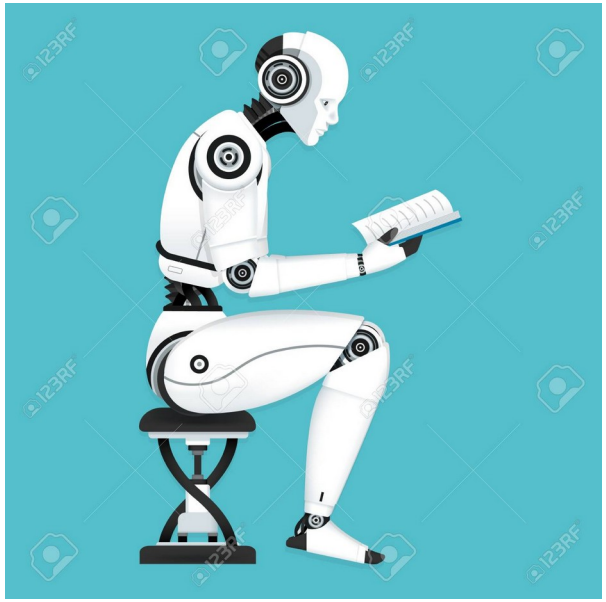
Machine Learning

CS60050



An Overview

What is Machine Learning?

- Optimize a performance criterion using example data or past experience.
- Role of Statistics: inference from a sample
- Role of Computer science: efficient algorithms to
  - Solve an optimization problem
  - Represent and evaluate the model for inference
- Learning is used when:
  - Human expertise does not exist (navigating on Mars),
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes with time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)
- There is no need to “learn” to calculate payroll

What we talk when we talk about “Learning”

- Learning general models from a data of particular examples
- Data is cheap and abundant (data warehouses, data marts); knowledge is expensive and scarce.
- Example in retail: Customer transactions to consumer behavior:

  People who bought “Da Vinci Code” also bought “The Five People You Meet in Heaven” (www.amazon.com)

- Build a model that is *a good and useful approximation* to the data.

Types of Learning Tasks

- Association
- Supervised learning
  - Learn to predict output when given an input vector
- Reinforcement learning
  - Learn action to maximize payoff
    - Payoff is often delayed
    - Exploration vs. exploitation
    - Online setting
- Unsupervised learning
  - Create an internal representation of the input e.g. form clusters; extract features
    - How do we know if a representation is good?
  - Big datasets do not come with labels.

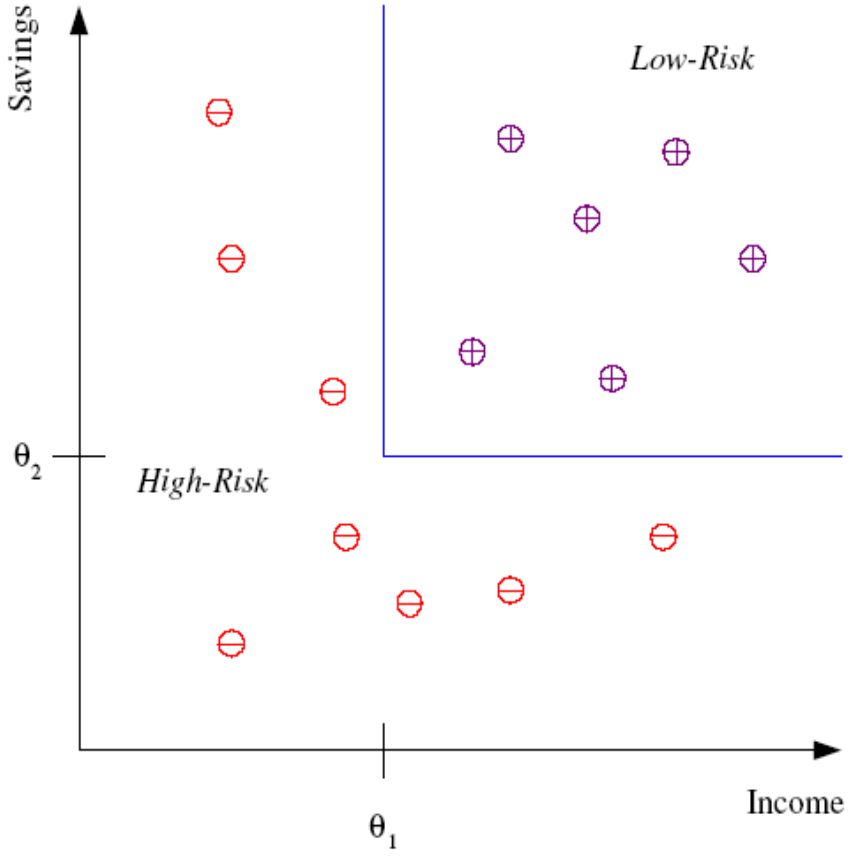
Learning Associations

- Basket analysis:
  - P (Y | X) probability that somebody who buys X also buys Y where X and Y are products/services.

- Example:
  - *P* (chips | cold-drinks) = 0.7

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

Classification: Applications

- a.k.a. Pattern Recognition
- Face recognition:
  - Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition:
  - Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- ...

