Supervised Learning

These slides were assembled by Eric Eaton, with grateful acknowledgement of the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution. Please send comments and corrections to Eric.

The Badges Game

+ Naoki Abe

- Eric Baum

Background:

- Pre-registered attendees at the 1994 Machine Learning Conference received a name badge labeled with a "+" or "-"
- The label is based only upon the name
- There are 294 examples (210 positive and 84 negative)

What function was used to generate the +/- labeling?

Training Data

+ Naoki Abe
- Myriam Abramson
+ David W. Aha
+ Kamal M. Ali
- Eric Allender
+ Dana Angluin
- Chidanand Apte
+ Minoru Asada
+ Lars Asker
+ Javed Aslam
+ Jose L. Balcazar
- Cristina Baroglio

+ Peter Bartlett
- Eric Baum
+ Welton Becket
- Shai Ben-David
+ George Berg
+ Neil Berkman
+ Malini Bhandaru
+ Bir Bhanu
+ Reinhard Blasig
- Avrim Blum
- Anselm Blumer
+ Justin Boyan

+ Carla E. Brodley
+ Nader Bshouty
- Wray Buntine
- Andrey Burago
+ Tom Bylander
+ Bill Byrne
- Claire Cardie
+ John Case
+ Jason Catlett
- Philip Chan
- Zhixiang Chen
- Chris Darken

+ Naoki Abe
+ Kamal M. Ali
- Chidanand Apte
+ Javed Aslam
+ Timothy P. Barber
+ Peter Bartlett
- Shai Ben-David
+ Malini Bhandaru
- Avrim Blum
+ Carla E. Brodley
- Andrey Burago
- Claire Cardie
+ Jason Catlett
+ Mark Changizi
+ Wan P. Chiang
+ William Cohen
- Antoine Cornuejols
+ Lindley Darden
- Brian D. Davidson
- Scott E. Decatur
- Thomas G. Dietterich
+ Harris Drucker
- Thomas Ellman
+ Bob Evans
- Usama Fayyad
+ David Finton
+ Seth Flanders
+ Judy A. Franklin
+ Merrick L. Furst
+ Ricard Gavalda
+ David Gillman
+ Paul W. Goldberg
+ Geoffrey Gordon
+ William A. Greene
+ Tal Grossman
+ Earl S. Harris Jr.
+ Lisa Hellerstein
+ Haym Hirsh
+ Jiarong Hong
+ Masayuki Inaba
+ Jeff Jackson
- Klaus P. Jantke
+ Randolph Jones
+ Bala Kalyanasundaram
+ Michael Kearns
+ Dennis F. Kibler
- Jyrki Kivinen
+ Ron Kohavi
+ Daniel Kortenkamp

- Myriam Abramson
- Eric Allender
+ Minoru Asada
+ Haralabos Athanassiou
+ Michael W. Barley
- Eric Baum
+ George Berg
+ Bir Bhanu
- Anselm Blumer
+ Nader Bshouty
+ Tom Bylander
+ Richard A. Caruana
+ Nicolo Cesa-Bianchi
+ Pang-Chieh Chen
- Steve A. Chien
+ David Cohn
+ Mark W. Craven
- Chris Darken
+ Michael de la Maza
+ Gerald F. DeJong
+ Michael J. Donahue
- Chris Drummond
+ Tapio Elomaa
- Claudio Facchinetti
+ Aaron Feigelson
+ John Fischer
+ Lance Fortnow
+ Yoav Freund
+ Jean Gabriel Ganascia
+ Melinda T. Gervasio
- Attilio Giordana
+ Sally Goldman
+ Jonathan Gratch
+ Russell Greiner
+ Margo Guertin
+ David Haussler
+ David Helmbold
+ Jonathan Hodgson
- Chun-Nan Hsu
- Drago Indjic
+ Sanjay Jain
+ Nathalie Japkowicz
+ Michael I. Jordan
- Thomas E. Kammeyer
+ Neela Khan
+ Jorg-Uwe Kietz
- Emanuel Knill
+ Pascal Koiran
+ Matevz Kovacic

+ David W. Aha
+ Dana Angluin
+ Lars Asker
+ Jose L. Balcazar
- Cristina Baroglio
+ Welton Becket
+ Neil Berkman
+ Reinhard Blasig
+ Justin Boyan
- Wray Buntine
+ Bill Byrne
+ John Case
- Philip Chan
- Zhixiang Chen
+ Jeffery Clouse
- Clare Bates Congdon
+ Robert P. Daley
- Bhaskar Dasgupta
- Olivier De Vel
+ Kan Deng
+ George A. Drastal
+ Hal Duncan
+ Susan L. Epstein
+ Tom Fawcett
+ Nicolas Fiechter
+ Paul Fischer
- Ameur Foued
+ Johannes Furnkranz
+ William Gasarch
+ Yolanda Gil
+ Kate Goelz
+ Diana Gordon
+ Leslie Grate
+ Marko Grobelnik
+ Tom Hancock
+ Matthias Heger
+ Daniel Hennessy
+ Robert C. Holte
+ Kazushi Ikeda
+ Nitin Indurkhya
+ Wolfgang Janko
+ George H. John
+ Leslie Pack Kaelbling
- Grigoris Karakoulas
+ Roni Khardon
- Efim Kinber
- Craig Knoblock
+ Moshe Koppel
- Stefan Kramer

