Introduction to Machine Learning

Module 1: Introduction

Part A: Introduction

Sudeshna Sarkar

IIT Kharagpur

Overview of Course

1. Introduction
2. Linear Regression and Decision Trees
3. Instance based learning

   Feature Selection
4. Probability and Bayes Learning
5. Support Vector Machines
6. Neural Network
7. Introduction to Computational Learning Theory
8. Clustering

Module 1

1. Introduction
   a) **Introduction**
   b) Different types of learning
   c) Hypothesis space, Inductive Bias
   d) Evaluation, Training and test set, cross-validation
2. Linear Regression and Decision Trees
3. Instance based learning
   Feature Selection
4. Probability and Bayes Learning
5. Support Vector Machines
6. Neural Network
7. Introduction to Computational Learning Theory
8. Clustering

Machine Learning History

- 1950s:
  - Samuel's checker-playing program
- 1960s:
  - Neural network: Rosenblatt's perceptron
  - Minsky & Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Expert systems and knowledge acquisition bottleneck
  - Quinlan’s ID3
  - Natural language processing (symbolic)

Machine Learning History

- 1980s:
  - Advanced decision tree and rule learning
  - Learning and planning and problem solving
  - Resurgence of neural network
  - Valiant's PAC learning theory
  - Focus on experimental methodology
- 90's ML and Statistics
  - Data Mining
  - Adaptive agents and web applications
  - Text learning
  - Reinforcement learning
  - Ensembles
  - Bayes Net learning

- 1994: Self-driving car road test
- 1997: Deep Blue beats Gary Kasparov

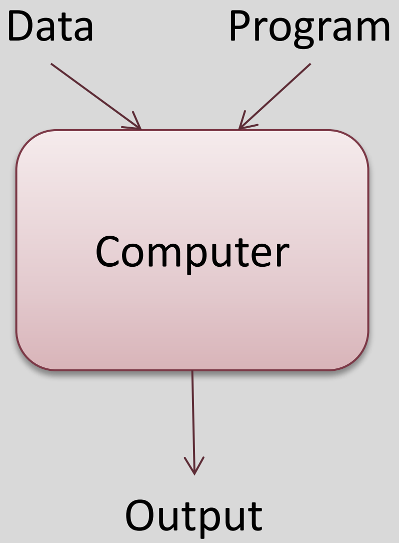
Machine Learning History

- Popularity of this field in recent time and the reasons behind that
  - New software/ algorithms
    - Neural networks
    - Deep learning
  - New hardware
    - GPU’s
  - Cloud Enabled
  - Availability of Big Data

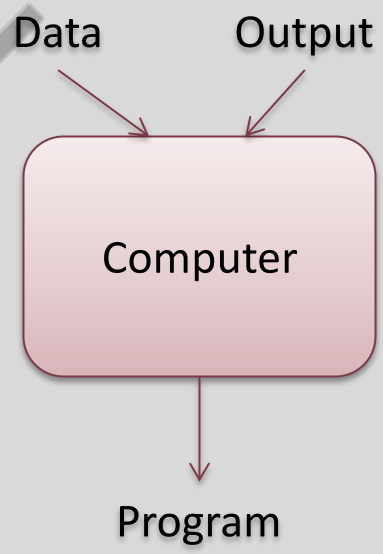
- 2009: Google builds self driving car
- 2011: Watson wins Jeopardy
- 2014: Human vision surpassed by ML systems

Programs vs learning algorithms

Algorithmic solution



Machine learning solution



Machine Learning : Definition

- Learning is the ability to improve one's behaviour based on experience.
- Build computer systems that automatically improve with experience
- What are the fundamental laws that govern all learning processes?
- Machine Learning explores algorithms that can
  - learn from data / build a model from data
  - use the model for prediction, decision making or solving some tasks

Machine Learning : Definition

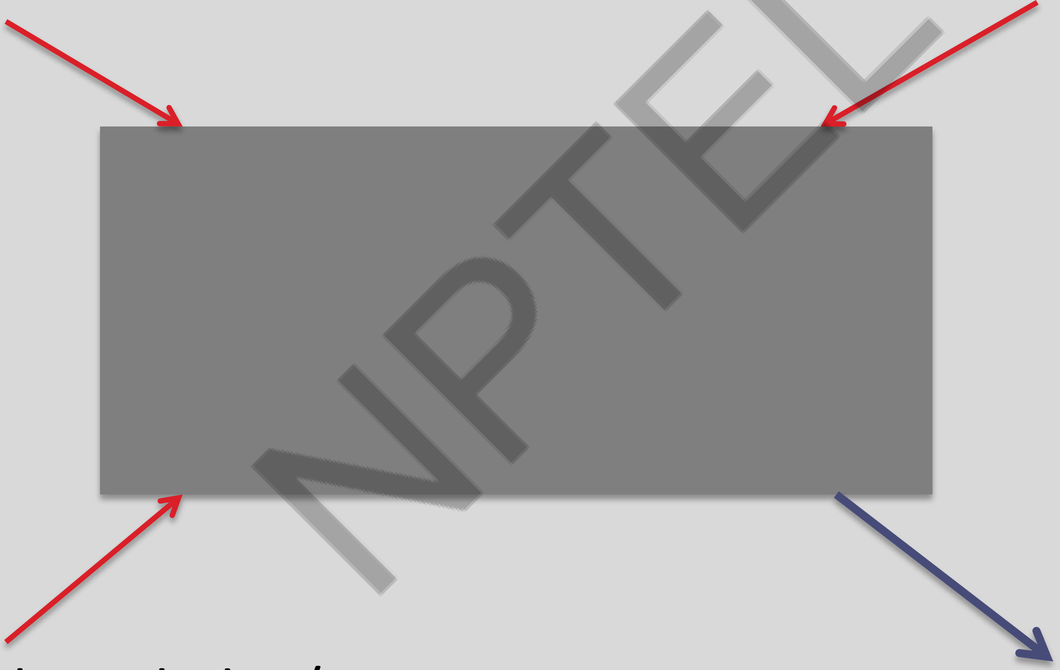
- A computer program is said to learn from *experience E* with respect to some *class of tasks T* and *performance measure P*, if its performance at tasks in T, as measured by P, improves with experience E.

  [Mitchell]

Components of a learning problem

- Task: The behaviour or task being improved.
  - For example: classification, acting in an environment
- Data: The experiences that are being used to improve performance in the task.
- Measure of improvement :
  - For example: increasing accuracy in prediction, acquiring new, improved speed and efficiency

Black-box Learner



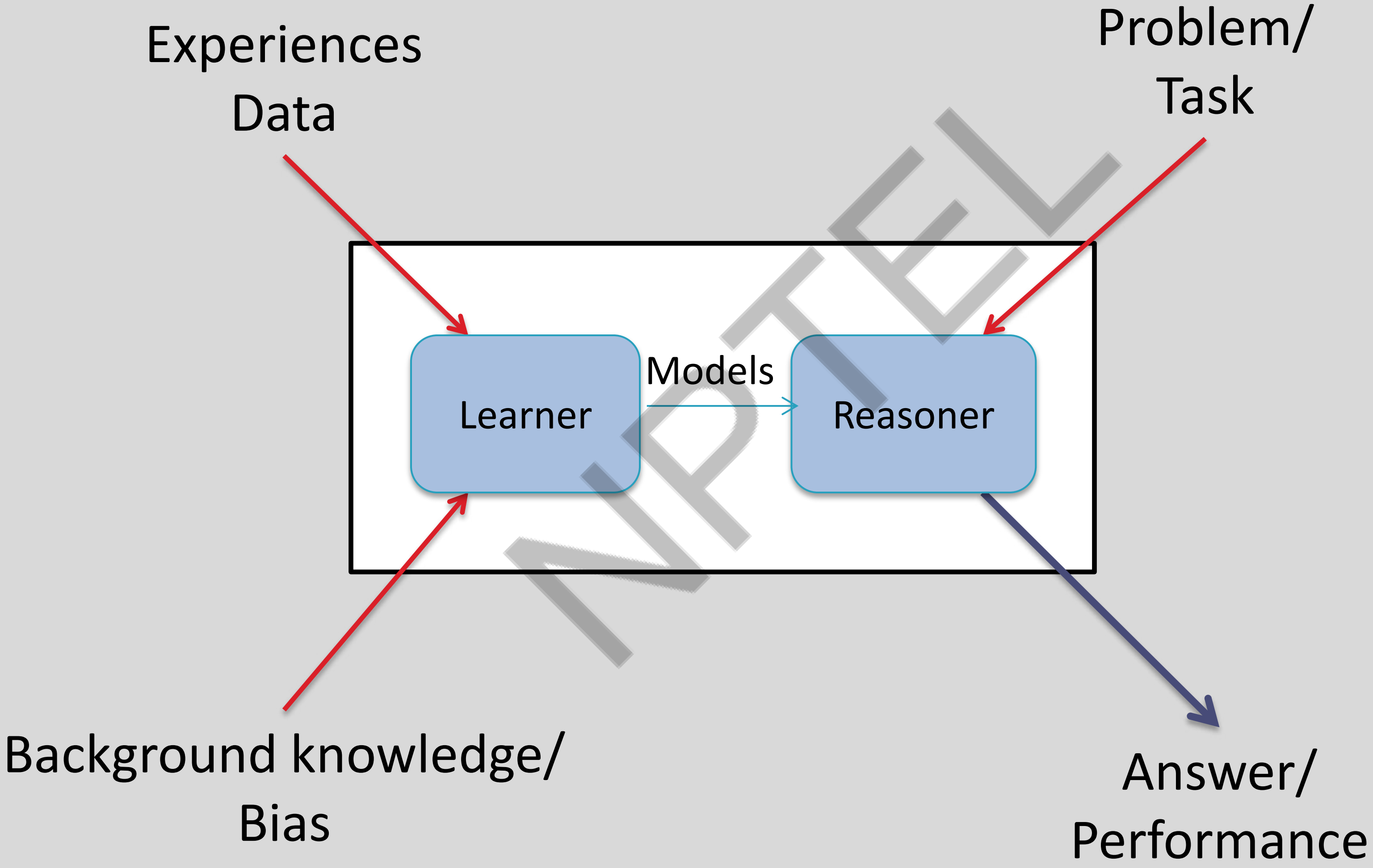
Experiences
Data

Problem/
Task

Background knowledge/
Bias

Answer/
Performance

Learner


Experiences
Data
Problem/
Task
Learner
Models
Reasoner
Background knowledge/
Bias
Answer/
Performance

Many domains and applications

Medicine:

- Diagnose a disease
  - Input: symptoms, lab measurements, test results, DNA tests,
  - Output: one of set of possible diseases, or “none of the above”
- Data: historical medical records
- Learn: which future patients will respond best to which treatments

Many domains and applications

Vision:

- say what objects appear in an image
- convert hand-written digits to characters 0..9
- detect *where* objects appear in an image

Many domains and applications

Robot control:

- Design autonomous mobile robots that learn from experience to
  - Play soccer
  - Navigate from their own experience

Many domains and applications

NLP:

- detect where entities are mentioned in NL
- detect what facts are expressed in NL
- detect if a product/movie review is positive, negative, or neutral

Speech recognition

Machine translation

Many domains and applications

Financial:

- predict if a stock will rise or fall
- predict if a user will click on an ad or not