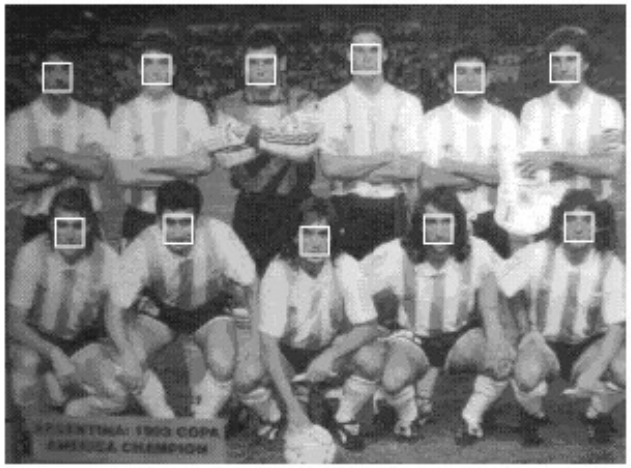
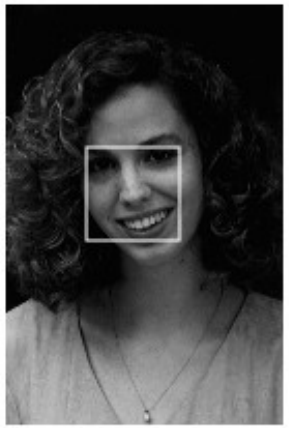
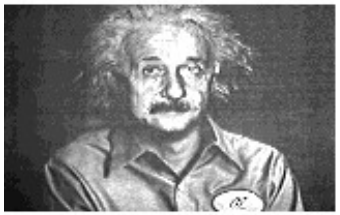
Face Recognition

Training examples of a person

Test images

The Role of Learning: *Face Recognition*



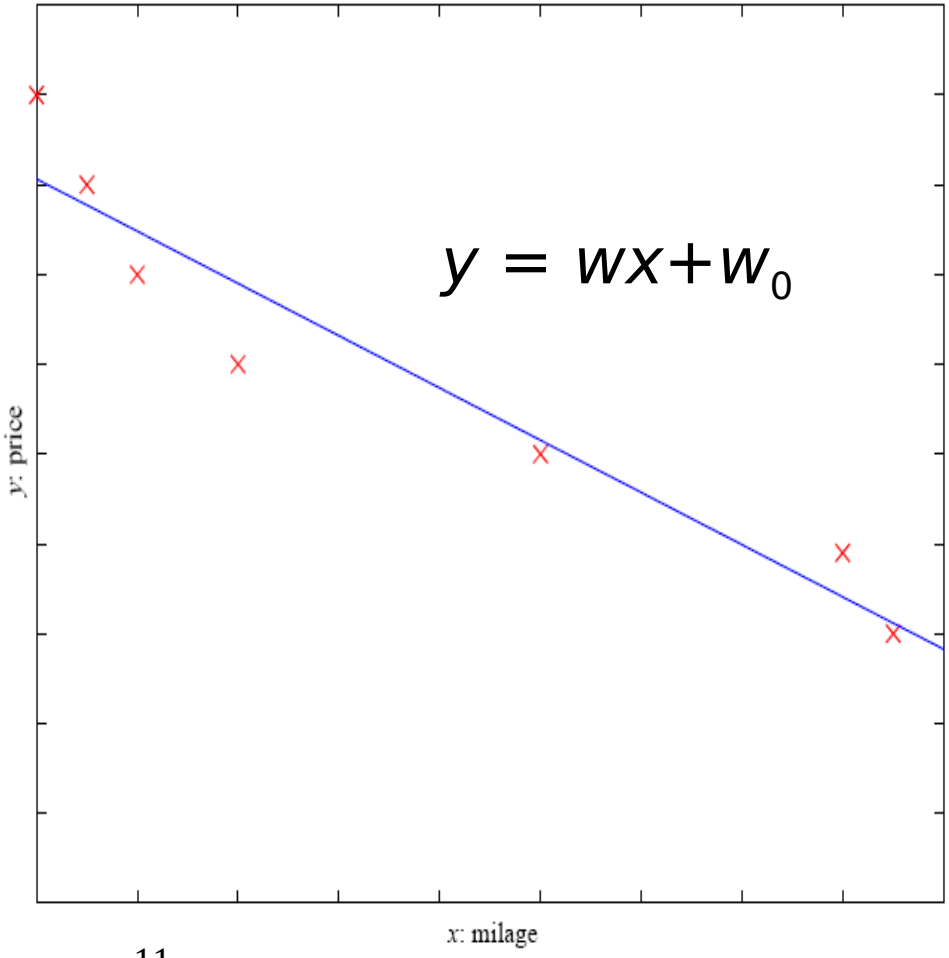







The Role of Learning: *Object Recognition*

Regression

- Example: Price of a used car
  - x : car attributes
  - y : price
  - $y = g\,(x, \theta)$
  - $g\,(\,)$ model
  - θ parameters



