E0 270 Machine Learning

Course Overview & Lecture 1

Shivani Agarwal / Indrajit Bhattacharya
January Term 2012
Department of Computer Science & Automation
Indian Institute of Science

Course Staff

Instructors:

Dr. Shivani Agarwal

Dr. Indrajit Bhattacharya

TAs:

Adway Mitra

Arun Rajkumar

Raman Sankaran

Goutham Tholpadi

Class Meetings

Lectures:

T-Th 11:30am – 1:00pm

CSA Lecture Hall (Room 117)

Tutorial/Discussion Sessions:

Will be announced on an ongoing basis

Office Hours:

Updated schedule on course website

Course Website

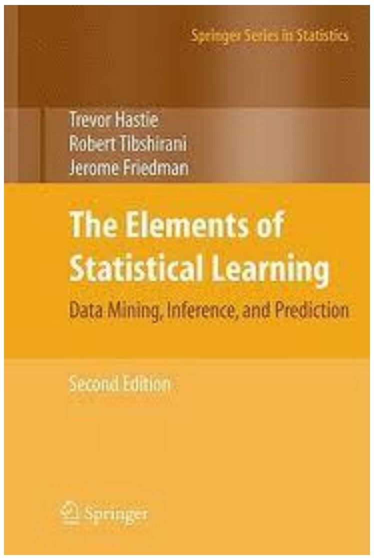
http://drona.csa.iisc.ernet.in/~e0270/Jan-2012/

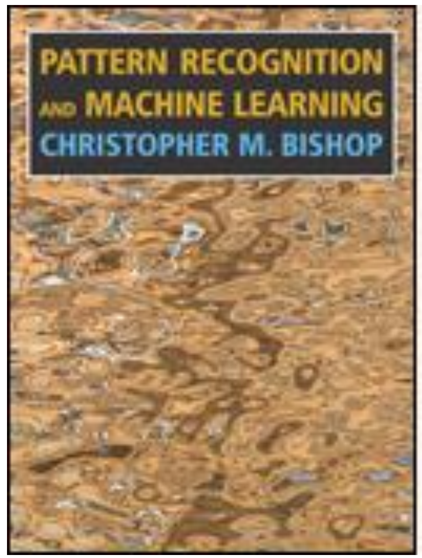
Check regularly for announcements and updates!

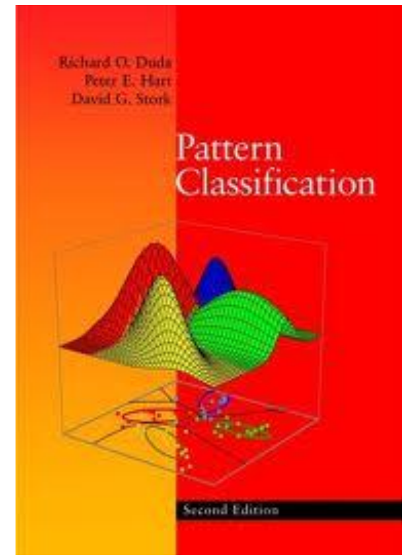
Goals of the Course

- Give you a strong foundation in machine learning
- Enable you to apply machine learning techniques to real problems
- Prepare you for advanced coursework/research in machine learning and related fields

References







See course website for details of these books

Pointers to additional references will be provided

Prerequisites

Required if you want to credit the course:

- **E0 232 Probability and Statistics** or equivalent course elsewhere, with a grade of B or higher

Desirable:

- Some background in linear algebra
- Some background in optimization

Grading Policy

5-6 Assignments	~25%
2 Midterms	~30%
Final Exam	~40%
Participation/Attendance	~ 5%

Assignments

- Assignments are for your learning. You can work in groups, but write your own code and solutions.*
- You are allowed up to 2 late submissions, late by up to 2 days each (with advance email to TA). No further late submissions will be allowed.#
- We will drop your assignment with the lowest grade.
- Assignment 1 out today. Due Jan 24.

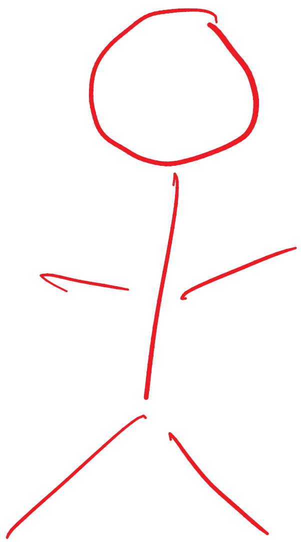
______

* See academic honesty policy on website.

Except in the case of a documented medical/personal emergency, which must be supported by a medical certificate or signed letter submitted to the instructors and to the CSA office.