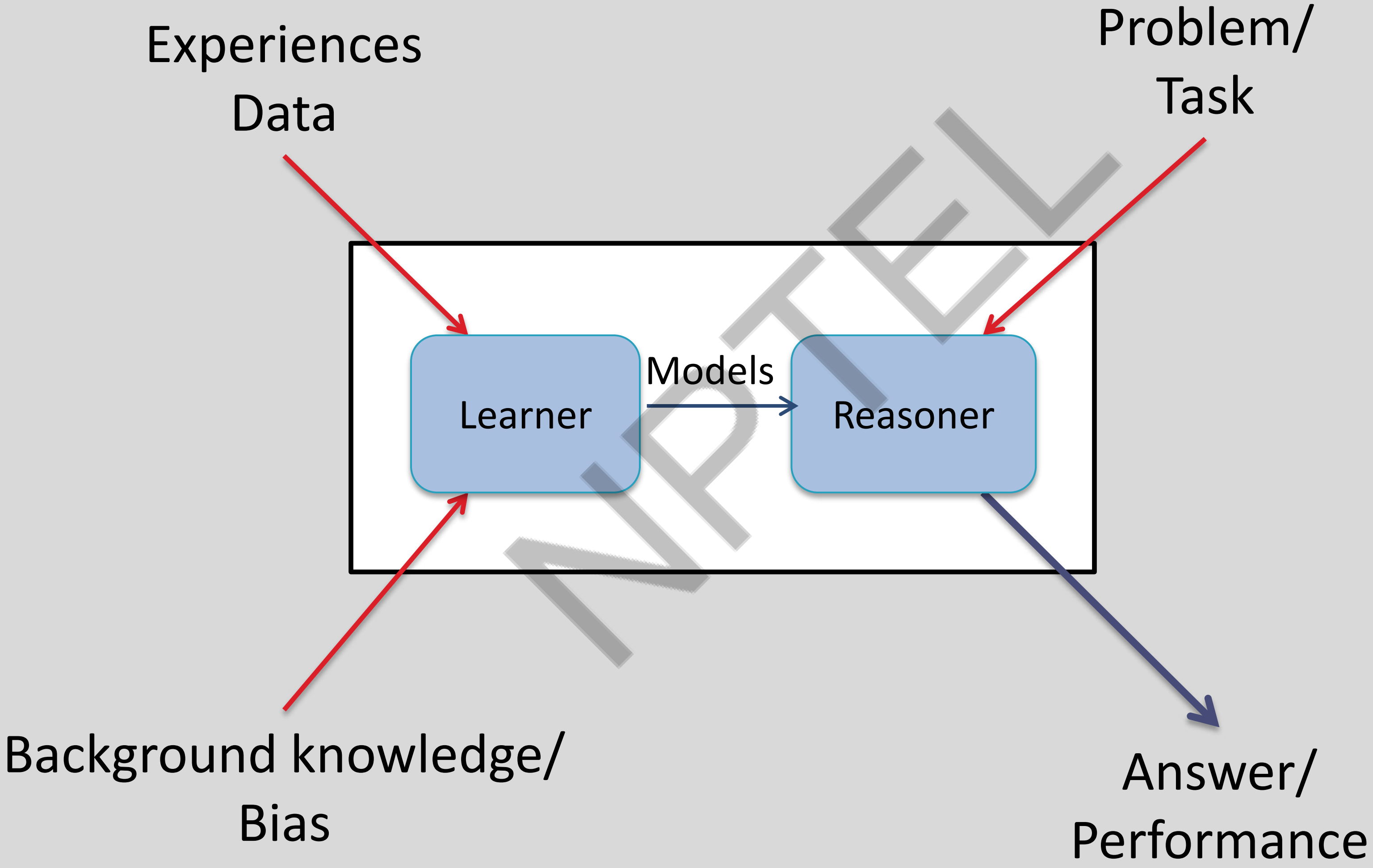
Application in Business Intelligence

- Forecasting product sales quantities taking seasonality and trend into account.
- Identifying cross selling promotional opportunities for consumer goods.
- …

Some other applications

- Fraud detection : Credit card Providers
- determine whether or not someone will default on a home mortgage.
- Understand consumer sentiment based off of unstructured text data.
- Forecasting women’s conviction rates based off external macroeconomic factors.

Learner



Experiences
Data

Problem/
Task

Learner

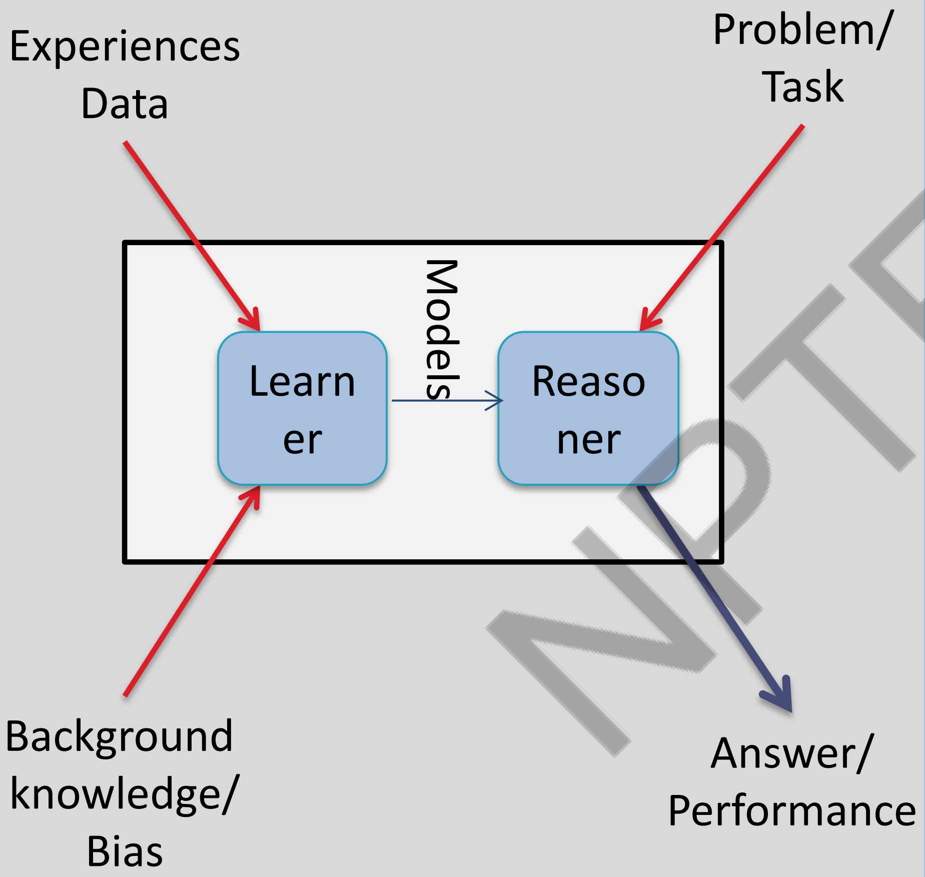
Models

Reasoner

Background knowledge/
Bias

Answer/
Performance

Design a Learner



1. Choose the training experience
2. Choose the target function (that is to be learned)
3. Choose how to represent the target function
4. Choose a learning algorithm to infer the target function

Choosing a Model Representation

- The richer the representation, the more useful it is for subsequent problem solving.
- The richer the representation, the more difficult it is to learn.

- Components of Representation
  - Features
  - Function class / hypothesis language

Foundations of Machine Learning

Module 1: Introduction

Part B: Different types of learning

Sudeshna Sarkar

IIT Kharagpur

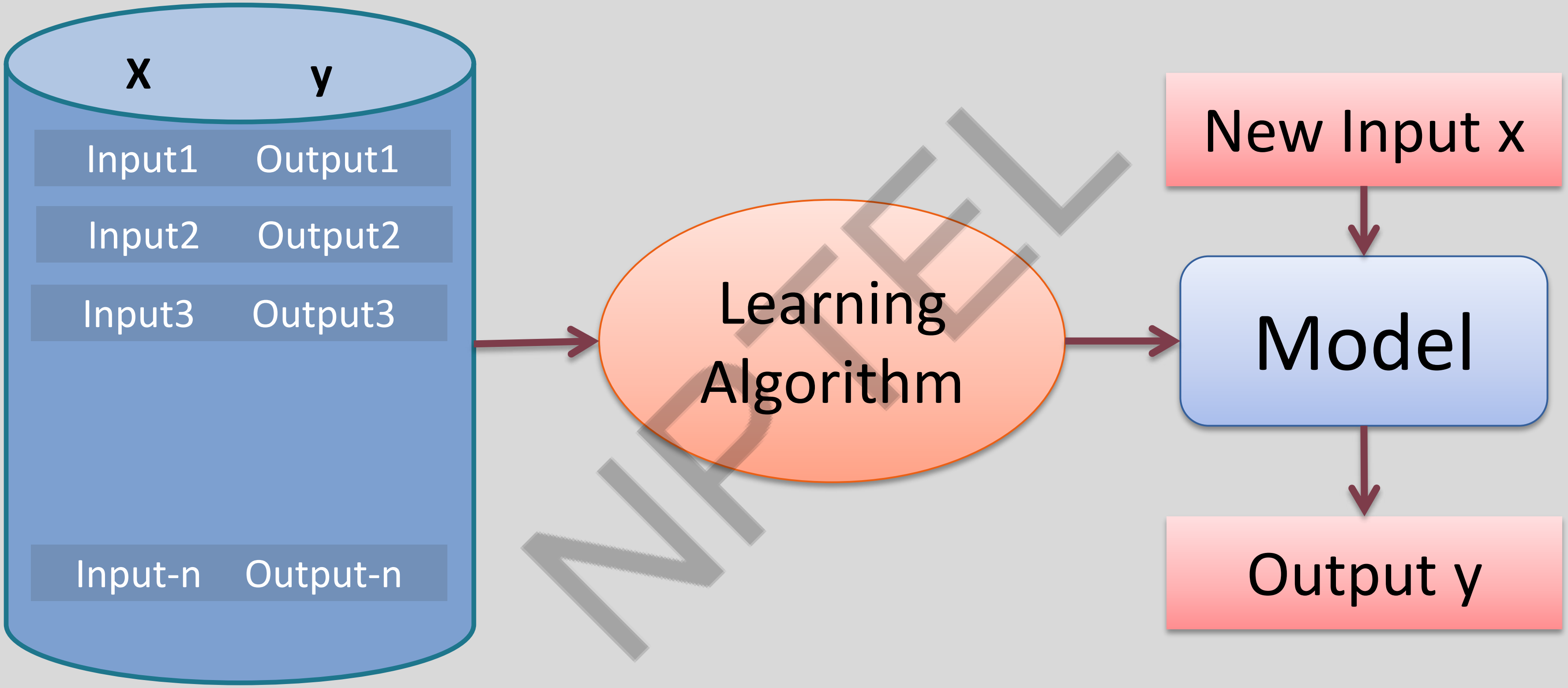
Module 1

1. Introduction
    a) Introduction
    b) **Different types of learning**
    c) Hypothesis space, Inductive Bias
    d) Evaluation, Training and test set, cross-validation
2. Linear Regression and Decision Trees
3. Instance based learning
   Feature Selection
4. Probability and Bayes Learning
5. Neural Network
6. Support Vector Machines
7. Introduction to Computational Learning Theory
8. Clustering

