Machine Learning

Javier Béjar

LSI - FIB

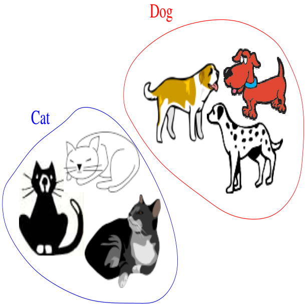
Term 2012/2013

1. Introduction to Inductive learning
2. Search and inductive learning
3. Hypothesis Spaces
4. Formal View
5. Model Selection
6. Model Performance

1. Introduction to Inductive learning
2. Search and inductive learning
3. Hypothesis Spaces
4. Formal View
5. Model Selection
6. Model Performance

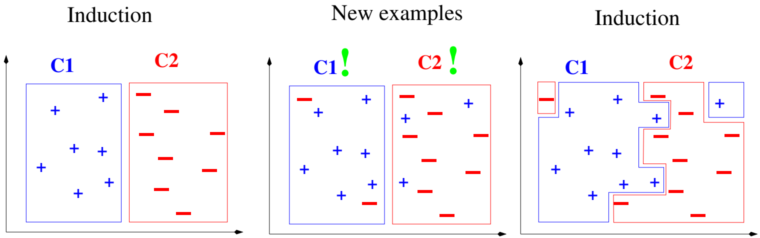
Inductive learning

- Is the area with the larger number of methods
- **Goal:** To discover general concepts from a limited set of examples (experience)
- It is based on the search of similar characteristics among examples (common patterns)
- All its methods are based on *inductive reasoning*



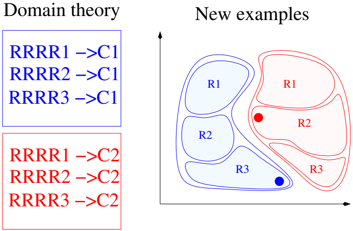
Inductive reasoning vs Deductive reasoning

- Inductive reasoning
  - It obtains general knowledge (a model) from specific information
  - The knowledge obtained is new
  - Its not truth preserving (new information can invalidate the knowledge obtained)
  - It has no well founded theory (heuristic)



Inductive reasoning vs Deductive reasoning

- Deductive reasoning
  - It obtains knowledge using well established methods (logic)
  - The knowledge is not new (it is implicit in the initial knowledge)
  - New knowledge can not invalidate the knowledge already obtained
  - Its basis is mathematical logic

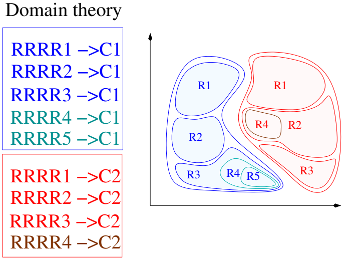


Domain theory

RRRR1 –>C1
RRRR2 –>C1
RRRR3 –>C1

RRRR1 –>C2
RRRR2 –>C2
RRRR3 –>C2

New examples



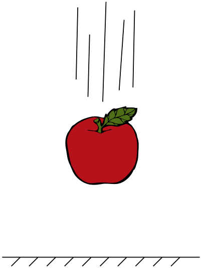
Domain theory

RRRR1 –>C1
RRRR2 –>C1
RRRR3 –>C1
RRRR4 –>C1
RRRR5 –>C1

RRRR1 –>C2
RRRR2 –>C2
RRRR3 –>C2
RRRR4 –>C2

Inductive learning

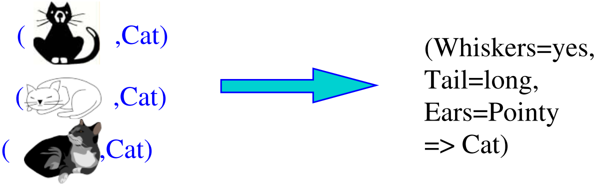
- From a formal point of view the knowledge that we obtain is invalid
- We assume that a limited number of examples represents the characteristics of the concept that we want to learn
- Just only one counterexample invalidates the result
- But, most of the human learning is inductive!



$$F = G\frac{m_1 \cdot m_2}{r^2}$$

Types of inductive learning

- Supervised inductive learning
  - Examples are usually described by pairs attribute-value
  - Each example is labeled with the concept it belongs to
  - Learning is performed by contrast among concepts
  - A set of heuristics allows to generate different hypotheses
  - A criteria of preference (*bias*) is used to choose the most *suitable* hypothesis for the examples
  - **Result:** The concept or concepts that describe better the examples



(,Cat)

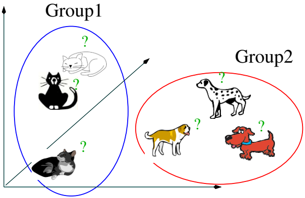
(,Cat)

(,Cat)