+ Martinch Krikis
- Stephen Kwek
- Steffen Lange
+ Wee Sun Lee
- Charles X. Ling
- Phil Long
+ Rich Maclin
+ Yishay Mansour
- Oded Maron
+ Toshiyasu Matsushima
- R. Andrew McCallum
+ Michael A. meystel
+ Tom M. Mitchell
- Andrew W. Moore
- Stephen Muggleton
+ Filippo Neri
+ Nikolay Nikolaev
+ Dan Oblinger
+ David W. Opitz
- Ed Pednault
+ Aurora Perez
- Krishnan Pillaipakkamnatt
+ Lorien Y. Pratt
- J. R. Quinlan
- R. Bharat Rao
+ Michael Redmond
+ Ronald L. Rivest
+ Robert S. Roos
+ Dan Roth
- Stuart Russell
+ William Sakas
- Claude Sammut
+ Mark Schwabacher
+ Sebastian Seung
+ Daniel L. Silver
+ Mona Singh
+ David B. Skalak
+ Donna Slonim
+ Von-Wun Soo
- Frank Stephan
+ Joe Suzuki
- Irina Tchoumatchenko
+ Tatsuo Unemi
+ Karsten Verbeurgt
+ Manfred Warmuth
- Thomas Wengerek
+ Robert Williamson
+ Takefumi Yamazaki
- Thomas Zeugmann

+ Martin Kummer
+ Wai Lam
+ Pat Langley
+ Moshe Leshno
+ Michael Littman
+ Wolfgang Maass
- Sridhar Mahadevan
+ Mario Marchand
+ Maja Mataric
- Stan Matwin
- L. Thorne McCarty
- Steven Minton
+ Dunja Mladenic
+ Johanne Morin
+ Patrick M. Murphy
- Craig Nevill-Manning
- Steven W. Norton
+ Jong-Hoon Oh
+ Sandra Panizza
+ Jing Peng
+ Bernhard Pfahringer
+ Roberto Piola
- Armand Prieditis
+ John Rachlin
- Priscilla Rasmussen
+ Patricia J. Riddle
+ Huw Roberts
+ Justinian Rosca
+ James S. Royer
+ Lorenza Saitta
+ Marcos Salganicoff
+ Cullen Schaffer
+ Michele Sebag
- Arun Sharma
- Glenn Silverstein
+ Satinder Pal Singh
+ Sean Slattery
+ Carl H. Smith
- Thomas G. Spalthoff
+ Mandayam T. Suraj
- Prasad Tadepalli
- Brian Tester
- Lyle H. Ungar
+ Paul Vitanyi
+ Gary Weiss
- Bradley L. Whitehall
+ Janusz Wnek
+ Holly Yanco
+ Jean-Daniel Zucker

- Eyal Kushilevitz
+ Ken Lang
+ Mary Soon Lee
+ Long-Ji Lin
+ David Loewenstern
- Bruce A. MacDonald
- J. Jeffrey Mahoney
- Shaul Markovitch
+ David Mathias
- Eddy Mayoraz
- Alexander M. Meystel
+ Nina Mishra
+ David Montgomery
+ Hiroshi Motoda
- Sreerama K. Murthy
- Andrew Y. Ng
+ Joseph O'Sullivan
- Arlindo Oliveira
+ Barak A. Pearlmutter
+ Fernando Pereira
+ David Pierce
+ Leonard Pitt
+ Foster J. Provost
+ Vijay Raghavan
+ Joel Ratsaby
+ Lance Riley
+ Dana Ron
+ John R. Rose
+ Ronitt Rubinfeld
+ Yoshifumi Sakai
- Steven Salzberg
+ Robert Schapire
+ Gary M. Selzer
+ Jude Shavlik
+ Yoram Singer
+ Kimmen Sjolander
+ Robert Sloan
+ Sonya Snedecor
+ Mark Staley
+ Richard S. Sutton
+ Hiroshi Tanaka
- Chen K. Tham
+ Paul Utgoff
+ Xuemei Wang
- Sholom Weiss
- Alma Whitten
+ Kenji Yamanishi
+ John M. Zelle
+ Darko Zupanic