<table>
<tr><th></th><th>ME I</th><th>ME II</th><th>PhD I-II</th><th>PhD III+</th><th>Other</th></tr>
<tr><td>CSA</td><td>~~||||~~ ~~|||~~ |
~~||||~~ ~~||||~~
~~||||~~ |</td><td>|</td><td>~~||||~~</td><td></td><td>|||</td></tr>
<tr><td>EE</td><td>||||</td><td>|||</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>ECE</td><td></td><td>||||</td><td>~~||||~~</td><td>|</td><td>|</td></tr>
<tr><td>Other</td><td></td><td>~~||||~~ |</td><td>||</td><td>|</td><td></td></tr>
</table>

Why Study Machine Learning?



Wouldn't it be fun to see to what extent human learning can be implemented in machines, and in the process, perhaps gain insights into how humans learn in the first place?

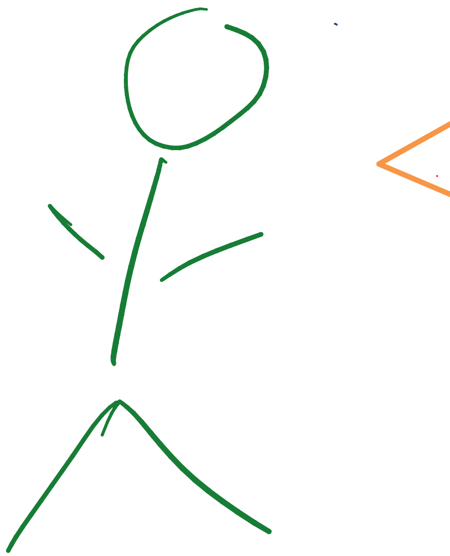
Cognitive Scientist

Why Study Machine Learning?

Mathematician

Lots of cool math.
Lots of rich theory.
Applications too!

Why Study Machine Learning?



I can use my knowledge of computer science to help design algorithms that can learn from data in almost every field. I can finally show my friends in astronomy, biology, finance, … how cool computer science is!

Computer Scientist

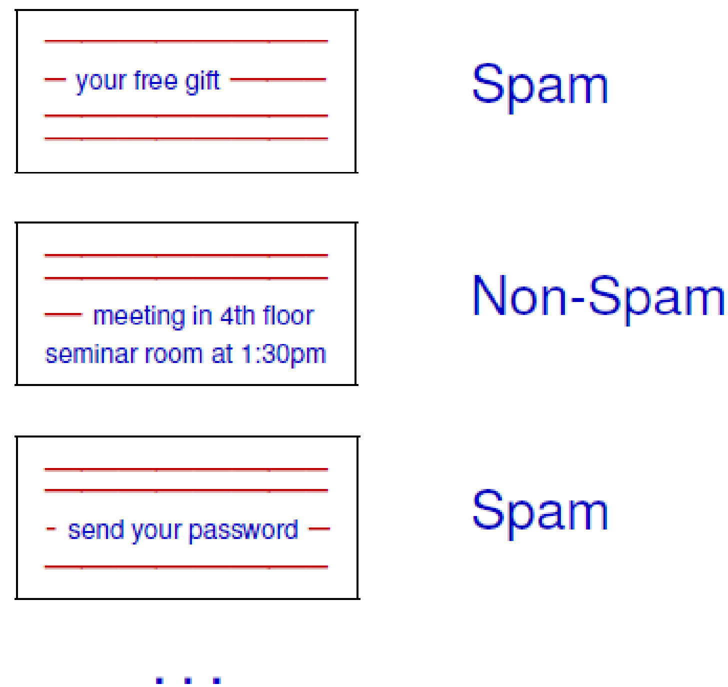
Why Study Machine Learning?

In this course:

- Predictive models needed in many areas of science, engineering, business
- Data everywhere: astronomy, biology, climate modeling, drug discovery, finance, geology, Web,…
- Would like to understand how to design and analyze algorithms that can automatically "learn" predictive models from this data

Example 1: Email Spam Filter

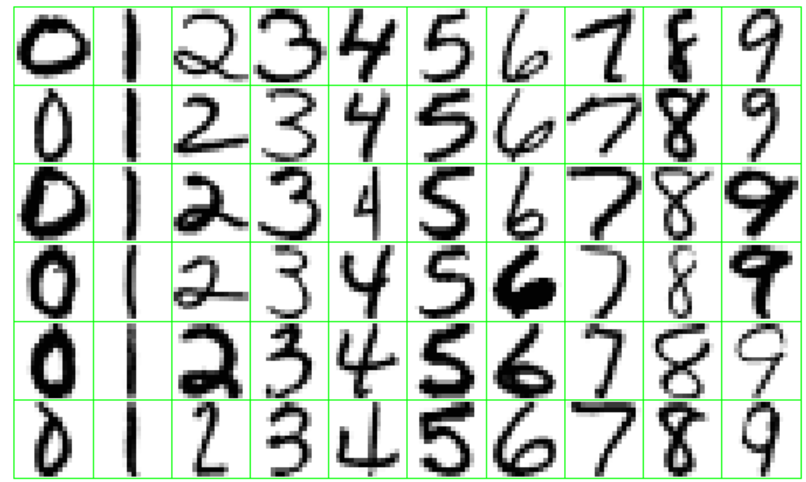
- Would like to build an email filter that can predict whether a new message is spam or non-spam
- Have data containing previous examples of messages labeled as spam or non-spam
- Can we "learn" an email filter from this data?



