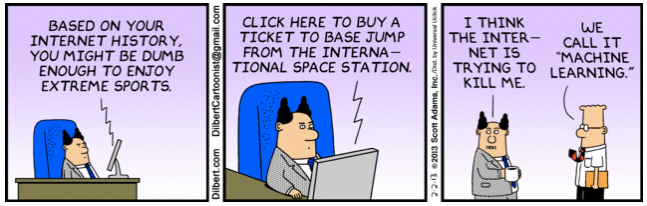
CS545 Machine Learning

Course Introduction



1

Machine learning and related fields

Machine learning: the construction and study of systems that learn from data.

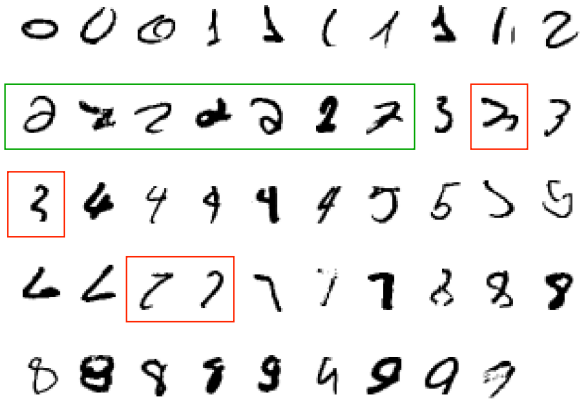
Pattern recognition: the same field, different practitioners

Data mining: using algorithms (often ML) to discover patterns in a data

Statistics and probability: a lot of algorithms have a probabilistic flavor

2

Example problem: handwritten digit recognition



Tasks best solved by a learning algorithm

Recognizing patterns and anomalies:

- Face recognition
- Handwritten or spoken words
- Medical images
- Unusual credit card transactions
- Unusual patterns of sensor readings (in nuclear power plants or car engines)
- Stock prices

web-based examples of machine learning

Spam filtering, fraud detection:

- The enemy adapts so we must adapt too.

Recommendation systems (amazon, netflix):

- Lots of noisy data. Million dollar prize!

Information retrieval:

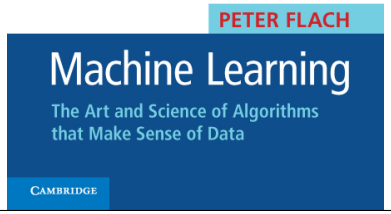
- Find documents or images with similar content.

Course Objectives

The machine learning toolbox

- Formulating a problem as an ML probler..
- Understanding a variety of ML algorithms
- Running and interpreting ML experiments
- Understanding what makes ML work – theory and practice

The textbook:



Grading

Assignments are 100% of the grade

Around 5 assignmnets, worth 80%

- Combination of implementation, running ML experiments, and theory questions

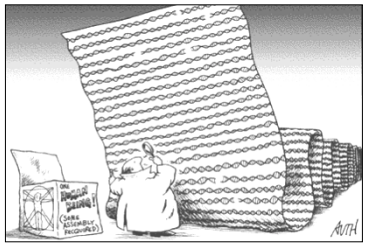
A "project" assignment worth 20%

- You choose what you want to work on!

Course staff

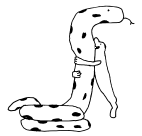
Asa Ben-Hur
(instructor)

Navini Dantanarayana
(TA)



9

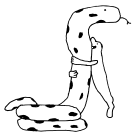
Implementation: Python



Why Python for ML?

- A concise and intuitive language
- An interpreted language – allows for interactive data analysis
- Simple, easy to learn syntax
- Highly readable, compact code
- Supports object oriented and functional programming
- Libraries for plotting and vector/matrix computation
- Strong support for integration with other languages (C,C++,Java)

Implementation: Python



Why Python for ML?

- Dynamic typing and garbage collection
- Cross-platform compatibility
- Free
- Language of choice for many ML researchers

Why I love Python

I am more productive!

- Machine performance vs. programmer performance

Makes programming fun!

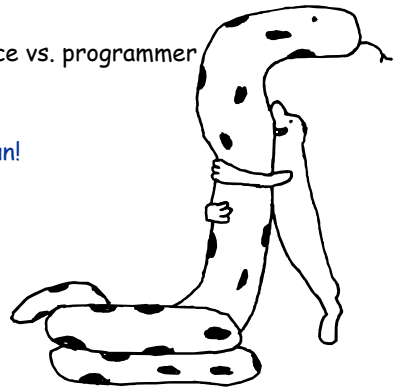


image from: ftp://www.mindview.net/pub/eckel/LovePython.zip

Which version?

2.x or 3.x? Stick with 2.x for now.

Python 3 is a non-backward compatible version that removes a few "warts" from the language.

13

Does anyone else use python?

One of the three "official languages" in google.

