

# Laboratorio di Intelligenza Artificiale

---

- Obiettivi formativi
- Modalità e Valutazione
- Proposte di progetti

E. Lamma, M. Gavanelli, F. Riguzzi, R. Zese, E. Bellodi  
*nome.cognome@unife.it*

# Laboratorio di Intelligenza Artificiale

---

- Laboratorio di 3 CFU di ambito F, associabile ai moduli/insegnamenti AA2020/21 di:
  - Fondamenti di Intelligenza Artificiale
  - Data mining and Analytics
  - Intelligenza Artificiale per l'ottimizzazione vincolata
  - Deep learning (per chi ha seguito AA2019/20, o lo seguirà il prossimo AA)
- Docenti: Lamma, Riguzzi, Gavanelli - Zese

# Obiettivi formativi

---

- Progettare e sviluppare un'applicazione di IA usando sistemi e linguaggi di Intelligenza Artificiale
- Due coordinate:
  - problema da affrontare
  - strumento/ientrambi da concordare
- Progetto da concordare con uno o più docenti
- Svolgimento anche in **gruppo** (max 3 persone), combinabile anche con progetti di altri corsi



# Modalità e valutazione

---

- Il progetto viene assegnato dopo aver sostenuto con successo l'esame di Fondamenti di IA/oppure Deep learning, o durante il II semestre
- La data di discussione va concordata con uno dei docenti (che inviterà anche gli altri docenti a partecipare)
- Alla discussione del progetto, lo studente/il gruppo è tenuto a:
  - inviare almeno 2 giorni prima una relazione sintetica del lavoro svolto (max. 6 pagine);
  - effettuare una presentazione della durata di 15 minuti ca. (max. 10 slide);
  - mostrare una demo del sistema realizzato.

# Problemi (esempi)

---

- Ricerca di soluzioni
  - Soluzione automatica del problema del “parcheggio”
- Giochi a due avversari (*any*), più team, ogni team sviluppa la sua strategia di gioco
  - Giochi - forza 4 (alfa-beta)
- Ragionatori dedicati (ontologie, probabilità)
- Problemi a vincoli
  - TSP, problema del commesso viaggiatore
  - Scheduling, vincoli dedicati

# Problemi (esempi)

---

- Deep learning e deep network
  - Riconoscimento da immagini
  - Classificazione da immagini
- Classificazione di testi (giochi matematici)
- Agenti conversazionali, chatbot

## (Alcuni) strumenti

---

- AIMA.search, per modellazione e soluzione problemi di ricerca
- <https://artint.info/AIPython/> <https://aispace2.github.io/AISpace2/>
- Prolog
- Protegé (editor di ontologie)
- Pellet (ragionatore ontologico)

### *Ma anche:*

- Ragionamento probabilistico (cplint)
- Apprendimento automatico (Weka, cplint, etc)
- Linguaggi a vincoli (Eclipse, miniZinc, Answer Set Programming, etc)
- Tensorflow, MS Cognitive Toolkit CNTK, etc
- DialogFlow

# Proposte di progetti

---

Knowledge Representation (KR)

Ragionamento automatico (AR)

Machine learning (ML)

Deep learning (DL)

Analisi testi (NLP)

Constraint Processing (CP)

