# Learning in AI Planning

Dra. Mª Dolores Rodríguez Moreno

# Objectives

**Specific Objectives**

- Understand what ML is
- Know main ML techniques in AI Planning

**Source**

- Daniel Borrajo. Slides. Aprendizaje automático en Planificación automática.
- Stuart Russell & Peter Norvig (2009). Artificial Intelligence: A Modern Approach. Chapter 3, 10. (3rd Edition). Ed. Pearsons

# Outline

- ML
- Learning agents
- ML in planning
- Learning by macro-operators
- Explanation-Based Learning (EBL)
- Learning by Analogy
- Portfolios
- Conclusions

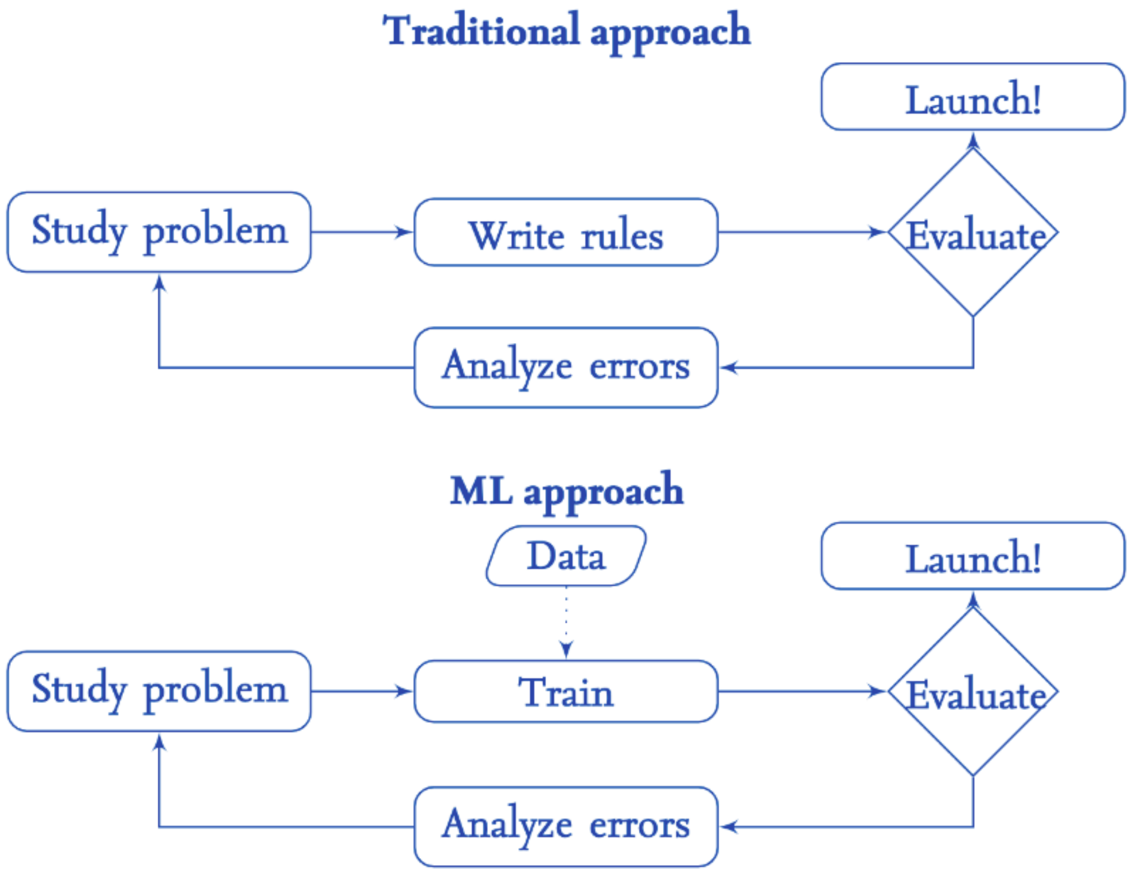
# Definition

| ML definition |
| --- |
| ML is the science (and art) of programming computers so they can learn form data A. Géron, 2017 |

