca a profondità limitata	No	No	$O(b^l)$	$O(bd)$.
Ricerca con approfondimento iterativo	Si	Solo se il costo di cammino è monotonicamente decrescente.		
Ricerca bidirezionale	Si			

In tutti i casi, la complessità di questi algoritmi è spesso eccessiva per risolvere problemi reali!

Riassunto e conclusioni

Un breve riassunto di quanto fatto, algoritmi informati

Algoritmi che sfruttano conoscenze specifiche del problema. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione in **ambienti osservabili, discreti, noti e deterministici**.

Algoritmi informati	Completezza	Ottimalità	Complessità temporale	Complessità spaziale
Best-first greedy	No	No	$O(b^m)$	$O(b^m)$
A*	Solo se il numero di nodi è finito e il loro costo è minore o uguale a C^*	Solo se l'euristica è ammissibile e consistente	$O(b^e)$.	$O(b^m)$
Beam Search	No			$O(kd)$.
IDA*	Solo se il numero di nodi è finito e il loro costo è minore o uguale a C^*	Solo se l'euristica è ammissibile e consistente	$O(b^e)$.	$O(bd)$
Best-first ricorsiva	No	Solo se l'euristica è ammissibile	Dipende dalla funzione euristica	$O(bd)$.
Simplified Memory Bounded A*	Solo se la soluzione è raggiungibile	Solo se la soluzione raggiungibile è quella ottima	Potrebbe esplodere a causa del continuo passaggio da un cammino all'altro	Fino ad esaurimento memoria

IDA* sembra essere l'algoritmo che offre il migliore compromesso tra efficacia ed efficienza.

Riassunto e conclusioni

Un breve riassunto di quanto fatto, algoritmi informati

Algoritmi che sfruttano conoscenze specifiche del problema. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione in **ambienti osservabili, discreti, noti e deterministici**.

Algoritmi informati	Completezza	Ottimalità	Complessità temporale	Complessità spaziale
Best-first greedy	No	No	$O(b^m)$	$O(b^m)$
A*	Solo se il numero di nodi è finito e il loro costo è minore o uguale a C^*	Solo se l'euristica è ammissibile e consistente	$O(b^ε)$.	$O(b^m)$
Beam Search	No	No	$O(b^k)$	$O(kd)$.
IDA*	Solo se il numero di nodi è finito e il loro costo è minore o uguale a C^*	Solo se l'euristica è ammissibile e consistente	$O(b^ε)$.	$O(bd)$
Best-first ricorsiva	No	Solo se l'euristica è ammissibile e consistente	$O(b^ε)$.	$O(b^m)$
Simplified Memory Bounded A*	Solo se la soluzione è raggiungibile	Solo se la soluzione è raggiungibile e l'euristica è ammissibile e consistente	$O(b^ε)$.	$O(b^m)$

Definire un'euristica con la quale rinforzare la ricerca non è facile e, anzi, richiede esperienza e/o evidenze empiriche per essere efficace!

Riassunto e conclusioni

Un breve riassunto di quanto fatto, ricerca locale

Algoritmi che **non** effettuano una ricerca esaustiva dello spazio, **ma migliorano iterativamente una soluzione applicando modifiche locali**. Possono essere usati per tutte le tipologie di problemi in cui lo scopo è ricercare una soluzione, anche se non ottimale, in tempi ragionevoli.

Per definizione, **non garantiscono l'ottimalità**, sebbene riescano a trovare quasi sempre una soluzione (sono, perciò, **quasi-completi**). La loro complessità dipende da molti fattori che riguardano sia il problema per il quale sono sviluppato sia la loro progettazione.

Hill-Climbing

Semplice da implementare

Anche le varianti del classico Hill-Climbing sono semplici modifiche all'algoritmo di base.

Spesso non ottimo

Plateau, massimi/minimi locali e creste non consentono spesso di trovare una soluzione ottima.

Simulated Annealing

Semplice da implementare

Ha semplicemente una componente di esplorazione casuale in più rispetto ad HC.

Non facile da configurare

Le prestazioni dipendono fortemente dai parametri (velocità di raffreddamento e temperatura iniziale).

