Learning in AI Planning

Dra. Mª Dolores Rodríguez Moreno

Universidad de Alcalá ISG

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
- Zimmerman and Kambhampati. Learning-Assisted Automated Planning Looking Back, Taking Stock, Going Forward. AI Magazine, Vol. 4 (2), 2003

Outline

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

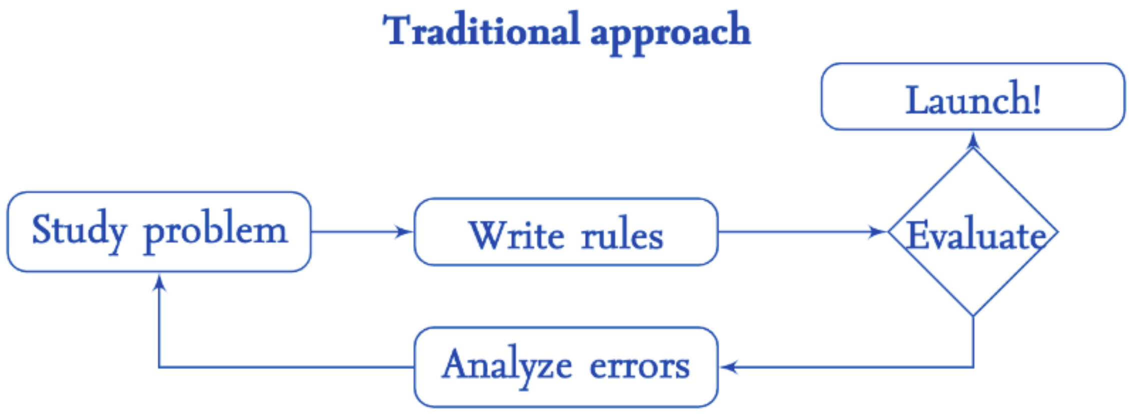
Definition

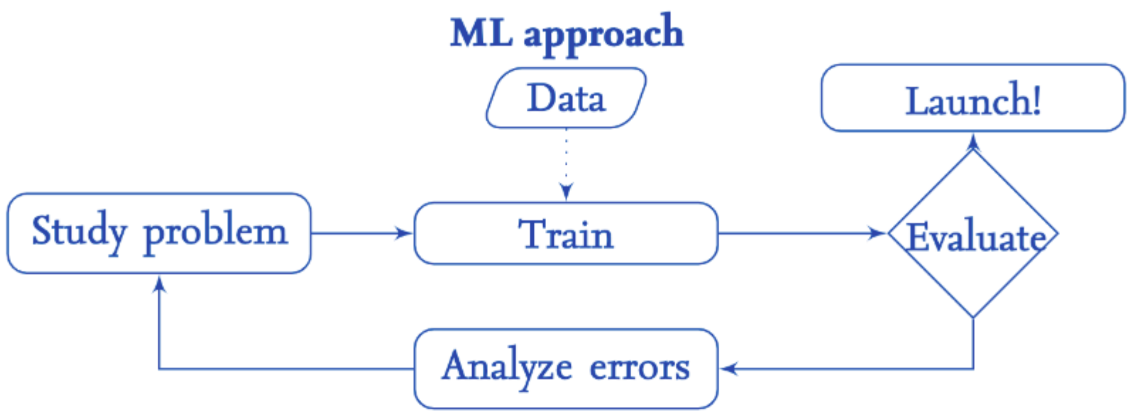
ML definition
ML is the science (and art) of programming computers so they can learn form data A. Géron, 2017

Alternative definitions

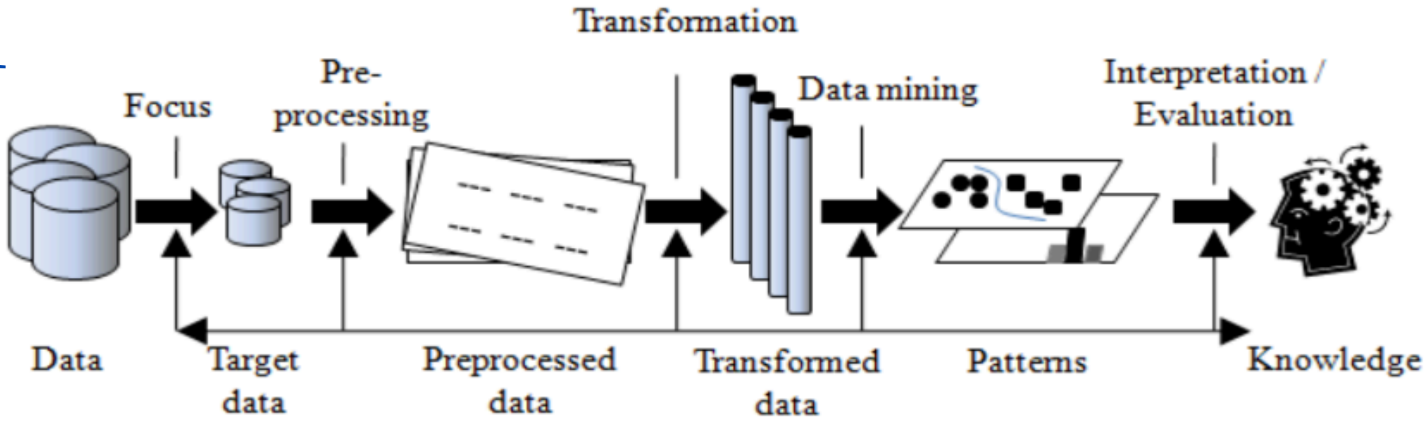
- Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel, 1959
- A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves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