– Supervised learning
– Binary Classification

Example 2: Handwritten Digit Recognition

- Would like to build a model that can automatically recognize handwritten digits from images
- Have data containing examples of such images labeled with the correct digit (0,1,…,9)
- Can we "learn" an accurate recognition model from this data?

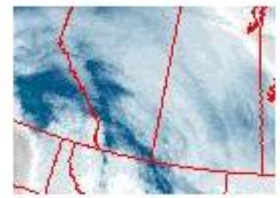


– Supervised Learning

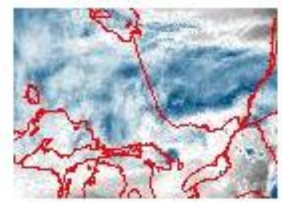
– Multiclass classification

Example 3: Weather Forecasting

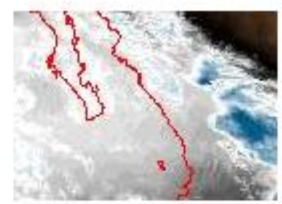
- Would like to build a forecasting model which, given a satellite image showing water vapor in a region, can predict the amount of rainfall in the coming week
- Have data containing examples of such images recorded in the past, together with the amount of rainfall observed in the following weeks
- Can we "learn" a forecasting model from this data?



22.1 mm



134.3 mm



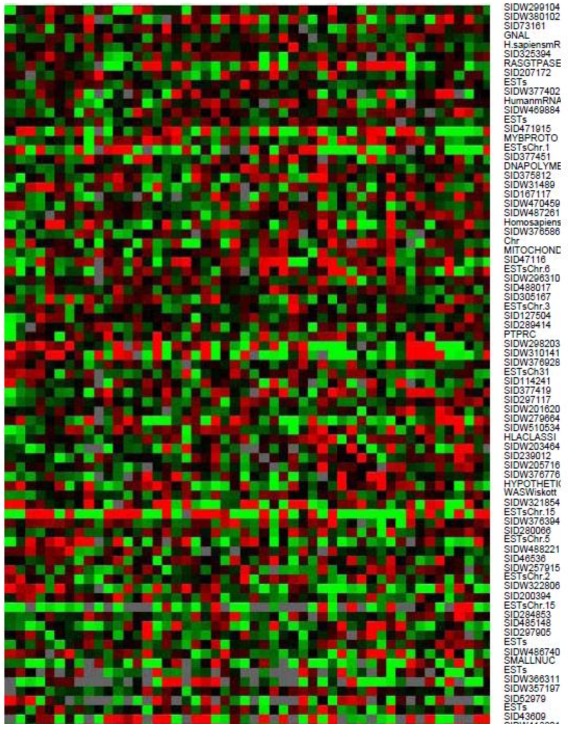
47.9 mm

. . .

- Supervised Learning
- Regression

Example 4: Gene Expression Analysis

- Would like to identify genetic patterns in patients, such as groups of genes that have similar behavior, or groups of patients that have a similar form of genetic disease
- Have microarray expression data containing expression levels of thousands of genes in various patients
- Can we "learn" genetic patterns from this data?