Algoritmi genetici

Flessibilità

Oltre ad essere facili da implementare, sono estremamente flessibili.

Flessibilità

La flessibilità può essere un problema, poiché potrebbe rendere complessa la loro progettazione.

Riassunto e conclusioni

Un breve riassunto di quanto fatto, ricerca con avversari

Algoritmi che possono essere utilizzati in **ambienti multi-agente**, dove due o più avversari alternano le proprie mosse fino al raggiungimento di un obiettivo.

Abbiamo per lo più trattato i **giochi a somma zero e con informazione perfetta**, ovvero situazioni in cui gli avversari conoscono le loro possibili mosse e in cui i valori di utilità finali sono uguali ma di segno opposto - ad esempio, se uno vince l'altro perde o, al massimo, i due avversari pareggiano.

Algoritmo Minimax

Semplice da modellare

I valori minimax dei due avversari sono calcolati sulla base dell'utilità di un nodo per un giocatore. Anche le versioni alternative (quelle stocastiche) sono semplici da modellare rispetto alla versione di base.

Complessità temporale

La rappresentazione dell'albero di gioco è spesso da considerare solo un costrutto teorico che non consente di trattare i problemi in maniera efficiente.

Algoritmo Alfa-Beta

Più efficienza

La potatura consente di tagliare interi sotto-alberi dell'albero di gioco, aumentando l'efficienza.

Ordinamento delle mosse

L'efficacia dipende fortemente dall'ordinamento delle mosse, che non è facile da calcolare.

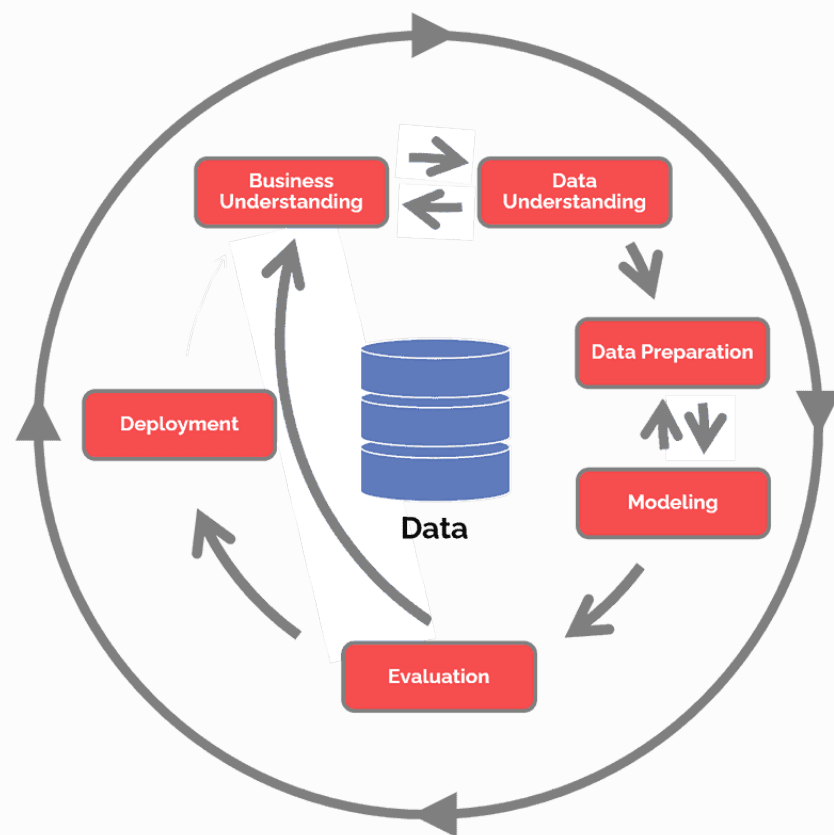
Nella maggior parte dei casi, tuttavia, conviene usare versioni euristiche di questi algoritmi per arrivare alla definizione di decisioni imperfette in tempo reale. **L'algoritmo H-Minimax** è perciò da considerare la versione migliore per la definizione di giochi.