Supervised Learning: Applications

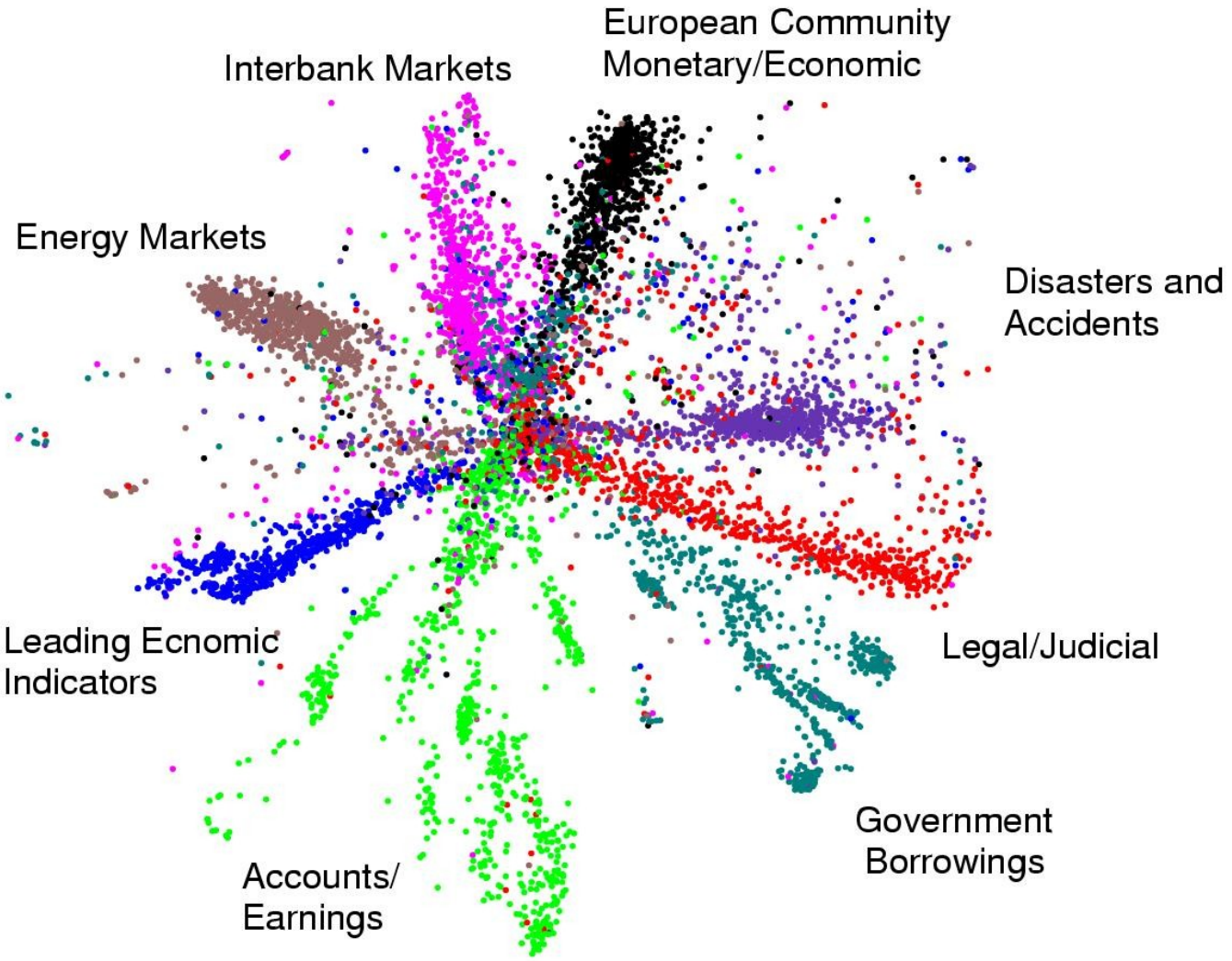
- Prediction of future cases:
  - Use the rule to predict the output for future inputs
- Knowledge extraction:
  - The rule is easy to understand
- Compression:
  - The rule is simpler than the data it explains
- Outlier detection:
  - Exceptions that are not covered by the rule, e.g., fraud

Unsupervised Learning

- Learning “what normally happens”
- Clustering: Grouping similar instances

- Example applications
  - Customer segmentation in CRM (customer relationship management)
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Displaying Structure of a Set of Documents