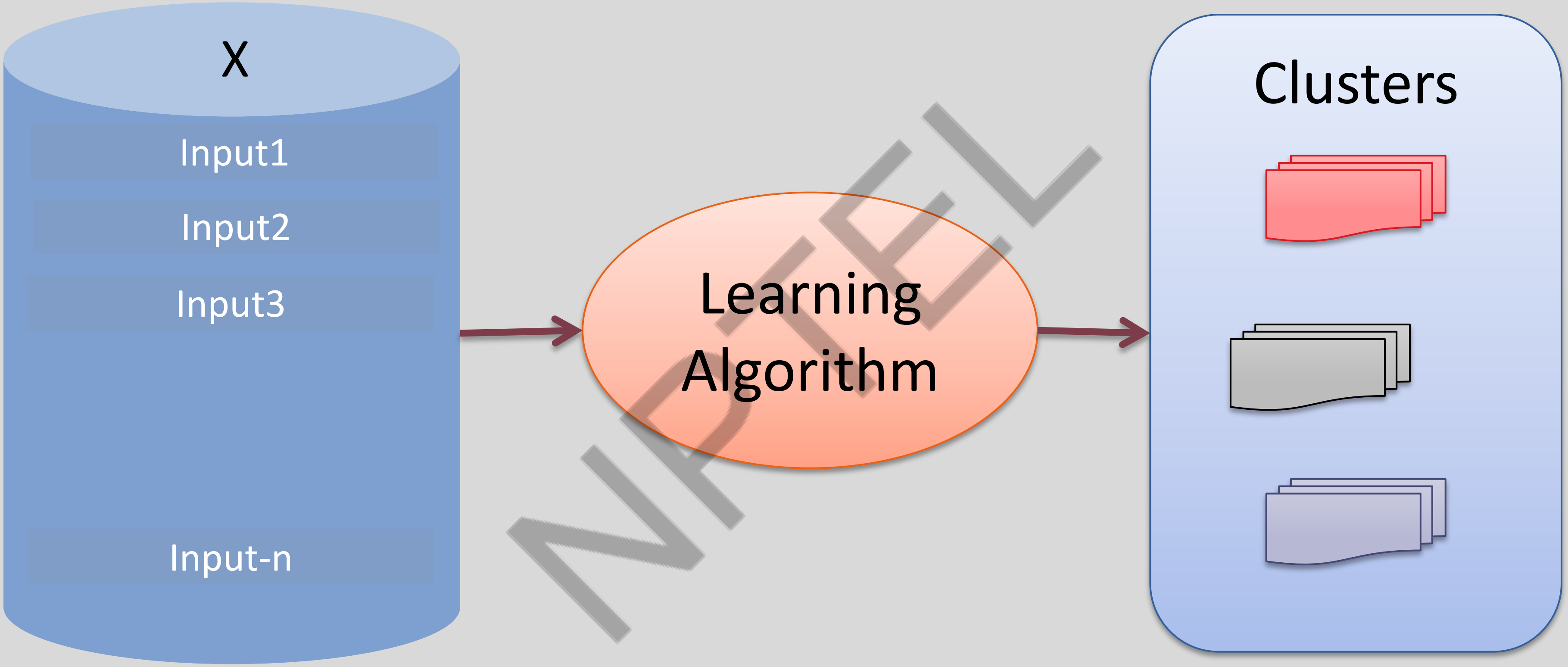
Broad types of machine learning

- Supervised Learning
  - X,y (pre-classified training examples)
  - Given an observation x, what is the best label for y?
- Unsupervised learning
  - X
  - Given a set of x’s, cluster or summarize them
- Semi-supervised Learning
- Reinforcement Learning
  - Determine what to do based on rewards and punishments.

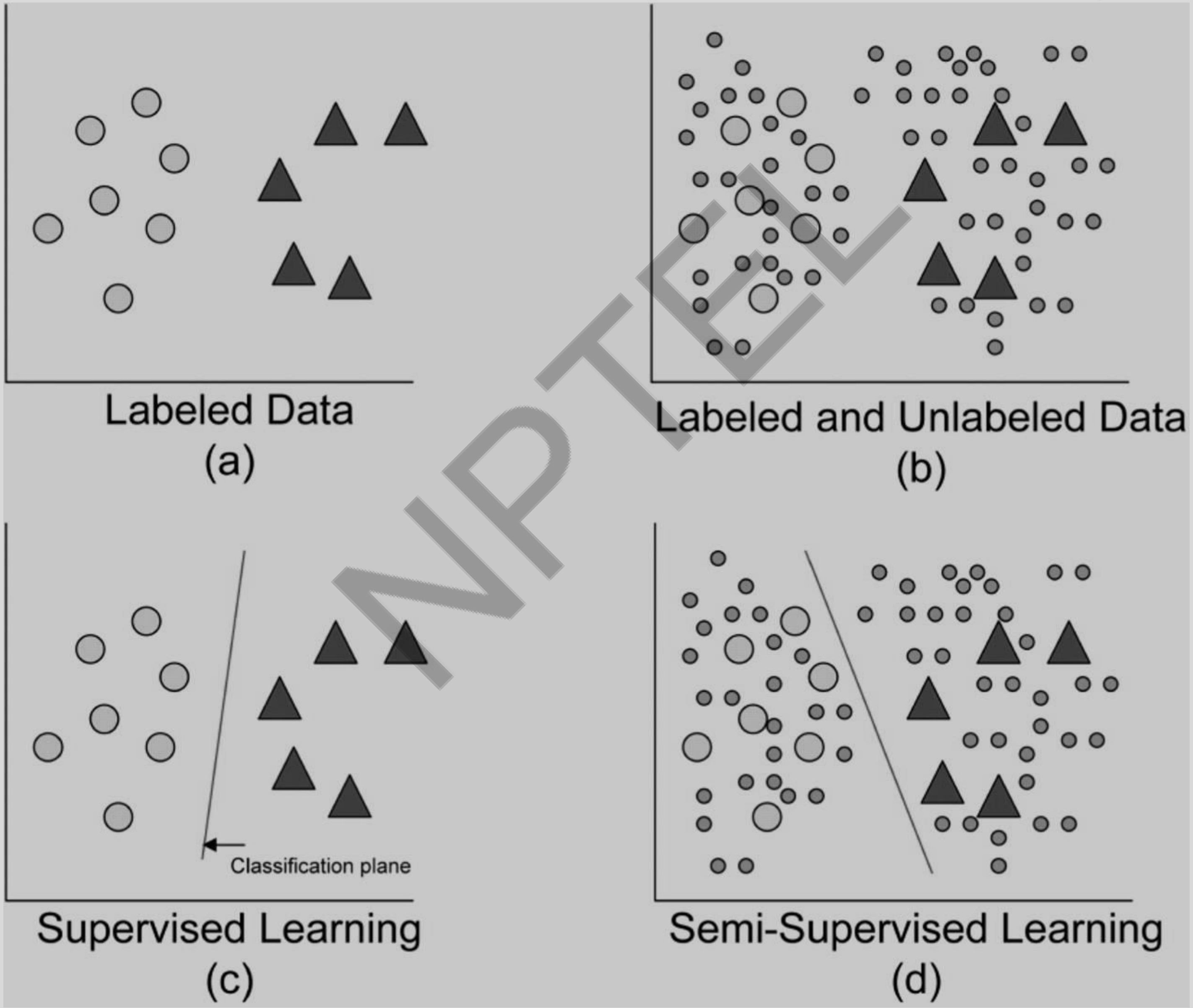
Supervised Learning



Unsupervised Learning



Semi-supervised learning



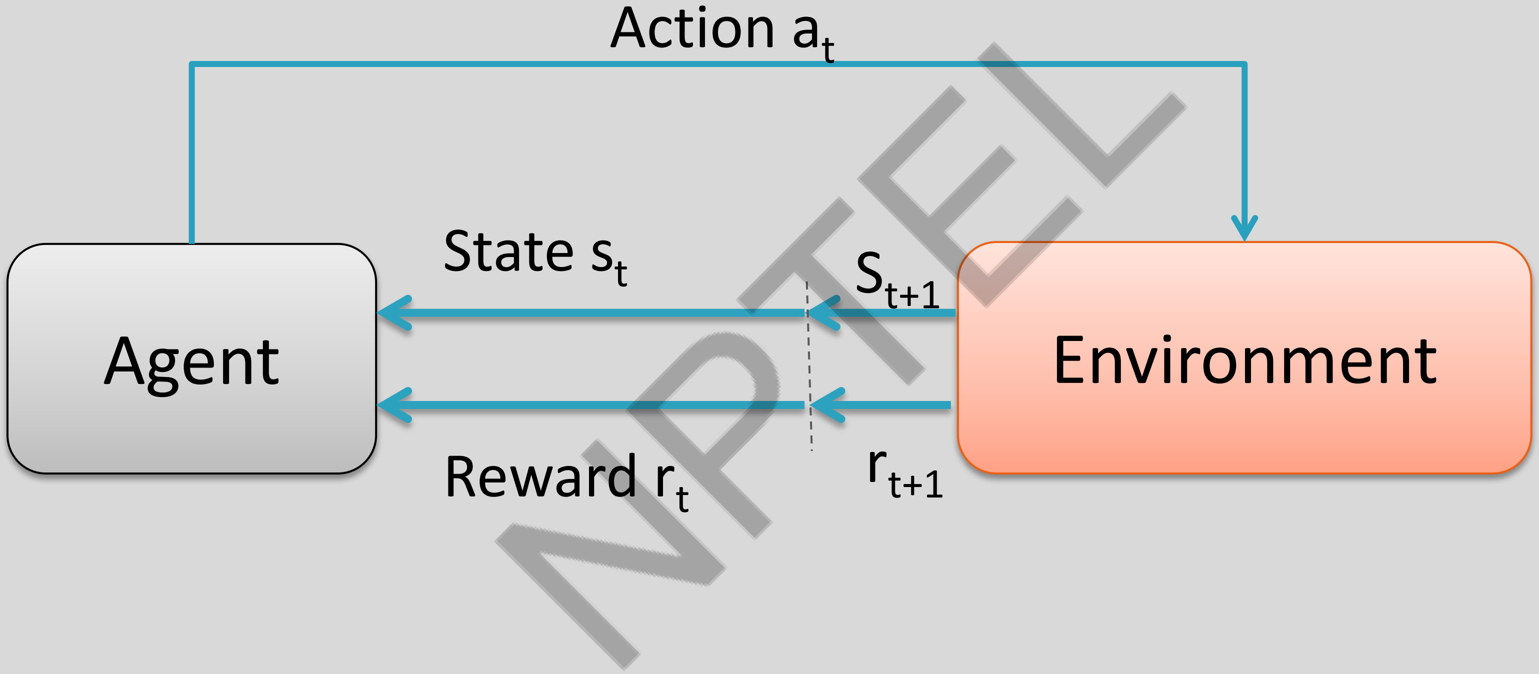
Labeled Data
(a)

Labeled and Unlabeled Data
(b)