Riassunto e conclusioni

Un breve riassunto di quanto fatto, problemi di apprendimento

Esistono diverse tipologie di algoritmi. Vengono solitamente classificati in algoritmi di apprendimento **supervisionato** e **non supervisionato**. I primi richiedono la specifica di un insieme di dati per cui variabili indipendenti e variabile dipendente sono noti. I secondi non richiedono un insieme di dati etichettato, ma mirano a trovare pattern all'interno dei dati di partenza.

Dato un problema di partenza, dovremo innanzitutto selezionare l'algoritmo da utilizzare sulla base di una fase di business e data understanding, la quale potrà fornirci indicazioni sui migliori algoritmi da utilizzare nel contesto specifico.



A prescindere dalla selezione effettuata, qualsiasi algoritmo ha necessità di ingegnerizzazione: il **machine learning engineering** è la branca dell'ingegneria del software che mira a definire i passi necessari per utilizzare un algoritmo di apprendimento in contesti reali.

Abbiamo parlato del CRISP-DM, il quale fornisce un modello di ciclo di vita che distingue le diverse fasi ingegneristiche che portano dai requisiti di partenza al rilascio di strumenti di machine learning.

Abbiamo discusso della particolare importanza della **data engineering**, la quale fornisce gli strumenti metodologici per comprendere, gestire, pulire e risolvere potenziali problemi nei dati —> è qui che i principali requisiti non-funzionali (sicurezza, privacy, etica) di un algoritmo di machine learning vengono identificati e diagnosticati! Senza una buona fase di data engineering riusciremo esclusivamente a sviluppare un prototipo senza alcuna valenza pratica!

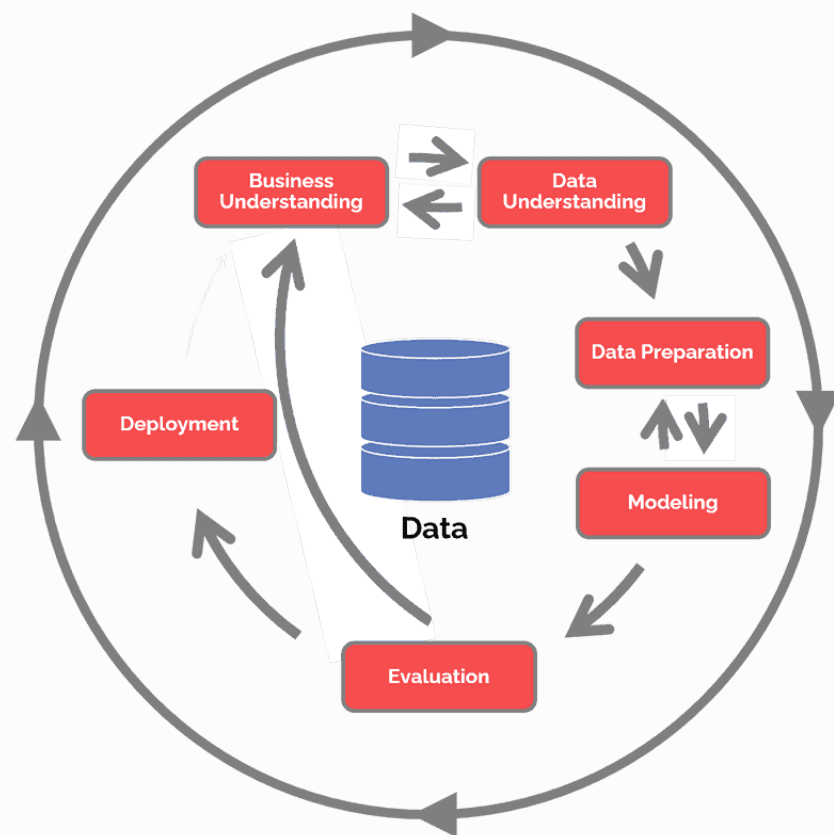
Solo dopo arriva la fase di **feature engineering**.