Test Data

? Shivani Agarwal
? Chris Callison-Burch
? Eric Eaton
? Peter Stone
? Matthew Taylor

Labeled Test Data

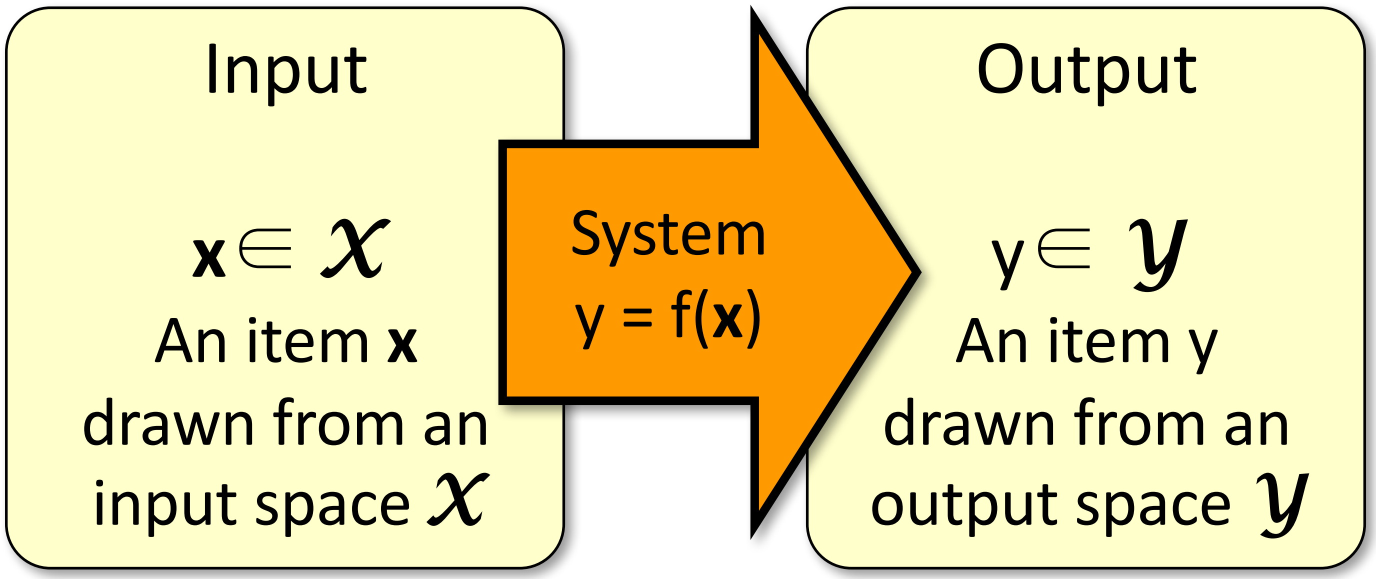
- Shivani Agarwal
- Chris Callison-Burch
- Eric Eaton
+ Peter Stone
+ Matthew Taylor

What is Learning?

- The Badges Game is an example of a key learning protocol: supervised learning
- First question: Are you sure you got it? Why?
- Issues:
  - Which problem was easier: prediction or modeling?
  - Representation
  - Problem setting
  - Background Knowledge
  - When did learning take place?

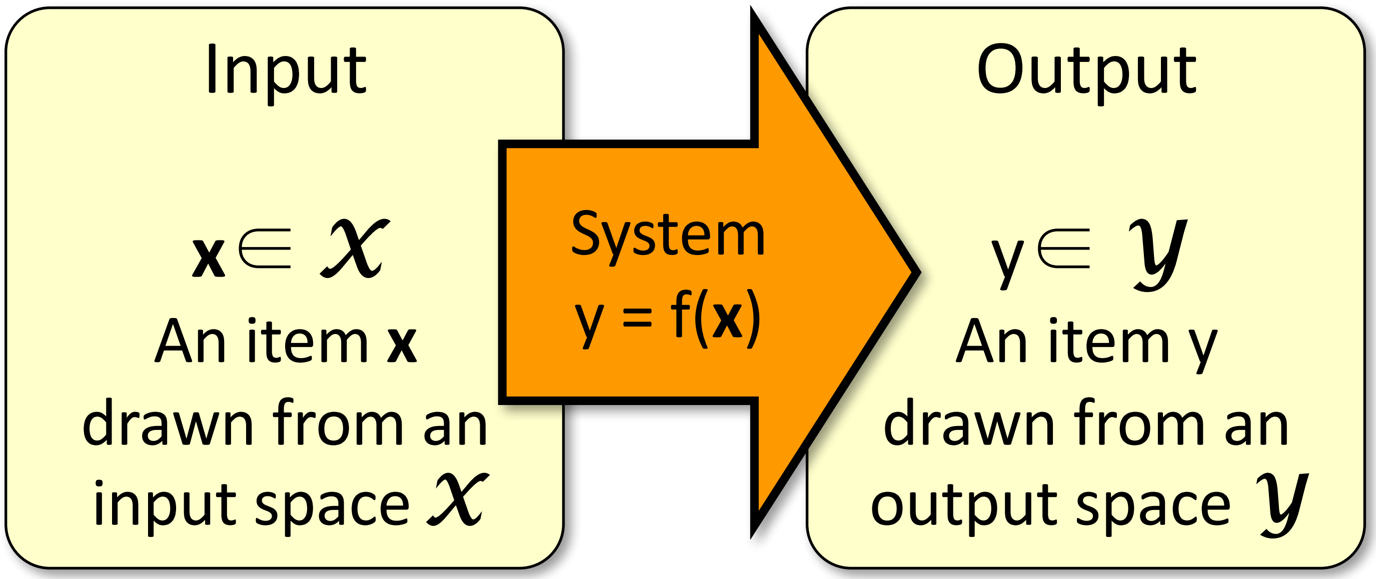
Algorithm: can you write a program that takes this data as input and predicts the label for your name?