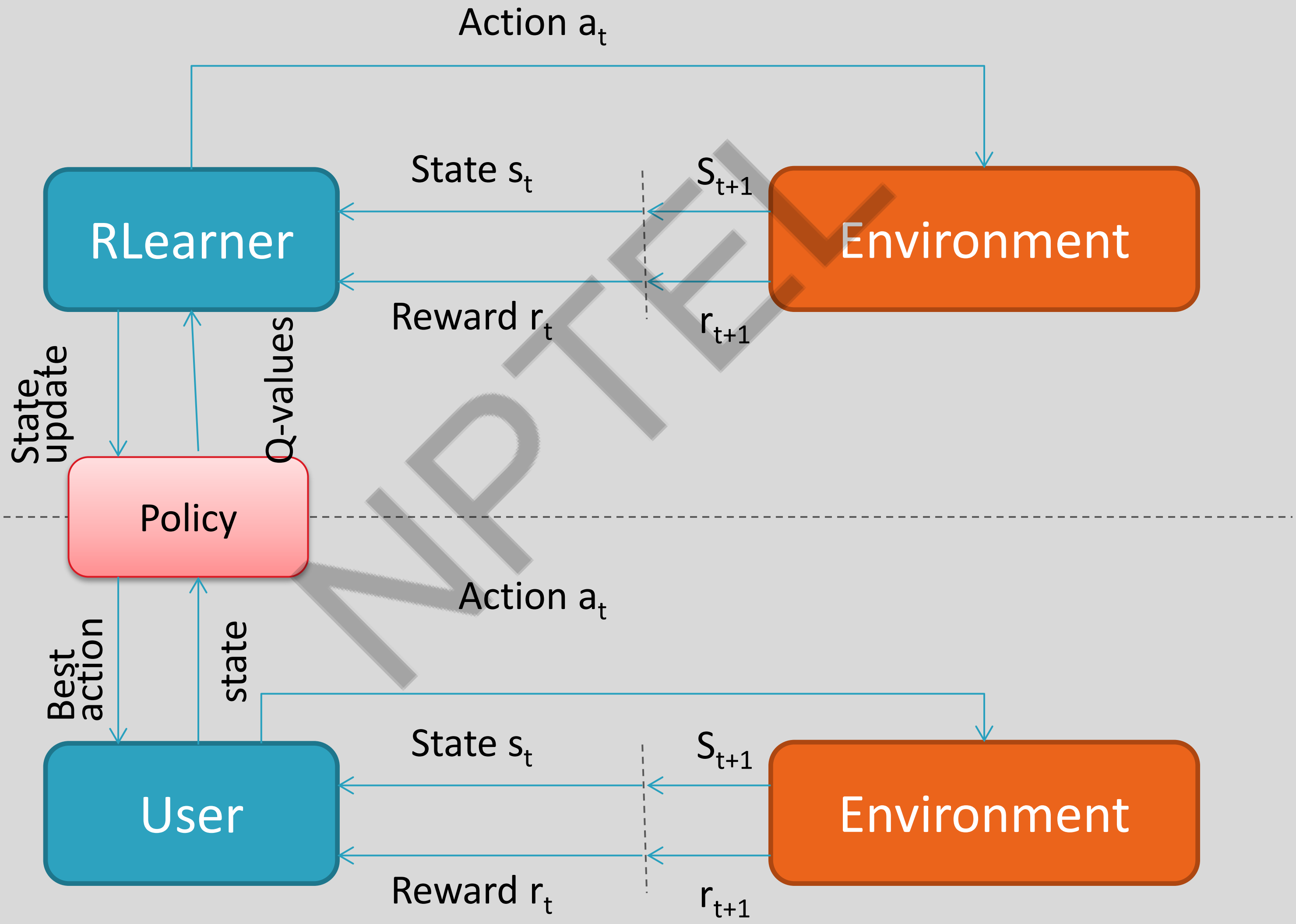
Supervised Learning
(c)

Semi-Supervised Learning
(d)

Reinforcement Learning



Reinforcement Learning



Supervised Learning

Given:

- a set of input features $X_1, \dots, X_n$
- A target feature Y
- a set of training examples where the values for the input features and the target features are given for each example
- a new example, where only the values for the input features are given

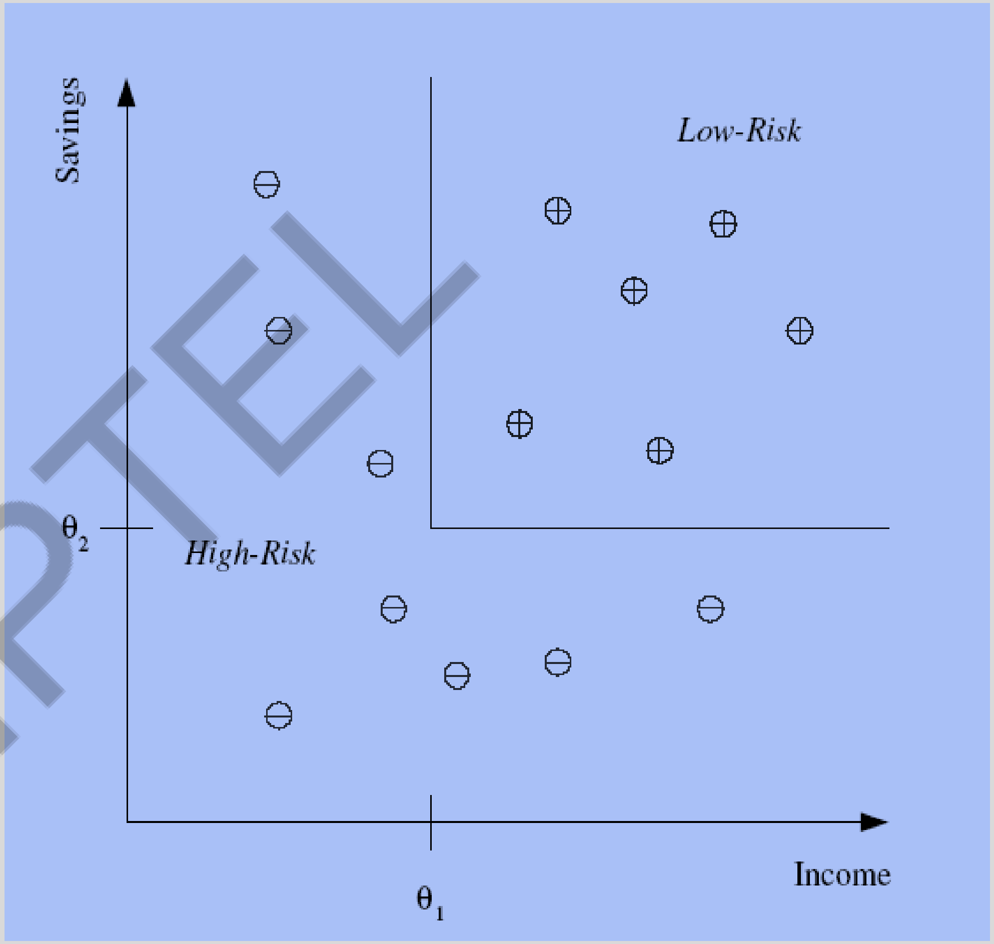
Predict the values for the target features for the new example.

- classification when Y is discrete
- regression when **Y** is continuous

Classification

Example: Credit scoring

Differentiating between low-risk and high-risk customers from their *income* and *savings*



Regression

Example: Price of a used car

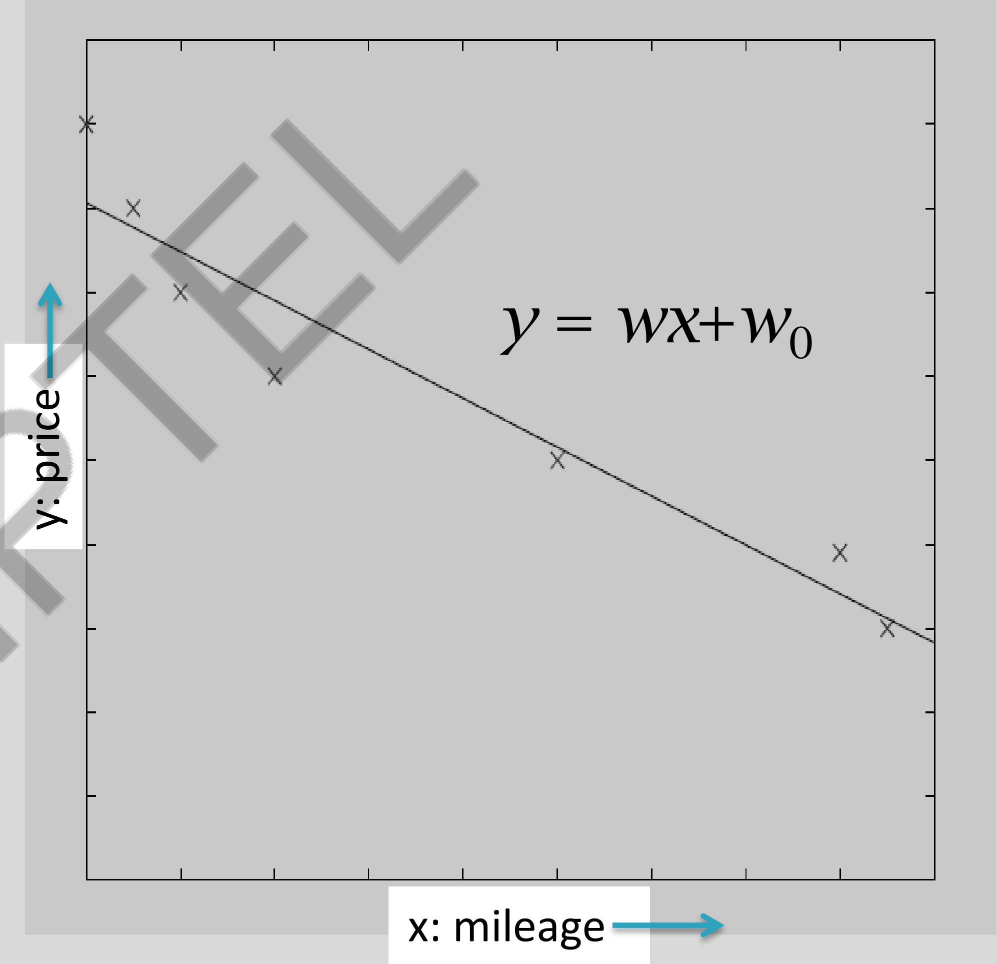
x : car attributes

y : price

$$y = g(x, \theta)$$

$g(\,)$ model,

θ parameters



Features

- Often, the individual observations are analyzed into a set of quantifiable properties which are called features. May be
  - categorical (e.g. "A", "B", "AB" or "O", for blood type)
  - ordinal (e.g. "large", "medium" or "small")
  - integer-valued (e.g. the number of words in a text)
  - real-valued (e.g. height)

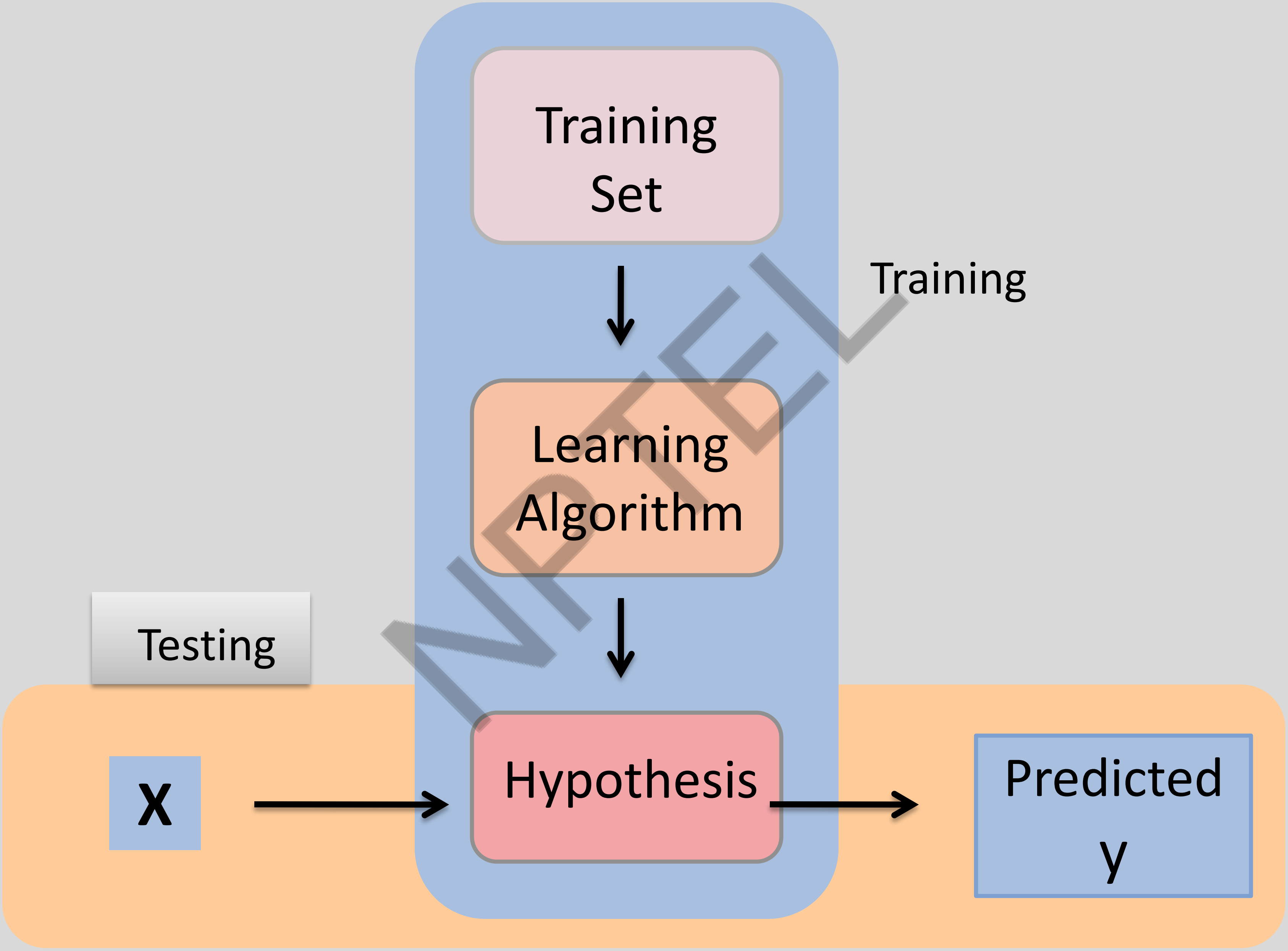
Example Data

Training Examples:

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	Home
e2	reads	unknown	new	short	Work
e3	skips	unknown	old	long	Work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

New Examples:

e7	???	known	new	short	work
e8	???	unknown	new	short	work



Training Set

Learning Algorithm

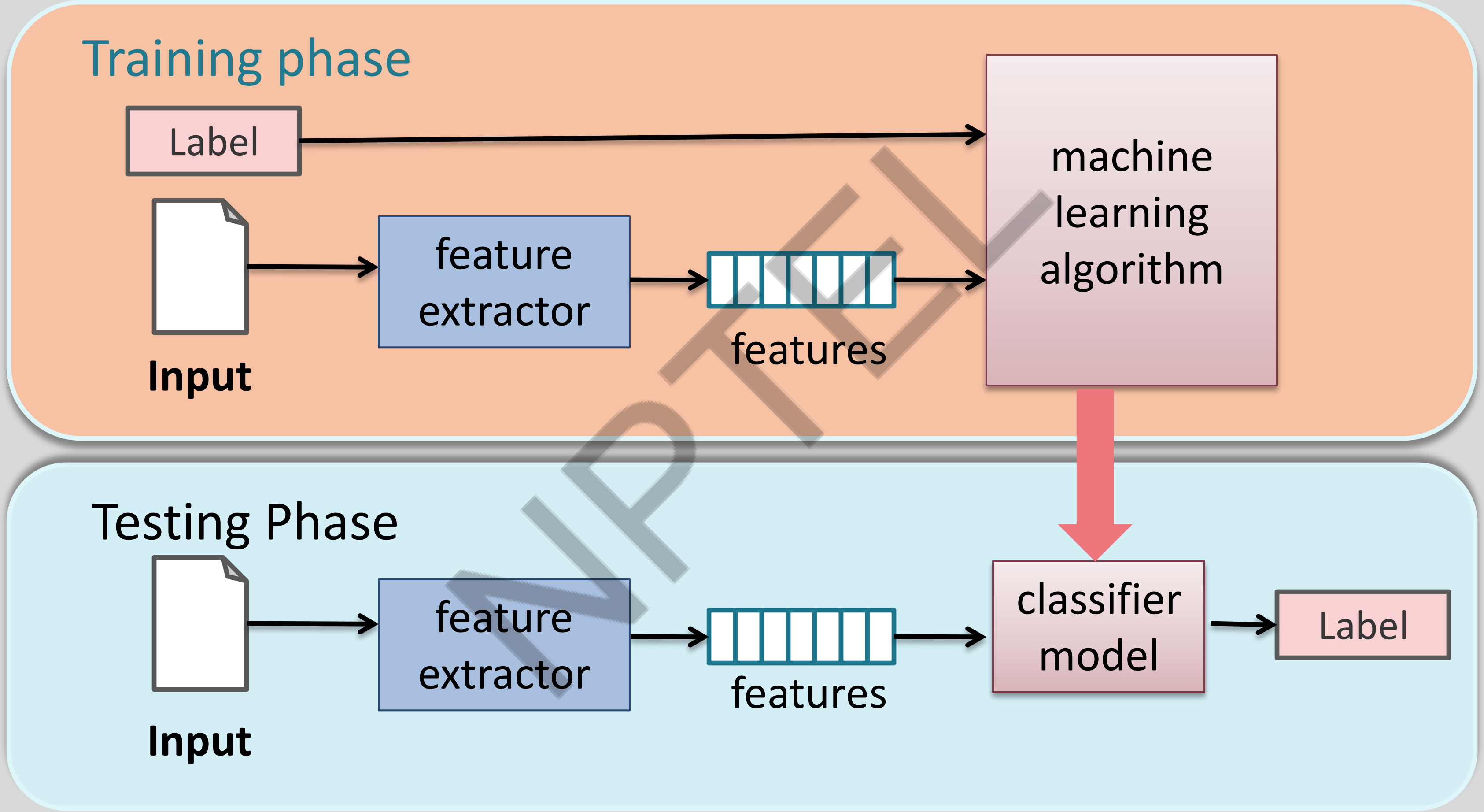
Hypothesis

Training

Testing

x

Predicted y


Training phase
Label
machine learning algorithm
feature extractor
features
Input
Testing Phase
feature extractor
features
classifier model
Label
Input

Classification learning

- Task *T:*
  - input:
  - output:
- Performance metric *P:*
- Experience *E:*

Classification learning

- Task *T*:
  - – input: a set of *instances* $d_1,\dots,d_n$
    - an instance has a set of *features*
    - we can represent an instance as a vector $\mathbf{d}=<x_1,\dots,x_n>$
  - – output: a set of *predictions* $\hat{y}_1,\dots,\hat{y}_n$
    - one of a fixed set of constant values:
      - – *{+1,-1}* or *{cancer, healthy}*, or *{rose, hibiscus, jasmine, …}*, or …
- Performance metric *P:*
- Experience *E:*

Classification Learning

Task	Instance	Labels
medical diagnosis	**patient record:** blood pressure diastolic, blood pressure systolic, age, sex (0 or 1), BMI, cholesterol	{-1,+1} = low, high risk of heart disease
finding entity names in text	**a word in context:** capitalized (0,1), word-after-this-equals-Inc, bigram-before-this-equals-acquired-by, …	{first,later,outside} = first word in name, second or later word in name, not in a name
image recognition	**image:** 1920*1080 pixels, each with a code for color	{0,1} = no house, house

Classification learning

we care about performance on the *distribution*, not the *training data*

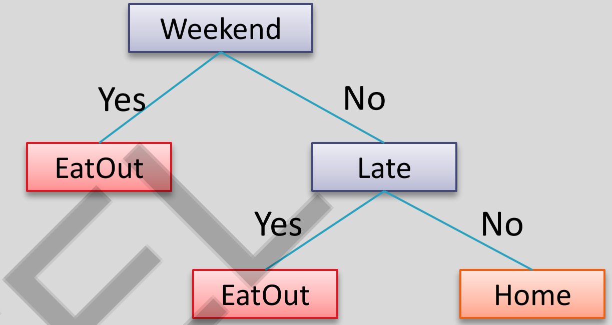
- Task *T:*
  - input: a set of *instances* $d_1,\ldots,d_n$
  - output: a set of *predictions* $\hat{y}_1,\ldots,\hat{y}_n$
- Performance metric *P:*
  - Prob (wrong prediction) on examples from *D*
- Experience *E:*
  - a set of *labeled examples (x,y)* where *y* is the true label for *x*
  - ideally, examples should be *sampled* from some fixed distribution *D*

Classification Learning

Task	Instance	Labels	Getting data
medical diagnosis	**patient record:** lab readings	risk of heart disease	wait and look for heart disease
finding entity names in text	**a word in context:** capitalized, nearby words, ...	{first,later,outside}	text with manually annotated entities
image recognition	**image:** pixels	no house, house	hand-labeled images