Riassunto e conclusioni

Un breve riassunto di quanto fatto, problemi di apprendimento

Esistono diverse tipologie di algoritmi. Vengono solitamente classificati in algoritmi di apprendimento **supervisionato** e **non supervisionato**. I primi richiedono la specifica di un insieme di dati per cui variabili indipendenti e variabile dipendente sono noti. I secondi non richiedono un insieme di dati etichettato, ma mirano a trovare pattern all'interno dei dati di partenza.

Dato un problema di partenza, dovremo innanzitutto selezionare l'algoritmo da utilizzare sulla base di una fase di business e data understanding, la quale potrà fornirci indicazioni sui migliori algoritmi da utilizzare nel contesto specifico.



Nel **feature engineering** passiamo dai dati alle caratteristiche, identificando le metriche/variabili indipendenti più rilevanti per il problema di classificazione/regressione/clustering considerato.

Questa fase si basa su (1) analisi dei dati e delle relazioni tra questi e (2) identificazione delle caratteristiche più rilevanti per la predizione.

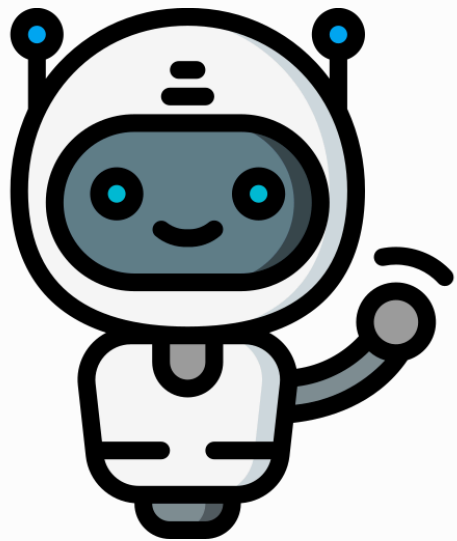
Gli algoritmi di apprendimento entrano in gioco solo adesso, ovvero dopo esserci assicurati che i dati di partenza: la sperimentazione empirica, abilitata da una buona scelta dei metodi di valutazione e delle metriche di valutazione da considerare, ci aiuterà a confrontare diversi modelli. Si sceglierà quello più performante o, in alcuni casi, quello che fornisce il miglior compromesso tra accuratezza delle predizioni e aspetti non funzionali (es., fairness, sicurezza, e altro).

Il tutto ciò considerando gli elementi di base della teoria dell'apprendimento, ovvero **bias, varianza e il loro compromesso**.

Riassunto e conclusioni

Un breve riassunto di quanto fatto, intelligenza artificiale at work

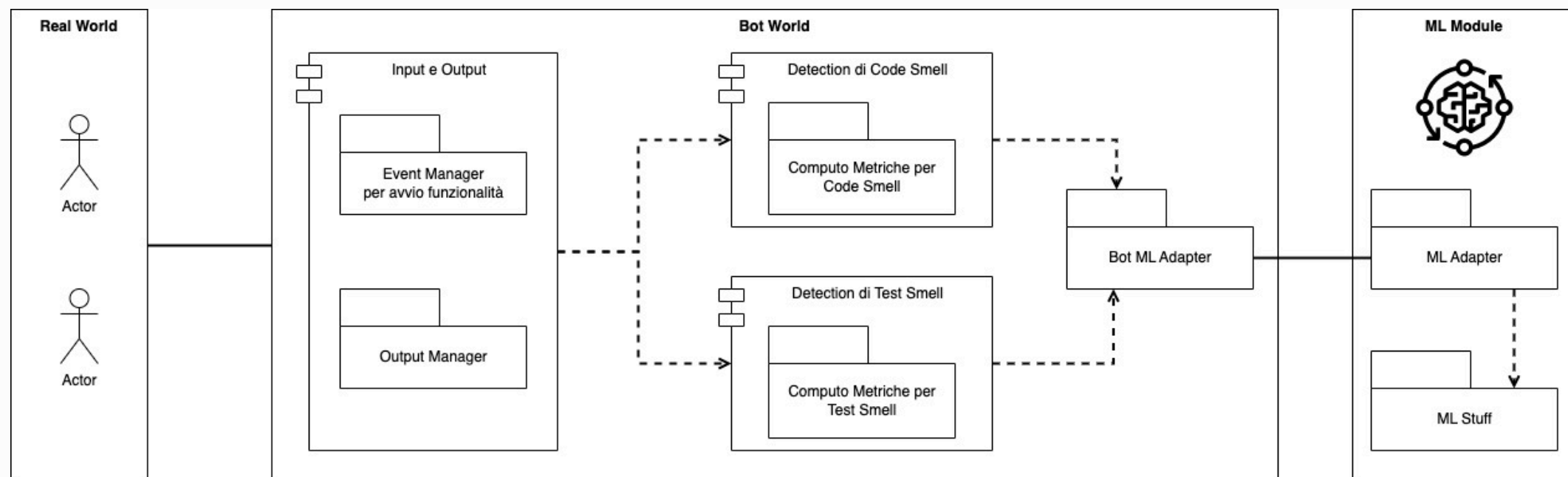
Tuttavia, progettare e sviluppare un algoritmo è, ahimè, solo parte del problema! Questi algoritmi devono necessariamente essere inclusi in sistemi software più grandi e complessi.