# Sistema a regole per spiegare messaggi d'errore dei compilatori (KR+NLP)

---

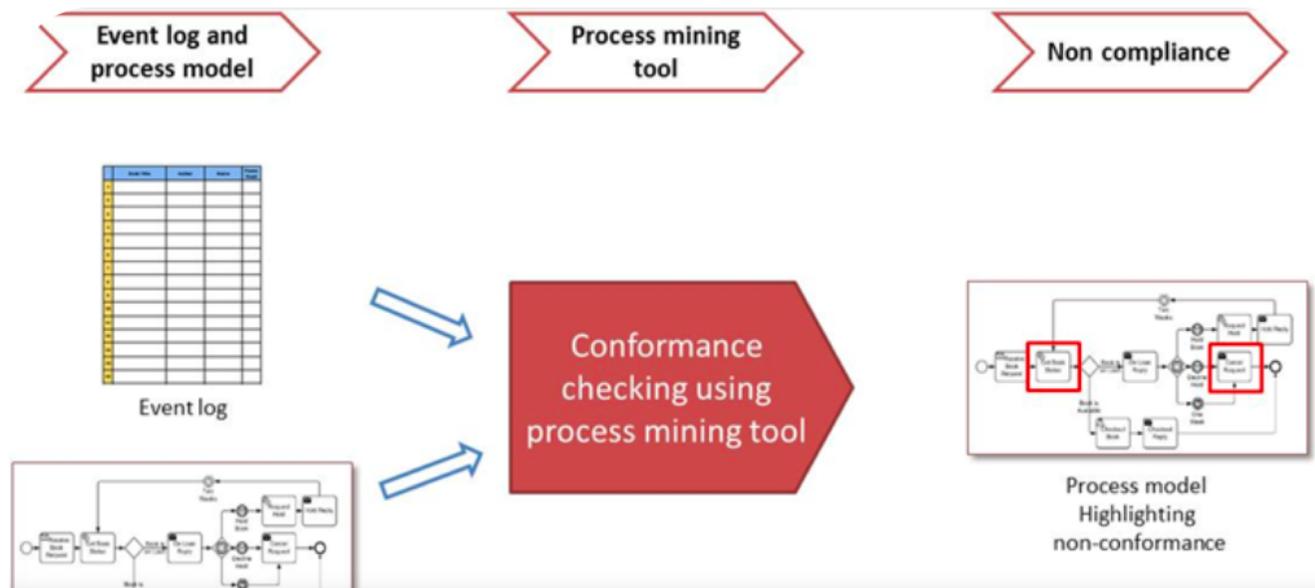
- Sistema per correzione semi-automatica dei programmi C (o Java) degli studenti del I anno
- In alcuni casi, consegnano programmi che non si compilano (o con warning di compilazione)
- Dati
  - messaggio di errore
  - programma
- stampare un messaggio intuitivo per lo studente
- **warning: implicit declaration of function 'f'**
- *“Ordine funzioni errato”*
- **error: 'Q' is a pointer; did you mean to use '->'?**
- *“Controllare priorità degli operatori \* e .”*

```
typedef struct
{   int x,y; } punto;

main()
{   punto P;
    f(&P);
}

void f(punto *P)
{   *P.x=0;
}
```

# Business Process Management (ML / KR+AR)



## Workflow Mining: A Survey of Issues and Approaches

W.M.P. van der Aalst<sup>1</sup>, B.F. van Dongen<sup>1</sup>, J. Herbst<sup>2</sup>, L. Maruster<sup>1</sup>, G. Schimm<sup>3</sup>, and A.J.M.M. Weijters<sup>1</sup>

- Workflow mining extracts information about processes from transaction logs
- Process mining for the method of distilling a structured process description from a set of real executions.

# Business Process Management (ML)

case identifier	task identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

Table 2. A workflow log.

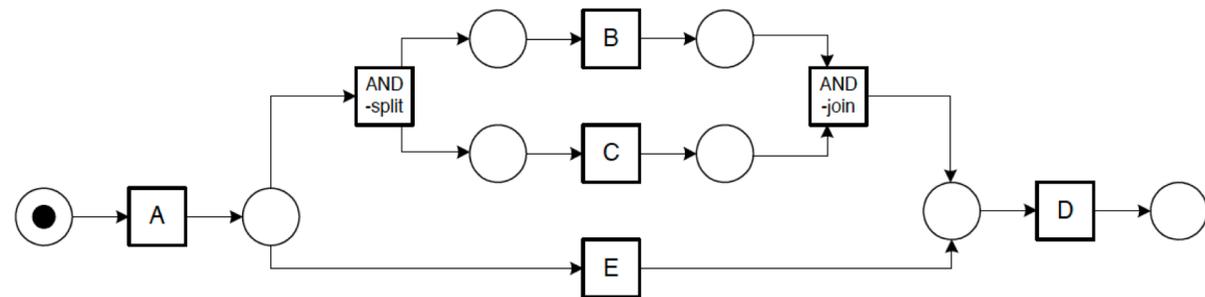


Fig. 3. A process model corresponding to the workflow log.