Supervised Learning



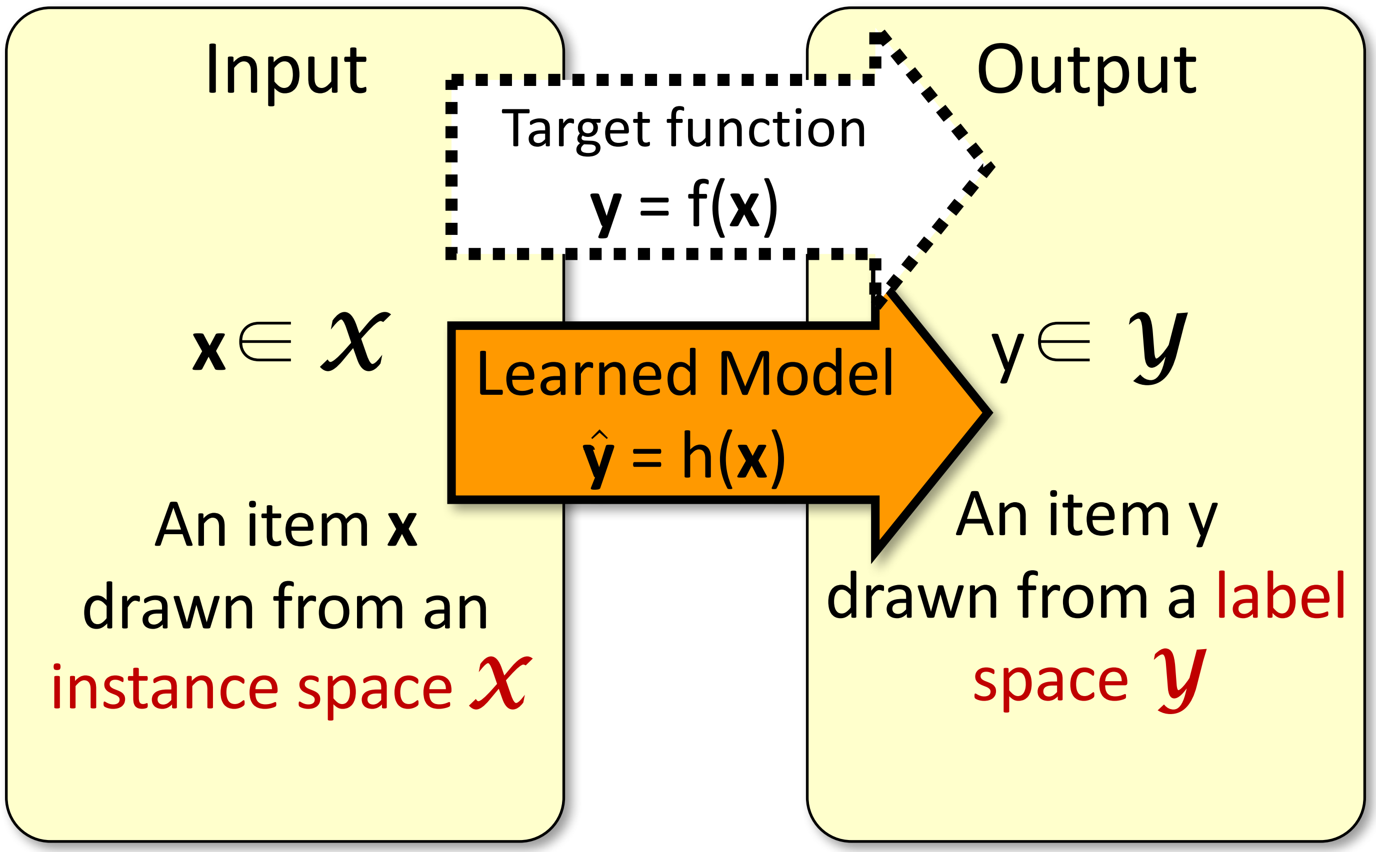
- We consider systems that apply an unknown function f() to input items **x** and return an output **y** = f(**x**).