- Unsupervised Learning
- Clustering

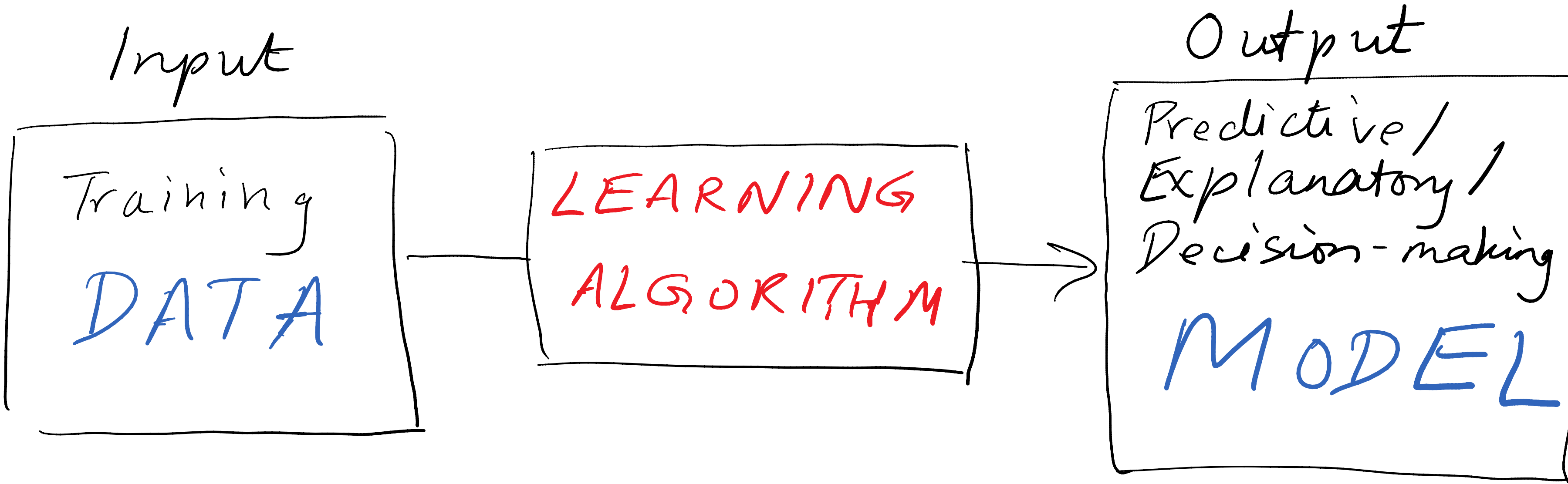
Many More Examples…

- Speech recognition (recognizing speech)
- Computer vision (recognizing objects)
- Natural language processing (recognizing/modeling sentence structure)
- Stock markets (predicting stock prices)
- Recommender systems (predicting customer behavior)
- Drug discovery (predicting activities of compounds)
- Cache systems (predicting file usage)
- Defense/security (predicting/identifying intrusions)
- Equipment design/maintenance (predicting equipment failures)
- …

Course Plan

Part I	
Supervised learning problems and algorithms	classification, regression, multiclass classification, etc; nearest neighbor, decision trees, SVMs, least squares, AdaBoost
Learning theory	generalization error, VC-dimension, PAC learning
Part II	
Probabilistic models for supervised and unsupervised learning	naïve Bayes, logistic regression, k-means, maximum likelihood, EM algorithm, Bayesian inference
Probabilistic graphical models	HMMs, Bayesian networks, MRFs, etc
Part III	
Additional learning settings	online learning, semi-supervised learning, etc

TYPICAL LEARNING PROBLEM



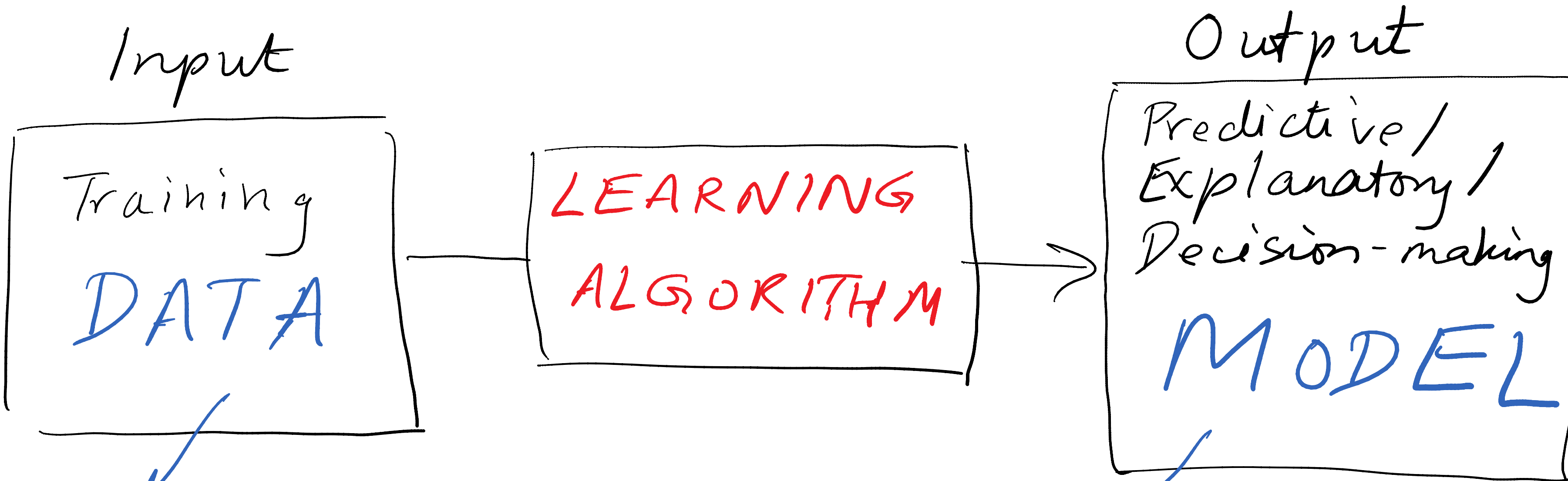
To specify a learning problem:

- Specify form of desired model
- Specify form of training data

To give a solution to a learning problem:

- Design a learning algorithm that given specified form of data, learns specified form of model.

BINARY CLASSIFICATION



Data: Examples of objects seen in the past labeled with the correct label (emails labeled as spam/non-spam, images labeled as face/non-face, etc.)

Model: Classifier that can classify objects into one of two categories or "classes" (e.g. classify email messages as spam/non-spam; images as face/non-face; tissue samples as cancer/non-cancer, etc.)

BINARY CLASSIFICATION

Instance space $\mathcal{X}$ (set of all possible 'objects' or 'instances' to be classified — e.g. if emails represented as 1000-dimensional vectors denoting presence/absence of 1000 words, then $\mathcal{X} = \{0,1\}^{1000}$)

Label/outcome space $\mathcal{Y} = \{\pm 1\}$ (e.g. $+1$: spam, -1 : non-spam)

Input: Training DATA → LEARNING ALGORITHM → Output: Predictive/ Explanatory/ Decision-making MODEL

Training data (or training sample):

$$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$$

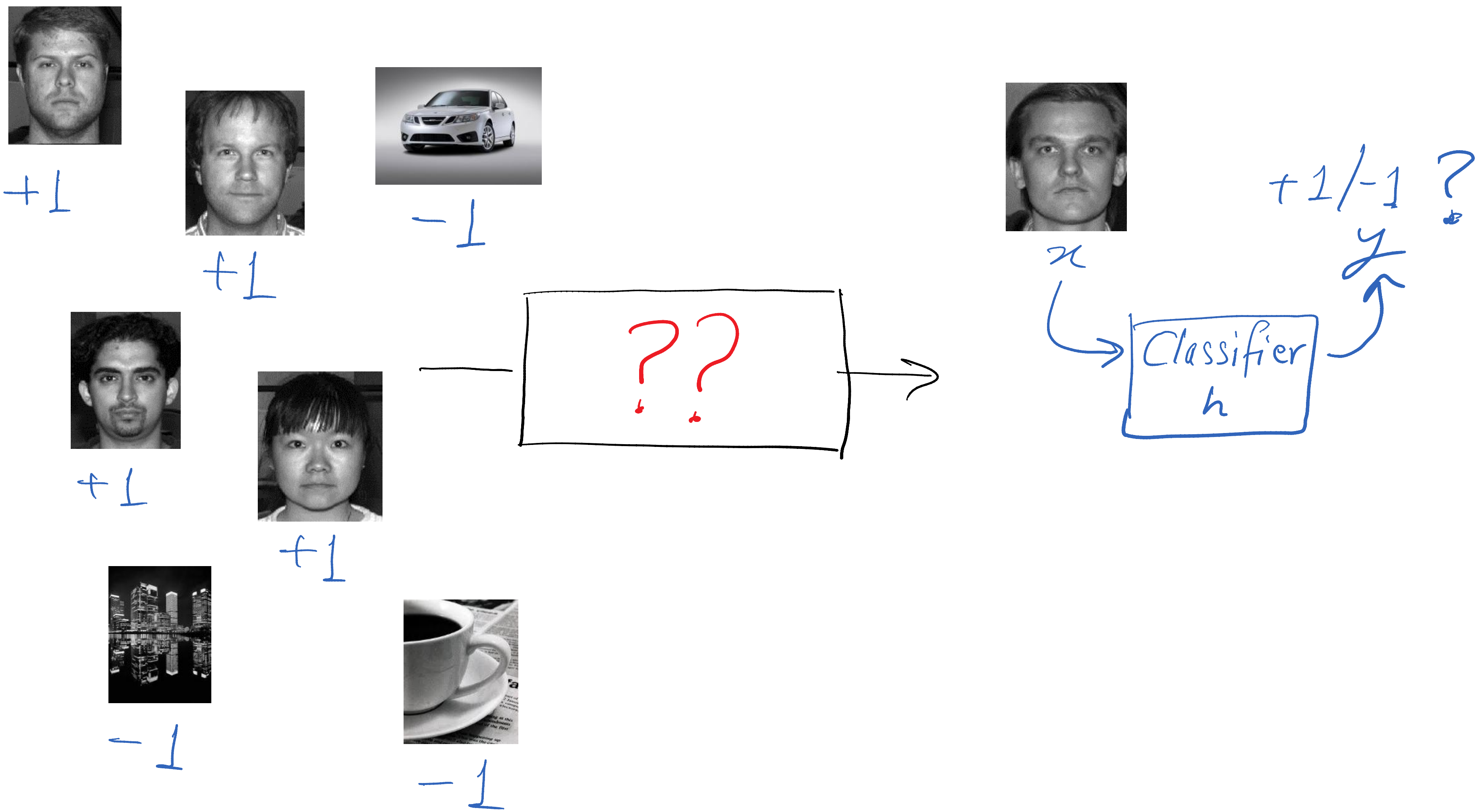
(e.g. x_i = example email, represented as a vector
y_i = label, $+1$ if x_i is spam,
-1 if x_i is non-spam)