Peter Norvig, Director of Research at Google:

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolved. Today dozens of Google engineers use Python, and we're looking for more people with skils in this language"

ML in Python

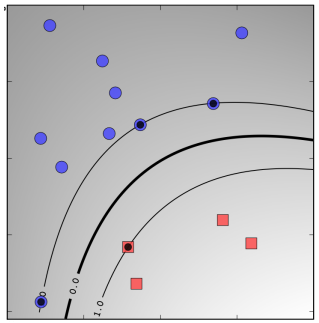
We will use PyML, which was written by the instructor

Available on sourceforge:
http://pyml.sf.net

Also:

NumPy: operations on arrays and matrices

Matplotlib: plotting library



15

How will we learn Python?

- Overview of Python/PyML in lecture.
- Course website has links to Python tutorials and other resources

16

Labeled data

E-mail	x_1	x_2	Spam?
1	1	1	1
2	1	0	-1
3	0	1	-1
4	0	0	1
5	0	0	-1

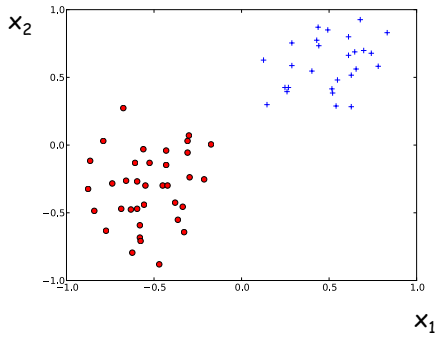
x_1 and x_2 are two characteristics of emails (e.g. the presence of the word "viagara"). These are called features

Spam? Is the label associated with the each email

This is a binary classification problem

17

Binary classification

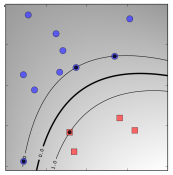


Scatter plot of labeled data with two features (dimensions)

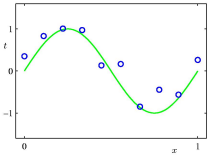
18

ML tasks

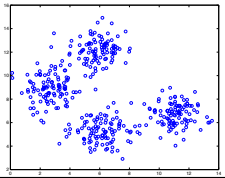
Classification: discrete/categorical labels



Regression: continuous labels

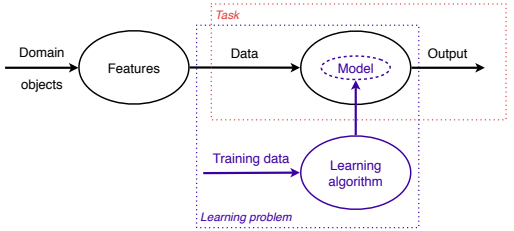


Clustering: no labels



19

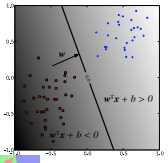
Using ML to address a learning task



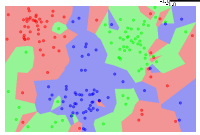
20

Types of models

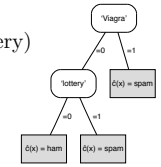
Geometric
- Ridge-regression, SVM, perceptron



Distance-based
- K-nearest-neighbors



Probabilistic
- Naïve-bayes $P(Y = \text{spam}|\text{Viagara}, \text{lottery})$



Logical models: Tree/Rule based
- Decision trees

21

Types of learning tasks

Supervised learning
- Learn to predict output given labeled examples

Unsupervised learning
- Data is unlabeled
- Create an internal representation of the input e.g. form clusters; extract features
- Most "big" datasets do not come with labels

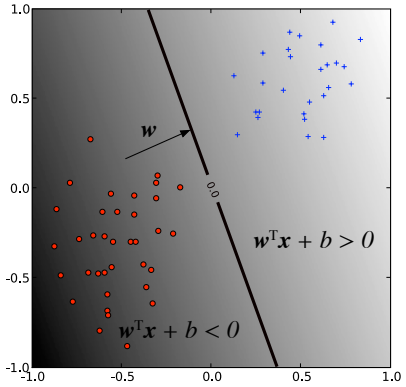
Reinforcement learning
- Learn action to maximize payoff
  - Not much information in a payoff signal
  - Payoff is often delayed
- not be covered in this course.

Human vs machine learning

Human	Machine
Observe someone, then repeat	Supervised Learning
Keep trying until it works (riding a bike)	Reinforcement Learning
Memorize	k-Nearest Neighbors
20 Questions	Decision Tree
A network of neurons with complex interconnections	Neural networks