(Whiskers=yes,
Tail=long,
Ears=Pointy
=> Cat)

Types of inductive learning

- Unsupervised inductive learning
  - Examples are not labeled
  - We want to discover a suitable way to cluster the objects
  - Learning is based on the similarity/dissimilarity among examples
  - Heuristic preference criteria will guide the search
  - **Result:** A partition of the examples and a characterization of the partitions



1. Introduction to Inductive learning

2. Search and inductive learning

3. Hypothesis Spaces

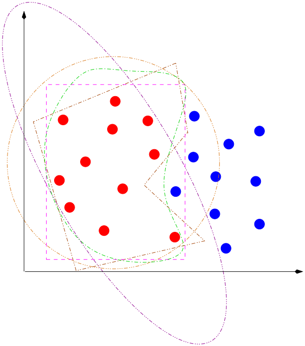
4. Formal View

5. Model Selection

6. Model Performance

Learning as search (I)

- The usual way to view inductive learning is as a search problem
- The goal is to discover a function/representation that summarizes the characteristics of a set of examples
- The space of search is all the possible concepts that can be built
- There are different ways to do the search



Learning as search (II)

- **Search space:** Language used to describe the concepts $\Longrightarrow$ Set of concepts that can be described
- **Search operators:** Heuristic operators that allow to explore the space of concepts
- **Heuristic function:** Preference function/criteria that guides the search (*Bias*)

1. Introduction to Inductive learning
2. Search and inductive learning
3. Hypothesis Spaces
4. Formal View
5. Model Selection
6. Model Performance

Hypothesis Spaces

- There are multiple models that we can use for inductive learning
- Each one of them has its advantages and disadvantages
- The most common are:
  - Logic formulas/rules
  - Probabilistic models (discrete/continuous) (parametric/non parametric)
  - General functions (linear/non linear)
- Depending on the problem some could model better its characteristics than others

Hypothesis space - Logic formulas

Logic as a model: The language used defines the size of the search space and the kind of concepts that can be learned. Represents hard linear decisions.

- Pure conjunctive/disjunctive formulas (2^n)

$$(A \vee \neg B \vee C \vee \neg D \vee E)$$

- k-term-CNF/k-term-DNF (2^{kn})

$$(A \vee \neg B \vee D \vee \neg E) \wedge (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee D \vee E) \wedge \ldots$$

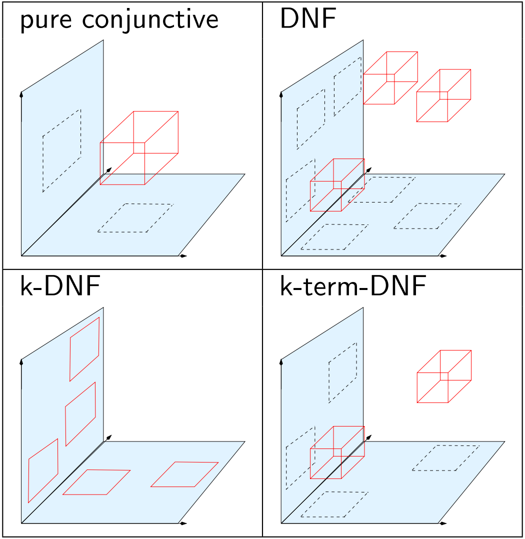
- k-CNF/k-DNF (2^{n^k})

$$(A \vee \neg B \vee C) \wedge (A \vee B \vee \neg D) \wedge (A \vee \neg B \vee E) \wedge \ldots$$

- CNF/DNF (2^{2^n})

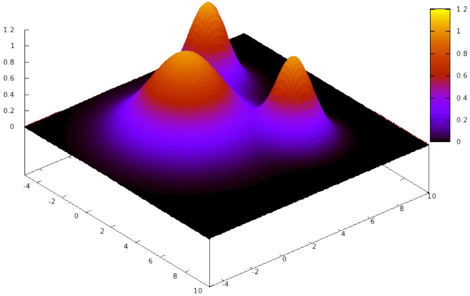
$$(A \vee \neg B \vee D \vee \neg E) \wedge (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee D \vee E)$$

Hypothesis space - Logic formulas



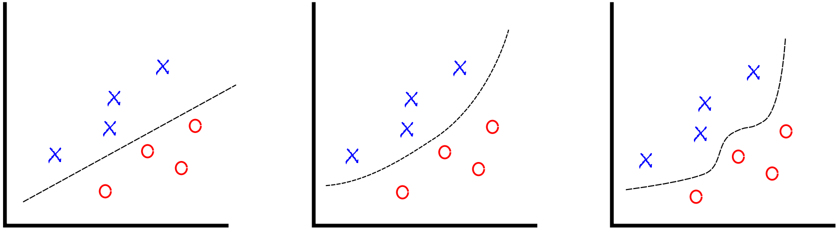
Hypothesis space - Probabilistic functions