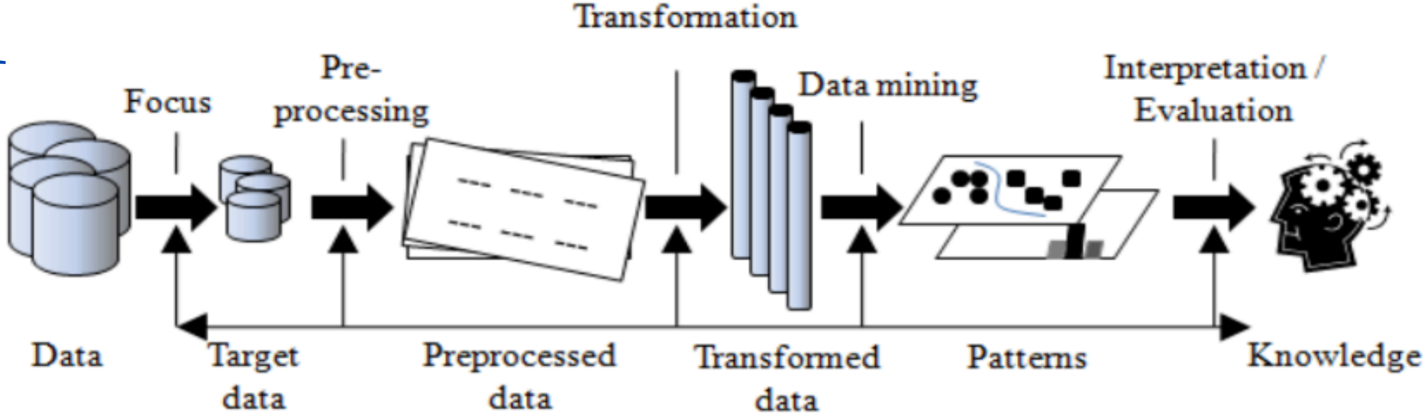
Alternative definitions

- Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel, 1959
- A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience. E. Tom Mitchell, 1997

# Definition


Traditional approach
Launch!
Study problem
Write rules
Evaluate
Analyze errors
ML approach
Data
Launch!
Study problem
Train
Evaluate
Analyze errors

# Definition




(Source)

Steps in any ML application:

1. Data adquisition
2. Selection, cleaning and transformation
3. Machine Learning
4. Learning evaluation
5. Explotation

The goal in ML is to get a representation of those patterns

# Outline

- ML
- **Learning agents**
- ML in planning
- Learning by macro-operators
- Explanation-Based Learning (EBL)
- Learning by Analogy
- Portfolios
- Conclusions

## Purpose

- Learning is essential for unknown environments
  - i.e., when designer lacks omniscience
- Learning is useful as a system construction method
  - i.e., expose the agent to reality rather than trying to write it down
- Learning modifies the agent's decision mechanisms to improve performance

# Elements

- Design of a learning element is affected by
  - Which components of the performance element are to be learned
  - What feedback is available to learn these components
  - What representation is used for the components
- Type of feedback:
  - Supervised learning: correct answers for each example
  - Unsupervised learning: correct answers not given
  - Reinforcement learning: occasional rewards

# Classification

- Inductive L: inferring new knowledge from observations (not guaranteed correct)
  - Concept/Classification L: identify characteristics of class members (e.g., what makes a CS class fun, what makes a customer buy, etc.)
  - Unsupervised L: examine data to infer new characteristics (e.g., break chemicals into similar groups, infer new mathematical rule, etc.)
  - Reinforcement L: learn appropriate moves to achieve goals (e.g., win a game of Checkers, perform a robot task, etc.)
- Deductive L: recombine existing knowledge to effectively solve problems
- Analytical L: reuse the acquired knowledge when presented with similar problems in the future (avoid bad decisions)