Model for predicting label of new instances:

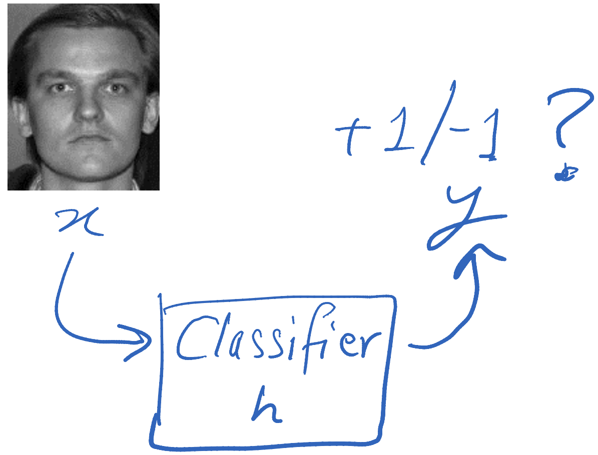
$$x \in \mathcal{X} \longrightarrow \boxed{h} \longrightarrow y \in \mathcal{Y}$$

h can be a function $h : \mathcal{X} \to \mathcal{Y}$ or an algorithm

A SIMPLE CLASSIFICATION ALGORITHM...

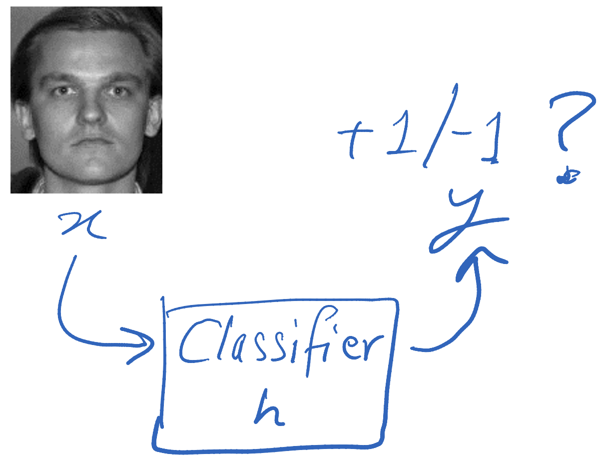


... NEAREST NEIGHBOR



h: given new instance $x \in \mathcal{X}$,
find its nearest neighbor
in the training sample, x_{i^*};
take y to be the label of
the nearest neighbor, $y = y_{i^*}$

... NEAREST NEIGHBOR



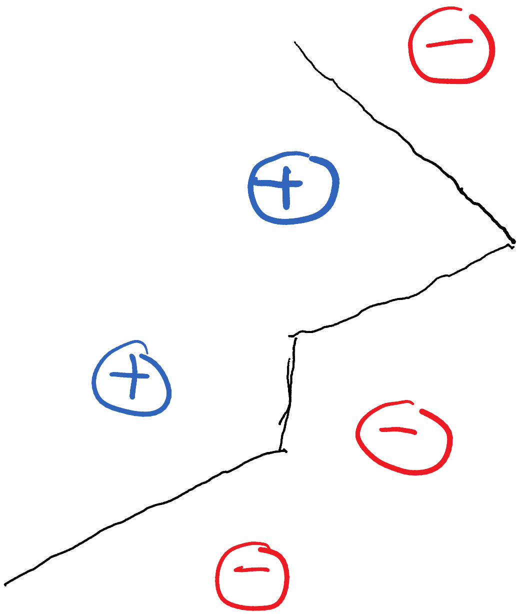
h: given new instance $x \in \mathcal{X}$, find its nearest neighbor in the training sample, x_{i^*}; take y to be the label of the nearest neighbor, $y = y_{i^*}$

requires distance $d(x, x')$ over $\mathcal{X}$;