- We assume that the attributes follow specific PDFs
- The complexity of the concepts depends on the PDF
- The size of the search space depends on the parameters needed to estimate the PDF
- It allows soft decisions (the prediction is a probability)



Hypothesis space - General functions

- We could assume a specific type of function or family of functions (linear, non linear)
- The complexity of the concepts depends on the function
- The size of the search space depends on the parameters of the function



1. Introduction to Inductive learning

2. Search and inductive learning

3. Hypothesis Spaces

4. Formal View

5. Model Selection

6. Model Performance

Supervised inductive learning - A formal view

- We can define formally the task of supervised inductive learning as:
  - Given a set $\mathcal{E}$ of n examples described by input-output pairs:

    $$(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$$

    each one generated by an unknown function, $y = f(x)$ with probability $P(x, y)$,
  - given a hypothesis space $\mathcal{H} = \{h(x; \alpha) | \alpha \in \Pi\}$, where Π is a set of parameters
  - and a *loss function* $L(y, h(x; \alpha))$ (the cost of being wrong),
  - discover the function h_α that approximates f
- The learning process has to search in the space of hypothesis the function h that performs well for all possible values (even the ones we have not seen)

Supervised inductive learning - A formal view

- This formalization allows to compute the **true error** of the hypothesis h_α as the expected loss:

$$E_{true}(h_\alpha) = \int L(y, h_\alpha(x))P(x, y)dx\ dy^{1}$$

- Because we only have a subset of examples in practice we just can compute the **empirical error**:

$$E_{emp}(h_\alpha) = \frac{1}{n} \sum_{(x,y)\in\mathcal{E}} L(y, h_\alpha(x))$$

- The goal is then to solve

$$h_\alpha^* = \arg \min_{h_\alpha \in \mathcal{H}} E_{emp}(h_\alpha)$$

[1] Weighted average of the errors for all the possible values.

Supervised inductive learning - A formal view

- Usually x is a vector of attribute-value pairs, where each attribute is discrete or continuous
- More complex representations for x are also possible
- Depending on the values of y we can distinguish two different tasks:
  - If y is a finite set of values we have **classification** (in the case of only two values it is usually called **binary classification**)
  - If y is a number we have **regression**
- The usual loss functions for these tasks are:
  - 0/1 loss $L_{0/1}(y,\hat{y}) = 0$ if $y = \hat{y}$, else 1 (classification)
  - Square error loss $L_2(y,\hat{y}) = (y - \hat{y})^2$ (regression)
  - Absolute error loss $L_1(y,\hat{y}) = |y - \hat{y}|$ (regression)

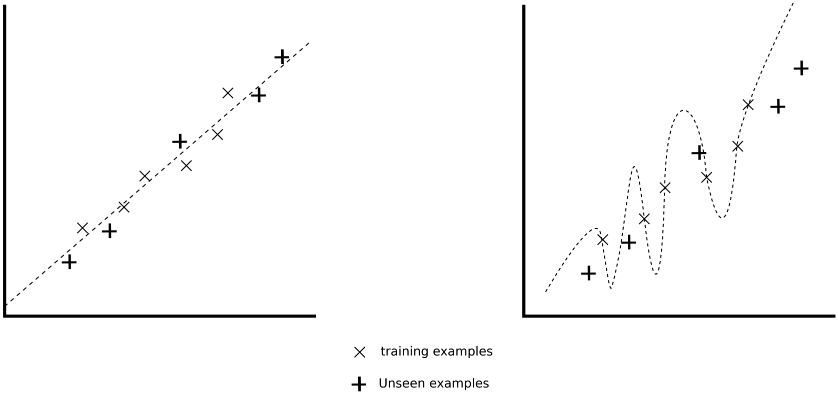
1. Introduction to Inductive learning
2. Search and inductive learning
3. Hypothesis Spaces
4. Formal View
5. **Model Selection**
6. Model Performance

Supervised inductive learning - Learning the hypothesis

- The choice of the space of functions $\mathcal{H}$ depends on the problem and it is a parameter, but we need to have a compromise:
  - A hypothesis space too complex needs more time to be explored
  - A hypothesis space too simple could not contain a good approximation of f
- Minimizing the empirical error usually does not gives enough information for the search
- Usually the more complex is h_α the better we can reduce the error
- We also have to keep in mind that we want to predict the unseen examples too:
  - A complex hypothesis could fit too well the seen data and not to generalize well (**overfitting**)

Complex hypothesis vs Overfitting

- Fitting perfectly the training data can lead to poor generalization



Supervised inductive learning - Model selection

- A usual criteria for guiding the learning process is, given a choice of different hypothesis, to prefer the **simpler one**
- This principle is called the **Ockam's Razor**