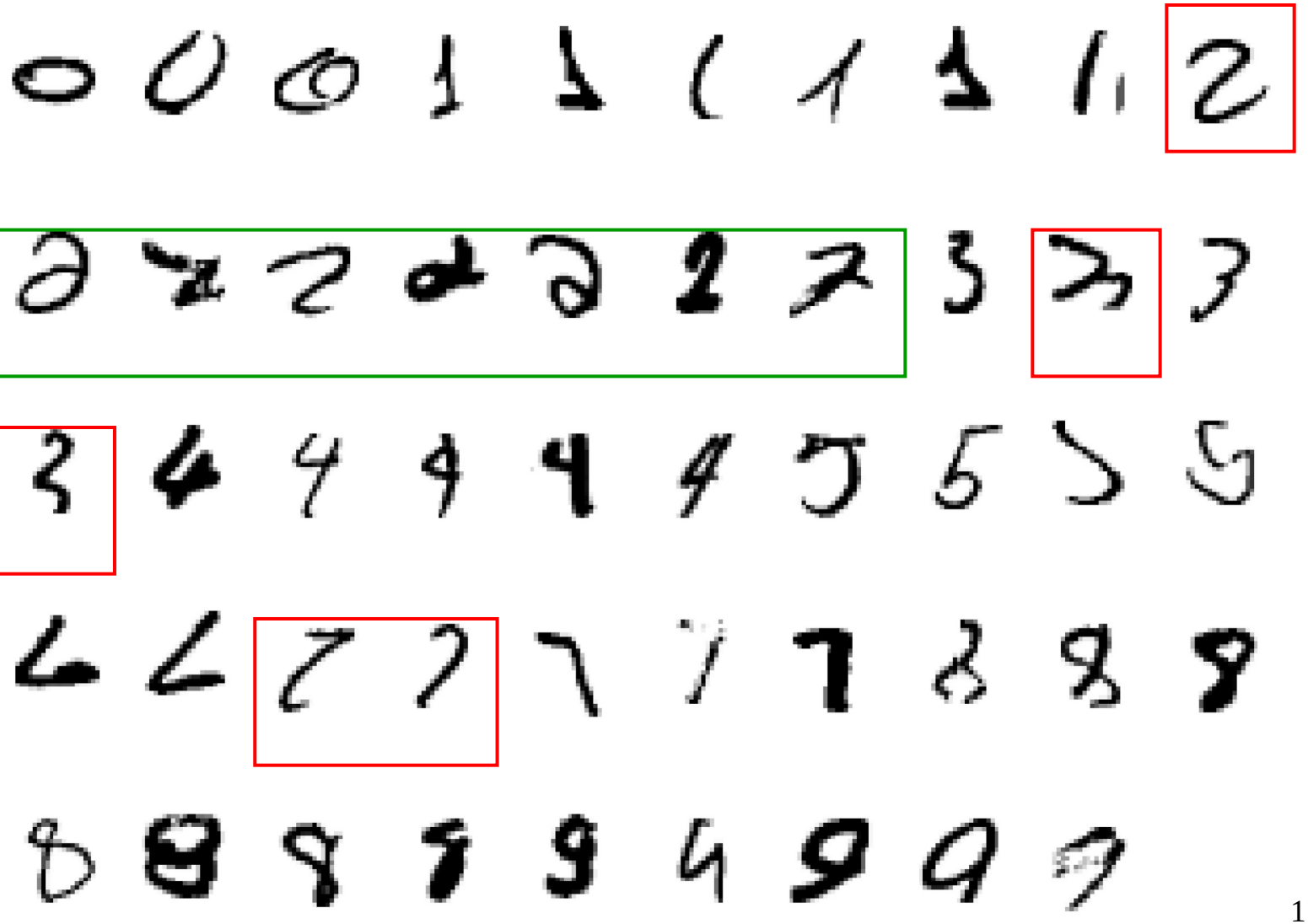
More Examples: *Netflix Review*

- Application: automatic product recommendation
- Importance: this is the modern/future shopping.
- Prediction goal: Based on past preferences, predict which movies you might want to watch
- *Data: Past movies you have watched*
- *Target: Like or don't-like*
- *Features: ?*

More Examples: *Zipcodes Digit Recognition*

- Application: automatic zipcode recognition
- Importance: this is modern/future delivery of small goods.
- Goal: Based on your handwritten digits, predict what they are and use them to route mail
- *Data: Black-and-white pixel values*
- *Target: Which digit*
- *Features: ?*

Which one makes a 2?



More Examples: *Google Advertisement*

- Application: automatic ad selection
- Importance: this is modern/future advertising.
- Prediction goal: Based on your search query, predict which ads you might be interested in
- *Data: Past queries*
- *Target: Whether the ad was clicked*
- *Features: ?*

More Examples: *Call Centers Connections*

- Application: automatic call routing
- Importance: this is modern/future customer service.
- Prediction goal: Based on your speech recording, predict which words you said
- *Data: Past recordings of various people*
- *Target: Which word was intended*
- *Features: ?*

More Examples: *Stock Market*

- Application: automatic program trading
- Importance: this is modern/future finance.
- Prediction goal: Based on past patterns, predict whether the stock will go up
- *Data: Past stock prices*
- *Target: Up or down*
- *Features: ?*

Web-based Examples of ML

- The web contains a lot of data. Tasks with very big datasets often use machine learning
  - especially if the data is noisy or non-stationary.
- Spam filtering, fraud detection:
  - The enemy adapts so we must adapt too.
- Recommendation systems:
  - Lots of noisy data. Million dollar prize!
- Information retrieval:
  - Find documents or images with similar content.

What is Machine Learning?

- **[Mitchell 1997]** ***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.***
- **Examples of the task, T**
  - Classification – to learn function $f: \mathbb{R}^n \rightarrow \{1, \ldots, k\}$
  - Classification with missing inputs
  - Regression – to learn function $f: \mathbb{R}^n \rightarrow \mathbb{R}$
    (predict weather, temp, stock market)
  - Transcription – for e.g., learning speech to text or image to text conversion
  - Machine translation – for e.g., translate English sentences to German
  - Structured output – for e.g., mark roads in aerially captured map
  - Anomaly detection
  - Denoising
  - Density estimation – to learn function $p_{model} : \mathbb{R}^n \rightarrow \mathbb{R}$ where $p_{model}(x)$ can be interpreted as a probability density function

What is Machine Learning?

- **[Mitchell 1997]** ***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.***
- **Performance measure, P**
  - For tasks like transcription and classification, measure accuracy of model
    - Accuracy = the proportion of examples for which the model produces the correct output
    - Error rate = the proportion of examples for which the model produces incorrect output
  - Data = { Training set, Test set }
    - The model is trained using the training set
    - Performance is assessed using the test set
  - Deciding the performance measure is not always very straight forward
    - For example, when performing a regression task, should we penalize the system more if it frequently makes medium sized mistakes or if it rarely makes very large mistakes

What is Machine Learning?

- **[Mitchell 1997]** ***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.***
- **The Experience, E**
  - Unsupervised Learning Algorithms experience a dataset containing many features, then learn useful properties of the structure of the dataset (such as anomaly detection, denoising, etc)
    - Learning the probability distribution p(x) by observing several examples of a random vector x.
  - Supervised Learning Algorithms experience a dataset containing features, but each example is also associated with a label or target
    - Observing several examples of a random vector x and an associated value or vector y, and then estimating p(y|x)

Supervised and Unsupervised Learning are NOT formally different

- The chain rule of probability states that for a vector $x \in \mathbb{R}^n$, the joint probability distribution can be decomposed as:

$$p(x) = \Pi^{n}_{i=1}\, p(x_i \mid x_1, x_2, \ldots, x_{i-1})$$

- The decomposition means that we can solve the problem of learning p(x) by splitting it into n supervised learning problems

- Alternatively we can solve the supervised learning problem p(y|x) by using the traditional unsupervised learning technologies to learn the joint distribution p(x, y), then inferring:

$$p(y|x) = p(x,y) / \Sigma_{y'}\, p(x,y')$$

What is a Learning Problem?

- Learning involves performance improving
  - at some task *T*
  - with experience *E*
  - evaluated in terms of performance measure *P*
- Example: learn to play checkers
  - Task *T*: playing checkers
  - Experience *E*: playing against itself
  - Performance *P*: percent of games won
- What exactly should be learned?
  - How might this be represented?
  - What specific algorithm should be used?

Develop methods, techniques and tools for building intelligent learning machines, that can solve the problem in combination with an available data set of training examples.

When a learning machine improves its performance at a given task over time, without reprogramming, it can be said to have learned something.