Nel contesto di questo corso, abbiamo approfondito due casi d'uso particolarmente utilizzati oggi. I **bot** e i **conversational agent**, ovvero degli assistenti virtuali che includono componenti di intelligenza artificiale e sono in grado di interagire con gli utenti tramite interfacce di **natural language processing**.

Abbiamo approfondito l'architettura dei bot, apprendendo come è possibile combinare un bot con un sistema software e come un sistema software può interfacciarsi con un sistema di intelligenza artificiale.

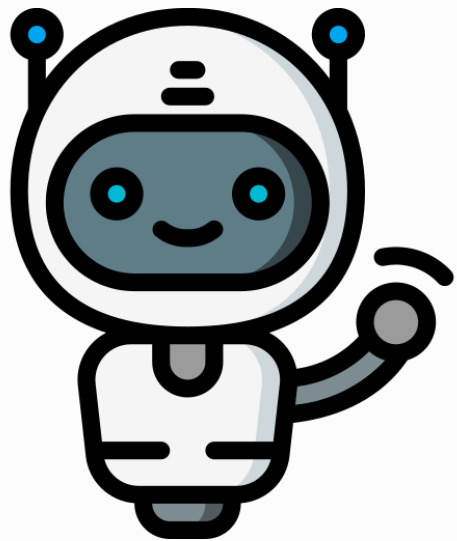
Abbiamo infine approfondito l'ambito dello **smart development**.



Riassunto e conclusioni

Un breve riassunto di quanto fatto, intelligenza artificiale at work

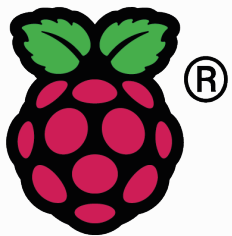
Tuttavia, progettare e sviluppare un algoritmo è, ahimè, solo parte del problema! Questi algoritmi devono necessariamente essere inclusi in sistemi software più grandi e complessi.



Nel contesto di questo corso, abbiamo approfondito due casi d'uso particolarmente utilizzati oggi. I **bot** e i **conversational agent**, ovvero degli assistenti virtuali che includono componenti di intelligenza artificiale e sono in grado di interagire con gli utenti tramite interfacce di **natural language processing**.

Abbiamo approfondito l'architettura dei bot, apprendendo come è possibile combinare un bot con un sistema software e come un sistema software può interfacciarsi con un sistema di intelligenza artificiale.

Abbiamo infine approfondito l'ambito dello **smart development**.



Nel secondo caso, i **sistemi IoT**. Abbiamo in particolare appreso come un sistema basato su intelligenza artificiale può interagire con i **big data**, acquisendo dati da sensori esterni e assicurandosi che questi siano di qualità sufficiente per poter essere utilizzati a fini predittivi e/o smart.

Abbiamo anche identificato gli elementi principali di **architetture IoT**, comprendendo il modo in cui i dispositivi IoT possono interagire con l'ambiente circostante.

Abbiamo inoltre approfondito come poter **simulare l'interazione con smart object**, così da poter abilitare la valutazione dell'efficacia ed efficienza di algoritmi di intelligenza artificiale in assenza di dispositivi fisici (oltre che abilitare il testing di unità degli algoritmi di intelligenza artificiale!)