# Outline

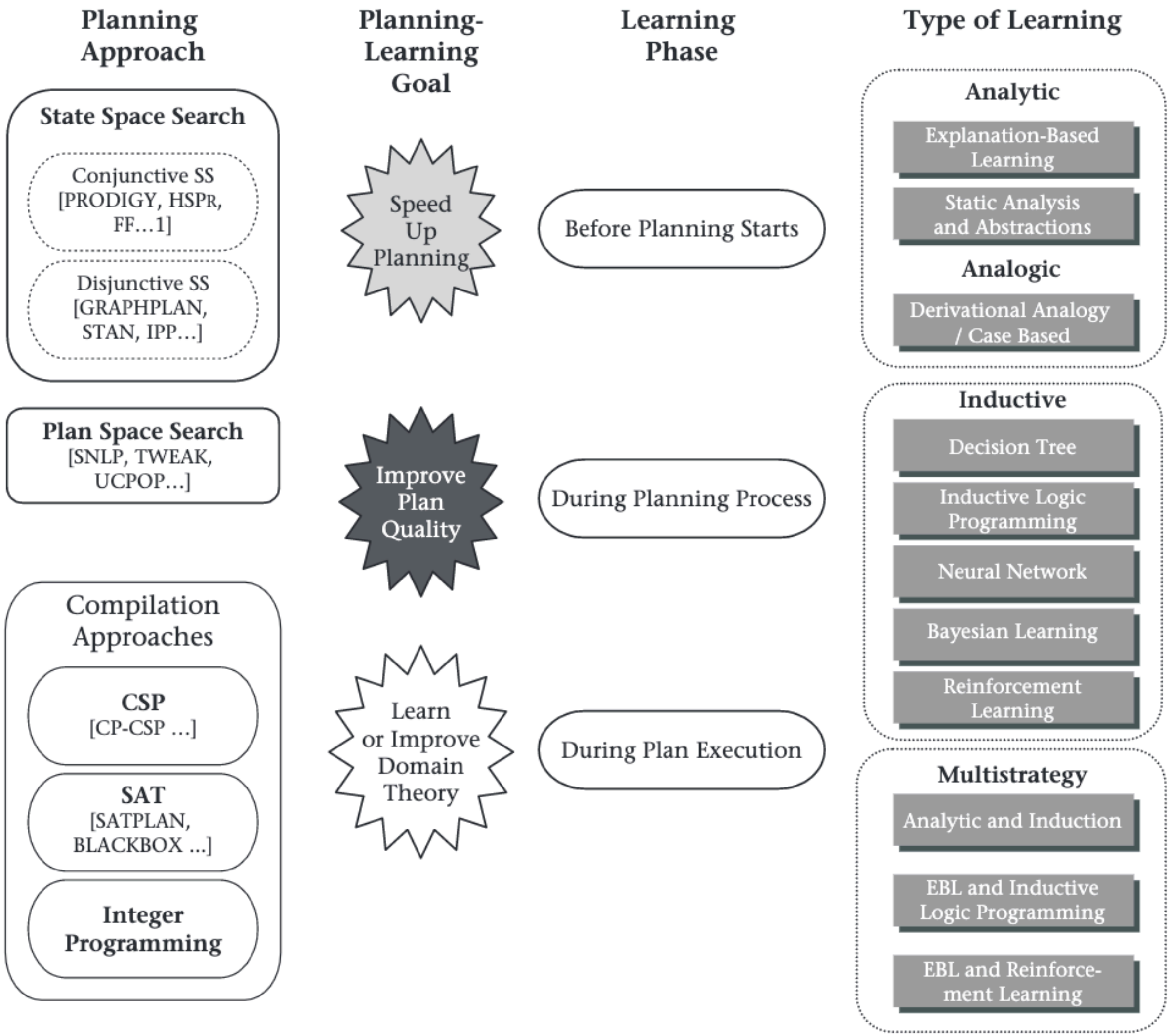
- ML
- Learning agents
- **ML in planning**
- Learning by macro-operators
- Explanation-Based Learning (EBL)
- Learning by Analogy
- Portfolios
- Conclusions

# ML in Planning (I)

- Planners efficiency have speed up
- Where can we use learning?