Learning Examples

- Example-1 (from Automated Game Playing field):
  - Problem: learn to play checkers
  - Task T: playing checkers
  - Performance P: percent of games won
  - Experience E: playing against itself

- Example-2 (from Machine/Computer Vision field):
  - Problem: learn to recognise objects from a visual scene or an image
  - Task T: identify all objects
  - Performance P: accuracy (e.g. number of objects correctly recognized)
  - Experience E: a database of objects recorded

Components of a Learning Problem

- **Task: the behavior or task that's being improved, e.g.** classification, object recognition, acting in an environment.
- **Data: the experiences that are being used to improve** performance in the task.
- **Measure of improvements: How can the improvement** be measured? Examples:
  - Provide more accurate solutions (e.g. increasing the accuracy in prediction)
  - Cover a wider range of problems
  - Obtain answers more economically (e.g. improved speed)
  - Simplify codified knowledge
  - New skills that were not presented initially

Learning = Generalization

H. Simon:

Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time.”

The ability to perform a task in a situation which has never been encountered before

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.
  - If the data is noise-free, each training example rules out a region of hypothesis space.
  - If the data is noisy, each training example scales the posterior probability of each point in the hypothesis space in proportion to how likely the training example is given that hypothesis.

- The art of supervised machine learning is in:
  - Deciding how to represent the inputs and outputs
  - Selecting a hypothesis space that is powerful enough to represent the relationship between inputs and outputs but simple enough to be searched.

Generalization

- The real aim of supervised learning is to do well on test data that is not known during learning.
- Choosing the values for the parameters that minimize the loss function on the training data is not necessarily the best policy.
- We want the learning machine to model the true regularities in the data and to ignore the noise in the data.
  - But the learning machine does not know which regularities are real and which are accidental quirks of the particular set of training examples we happen to pick.
- So how can we be sure that the machine will generalize correctly to new data?

Goodness of Fit vs. Model Complexity

- It is intuitively obvious that you can only expect a model to generalize well if it explains the data surprisingly well given the complexity of the model.

- If the model has as many degrees of freedom as the data, it can fit the data perfectly but so what?

- There is a lot of theory about how to measure the model complexity and how to control it to optimize generalization.

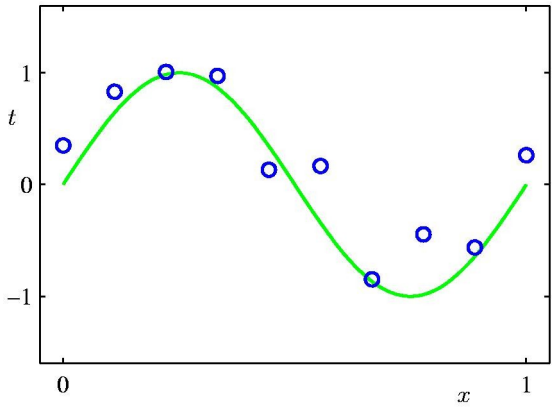
A Sampling Assumption

- Assume that the training examples are drawn independently from the set of all possible examples.
- Assume that each time a training example is drawn, it comes from an independent identical distribution (i.i.d)
- Assume that the test examples are drawn in exactly the same way – i.i.d. and from the same distribution as the training data.
- These assumptions make it very unlikely that a strong regularity in the training data will be absent in the test data.

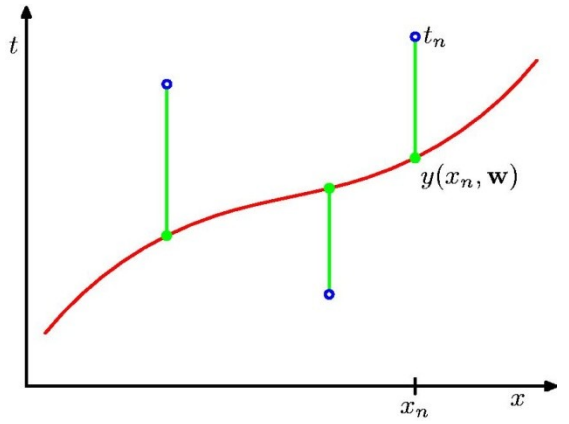
A Simple Example: Fitting a Polynomial

- The green curve is the true function (which is not a polynomial)
- The data points are uniform in x but have noise in y.