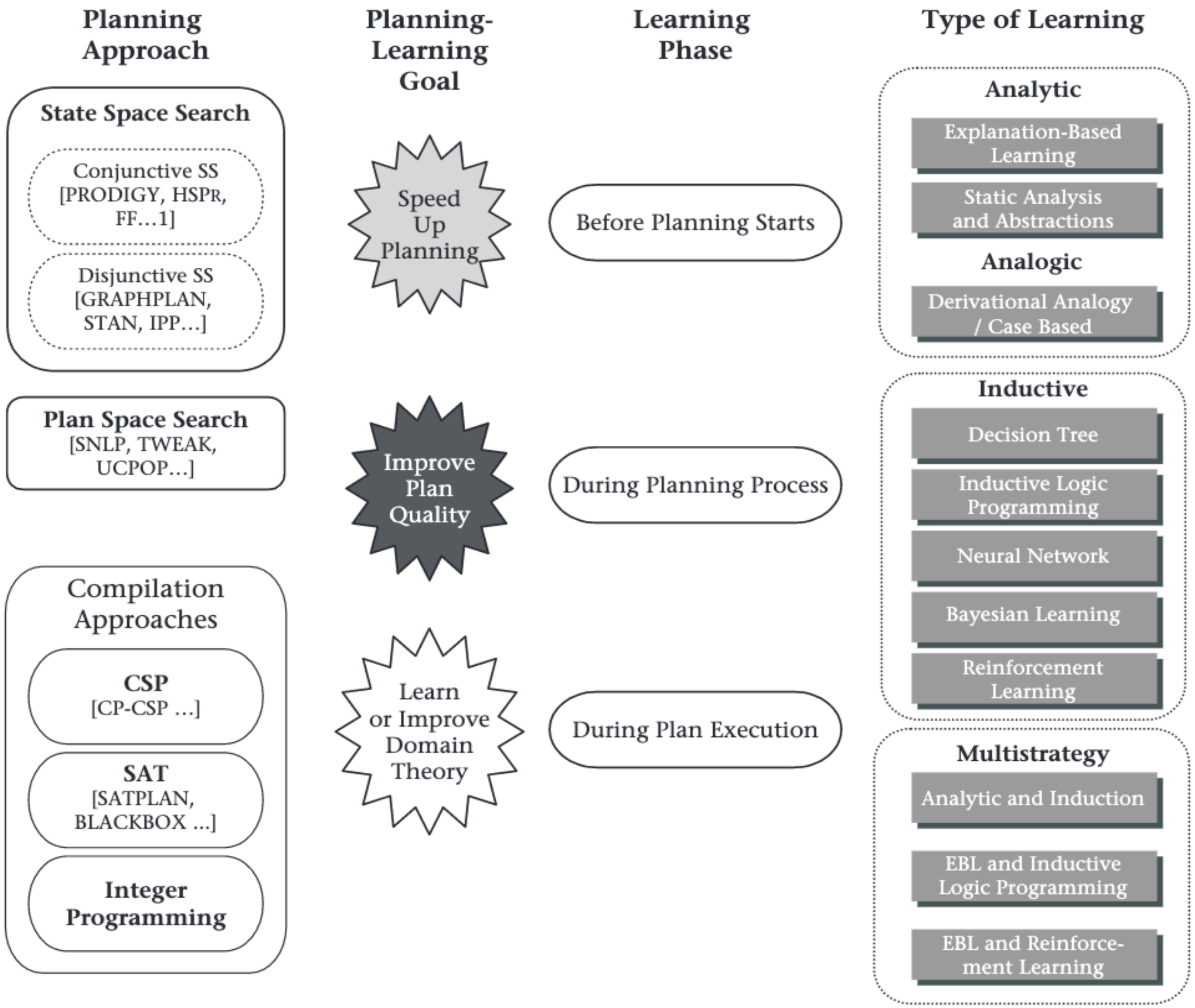
ML in Planning (I)

- When an exact model is unavailable (a nonclassical problem)
  - There are advantages for planners to evolve its domain theory by learning

- Where can we use learning in planning?