# ML in Planning (II)

- Inductive methods
  - Need lots of examples
  - Extract a general description of a *concept,* that is, learn general heuristics
- Deductive/Analytics methods
  - Learn correct knowledge
  - Explain and analyse an example, then learn very specific heuristics
  - Generalize instantiated explanation to apply to other instances
  - Need correct and complete theories
- Analogy methods: CBR



Planning Approach
State Space Search
Conjunctive SS [PRODIGY, HSPR, FF…1]
Disjunctive SS [GRAPHPLAN, STAN, IPP…]
Plan Space Search [SNLP, TWEAK, UCPOP…]
Compilation Approaches
CSP [CP-CSP …]
SAT [SATPLAN, BLACKBOX …]
Integer Programming
Planning-Learning Goal
Speed Up Planning
Improve Plan Quality
Learn or Improve Domain Theory
Learning Phase
Before Planning Starts
During Planning Process
During Plan Execution
Type of Learning
Analytic
Explanation-Based Learning
Static Analysis and Abstractions
Analogic
Derivational Analogy / Case Based
Inductive
Decision Tree
Inductive Logic Programming
Neural Network
Bayesian Learning
Reinforcement Learning
Multistrategy
Analytic and Induction
EBL and Inductive Logic Programming
EBL and Reinforce-ment Learning


Source

# Outline

- ML
- Learning agents
- ML in planning
- **Learning by macro-operators**
- Explanation-Based Learning (EBL)
- Learning by Analogy
- Portfolios
- Conclusions

# Macro-operators (I)

- Building an action model from scratch is a task that is exceedingly difficult and time-consuming even for domain experts
- Some approaches have been explored to learn action models from examples
- A common feature of these works is that they require states just before or after each action to be known
- Statistical and logical inferences can then be made to learn the actions' preconditions and effects

# Macro-operators (II)

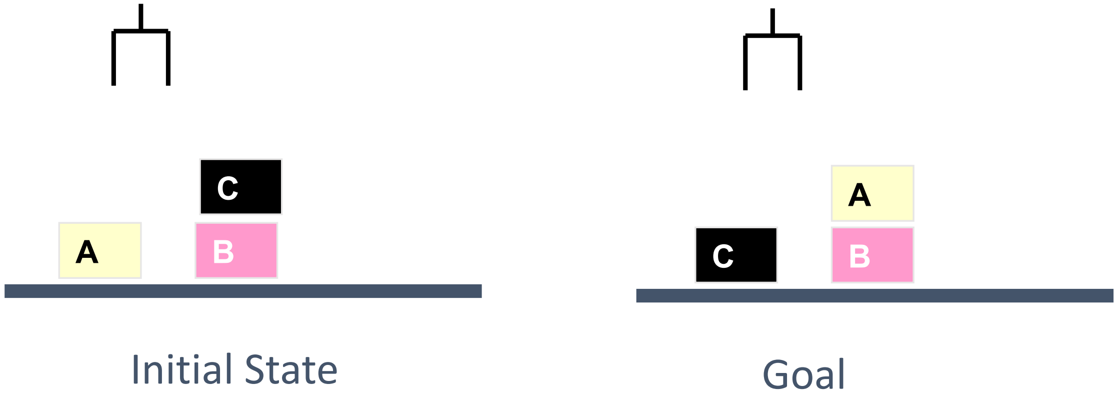
- First idea to apply learning to planning
- Learning started being applied to the planner STRIPS
- Originally conceived for two-fold purpose:
  - Learning sequences of actions
  - Monitoring execution of plans
- Key idea: create new operators by joining the descriptions of the individual operators that form a plan
- Creation of macro-operators through triangle tables