- Apprendere da dati (tracce o log di eventi), modelli simbolici in logica  
l'ordine, ovvero *integrity constraint* (implicazioni) della forma:

$$L_1, \dots, L_m \rightarrow A_1; \dots; A_n$$

- Applicare un algoritmo di ML, denominato PASCAL, sviluppato a UNIFE (apprende vincoli con probabilità)
- Dati del dominio medico, o altri domini

# Motore forward (KR+AR)

---

- Per un linguaggio del I ordine, con *integrity constraint* (implicazioni) della forma:

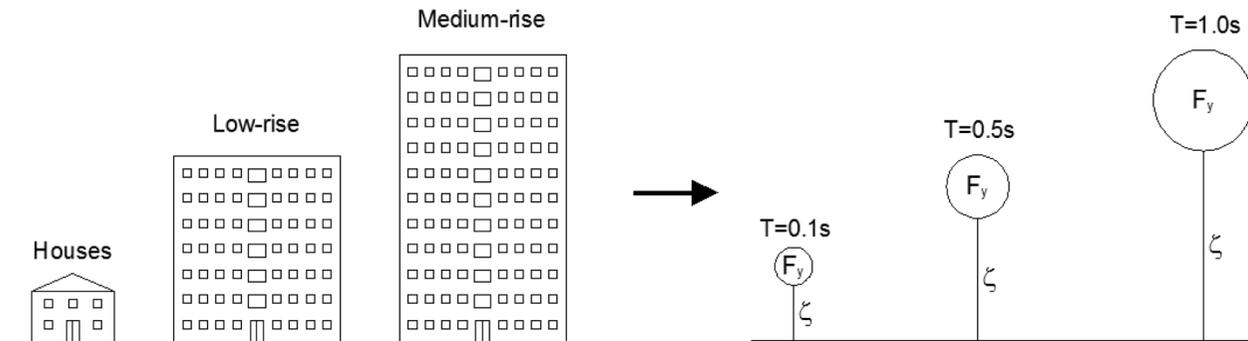
$$L_1, \dots, L_m \rightarrow A_1; \dots; A_n \quad (*)$$

- realizzare un motore inferenziale forward
- Utilizzare con Constraint Handling Rule (CHR):
  - semplificazione  $h_1, \dots, h_n \iff g_1, \dots, g_m \mid b_1, \dots, b_o.$
  - propagazione  $h_1, \dots, h_n \implies g_1, \dots, g_m \mid b_1, \dots, b_o.$
- Verificare la **conformance** di un nuovo log di eventi (ground) a una teoria espressa da vincoli del tipo (\*) più eventuale programma logico di background
- Integrare anche probabilità (altro progetto, a valle)

# Valutazione del rischio sismico (ML)

---

- Applicazione di ML a dati forniti da ingegneri civili di UNIFE



- MatLab: SVM, regressione
- Weka, alberi decisionali (DT, RF)