Representations

1. Decision Tree



Weekend

Yes

No

EatOut

Late

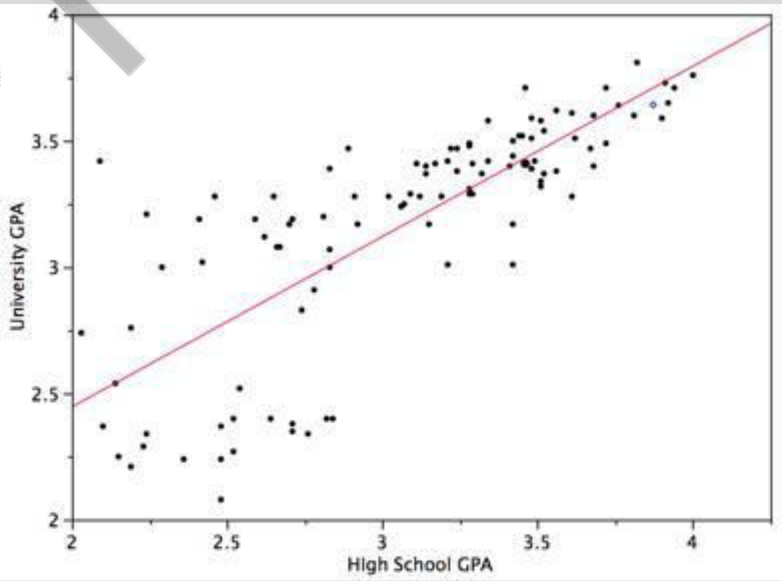
Yes

No

EatOut

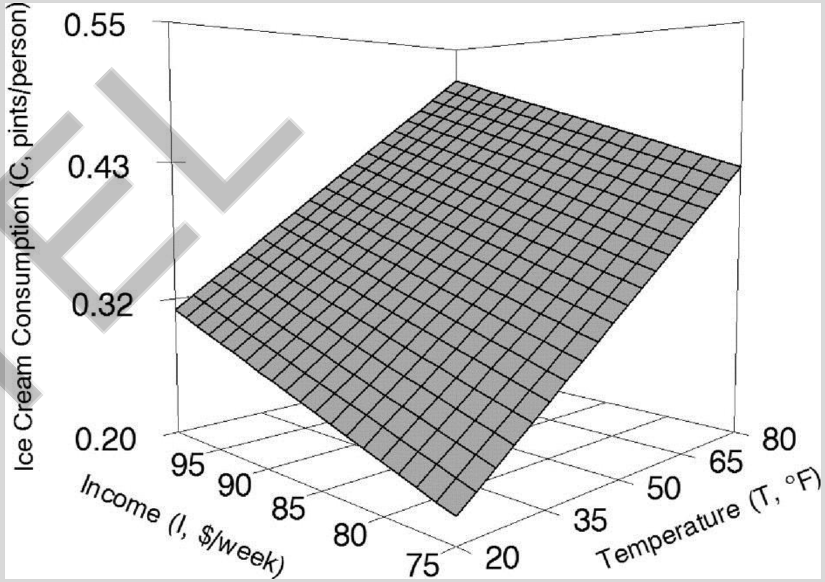
Home

2. Linear function

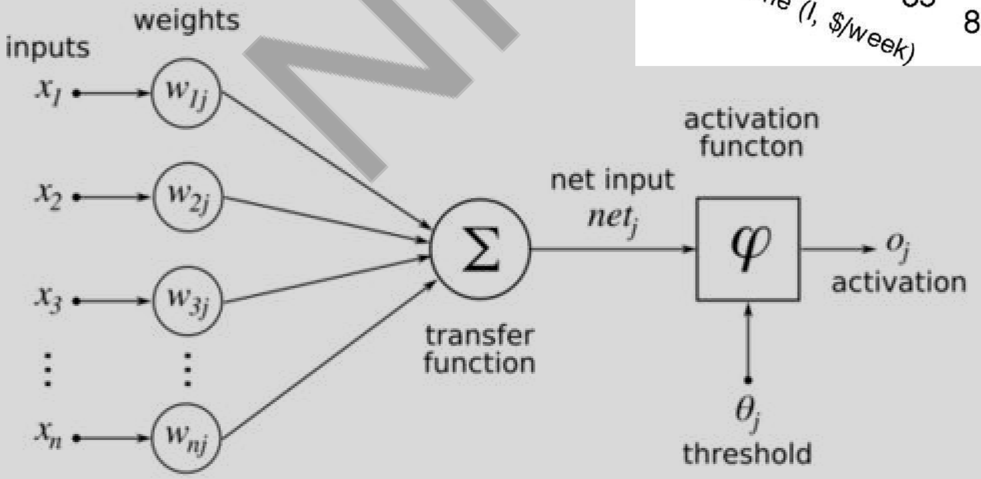


Representations

3. Multivariate linear function

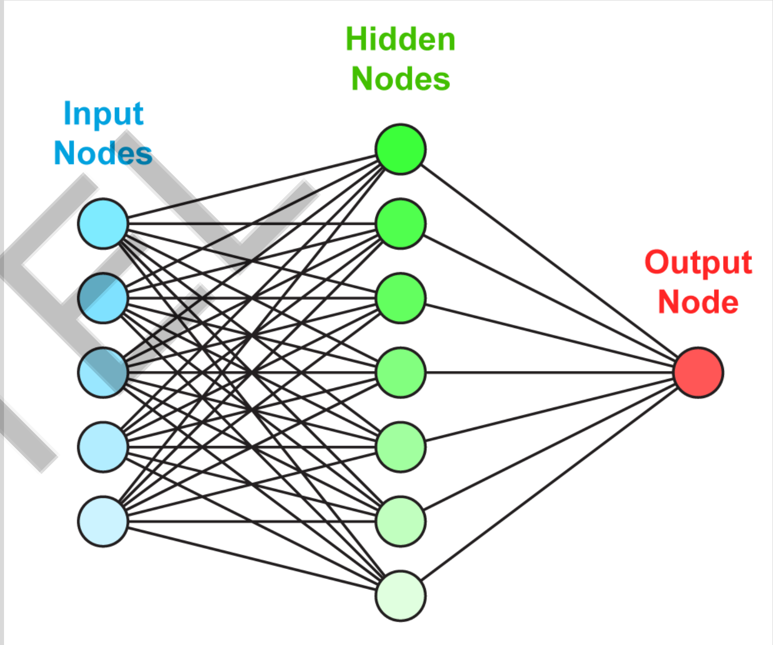


4. Single layer perceptron



Representations

5. Multi-layer neural network



Hypothesis Space

- One way to think about a supervised learning machine is as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

Terminology

- **Features:** The number of features or distinct traits that can be used to describe each item in a quantitative manner.
- **Feature vector:** n-dimensional vector of numerical features that represent some object
- **Instance Space X:** Set of all possible objects describable by features.
- **Example (x,y):** Instance x with label y=f(x).

Terminology

- **Concept c:** Subset of objects from X (c is unknown).
- **Target Function f:** Maps each instance x ∈ X to target label y ∈ Y
- **Example (x,y):** Instance x with label y=f(x).
- **Training Data S:** Collection of examples observed by learning algorithm.

  Used to discover potentially predictive relationships

Foundations of Machine Learning

Module 1: Introduction

Part c: Hypothesis Space and Inductive Bias

Sudeshna Sarkar

IIT Kharagpur

Inductive Learning or Prediction

- **Given** examples of a function *(X, F(X))*
  - **Predict** function *F(X)* for new examples *X*
- Classification

  F(X) = Discrete
- Regression

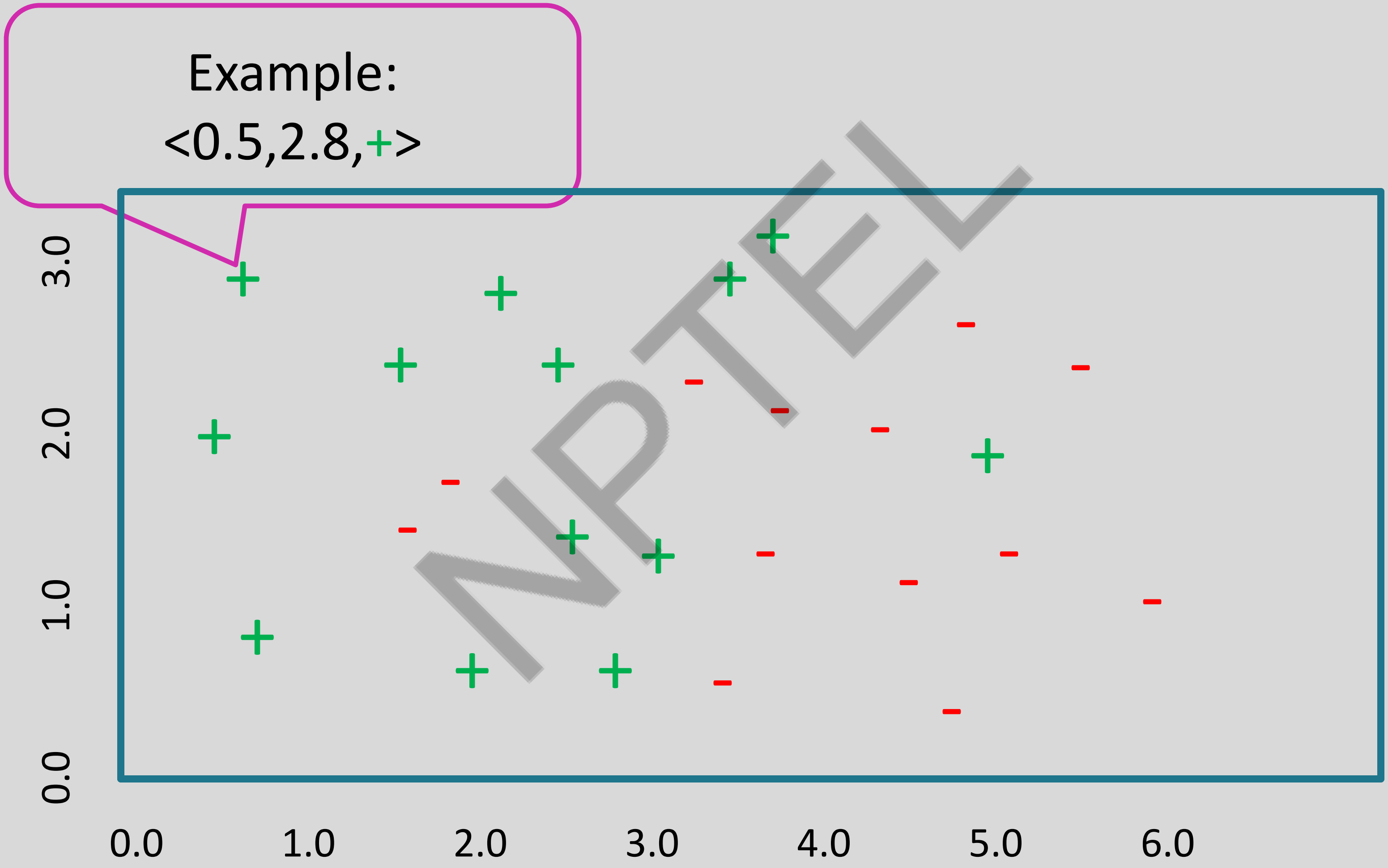
  F(X) = Continuous
- Probability estimation

  F(X) = Probability*(X):*

Features

- **Features**: Properties that describe each instance in a quantitative manner.
- **Feature vector**: n-dimensional vector of features that represent some object