# Macro-operators (III)

- Reuse of past experience (+)
- Re-planning from failures (+)
- Less search depth (+)
- Side-effect: learning operators sub-sequences (+)
- Increased branching factor (-)
- Need to consider utility (-)
  - More we learn less efficient

# Macro-operators: example

- Plan: UNSTACK(C,B), PUTDOWN(C), PICKUP(A), STACK(A,B)
- Macro-op
  - Preconditions: on(C,B), on(A,TABLE)
  - Effects: on(A,B), on(C,TABLE)
  - 4 actions
- Generalised by replacing instances for variables



# Outline

- ML
- Learning agents
- ML in planning
- Learning by macro-operators
- **Explanation-Based Learning (EBL)**
- Learning by Analogy
- Portfolios
- Conclusions

# EBL

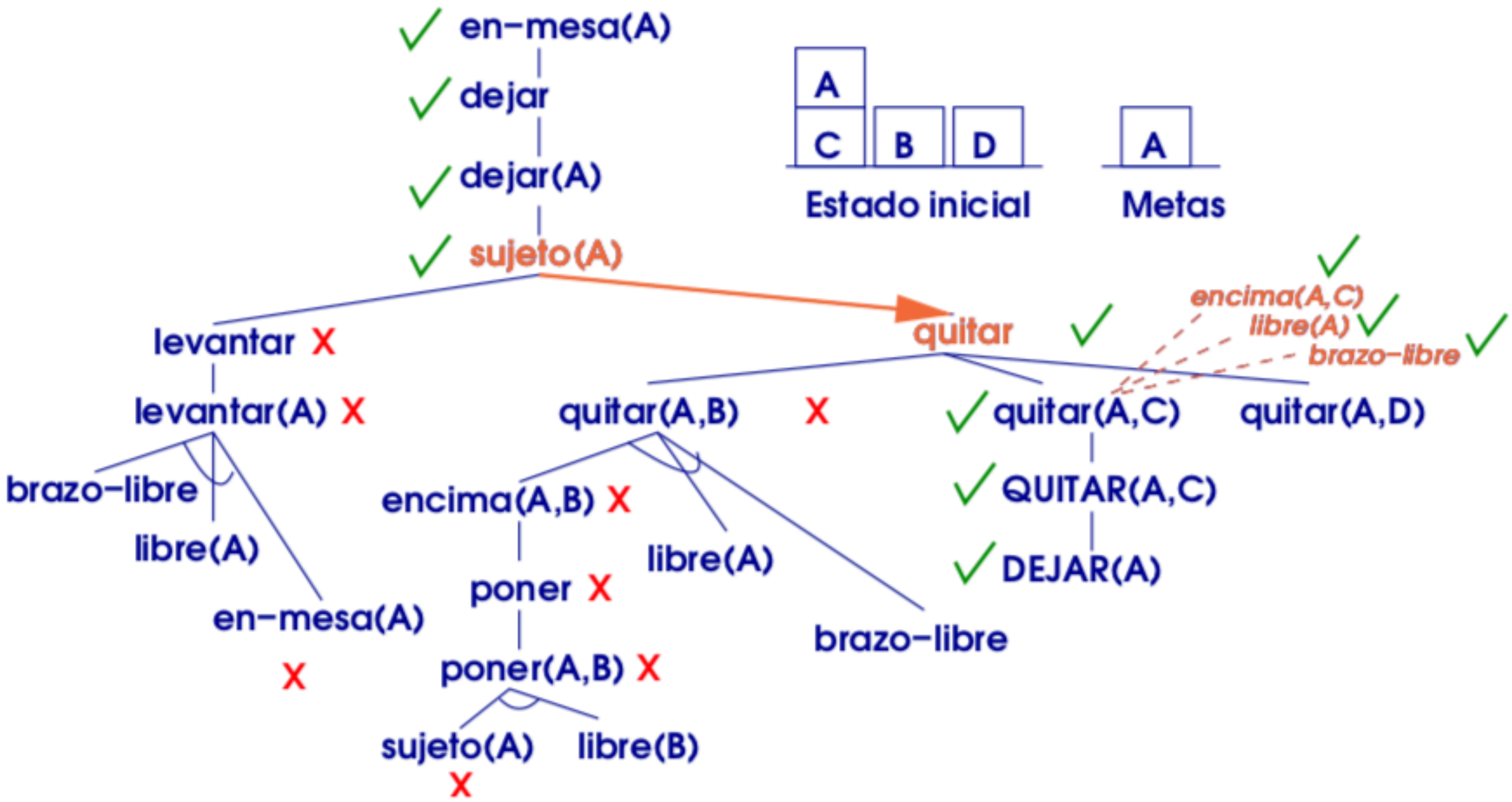
- By understanding why an example is a member of a concept, can learn the essential properties of the concept
- Trade-off the need to collect many examples for the ability to “explain” single examples (a “domain” theory)
- Given
  - Goal (e.g., some predicate calculus statement)
  - Situation Description (facts)
  - Domain Theory (inference rules)
  - Operationality Criterion
  - Use problem solver to justify, using the rules, the goal in terms of the facts.
- Generalize the justification as much as possible.
- The operationality criterion states which other terms can appear in the generalized result