Riassunto e conclusioni

Alcune note finali sui progetti e sugli esami

Documentazione progetto

- (1) **Non è previsto** un template per la documentazione del progetto!
- (2) **Bisogna rendere esplicita la metodologia** che ha portato alla definizione del progetto: ad esempio, dati gli algoritmi disponibili, perché ne è stato scelto uno piuttosto che un altro? Date diverse varianti dell'algoritmo selezionato, quale di queste meglio si adatta al problema? Quali sono, se necessarie, le strategie aggiuntive implementate per rendere gli algoritmi di base più efficaci/efficienti?
- (3) **Non c'è nessun vincolo** sul linguaggio di programmazione usato per l'implementazione. Anzi, se esistono delle librerie utili a semplificare il lavoro, **usatele!** Non è un esame di programmazione, ci interessa il ragionamento che porta alla definizione dell'agente, non che linguaggio avete usato per implementarlo.

Esami

- (1) Oltre alle date di Gennaio e Febbraio, **FIA avrà date anche nella sessione di recupero di Marzo/Aprile**. Tuttavia, non perdetevi l'occasione di completare il progetto nel prossimo mese e terminare l'esame nella sessione ordinaria.
- (2) Come detto più volte, la prova scritta può essere svolta in uno qualsiasi degli appelli di un semestre, **la discussione del progetto va fatta insieme al gruppo**.
- (3) Quando il progetto sarà terminato e vorrete discuterlo, per favore **inviate la documentazione prodotta prima della discussione**.

Riassunto e conclusioni

Alcune note finali sui progetti e sugli esami

Regole d'ingaggio

Che siano progetti combinati o non combinati, a Gennaio saranno premiati i migliori progetti. I vincitori saranno premiati con un award e una medaglia. Ci saranno tre progetti vincitori: uno votato da me, uno votato dai tutor e il terzo votato da voi!

Precondizione: Per ambire ad un FIA Award, **i progetti dovranno essere consegnati al pre-appello.**

Criteri di valutazione per il premio assegnato da me

- Qualità del progetto, in termini di solidità dei metodi applicati per la risoluzione del problema trattato;
- Qualità del report prodotto;
- Qualità dell'approccio allo sviluppo adottato: impariamo ad essere professionali ed approcciare lo sviluppo in maniera sistematica, tramite l'utilizzo appropriato di version control system;

Criteri di valutazione per il premio assegnato dai tutor

- Qualità della presentazione finale e qualità di esposizione del progetto;
- Innovatività ed impatto della tematica trattata nel contesto del progetto;

Criteri di valutazione per il premio assegnato da voi

- Premio vostro, criteri vostri!



Una veloce overview sulle possibili tesi in FIA

Possibili Argomenti su Tirocini e Tesi in FIA

Premessa. L'argomento di tesi non è assegnato/imposto, ma discusso e personalizzato sulla base degli interessi dello studente.

Lo scorso anno, gli oltre 100 tesisti che hanno deciso di portare avanti argomenti relativi all'intelligenza artificiale e all'ingegneria del machine learning si sono focalizzati su disparate tematiche, dall'uso di deep gravity fino a all'uso di IA per la prevenzione di attacchi terroristici e allo sviluppo di tool per smart development.

Per gli interessati: <https://github.com/SeSaLabUnisa/Tesi>.

In generale, la tesi (4 CFU) è generalmente associata al tirocinio (11 CFU), ma questo *non* è obbligatorio. Nel tirocinio ci si focalizza sull'apprendimento di argomenti vicini al corso, ma che vanno oltre quello che si è già studiato. Nella tesi le conoscenze acquisite si mettono in pratica - il periodo di tesi porta alla scrittura di un documento.

Sicurezza

- Cyberthreat Intelligence;
- Privacy e sicurezza di sistemi IoT;
- Analisi e predizione di vulnerabilità software.

Research-based ML

- Utilizzo di algoritmi di ricerca per la definizione di modelli predittivi.

AI & Mobile Dev.