Supervised Learning

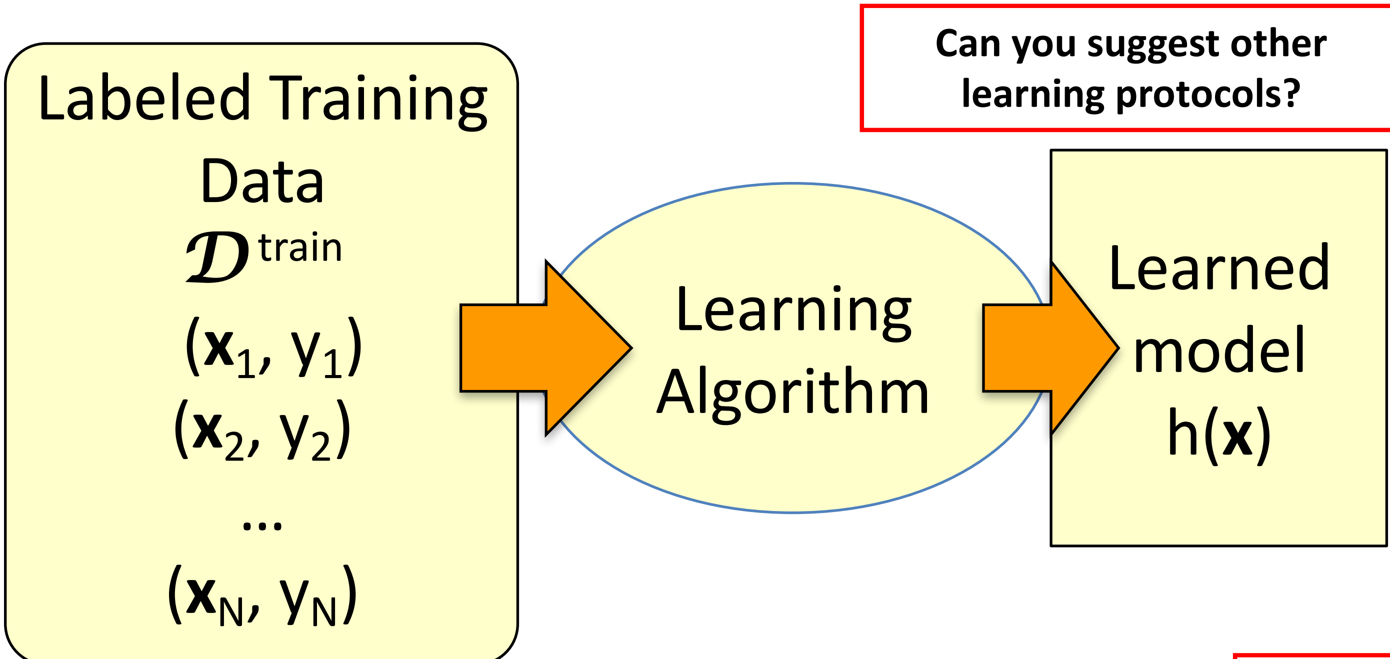


- In (supervised) machine learning, our goal is to learn a function h() from examples that approximates f()

Supervised learning



Supervised learning: Training



- Give the learner examples in $\mathcal{D}^{\text{train}}$
- The learner returns a model $h(\mathbf{x})$

h(x) is the model we'll use in our application

Function Approximation

Problem Setting

- Set of possible instances $\mathcal{X}$
- Set of possible labels $\mathcal{Y}$
- Unknown target function $f : \mathcal{X} \to \mathcal{Y}$
- Set of function hypotheses $H = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$

Input: Training examples of unknown target function f

$$\{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n} = \{\langle \boldsymbol{x}_1, y_1 \rangle, \ldots, \langle \boldsymbol{x}_n, y_n \rangle\}$$

Output: Hypothesis $h \in H$ that best approximates f

Based on slide by Tom Mitchell

Sample Dataset

- Columns denote features X_i
- Rows denote labeled instances $\langle \boldsymbol{x}_i, y_i \rangle$
- Class label denotes whether a tennis game was played

$\langle \boldsymbol{x}_i, y_i \rangle$

Predictors				Response
Outlook	**Temperature**	**Humidity**	**Wind**	**Class**
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Supervised learning: Testing

Labeled
Test Data
$\mathcal{D}^{\text{test}}$
$(\mathbf{x}'_1, y'_1)$
$(\mathbf{x}'_2, y'_2)$
…
$(\mathbf{x}'_M, y'_M)$

- Reserve some labeled data for testing

Supervised learning: Testing

Labeled Test Data

$\mathcal{D}^{\text{test}}$

$(\mathbf{x}'_1, y'_1)$

$(\mathbf{x}'_2, y'_2)$

...