# Deep neural networks (DL)

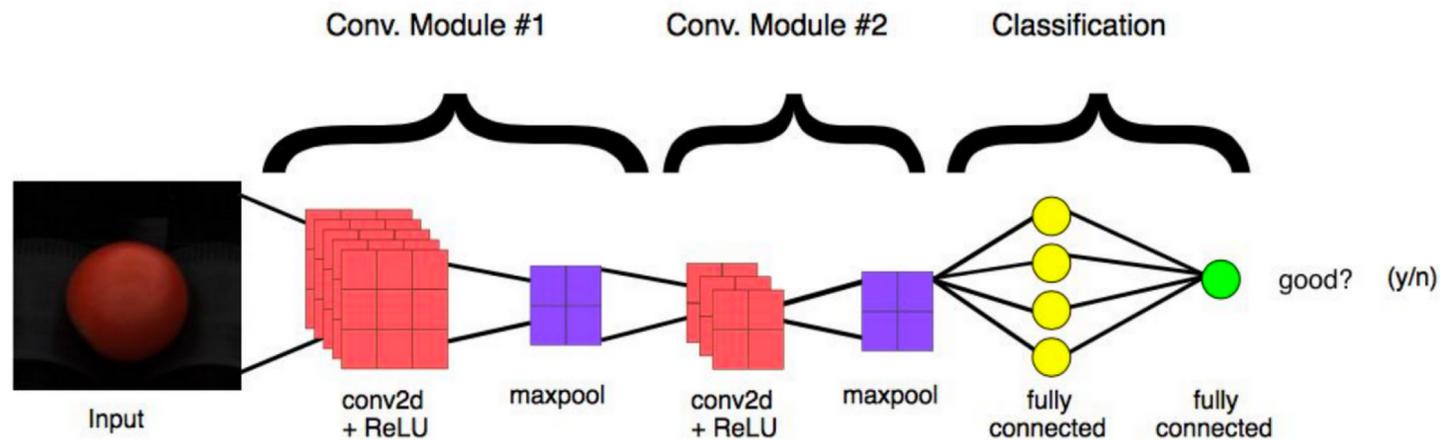
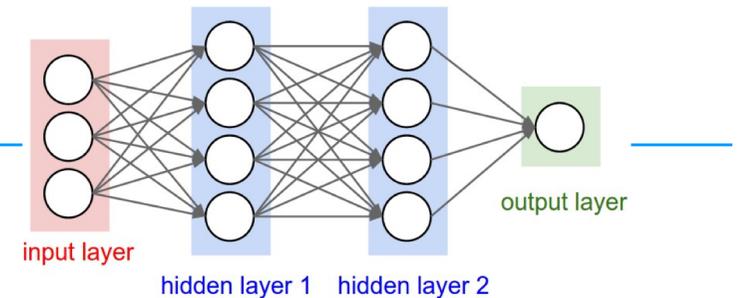


Figura 4. Esempio di una tipica rete convoluzione composta da due moduli di convoluzione e due layer fully connected nella parte finale.

- Per riconoscimento di immagini
- Per Natural Language Processing (NLP)