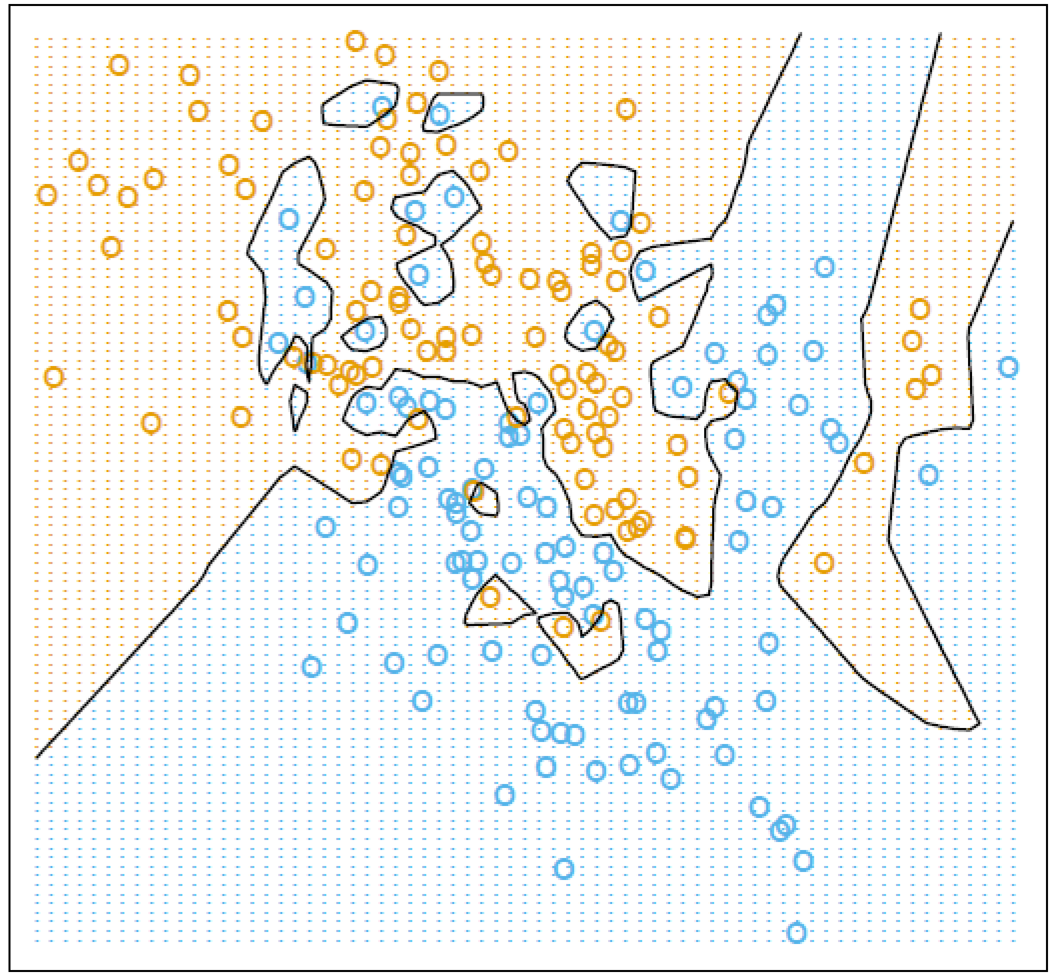
for $\mathcal{X} \subseteq \mathbb{R}^n$, Euclidean distance commonly used:

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2}$$

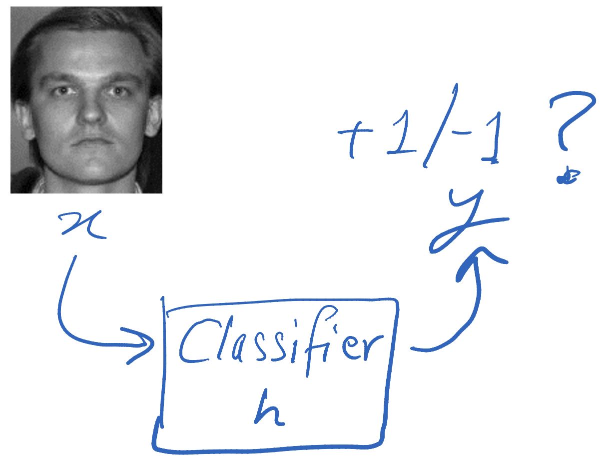
2-DIMENSIONAL EXAMPLE: $\mathcal{X} \subseteq \mathbb{R}^2$