$(\mathbf{x}'_M, y'_M)$

Raw Test Data

$\mathcal{X}^{\text{test}}$

$\mathbf{x}'_1$

$\mathbf{x}'_2$

....

$\mathbf{x}'_M$

Test Labels

$\mathcal{Y}^{\text{test}}$

y'_1

y'_2

...

y'_M

Supervised learning: Testing

- Apply the model to the raw test data
- Evaluate by comparing predicted labels against the test labels

Can you use the test data otherwise?

Raw Test Data
$\mathcal{X}^{\text{test}}$
$\mathbf{x}'_1$
$\mathbf{x}'_2$
....
$\mathbf{x}'_M$

Learned model
$h(\mathbf{x})$

Predicted Labels
$h(\mathcal{X}^{\text{test}})$
$h(\mathbf{x}'_1)$
$h(\mathbf{x}'_2)$
....
$h(\mathbf{x}'_M)$

Test Labels
$\mathcal{Y}^{\text{test}}$
y'_1
y'_2
...
y'_M

Supervised Learning : Examples

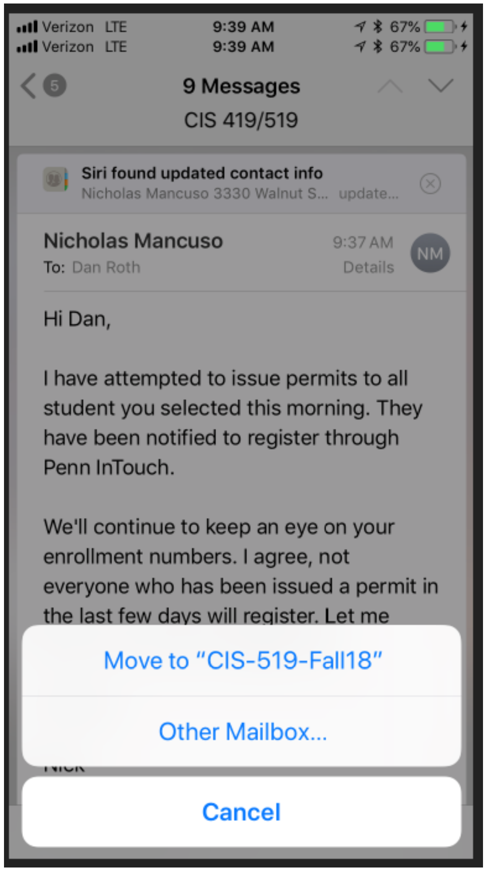
- Disease diagnosis
  - x: Properties of patient (symptoms, lab tests)
  - f : Disease (or maybe: recommended therapy)
- Part-of-Speech tagging
  - x: An English sentence (e.g., The can will rust)
  - f : The part of speech of a word in the sentence
- Face recognition
  - x: Bitmap picture of person's face
  - f : Name the person (or maybe: a property of)
- Automatic Steering
  - x: Bitmap picture of road surface in front of car
  - f : Degrees to turn the steering wheel

Many problems that do not seem like classification problems can be decomposed into classification problems.

Key Issues in Machine Learning

- Modeling
  - How to formulate application problems as machine learning problems?
  - How to represent the data?
  - Learning Protocols (**where is the data & labels coming from**?)
- Representation
  - What **functions** should we learn (hypothesis spaces) ?
  - How to map raw **input** to an instance space?
  - Any rigorous way to find these? Any general approach?
- Algorithms
  - What are good algorithms?
  - How do we define success?
  - Generalization vs. overfitting
  - The computational problem

Using supervised learning



- What is our instance space?
  - What kind of features are we using?
- What is our label space?
  - What kind of learning task are we dealing with?
- What is our hypothesis space?
  - What kind of functions (models) are we learning?
- What learning algorithm do we use?
  - How do we learn the model from the labeled data?
- What is our loss function/evaluation metric?
  - How do we measure success? What drives learning?

1. The instance space $\mathcal{X}$

Input

$\mathbf{x} \in \mathcal{X}$

An item **x** drawn from an instance space $\mathcal{X}$

Learned Model h(**x**)

Output

$y \in \mathcal{Y}$

An item y drawn from a label space $\mathcal{Y}$

- Designing an appropriate instance space $\mathcal{X}$ is crucial for how well we can predict y.