# Classificazione di difetti in odontoiatria (DL)

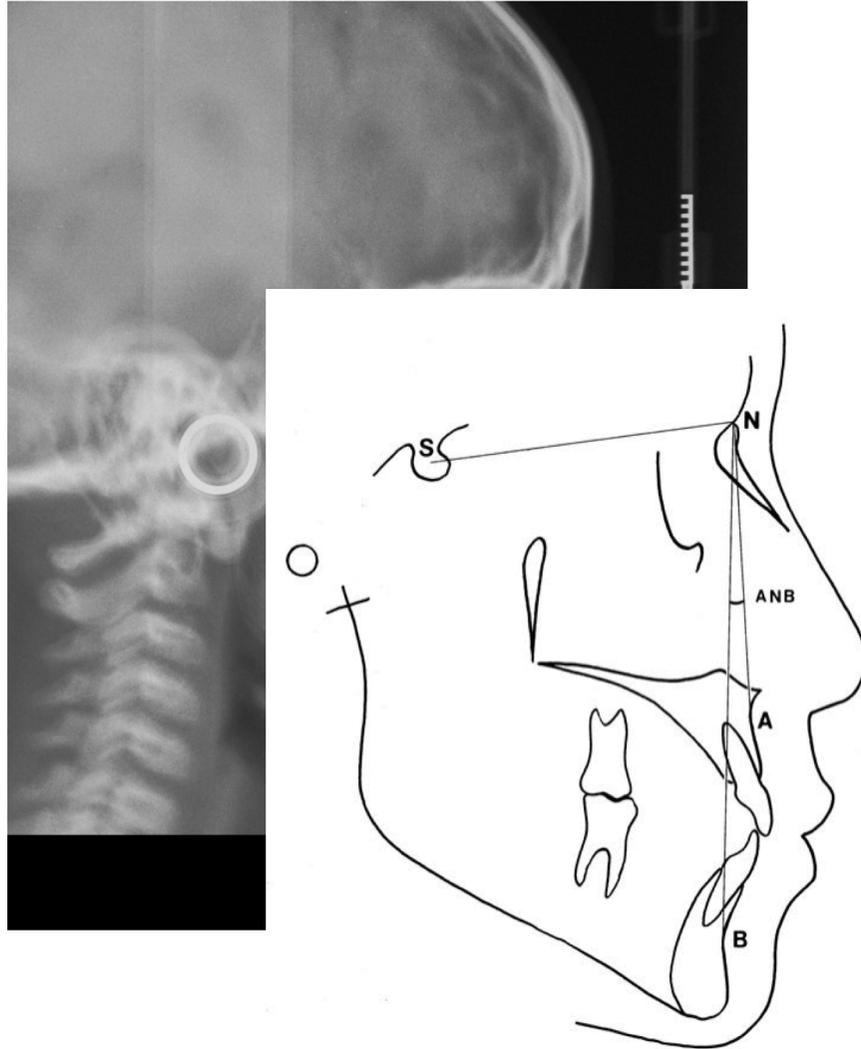
---



- In odontoiatria si analizzano immagini radiologiche per determinare la severità delle malocclusioni
- 3 classi (la differenza sta nel rapporto tra mascellare superiore e mandibola)
- Dataset fornito dal Dip. Medicina traslazionale e della Romagna
- oltre 1000 immagini

# Classificazione di difetti in odontoiatria (DL)

---



- In odontoiatria si analizzano immagini radiologiche per determinare la severità delle malocclusioni
- 3 classi (la differenza sta nel rapporto tra mascellare superiore e mandibola)
- Dataset fornito dal Dip. Medicina traslazionale e della Romagna
- oltre 1000 immagini

# Riconoscimento/classificazione difetti (DL)



[home](#) [chi siamo](#) [tecnologie frutti](#) [Qualità Coerente](#) [altre tecnologie](#) [risultati](#) [news](#) [contatti](#)



Unitec. Innovative linee, impianti e macchine per la lavorazione dell'Ortofrutta.

- Immagini di ortaggi, frutta, etc.

# Identificazione e classificazione difetti (DL+ML)

---

- Machine learning for Vision Inspection (ML4VI)

**bi-r3x**  
Big Data Innovation & Research Excellence



- Progetto finanziato che coinvolge:
  - Marposs
  - SACMI
  - Philip Morris Int.I, sede di Bologna
- Anche argomenti per tesi

# Analisi di Social Networks (NLP)

---

- Facebook For Developers
  - Sviluppo di programmi che, sfruttando le API di Facebook, permettono di collezionare qualsiasi tipo di dato disponibile pubblicamente in una pagina Facebook

facebook



Università degli  
Studi di Ferrara  
@unife.it

Home

Informazioni

Post

Gruppi

Eventi

Video

Foto

Recensioni

Community Netiquette

# AutoWeka per TSP Euclideo (ML + CP)

- Nel TSP Euclideo (problema del commesso viaggiatore, dove la distanza e` la distanza euclidea), una soluzione ottima non ha mai incroci.
- Abbiamo quindi implementato un nuovo vincolo che elimina gli incroci
- Questo va imposto per ogni coppia di vertici (città):  $O(n^2)$  vincoli
- Non tutti questi vincoli sono efficaci: alcuni vengono attivati molte volte e fanno poco pruning (overhead)
- Idea: cercare di apprendere quali vincoli faranno più pruning e imporre solo quelli. Uso tecniche di ML tramite il tool AutoWeka<sup>1</sup>



<sup>1</sup> <http://www.cs.ubc.ca/labs/beta/Projects/autoweka/#>

# cplint on SWISH: <http://cplint.eu/> (AR)

cplint on SWISH is a web application for probabilistic logic programming with a Javascript-enabled browser. [About](#) [Help](#) [Credits](#) [Dismiss](#)  
**New:** LEMUR (examples), coupon collector problem, random walk, Markov Logic Networks, BDD drawing: coin, path, dice, epidemic