from Bishop

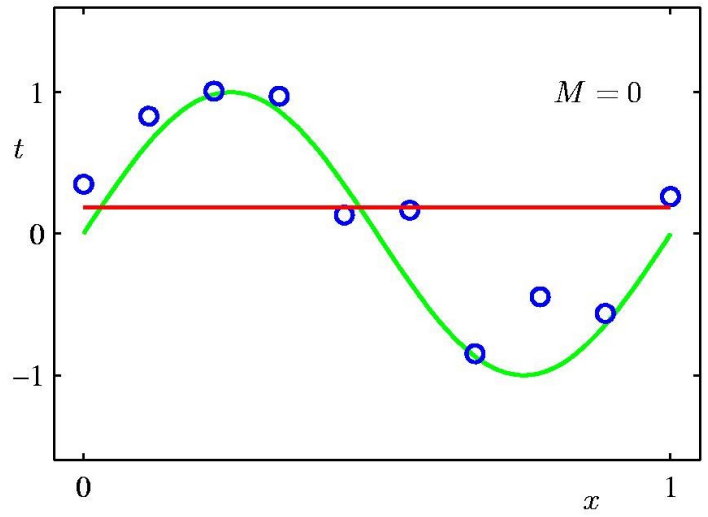


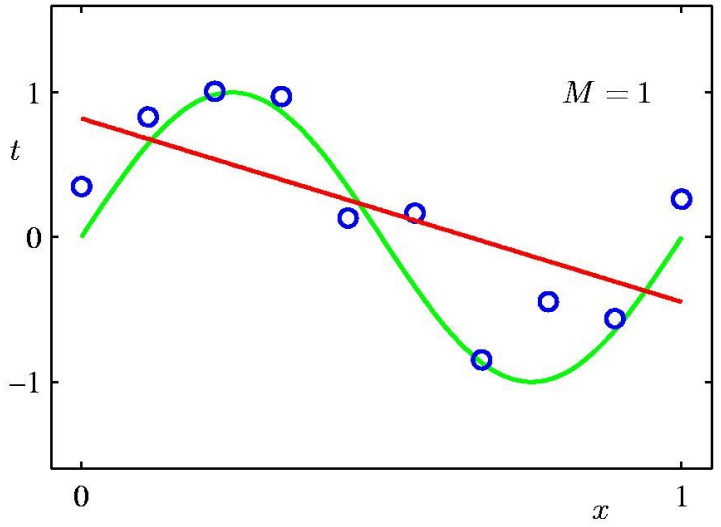
- We will use a loss function that measures the squared error in the prediction of y(x) from x. The loss for the red polynomial is the sum of the squared vertical errors.

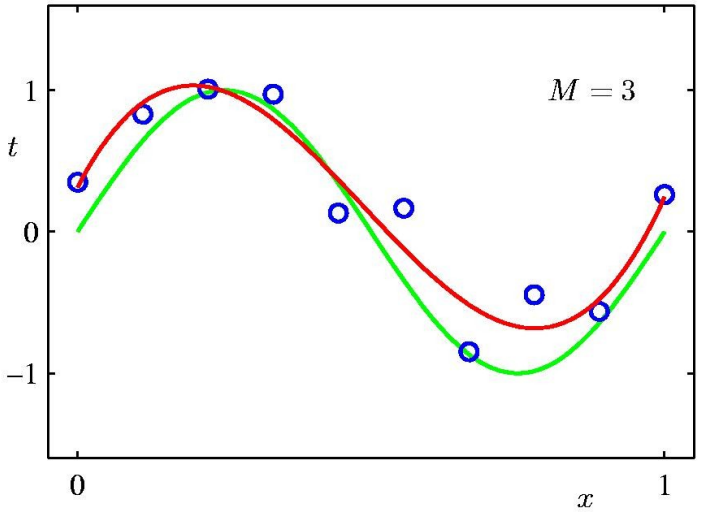


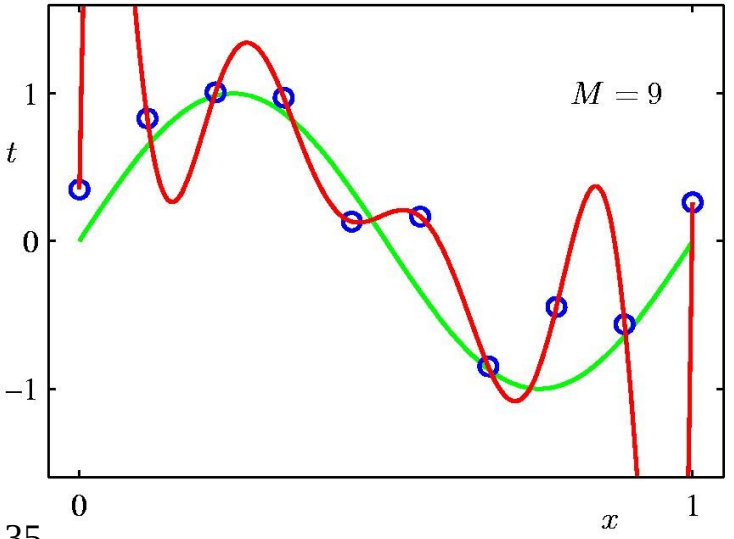
Some Fits to Data: *Which is the Best?*

from Bishop





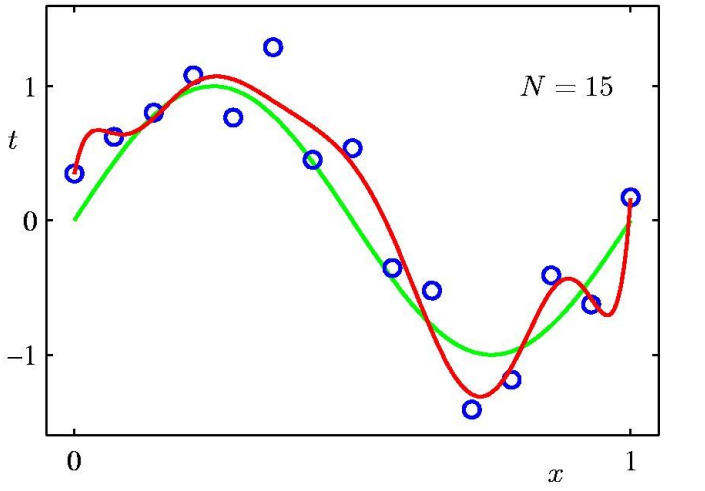




A simple way to Reduce Model Complexity

- If we penalize polynomials that have a high number of coefficients, we will get less wiggly solutions:

from Bishop



Occam's Razor

What Experience E to Use?

- Direct or indirect?
  - Direct: feedback on individual moves
  - Indirect: feedback on a sequence of moves
    - e.g., whether win or not
- Teacher or not?
  - Teacher selects board states
    - Tailored learning
    - Can be more efficient
  - Learner selects board states
    - No teacher
- Questions
  - Is training experience representative of performance goal?
  - Does training experience represent distribution of outcomes in world?

What Exactly Should be Learned?