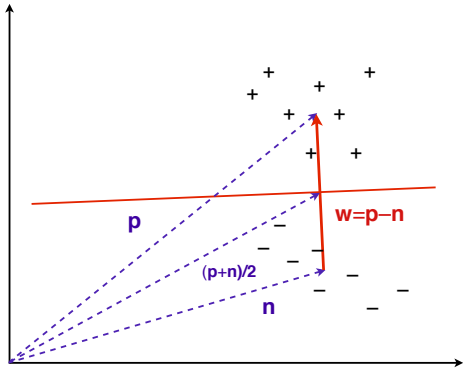
23

Linear classifiers



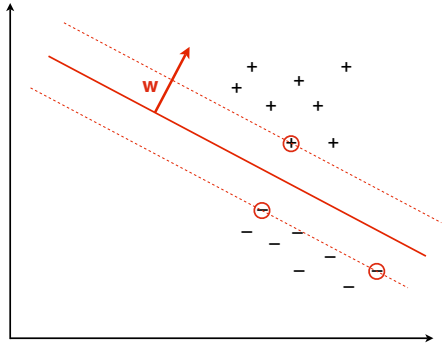
24

Closest centroid classifier



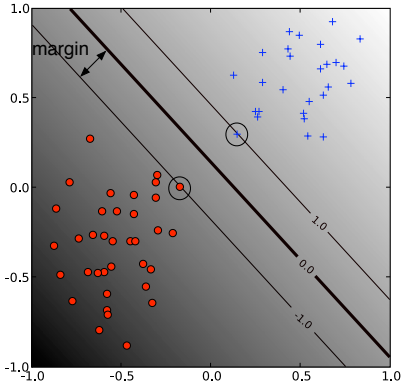
25

Large margin classification



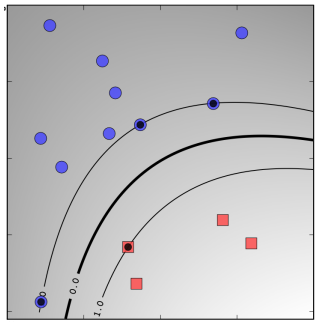
26

Large margin classification



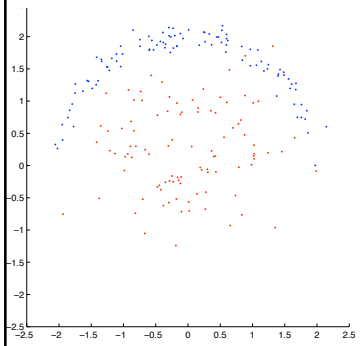
27

Non-linear large margin classifiers

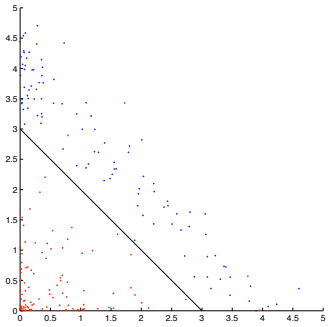


28

From linear to non-linear



Original data: not linearly separable



Transformed data: $(x', y') = (x^2, y^2)$

29

ML in Practice

Understanding the domain, and goals
Creating features, data cleaning and preprocessing
Learning models
Interpreting results
Consolidating and deploying discovered knowledge
Loop

Generalization

The real aim of supervised learning is to do well on test data that is not known during training.

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?

Trading off goodness of fit against model complexity

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?

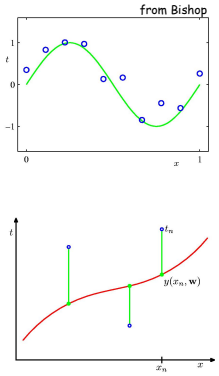
There is a lot of theory about how to measure model complexity and how to control it to optimize generalization.

A simple example: Fitting a polynomial

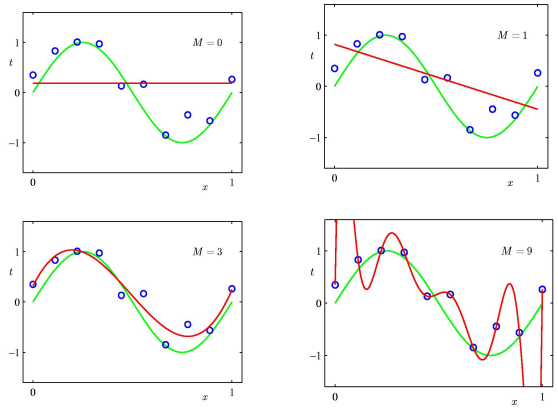
The green curve is the true function (which is not a polynomial)

The data points have noise in y.

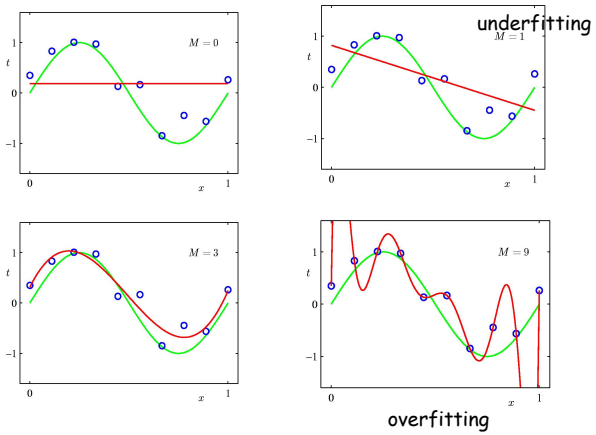
Measure of error (loss function) that measures the squared error in the prediction of y(x) from x. The loss for the red polynomial is the sum of the squared vertical errors.



Which model is best?



Which model is best?



Figures from: "Pattern Recognition and Machine Learning" by Christopher Bishop

What we'll cover

Supervised learning
- Linear classifiers
- Decision trees
- Probabilistic classifiers
- Neural networks
- Support vector machines
- Model ensembles

Unsupervised learning
- Clustering
- Dimensionality reduction

Running and interpreting ML experiments

36