Feature Space

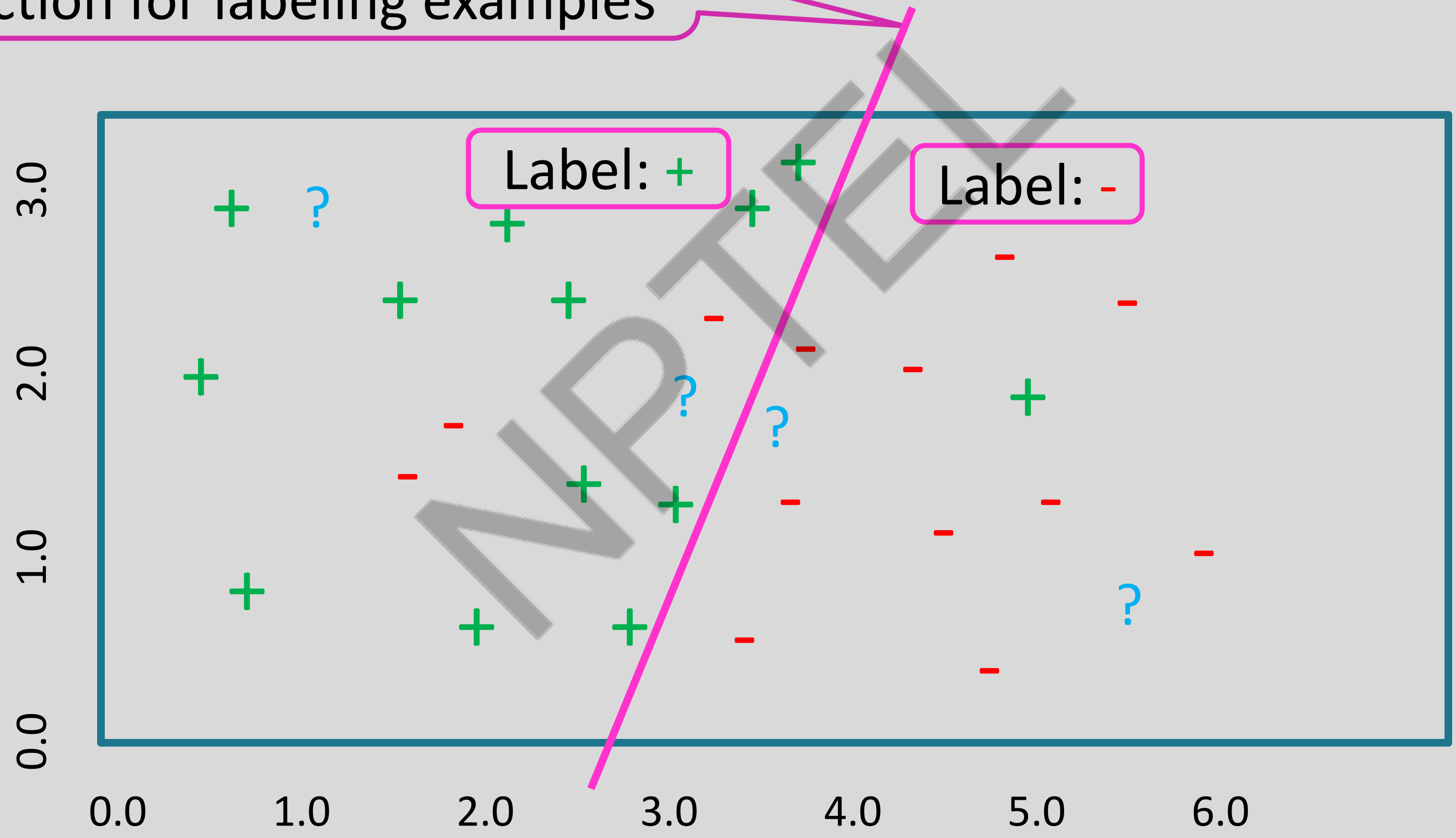


Example:
<0.5,2.8,+>

Slide by Jesse Davis: University of Washington

Terminology

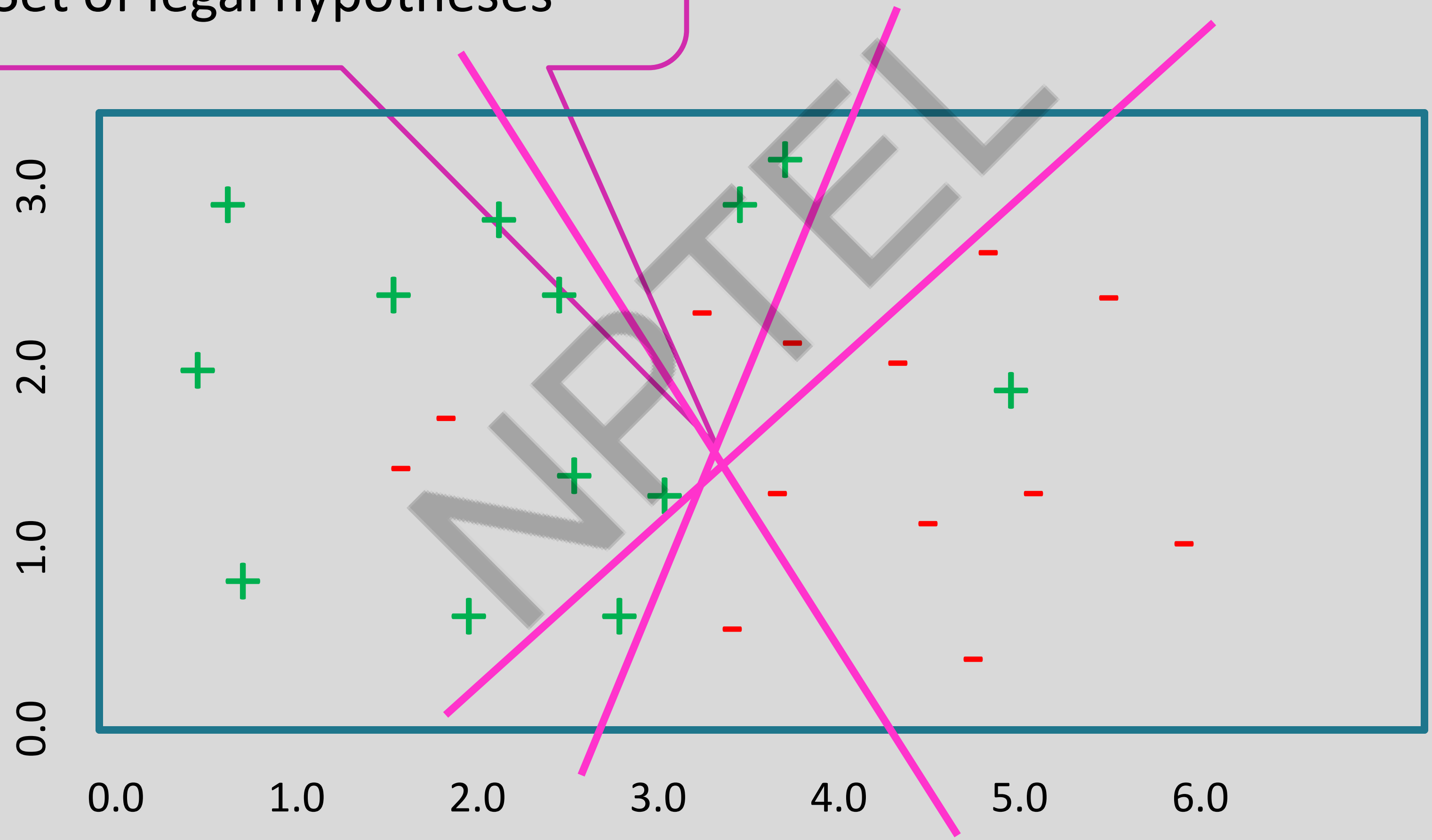
Hypothesis:
Function for labeling examples



Slide by Jesse Davis: University of Washington

Terminology

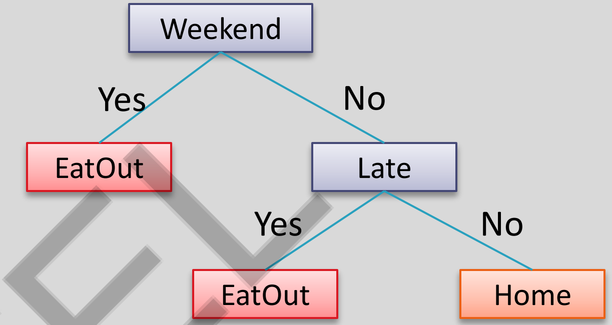
Hypothesis Space:
Set of legal hypotheses



Slide by Jesse Davis: University of Washington

Representations

1. Decision Tree



Weekend

Yes

No

EatOut

Late

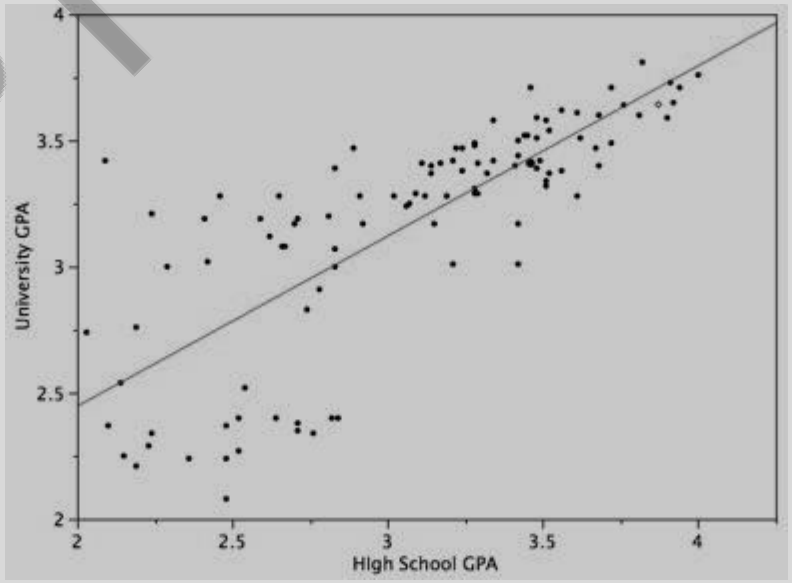
Yes

No

EatOut

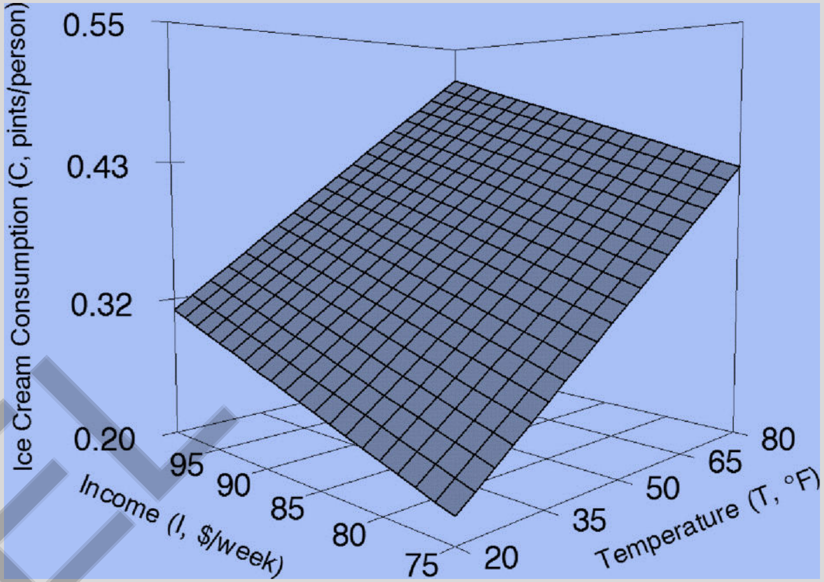
Home

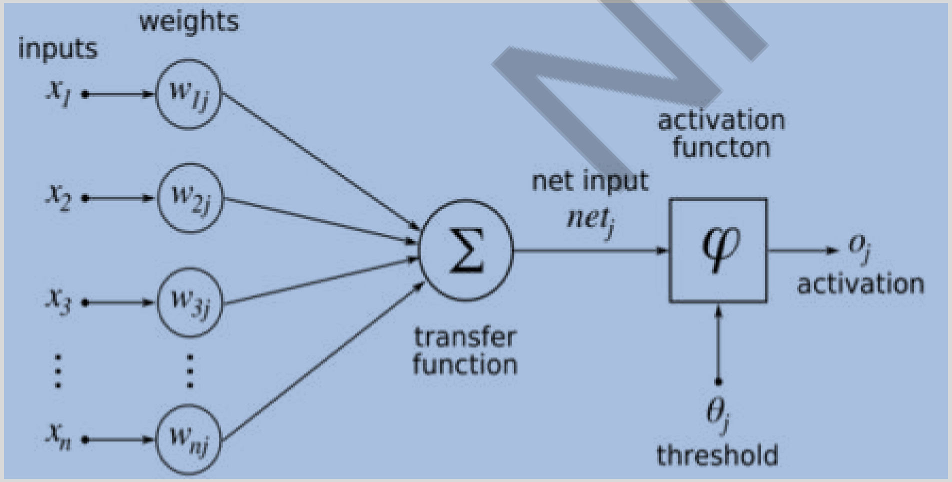
2. Linear function

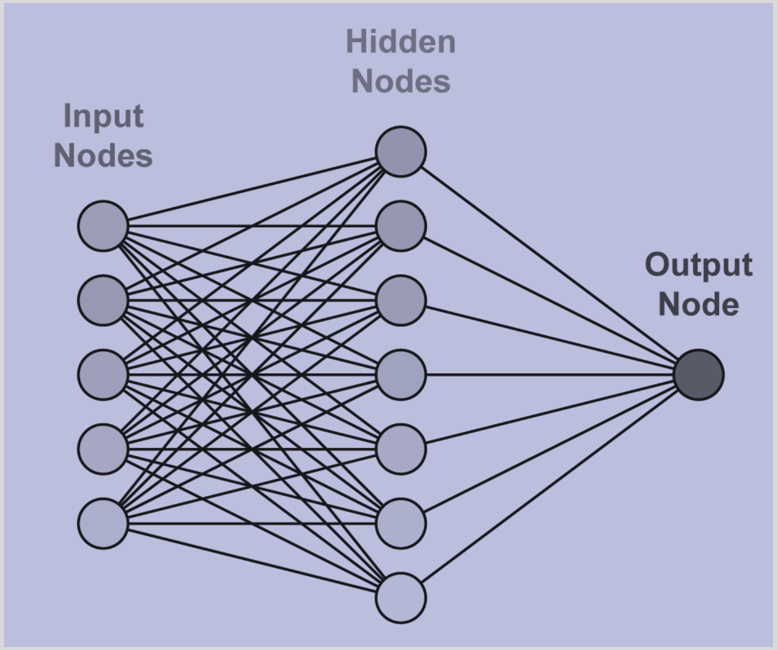


Representations

3. Multivariate linear function
4. Single layer perceptron
5. Multi-layer neural networks







Hypothesis Space

- The space of all hypotheses that can, in principle, be output by a learning algorithm.

- We can think about a supervised learning machine as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

Terminology

- **Example (x,y):** Instance x with label y.
- **Training Data S:** Collection of examples observed by learning algorithm.
- **Instance Space X**: Set of all possible objects describable by features.
- **Concept c:** Subset of objects from X (c is unknown).
- **Target Function f:** Maps each instance $x \in X$ to target label $y \in Y$

Classifier

- Hypothesis h: Function that approximates f.
- Hypothesis Space $\mathcal{H}$: Set of functions we allow for approximating f.
- The set of hypotheses that can be produced, can be restricted further by specifying a language bias.
- Input: Training set $\mathcal{S} \subseteq X$
- Output: A hypothesis $h \in \mathcal{H}$

Hypothesis Spaces

- If there are 4 (N) input features, there are 2^{16} $\left(2^{2^N}\right)$ possible Boolean functions.
- We cannot figure out which one is correct unless we see every possible input-output pair $2^4(2^N)$

Example

Hypothesis language

1. may contain representations of all polynomial functions from X to Y if X = $\mathcal{R}^n$ and Y = $\mathcal{R}$,
2. may be able to represent all conjunctive concepts over X when $X = B^n$ and $Y = B$ (with B the set of booleans).