- Playing checkers:
  - Alternating moves with well-defined rules
  - Choose moves using some function
  - Call this function the *Target Function*
- *Target function (TF):* function to be learned during a learning process
  - *ChooseMove*: Board → Move
  - *ChooseMove* is difficult to learn, e.g., with indirect training example
- A key to successful learning is to choose *appropriate* target function:
  - Strategy: ***reduce learning to search for TF***

- Alternative TF for checkers:
  - V : Board → $\mathbb{R}$
  - Measure "quality" of the board state
  - Generate all moves
    - choose move with largest value

A Possible Target Function V For Checkers

- In checkers, know all legal moves
  - From these, choose best move in any situation
- Possible V function for checkers:
  - **if** b is a final board state that is win, then $V(b) = 100$
  - if b is a final board state that is loss, then $V(b) = -100$
  - if b is a final board state that is draw, then $V(b) = 0$
  - if b is a not a final state in the game, then $V(b) = V(b')$, where b' is the best final board state that can be achieved starting from b and playing optimally until the end of the game
- This gives correct values, but is *not* operational
  - So may have to find good approximation to V
  - Call this approximation $\hat{V}$

How Might Target Function be Represented?

- Many possibilities (subject of course)
  - As collection of rules ?
  - As neural network ?
  - As polynomial function of board features ?
- Example of linear function of board features:
  - $\hat{V}(b): \; w_0 + w_1.bp(b) + w_2.rp(b) + w_3.bk(\text{b}) + w_4.rk(b) + w_5.bt(b) + w_6.rt(\text{b})$
    - $bp(b)$: number of black pieces on board b
    - $rp(b)$: number of red pieces on b
    - $bk(\text{b})$: number of black kings on b
    - $rk(b)$: number of red kings on b
    - $bt(b)$: number of red pieces threatened by black (i.e., which can be taken on black's next turn)
    - $rt(\text{b})$: number of black pieces threatened by red
- Generally, the more expressive the representation, the more difficult it is to estimate

Obtaining Training Examples

- $\hat{V}(b) = w_0 + w_1 bp(b) + w_2 rp(b) + w_3 bk(b) + w_4 rk(b) + w_5 bt(b) + w_6 rt(b)$
  - With *learned* function $\hat{V}(b)$:
  - Search over space of weights: estimate w_i
  - Training values that are needed $V_{\text{train}}(b)$
    - Some from prior experience; some generated
    - Example of training examples: (‹3,0,1,0,0,0›, +100)
- One rule for estimating training value

$$V_{\text{train}}(b) \leftarrow \hat{V}(\text{successor}(b))$$

  - successor(b) is for which it is program's turn to move
  - Used for intermediate values
  - good in practice
- Issue now of how to estimate weights w_i

Example of LMS Weight Update Rule

$$E = \sum_{(b, V_{\text{train}}) \in \text{training examples}} (V_{\text{train}}(b) - \hat{V}(b))^2$$

- Choose weights to minimize squared error
- Do repeatedly:
  - Select a training example b at random
    - 1. Compute $error(b) = V_{\text{train}}(b) - \hat{V}(b)$
    - 2. for each board feature x_i, update weight w_i

$$w_i \leftarrow w_i + c.\, x_i.error(b)$$ Gradient descent

    - 3. If error > 0, w_i increases and vice versa

Some Issues in Machine Learning

- What algorithms can approximate functions well (and when)?
- How does number of training examples influence accuracy?
- How does complexity of hypothesis representation impact learning?
- How does noisy data influence accuracy?
- What are the theoretical limits of learnability?
- How can prior knowledge of learner help?
- What clues can we get from biological learning systems?
- How can systems alter their own representations?

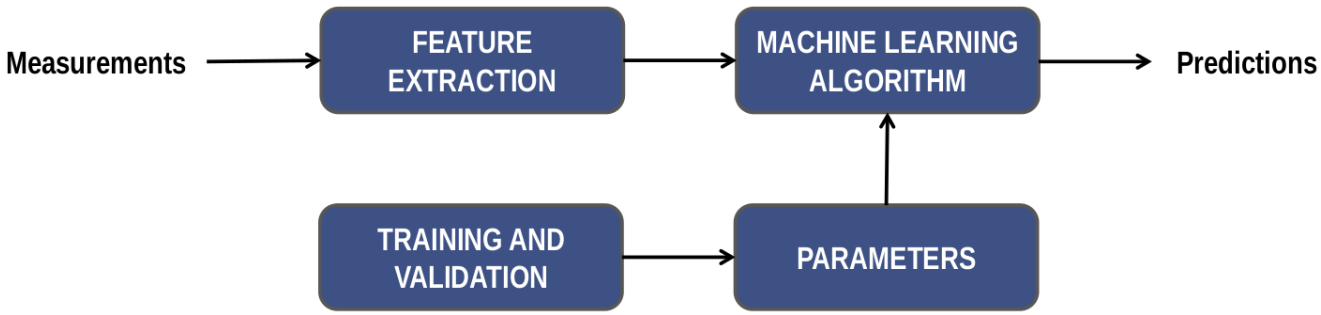
Learning Feedback

- Learning feedback can be provided by the system environment or the agents themselves.
  - Supervised learning: specifies the desired activities/objectives of learning – feedback from **a teacher**
  - Unsupervised learning: no explicit feedback is provided and the objective is to find out useful and desired activities on the basis of trial-and-error and self-organization processes – **a passive observer**
  - Reinforcement learning: specifies the utility of the actual activity of the learner and the objectives is to maximize this utility – feedback from **a critic**

Machine Learning Process

For (supervised) classification and regression (the most common tasks):

1. Algorithm selection: Choose an algorithm
2. Feature selection: Choose features that capture the important characteristics of the system
3. Training/model building: Use part of the labeled set to build model
4. Parameter optimization (cross-validation): Optimize the parameters using a second part of the labeled set to minimize the error rate
5. Test: Use the remainder of the dataset to validate and assess the performance of the tuned model
6. Apply



Ways of Learning

- Rote learning, i.e. learning from memory; in a mechanical way
- Learning from examples and by practice
- Learning from instructions/advice/explanations
- Learning by analogy
- Learning by discovery
- …

Inductive and Deductive Learning

- ***Inductive Learning:*** **Reasoning from a set of examples** to produce a general rules. The rules should be applicable to new examples, but there is no guarantee that the result will be correct.