- Definizione di algoritmi smart per lo sviluppo di applicazioni.

Smart development

- Creazione di conversational agents;
- Analisi predittive sullo sviluppo software.

AI & HCI

- Definizione di tecniche di IA per la risoluzione di problemi legati alle GUI.

Explainable AI

- Definizione di algoritmi che "spiegano" il funzionamento di ML.

Una veloce overview sulle possibili tesi in FIA

Possibili Argomenti su Tirocini e Tesi in FIA

Premessa. L'argomento di tesi non è assegnato/imposto, ma discusso e personalizzato sulla base degli interessi dello studente.

Lo scorso anno, gli oltre 100 tesisti che hanno deciso di portare avanti argomenti relativi all'intelligenza artificiale e all'ingegneria del machine learning si sono focalizzati su disparate tematiche, dall'uso di deep gravity fino a all'uso di IA per la prevenzione di attacchi terroristici e allo sviluppo di tool per smart development.

Per gli interessati: <https://github.com/SeSaLabUnisa/Tesi>.

In generale, la tesi (4 CFU) è generalmente associata al tirocinio (11 CFU), ma questo *non* è obbligatorio. Nel tirocinio ci si focalizza sull'apprendimento di argomenti vicini al corso, ma che vanno oltre quello che si è già studiato. Nella tesi le conoscenze acquisite si mettono in pratica - il periodo di tesi porta alla scrittura di un documento.

AI & Mobile Dev.

- Definizione di algoritmi smart per lo sviluppo di applicazioni.

Quantum SE

- Definizione di studi e algoritmi per lo sviluppo di applicazioni quantistiche.

Software engineering for Artificial Intelligence

- Definizione di studi e strumenti per il design, rilascio di sistemi di requisiti (non-)funzionali di sistemi di intelligenza artificiale.

Explainable AI

- Definizione di algoritmi che "spiegano" il funzionamento di ML.

Natural Language Processing

- Definizione e analisi di strumenti di natural language processing

Diversity and Inclusion

- Definizione compensativi per persone con Disturbi Specifici dell'Apprendimento o con disabilità

Una veloce overview sulle possibili tesi in FIA

Possibili Argomenti su Tirocini e Tesi in FIA

Premessa. L'argomento di tesi non è assegnato/imposto, ma discusso e personalizzato sulla base degli interessi dello studente.

Lo scorso anno, gli oltre 100 tesisti che hanno deciso di portare avanti argomenti relativi all'intelligenza artificiale e all'ingegneria del machine learning si sono focalizzati su disparate tematiche, dall'uso di deep gravity fino a all'uso di IA per la prevenzione di attacchi terroristici e allo sviluppo di tool per smart development.

Per gli interessati: <https://github.com/SeSaLabUnisa/Tesi>.

In generale, la tesi (4 CFU) è generalmente associata al tirocinio (11 CFU), ma questo *non* è obbligatorio. Nel tirocinio ci si focalizza sull'apprendimento di argomenti vicini al corso, ma che vanno oltre quello che si è già studiato. Nella tesi le conoscenze acquisite si mettono in pratica - il periodo di tesi porta alla scrittura di un documento.

Ottimizzazione e testing di requisiti non funzionali

- Uso di tecniche di IA per l'ottimizzazione di requisiti non funzionali di sistemi basati su IA.

Virtual/Augmented/Mixed Reality

- Uso di tecniche di IA per l'ottimizzazione di esperienze utente basate su VR/AR/MR.

Comprensione di programmi

- Definizione di tecniche di IA per comprendere automaticamente testi e codice sorgente.

Software engineering for Metaverse

- Ingegnerizzazione di metaversi a scopo educativo.

Una veloce overview sulle possibili tesi in FIA

Possibili Argomenti su Tirocini e Tesi in FIA

Premessa. L'argomento di tesi non è assegnato/imposto, ma discusso e personalizzato sulla base degli interessi dello studente.

Lo scorso anno, gli oltre 100 tesisti che hanno deciso di portare avanti argomenti relativi all'intelligenza artificiale e all'ingegneria del machine learning si sono focalizzati su disparate tematiche, dall'uso di deep gravity fino a all'uso di IA per la prevenzione di attacchi terroristici e allo sviluppo di tool per smart development.