1. The instance space $\mathcal{X}$

- When we apply machine learning to a task, we first need to define the instance space $\mathcal{X}$.
- Instances $x \in \mathcal{X}$ are defined by features:
  - Boolean features:
    - Is there a folder named after the sender?
    - Does this email contains the word 'class'?
    - Does this email contains the word 'waiting'?
    - Does this email contains the word 'class' and the word 'waiting'?
  - Numerical features:
    - How often does 'learning' occur in this email?
    - What long is email?
    - How many emails have I seen from this sender over the last day/week/month?
  - Bag of tokens
    - Just list all the **tokens** in the input

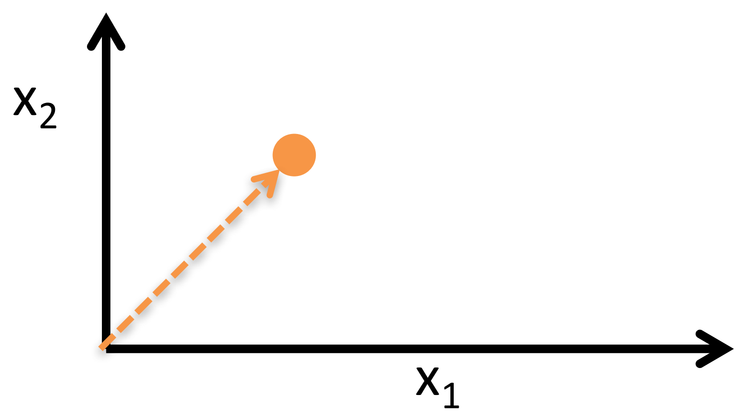
Does it add anything?

What's $\boldsymbol{x}$ for the Badges game?

- Possible features:
  - Gender
  - Name's country-of-origin
  - Length of their first or last name
  - Does the name contain letter 'x'?
  - How many vowels does their name contain?
  - Is the n-th letter a vowel?
  - Does the name have the same number of vowels and consonants?

$\mathcal{X}$ as a vector space

- $\mathcal{X}$ is an N-dimensional vector space (e.g. $\Re^N$)
  - Each dimension = one feature.
- Each **x** is a feature vector (hence the boldface **x**).
- Think of **x** = $[x_1 \ldots x_N]$ as a point in $\mathcal{X}$:



Good features are essential

- The choice of features is crucial for how well a task can be learned
  - In many application areas (language, vision, etc.), a lot of work goes into designing suitable features
  - This requires domain expertise
- Think about the badges game – what if you were focusing on visual features?
- We can't teach you what specific features to use for your task
  - But we will touch on some general principles

2. The label space $\mathcal{Y}$

Input

$\mathbf{x} \in \mathcal{X}$

An item **x** drawn from an instance space $\mathcal{X}$

Learned Model

h(**x**)

Output

$y \in \mathcal{Y}$

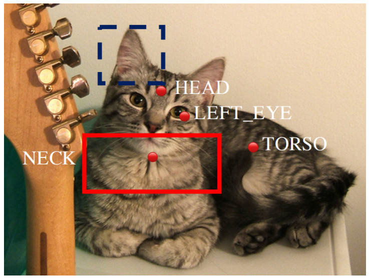
An item y drawn from a label space $\mathcal{Y}$

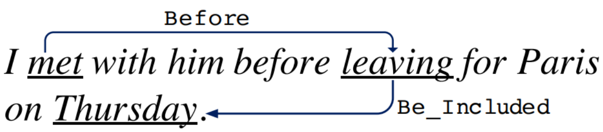
- The label space $\mathcal{Y}$ determines what *kind* of supervised learning task we are dealing with

Supervised learning tasks I

- Output labels y∈Y are categorical:
  - Binary classification: Two possible labels
  - Multi-class classification: k possible labels

- Output labels y∈Y are structured objects (sequences of labels, parse trees, etc.)
- Structure learning