- ***Deductive Learning:*** **Reasoning from a set of known** facts and rules to produce additional rules that are guaranteed to be true.

Assessment of Learning Algorithms

- The most common criteria for learning algorithms assessments:
  - Accuracy (e.g. % of correctly classified +'s and –'s)
  - Efficiency (e.g. examples needed, computational tractability)
  - Robustness (e.g. against noise, against incompleteness)
  - Special requirements (e.g. incrementality, concept drift)
  - Concept complexity (e.g. representational issues – examples & bookkeeping)
  - Transparency (e.g. comprehensibility for the human user)

Some Theoretical Settings

- Inductive Logic Programming (ILP)
- Probably Approximately Correct (PAC) Learning
- Learning as Optimization (Reinforcement Learning)
- Bayesian Learning
- …

Key Aspects of Learning

- Learner: who or what is doing the learning, e.g. an algorithm, a computer program.
- Domain: what is being learned, e.g. a function, a concept.
- Goal: why the learning is done.
- Representation: the way the objects to be learned are represented.
- Algorithmic Technology: the algorithmic framework to be used, e.g. decision trees, lazy learning, artificial neural networks, support vector machines

The Role of Learning

- **Learning is at the core of**
  - Understanding High Level Cognition
  - Performing knowledge intensive inferences
  - Building adaptive, intelligent systems
  - Dealing with messy, real world data
- **Learning has multiple purposes**
  - Knowledge Acquisition
  - integration of various knowledge sources to ensure robust behavior
  - Adaptation (human, systems)

An Owed to the Spelling Checker

I have a spelling checker.

It came with my PC

It plane lee marks four my revue

Miss steaks aye can knot sea.

Eye ran this poem threw it.

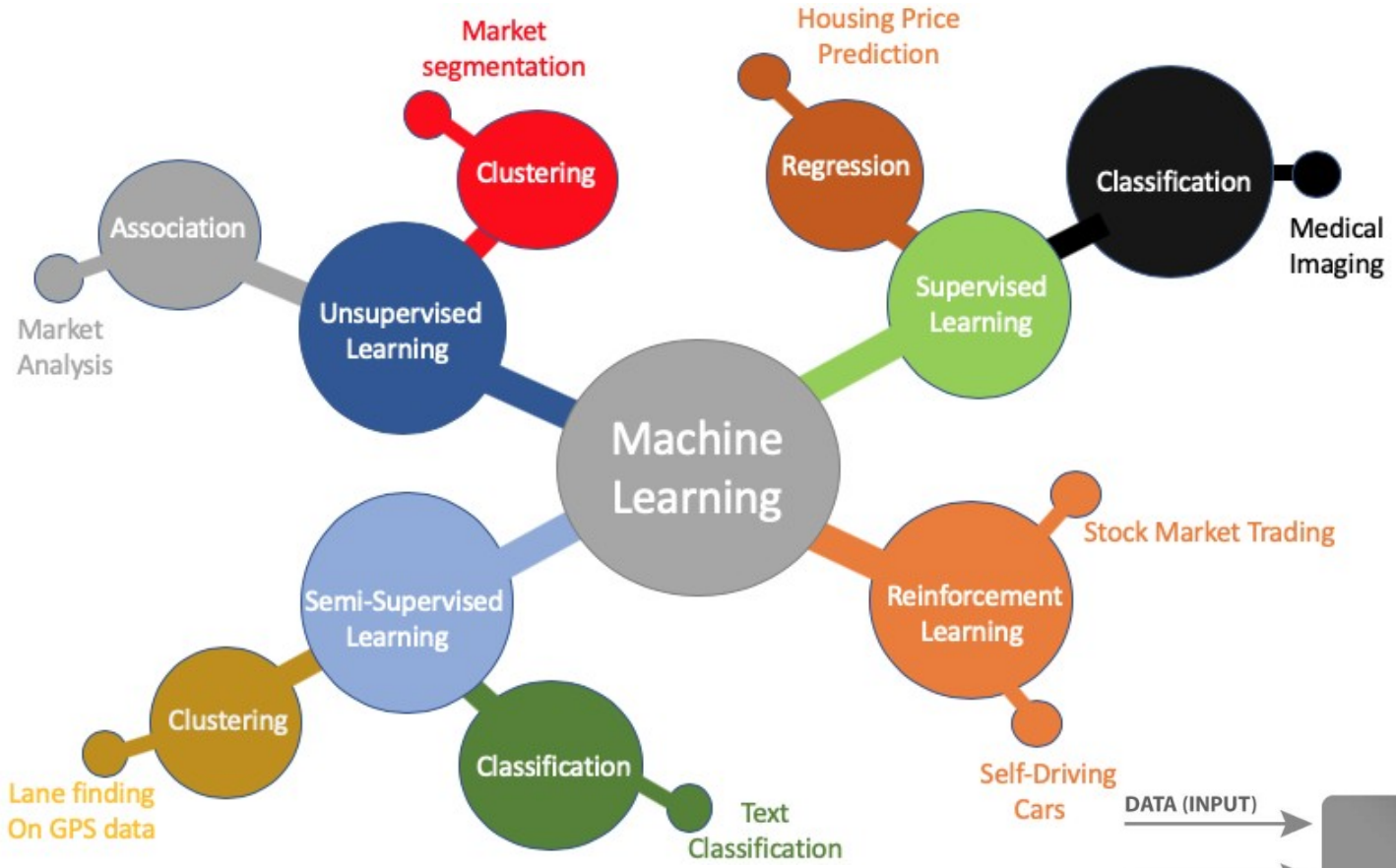
your sure reel glad two no.

Its vary polished in it's weigh

My checker tolled me sew.

……..

Thank You!



Machine Learning: *The Landscape*