  "*Plurality ought never be posited without necessity*"
- The justification of using this principle is that for a simple hypothesis the probability of it having unnecessary conditions is reduced
- The way to enforce this preference is usually called the **bias** o hypothesis **preference** of the learning algorithm

Model Selection - Error estimation

- In order to know how well our hypothesis generalizes we need to estimate the error for the unseen examples
- There are several methods for estimating this error:
  - **Holdout cross validation**: We split the data in a training set and a test set. The hypothesis h_α is obtained by minimizing the error in the training test and the expected error is estimated on the test set (usually a poor estimate)
  - **k-fold cross validation**: We split the data in k subsets and perform k learning processes with $1/k$ examples as test set and the rest as training set
  - **leave one out cross validation**: We repeat n times the learning process always leaving out one example to measure the error (useful when dataset is small)

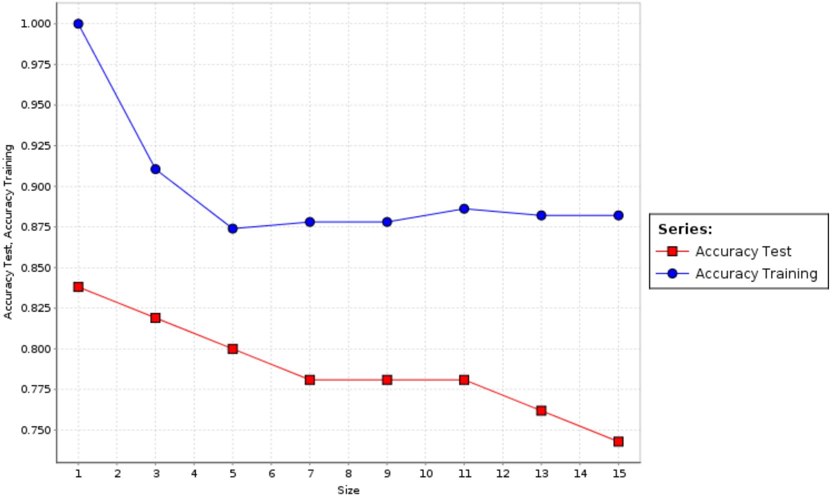
Model Selection - Selecting complexity

- If the complexity of the model is a parameter we can use the empirical error estimate to find the adequate value for our hypothesis

```
Function: crossvalidate-model-complexity (Learner,k,examples)
size $\leftarrow$ 1
repeat
    errorT[size], errorV[size] $\leftarrow$ cross-validation(Learner,size,k,examples)
    size ++
until errorT stops decreasing
best_size $\leftarrow$ min(errorV)
return Learner(best_size, examples)
```

- When the empirical error for the **training** converges we begin to overfit
- The size with the lower **test** error is the best hypothesis size

Model Selection - Selecting complexity



Model Selection - Selecting complexity

There are other methods for selecting the complexity of the model like:

- Regularization: We assume that we can somehow measure the complexity of the model ($Compl(h_\alpha)$), we look for the hypothesis that optimizes:

  $$h^*_\alpha = \arg\min_{h_\alpha \in \mathcal{H}} E_{emp}(h_\alpha) + \lambda Compl(h_\alpha)$$

  where λ is the weight for the complexity of the model
- Feature Selection: We determine what attributes are relevant during the search
- Minimum Description Length: We minimize the number of bits needed to encode the hypothesis vs the number of bits needed to encode the prediction errors in the examples (complexity and error are measured in the same units)

1. Introduction to Inductive learning
2. Search and inductive learning
3. Hypothesis Spaces
4. Formal View
5. Model Selection
6. Model Performance

Model Performance

- The measured empirical error is an estimate of how well our model generalizes
- In our measure of this error we are using all the data set
    - For example, each crossvalidation fold has a 90% of the data of the other folds
- In order to obtain a better estimate we need to have a separate test set
- A poor result in this test set could mean that our learning sample is biased and is not representative of the task

Model Performance - Uncertainty of the Estimate

The nature of the errors we commit has different sources:

- The specific space hypothesis, the best hypothesis is at a distance of the true function
- The specific sample of the task, different samples give different information
- The uncertainty of the values/representation of the sample
  - We are possibly using only a partial view of the task, some variables are not observed
  - We have possibly a corrupted version of the sample (the task labels are wrong)

Model Performance - Uncertainty of the Estimate

- Our error can be decomposed in:
  Error$=$ Systematic error (from the hypothesis space) $+$ Dependence on the specific sample $+$ Random nature of the process
- This error can also be described with what is called the Bias-Variance decomposition model
  - **Bias**: How much the average prediction deviates from the truth
  - **Variance**: How sensitive is our prediction to the sample used
- A specific expression for this decomposition can be computed for the different loss functions
- In general when the complexity of the model increases the bias decreases, but the variance increases
- The usual way to reduce both is to increase the size of the sample