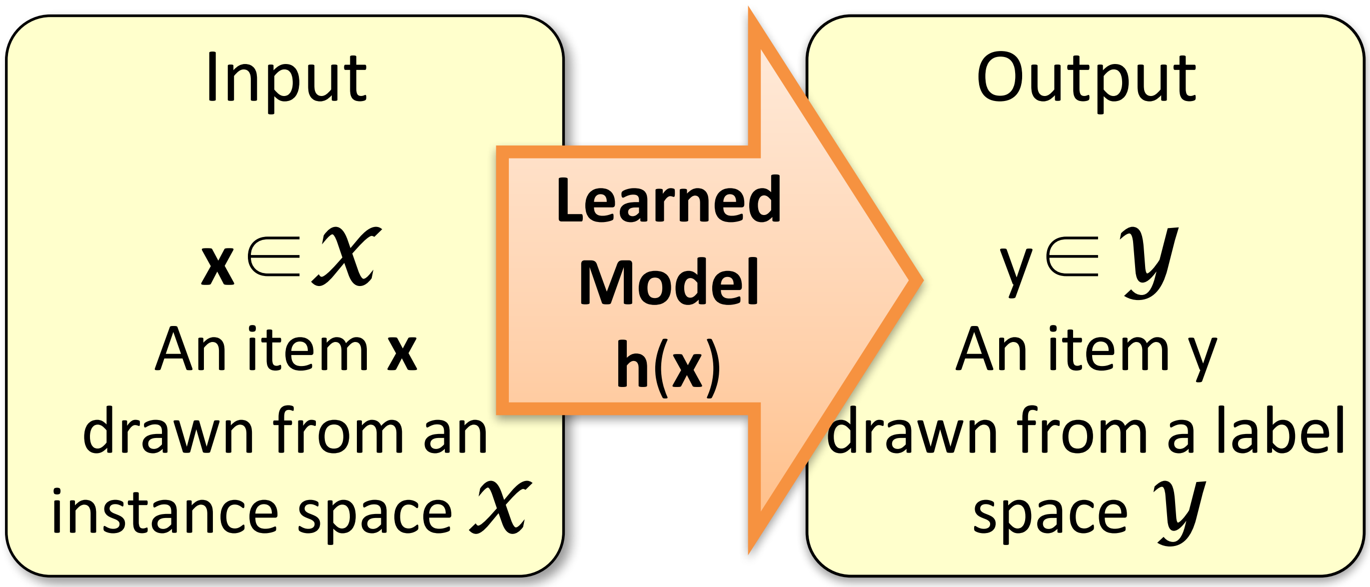




Supervised learning tasks II

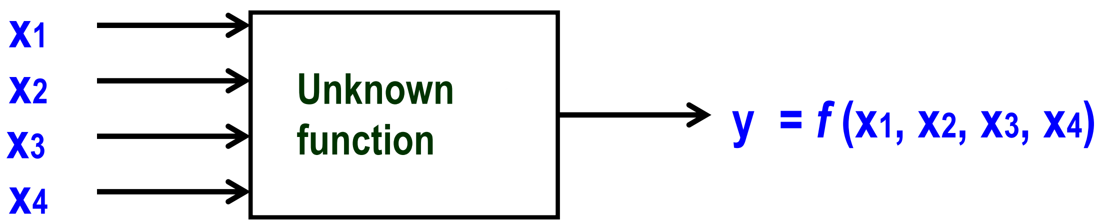
- Output labels $y \in Y$ are numerical:
  - Regression (linear/polynomial):
    - Labels are continuous-valued
    - Learn a linear/polynomial function f(x)
  - Ranking:
    - Labels are ordinal
    - Learn an ordering $f(x_1) > f(x_2)$ over input

3. The model h(**x**)



- We need to choose what *kind* of model we want to learn

A Learning Problem



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Can you learn this function? What is it?

Hypothesis Space

Complete Ignorance:

There are 2^{16} = 65536 possible functions over four input features.

We can't figure out which one is correct until we've seen every possible input-output pair.

After observing seven examples we still have 2^{9} possibilities for f

Is Learning Possible?

Example	x_1	x_2	x_3	x_4	y
1	0	0	0	0	?
2	0	0	0	1	?
	0	0	1	0	0
	1	1	0	1	?
	1	1	1	0	?
16	1	1	1	1	?

- There are $|\mathbf{Y}|^{|\mathbf{X}|}$ possible functions f(**x**) from the instance space **X** to the label space **Y.**
- Learners typically consider only a *subset* of the functions from **X** to **Y**, called the hypothesis space **H** . **H** $\subseteq |\mathbf{Y}|^{|\mathbf{X}|}$

General strategies for Machine Learning

- Develop flexible hypothesis spaces:
  - Decision trees, neural networks, nested collections.
  - **Constraining the hypothesis space is done algorithmically**
- Develop representation languages for restricted classes of functions:
  - Serve to limit the expressivity of the target models
  - E.g., Functional representation (n-of-m); Grammars; linear functions; stochastic models;
  - **Get flexibility by augmenting the feature space**
- In either case:
  - Develop algorithms for finding a hypothesis in our hypothesis space, that fits the data
  - And hope that they will generalize well

Key Issues in Machine Learning

- Modeling
  - How to formulate application problems as machine learning problems?
  - How to represent the data?
  - Learning Protocols (**where is the data & labels coming from**?)
- Representation
  - What **functions** should we learn (hypothesis spaces) ?
  - How to map raw **input** to an instance space?
  - Any rigorous way to find these? Any general approach?
- Algorithms
  - What are good algorithms?
  - How do we define success?
  - Generalization vs. overfitting
  - The computational problem