- Hypothesis language reflects an inductive bias that the learner has

Inductive Bias

- Need to make assumptions
  - Experience alone doesn't allow us to make conclusions about unseen data instances

- Two types of bias:
  - Restriction: Limit the hypothesis space
  - Preference: Impose ordering on hypothesis space

Inductive learning

- Inductive learning: Inducing a general function from training examples
  - Construct hypothesis *h* to agree with *c* on the training examples.
  - A hypothesis is consistent if it agrees with all training examples.
  - A hypothesis said to generalize well if it correctly predicts the value of *y* for novel example.
- *Inductive Learning is an Ill Posed Problem*:
  Unless we see all possible examples the data is not sufficient for an inductive learning algorithm to find a unique solution.

Inductive Learning Hypothesis

- Any hypothesis h found to approximate the target function c well over a sufficiently large set of training examples $\mathcal{D}$ will also approximate the target function well over other unobserved examples.

Learning as Refining the Hypothesis Space

- Concept learning is a task of searching an hypotheses space of possible representations looking for the representation(s) that best fits the data, given the bias.
- The tendency to prefer one hypothesis over another is called a **bias**.
- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.

Occam's Razor

- A classical example of Inductive Bias

- the simplest consistent hypothesis about the target function is actually the best

Some more Types of Inductive Bias

- Minimum description length: when forming a hypothesis, attempt to minimize the length of the description of the hypothesis.

- Maximum margin: when drawing a boundary between two classes, attempt to maximize the width of the boundary (SVM)

Important issues in Machine Learning

- What are good hypothesis spaces?
- Algorithms that work with the hypothesis spaces
- How to optimize accuracy over future data points (overfitting)
- How can we have confidence in the result? (How much training data – statistical qs)
- Are some learning problems computationally intractable?

Generalization

- Components of generalization error
  - Bias: how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - Variance: how much models estimated from different training sets differ from each other

Underfitting and Overfitting

- Underfitting: model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- Overfitting: model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Foundations of Machine Learning

Module 1: Introduction

Part D: Evaluation and Cross validation

Sudeshna Sarkar
IIT Kharagpur

Experimental Evaluation of Learning Algorithms

- Evaluating the performance of learning systems is important because:
  - Learning systems are usually designed to predict the class of "future" unlabeled data points.
- Typical choices for Performance Evaluation:
  - Error
  - Accuracy
  - Precision/Recall
- Typical choices for Sampling Methods:
  - Train/Test Sets
  - K-Fold Cross-validation

Evaluating predictions

- Suppose we want to make a prediction of a value for a target feature on example **x**:
  - y is the observed value of target feature on example ***x***.
  - $\hat{y}$ is the predicted value of target feature on example ***x***.
  - How is the error measured?

Measures of error

- Absolute error: $\frac{1}{n}|f(x) - y|$
- Sum of squares error: $\frac{1}{n}\sum_{i=1}^{n}(f(x) - y)^2$
- Number of misclassifications: $\frac{1}{n}\sum_{i=1}^{n}\delta(f(x), y)$

- $\delta(f(x), y)$ is 1 if *f(x)* $\neq$ y, and 0, otherwise.

Confusion Matrix

True class → Hypothesized class ↓	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

- Accuracy = (TP+TN)/(P+N)
- Precision = TP/(TP+FP)
- Recall/TP rate = TP/P
- FP Rate = FP/N

Sample Error and True Error

- The ***sample error*** of hypothesis *f* with respect to target function *c* and data sample *S* is:

$$error_s(f)= 1/n\ \Sigma_{x \in S}\delta(f(x),c(x))$$

- The ***true error*** (denoted $error_D(f)$) of hypothesis *f* with respect to target function *c* and distribution *D*, is the probability that *h* will misclassify an instance drawn at random according to *D*.

$$error_D(f)= Pr_{x \in D}[f(x) \neq c(x)]$$

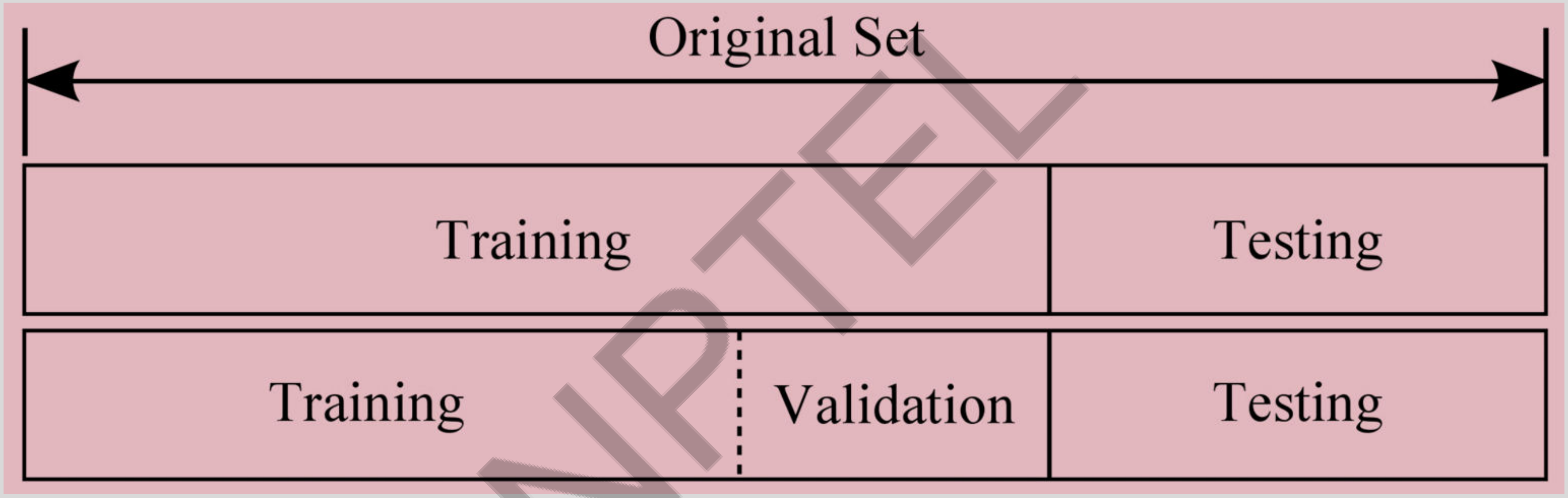
Why Errors

- Errors in learning are caused by:
  - Limited representation (representation bias)
  - Limited search (search bias)
  - Limited data (variance)
  - Limited features (noise)

Difficulties in evaluating hypotheses with limited data

- Bias in the estimate: The sample error is a poor estimator of true error
  - ==> test the hypothesis on an independent test set
- We divide the examples into:
  - **Training examples** that are used to train the learner
  - **Test examples** that are used to evaluate the learner
- Variance in the estimate: The smaller the test set, the greater the expected variance.

Validation set

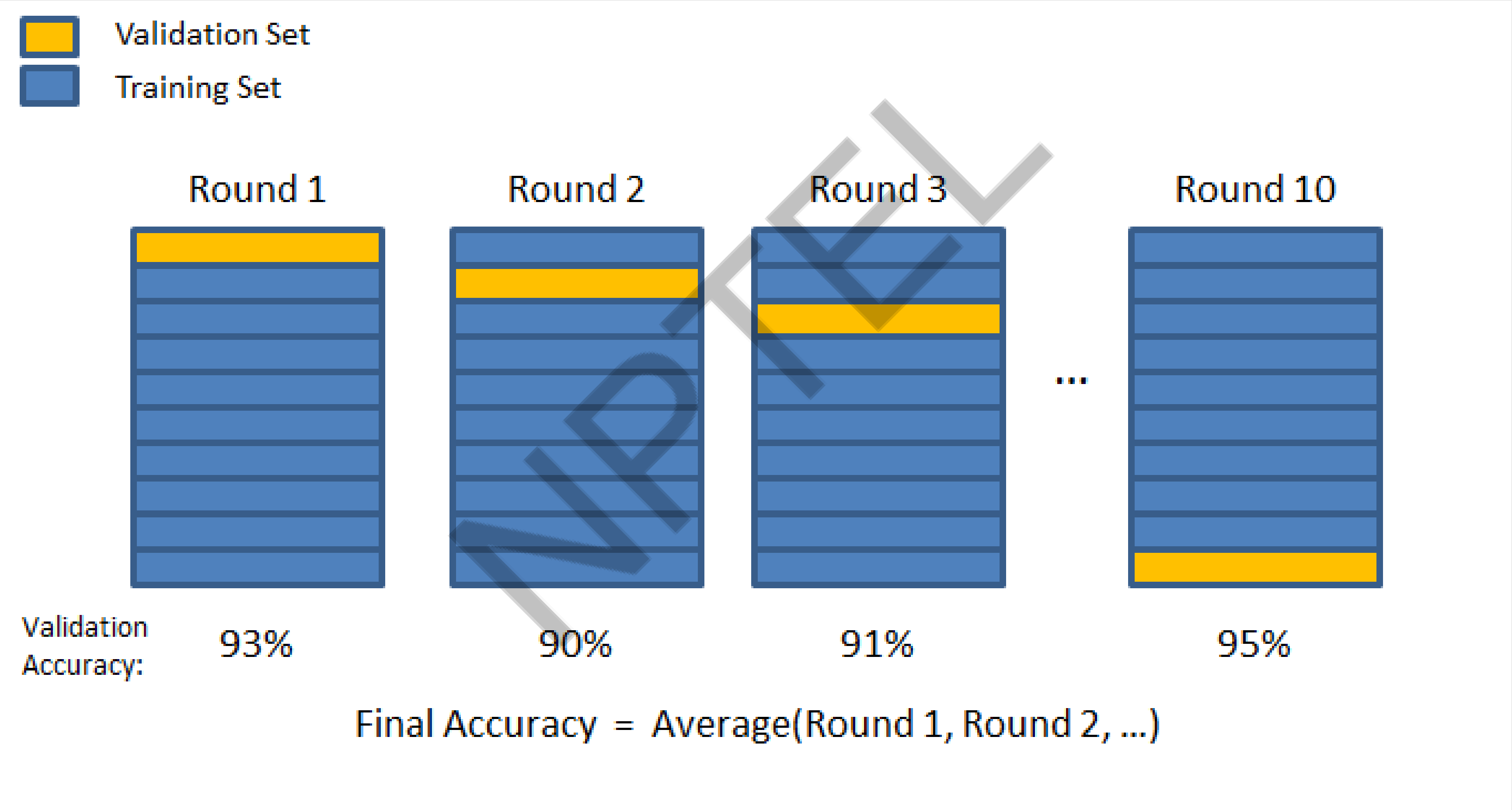


Validation fails to use all the available data

k-fold cross-validation

1. Split the data into k equal subsets
2. Perform k rounds of learning; on each round
   - 1/k of the data is held out as a test set and
   - the remaining examples are used as training data.
3. Compute the average test set score of the k rounds

K-fold cross validation



🟨 Validation Set

🟦 Training Set

Round 1 Round 2 Round 3 … Round 10

Validation Accuracy: 93% 90% 91% 95%

Final Accuracy = Average(Round 1, Round 2, …)

Trade-off

- In machine learning, there is always a trade-off between
  - complex hypotheses that fit the training data well
  - simpler hypotheses that may generalise better.
- As the amount of training data increases, the generalization error decreases.