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 Reinforcement 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)

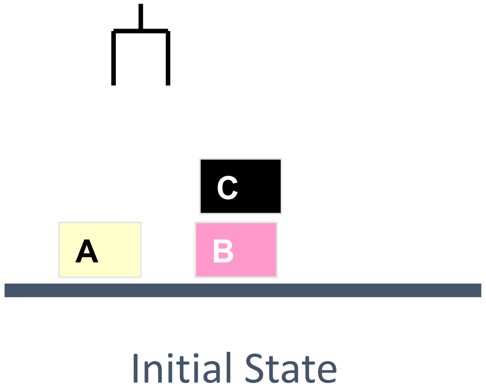
- First idea to apply learning to planning
- Learning started being applied to the planner STRIPS
- Originally conceived for two-fold purpose:
  - Learning sequences of actions
  - Monitoring execution of plans
- Key idea: create new operators by joining the descriptions of the individual operators that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

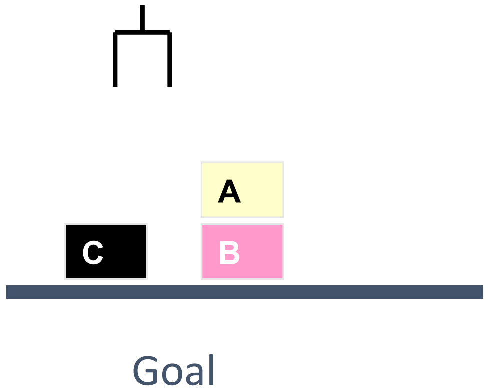
Macro-operators (III)

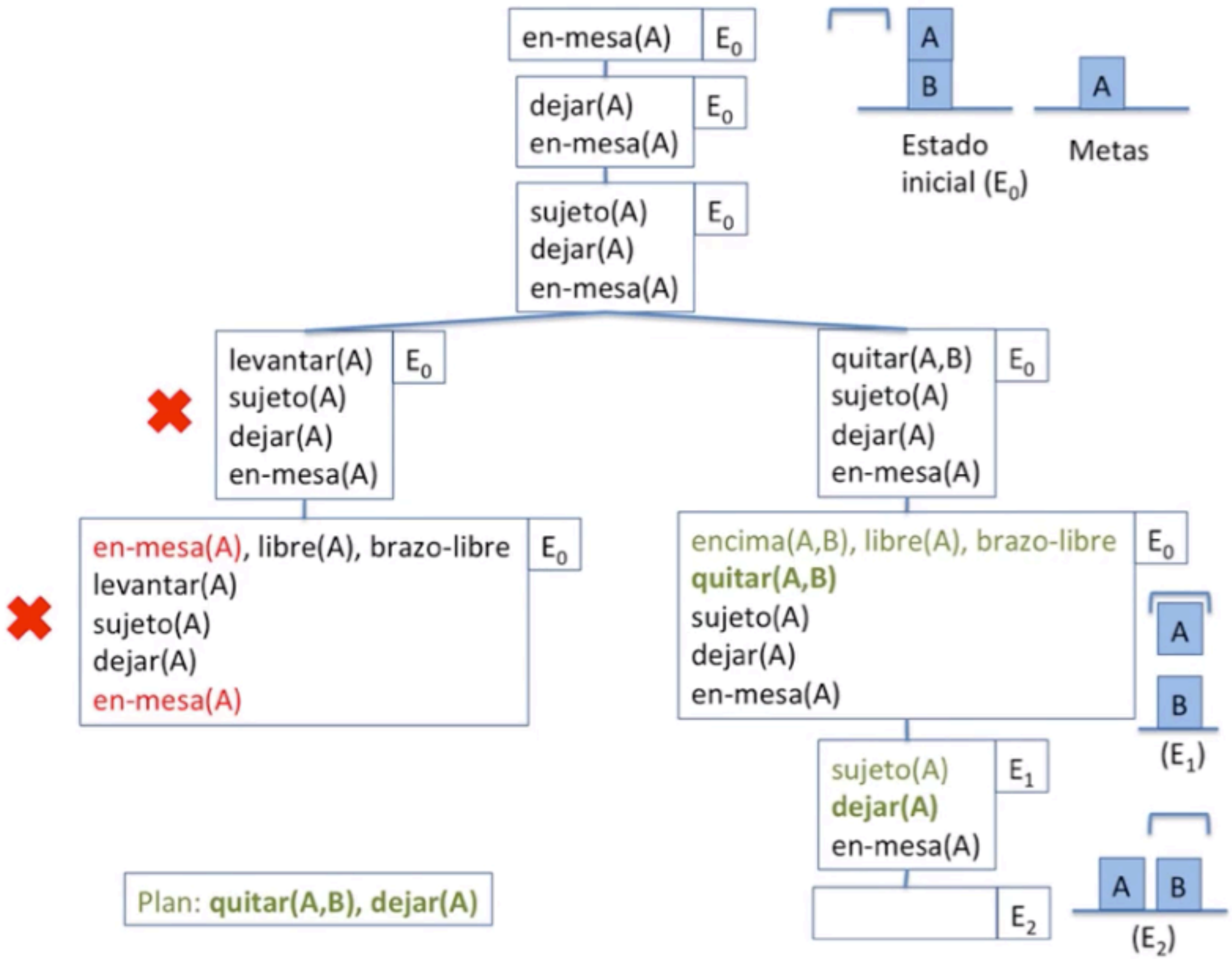
- Reuse of past experience (+)
- Re-planning from failures (+)
- Less search depth (+)
- Considered in addition to simple operators (-)
- Increased branching factor (-)
- Need to consider utility (-)
  - More we learn less efficient