Voronoi diagram



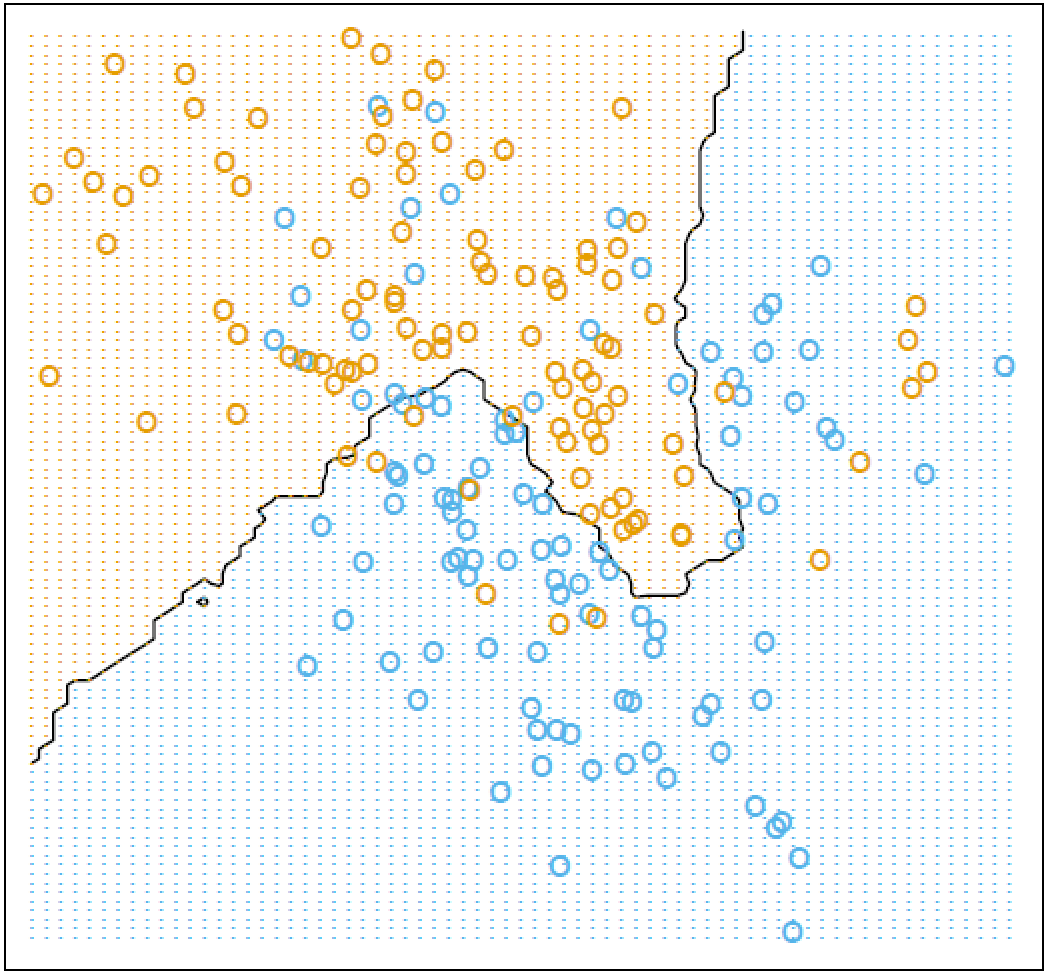
k-NEAREST NEIGHBOR (k-NN)



h: given new instance $x \in \mathcal{X}$, find its k nearest neighbors in the training sample, $x_{i_1^*}, x_{i_2^*}, \ldots, x_{i_k^*}$; take y to be the majority vote of the labels of these k nearest neighbors:

$$y = \begin{cases} +1, & \text{if } \sum_{j=1}^{k} y_{i_j^*} > 0 \\ -1, & \text{o/w.} \end{cases}$$

15-NN IN 2 DIMENSIONS



MEASURING PERFORMANCE/QUALITY OF LEARNED MODEL

- Interested in accuracy of predictions made by model on <u>new</u> instances/data
- Formal notions of accuracy generally involve probabilistic assumptions (will study later). In practice, typically have separate <u>test</u> data to evaluate quality
- For binary classification, can simply measure classification error on test data (fraction of test examples misclassified)

SUMMARY OF k-NN

BENEFITS

- Conceptually simple; easy to implement
- With some simple variations (k depends on m, $k_m \to \infty$, $\frac{k_m}{m} \to 0$ as $m \to \infty$), enjoys good theoretical properties

LIMITATIONS

- 'Memory-based' classifier; need to store all training data in memory
- Computational complexity of classification: for computing k nearest neighbors for each instance, $O(km)$; can be prohibitive for large m.

MODELING ISSUES

- Need to choose an appropriate distance measure

HOMEWORK (BEFORE NEXT LECTURE)

Assignment 1, problems 1-3.