# EBL

- Actual purpose: re-express target concept in a more operational manner (= efficiency)
- Inputs:
  - Target concept definition: decision to be made
  - Training example: the search episode with its successes and failures
  - Domain theory:
    - Operators used in the search and
    - Objects and possibly relationships in the world which may be used to build the explanation
  - Operationality criterion:
    - Describe concept using terms that are interpretable (efficiently) by the problem solver
    - Several possible criteria

# EBL

- Utility problem: the more rules learned, the slower the deliberation
  Possible solutions:
  - Perform utility analysis and discard low-utility rules
  - Heuristics to learn only effective knowledge
  - Incremental refinement of learned rules
- Factors influencing utility of control knowledge
  - Matching cost (cost of utilization)
  - Frequency of application
  - Savings every time it is applied


en-mesa(A)
dejar
dejar(A)
sujeto(A)
A
C
B
D
Estado inicial
A
Metas
levantar X
levantar(A) X
brazo-libre
libre(A)
en-mesa(A)
X
quitar
encima(A,C)
libre(A)
brazo-libre
quitar(A,B) X
encima(A,B) X
poner X
poner(A,B) X
sujeto(A)
X
libre(B)
libre(A)
brazo-libre
quitar(A,C)
quitar(A,D)
QUITAR(A,C)
DEJAR(A)

# Outline

- ML
- Learning agents
- ML in planning
- Learning by macro-operators
- Explanation-Based Learning (EBL)
- **Learning by Analogy**
- Portfolios
- Conclusions

# Analogy

- Can apply instance-based learning even when X$\Re$n
- $\longrightarrow$ need different "distance" metric
- Case-Based Reasoning is instance-based learning applied to instances with symbolic logic descriptions

# Analogy

- Humans solve problems combining background knowledge and past experience
- Databases: but they lack generality
- Case-based reasoning: past and new problems need only  to be similar for reuse
- PRODIGY/ANALOGY [Veloso, 1994b]

# Outline

- ML
- Learning agents
- ML in planning
- Learning by macro-operators
- Explanation-Based Learning (EBL)
- Learning by Analogy
- **Portfolios**
- Conclusions

# Portfolios

- Combine a set of planners with time to find the solution
- Autotuning: Determine the parameters that works in some domains
- Types:
  - Problem dependent: IBaCoP
  - Domain dependent: PbP
  - Domain independent: FDSS

# Outline

- ML
- Learning agents
- ML in planning
- Learning by macro-operators
- Explanation-Based Learning (EBL)
- Learning by Analogy
- Portfolios
- **Conclusions**

# Conclusions

- Learning needed for unknown environments, lazy designers
- Learning agent = performance element + learning element
- For supervised learning, the aim is to find a simple hypothesis approximately consistent with training examples
- Learning performance = prediction accuracy measured on test set