Per gli interessati: <https://github.com/SeSaLabUnisa/Tesi>.

In generale, la tesi (4 CFU) è generalmente associata al tirocinio (11 CFU), ma questo *non* è obbligatorio. Nel tirocinio ci si focalizza sull'apprendimento di argomenti vicini al corso, ma che vanno oltre quello che si è già studiato. Nella tesi le conoscenze acquisite si mettono in pratica - il periodo di tesi porta alla scrittura di un documento.

nted/Mixed Reality

IA per l'ottimizzazione di
basate su VR/AR/MR.

gineering for Metaverse

ione di metaversi a scopo
educativo.

Ulteriori incentivi: Discord channel e facilities.

Per gli interessati: <https://drive.google.com/file/d/1fDsvvfHJuHxX0ti2Xh1182ZYG2NmCLPJ/view?usp=sharing>.

sesalab@unisa.it 

sesa_lab   

<https://sesalabunisa.github.io/> 

Una veloce overview sulle possibili tesi in FIA

Possibili Argomenti su Tirocini e Tesi in FIA

Premessa. L'argomento di tesi non è assegnato/imposto, ma discusso e personalizzato sulla base degli interessi dello studente.

Lo scorso anno, gli oltre 100 tesisti che hanno deciso di portare avanti argomenti relativi all'intelligenza artificiale e all'ingegneria del machine learning si sono focalizzati su disparate tematiche, dall'uso di deep gravity fino a all'uso di IA per la prevenzione di attacchi terroristici e allo sviluppo di tool per smart development.

Per gli interessati: <https://github.com/SeSaLabUnisa/Tesi>.

In generale, la tesi (4 CFU) è generalmente associata al tirocinio (11 CFU), ma questo *non* è obbligatorio. Nel tirocinio ci si focalizza sull'apprendimento di argomenti vicini al corso, ma che vanno oltre quello che si è già studiato. Nella tesi le conoscenze acquisite si mettono in pratica - il periodo di tesi porta alla scrittura di un documento.



Possibilità di tirocinio esterno convenzionato, ad esempio con Kiranet. Per più info, potete scrivere a me o alla e-mail del lab.

sesalab@unisa.it 

sesa_lab   

<https://sesalabunisa.github.io/> 

Una veloce overview sulle possibili tesi in FIA

Possibili Argomenti su Tirocini e Tesi in FIA

Premessa. L'argomento di tesi non è assegnato/imposto, ma discusso e personalizzato sulla base degli interessi dello studente.

Lo scorso anno, gli oltre 100 tesisti che hanno deciso di portare avanti argomenti relativi all'intelligenza artificiale e all'ingegneria del machine learning si sono focalizzati su disparate tematiche, dall'uso di deep gravity fino a all'uso di IA per la prevenzione di attacchi terroristici e allo sviluppo di tool per smart development.

Per gli interessati: <https://github.com/SeSaLabUnisa/Tesi>.

In generale, la tesi (4 CFU) è generalmente associata al tirocinio (11 CFU), ma questo *non* è obbligatorio. Nel tirocinio ci si focalizza sull'apprendimento di argomenti vicini al corso, ma che vanno oltre quello che si è già studiato. Nella tesi le conoscenze acquisite si mettono in pratica - il periodo di tesi porta alla scrittura di un documento.

That's all folks!

Ci rivediamo, per chi volesse, alla magistrale in Software Engineering and IT Management!

sesalab@unisa.it 

sesa_lab   

<https://sesalabunisa.github.io/> 

Riassunto e conclusioni

Questionario di valutazione

Per finire, vi chiederei di compilare il seguente questionario sulla valutazione della parte di ricerca del corso. Il questionario è **anonimo** e **non influirà in nessun modo** sull'esame e sulla sua complessità! Serve esclusivamente a migliorare il corso.

Link: <https://forms.gle/QSPcK97PjXZNpZ289>





UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

Fondamenti di Intelligenza Artificiale

Lezione 20 - Riassunto e conclusioni