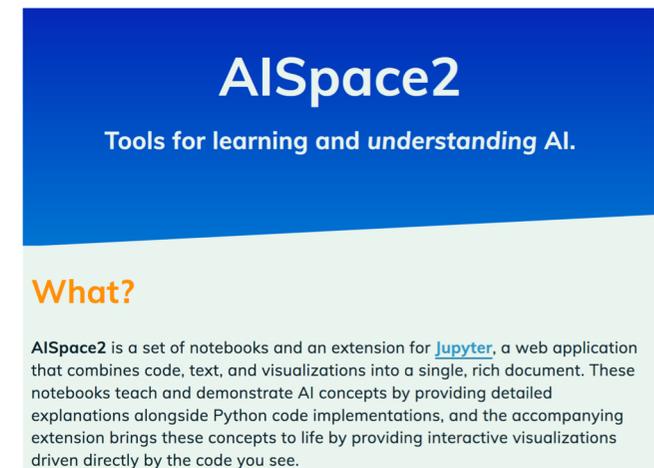
The screenshot shows the cplint on SWISH web application interface. At the top, there is a navigation bar with the logo, the title "cplint on SWISH", and menu items: File, Edit, Examples, Help, and Tutorial. A search box is located on the right side of the navigation bar. Below the navigation bar, there is a "New tab" button and a "Create a" section with "Program" and "Notebook" buttons. Underneath, there is a "based on" section with buttons for "Empty", "PITA", "MCINTYRE", "SLIPCOVER", "LEMUR", "ALEPH", "Student", and "CLP". Below this, there is an "Open source file containing" section with a search box and buttons for "Start of line", "Start of word", and "Anywhere". The main content area is divided into two sections. The top section features a large, stylized logo consisting of interlocking red and grey loops. The bottom section is a query input area with a placeholder text "Your query goes here ..." and buttons for "Examples", "History", and "Solutions". At the bottom right of the query input area, there is a checkbox for "table results" and a "Run!" button.

Reasoner Prolog esteso con probabilità  
Attività di sperimentazione (ad es. giochi )

# Confrontare AISpace2 con Belief propagation (AR)

---

- Confrontare AISpace2 <https://aispace2.github.io/AISpace2/> con <http://ml.unife.it/belief-propagation/> sulle reti bayesiane a <https://www.bnlearn.com/bnrepository/>
- Verificare fino a che dimensione di rete bayesiana funzionano i due sistemi
- Trovare limiti e pregi dei due sistemi



# Parallelizzazione reasoner ontologico Prolog (AR)

- SWI Prolog mette a disposizione predicati per parallelizzare l'esecuzione di programmi Prolog, gestendo diversi thread.
- Reasoner ontologico (TRILL on SWISH) scritto in Prolog

The screenshot shows the TRILL on SWISH web application interface. At the top, a banner reads: "TRILL on SWISH is a web application for a Javascript-enabled browser which embeds the tableau reasoners TRILL, TRILL<sup>P</sup> and TORNADO. [About](#) [Help](#) [Dismiss](#)". Below this is a navigation bar with "TRILL on SWISH" logo, "File", "Edit", "Examples", and "Help" menus. A search bar and a notification bell are also present. The main content area is divided into two panels. The left panel contains a "New tab" button, a "Create a" section with "Program" and "Notebook" buttons, and a "based on" section with "Empty", "TRILL", "TRILL<sup>P</sup>", "TORNADO", and "OwLRDF" options. Below this is a "Find files" search bar with "Filter" and "Type" dropdowns. A message states "No matching files" and provides instructions for new users. The right panel displays a large, abstract, pinkish-red geometric shape. At the bottom of the interface, there is a query input field with the placeholder text "?- Your query goes here ...", a "Run!" button, and a "table results" checkbox. Navigation buttons for "Examples", "History", and "Solutions" are also visible.

# Parallelizzazione reasoner ontologico Prolog (AR)

---

- Il ragionatore ontologico TRILL è scritto in SWI Prolog e, in fase di caricamento, deve analizzare gli assiomi (informazioni) contenuti in una ontologia, caricandola in memoria.
  - Il caricamento di ogni assioma è indipendente dagli altri.
- Obiettivo del progetto è la parallelizzazione del caricamento iniziale della ontologia, creando una coda di assiomi da analizzare che viene scorsa dai vari thread, in modo da gestire più assiomi contemporaneamente.

# Implementazione di nuovi vincoli in Programmazione Logica a Vincoli (CP)

---

- Nei linguaggi di programmazione logica a vincoli sono già presenti numerosi vincoli
- Altri vincoli possono essere aggiunti, per sfruttare algoritmi di propagazione nuovi o pubblicati in letteratura