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






en-mesa(A) E0
dejar(A)
en-mesa(A) E0
sujeto(A)
dejar(A)
en-mesa(A) E0
A
B
Estado inicial (E0)
A
Metas
levantar(A)
sujeto(A)
dejar(A)
en-mesa(A) E0
quitar(A,B)
sujeto(A)
dejar(A)
en-mesa(A) E0
en-mesa(A), libre(A), brazo-libre
levantar(A)
sujeto(A)
dejar(A)
en-mesa(A) E0
encima(A,B), libre(A), brazo-libre
quitar(A,B)
sujeto(A)
dejar(A)
en-mesa(A) E0
A
B
(E1)
sujeto(A)
dejar(A)
en-mesa(A) E1
E2
A B
(E2)
Plan: quitar(A,B), dejar(A)

Source

Example: Macro-operator

* encima(A,B) * libre(A) * brazo-libre	**QUITAR(A,B)**	
	* sujeto(A) libre(B)	**DEJAR(A)**
	libre(B)	brazo-libre en-mesa(A) libre(A)

Source

Outline

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

EBL (I)

- Involve use prior knowledge to explain ("prove") why the training example has a given label & use explanation to guide learning
- Inputs
  - Training example: the search episode with its successes and failures
  - Target concept definition: decision to be made
  - Domain theory:
    - Operators used in the search &
    - 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
- Popular in planning

EBL (II)

- Where to apply EBL in planning?
  - Improve speed & quality of the underlying planner
    - Learn "control knowledge" to speedup the search process (It..then rules)
    - Improve the quality of the solutions found by the search process
  - Develop domain models (e.g. action models)
- Dimensions of Variations
    - It was done from successes or failures
    - The explanations were based on complete/correct or partial domain theories
    - N# example: single or multiple (inductive learning is used in conjunction with EBL)
    - Planner search: space search (means-ends analysis), partial-order or heuristic search
- Utility problem: the more rules learned, the slower the deliberation

EBL example

```
(control-rule select-load-for-available-hoist
 (if (and (current-goal (available <h1>))
          (type-of-object <c1> crate)
          (type-of-object <h1> hoist)
          (type-of-object <l1> place)
          (type-of-object <s1> surface)))
 (then select operator load))
```

Outline

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

Analogy

- Reuse traces (cases) of solved problems & store them to avoid failure paths
- 2 approaches:
  - Transformational analogy: focus on the resultant sequence of actions, disregarding the reasons for selecting those actions
  - Derivational analogy: take into account intermediate information in addition to the resultant plan (I.e., formulation of subgoal structures, generation and subsequent rejection of alternatives, etc)
- Main emphasis: experience-driven problem solving
- Uses complex traces for justification for decisions and allows more flexible modification and reconstruction

CBR

- 4 Phases
  1. Retrieve: from memory cases relevant to solving it.
  2. Reuse: map the solution from the previous case to the target problem
  3. Revise: test the new solution
  4. Retain: once solution successfully adapted to the target problem, store as a new case in memory
- Main emphasis: retrieval and practically direct application
- Develops appropriate memory structures
- Bulk of work concentrates in retrieving from memory the right similar situation to the problem under consideration
- Unlike macroactions, cases can memorize goals information

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: a portfolio per problem (IBaCoP)
  - Domain dependent: a portfolio per domain (PbP)
  - Domain independent: same portfolio for domains and problems (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
- Classification
  - Inductive methods (lots of examples)
  - Deductive methods (explain an example, then learn very specific heuristics)
  - Analogy methods: CBR
- Learning in planning
  - Search Efficiency: Learn control knowledge to guide the planner through its search space
  - Domain models: Learn the preconditions and effects of the planning actions
  - Quality: Learn control knowledge for high quality plans
- Portfolios