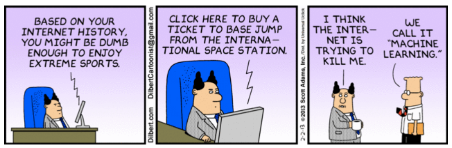
CS545 Machine Learning

Course introduction



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

Example problem: handwritten digit recognition

Tasks best solved by a learning algorithm

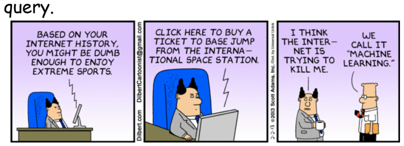
Recognizing patterns and anomalies:
- Face recognition
- Handwritten or spoken words
- Medical images
- Unusual credit card transactions
- Unusual patterns of sensor readings (in nuclear power plants or car engines)
- Stock prices

Examples of machine learning on the web

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 that are relevant to a query.



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 Book

Learning from data

http://amlbook.com

The website contains several e-chapters

Grading

Assignments + project are 100% of the grade

5 assignments, worth 80%

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

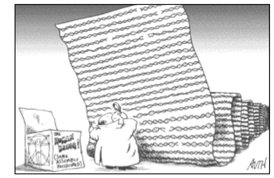
A "project" assignment worth 20%

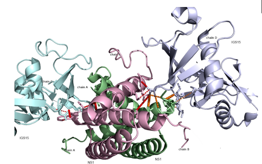
- You choose what you want to work on!

Course staff

Asa Ben-Hur
(instructor)

Basir Shariat
(TA)





About this course

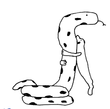
Course webpage:
http://www.cs.colostate.edu/~cs545

Slides/assignments are posted on the course webpage's schedule page.

~~RamCT~~ Canvas will be used for: ~~forums~~, grades

Piazza will be our primary communication tool

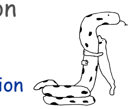
Implementation: Python



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 (other options: matlab, R)

Why Python

I am more productive!

- Machine performance vs. programmer performance

Makes programming fun!

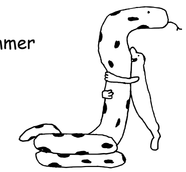


image from: ftp://www.mindview.net/pub/eckel/LovePython.zip

Which version?

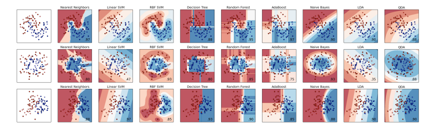
2.x or 3.x?

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

Doesn't matter at this point.

ML in Python

We will use scikit-learn
Download from : http://scikit-learn.org/



NumPy: operations on arrays and matrices

Matplotlib: plotting library

The Anaconda python distribution

Contains (almost) all the packages that will be used in this course.

ANACONDA

How will we learn Python?

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

Labeled data

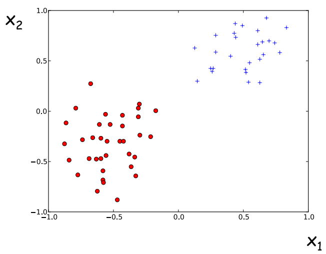
E-mail	$feature_1$	$feature_2$	Spam?
1	1	1	1
2	1	0	-1
3	0	1	-1
4	0	0	1
5	0	0	-1

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

Binary classification



Scatter plot of labeled data with two features (dimensions)

Another example: Credit approval

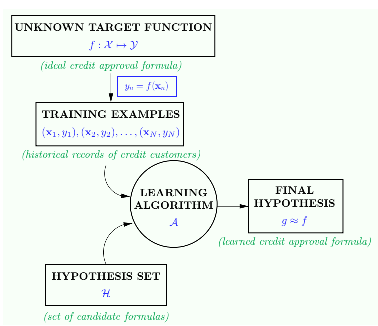
Should an applicant be approved for credit?

Feature	Value
Gender	Male
Salary	70,000
Debt	21,000
Years in job	1 year
…	

Key components of a learning problem

- Salary, debt, years in residence, … — *input* $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$.
- Approve credit or not — *output* $y \in \{-1, +1\} = \mathcal{Y}$.
- True relationship between $\mathbf{x}$ and y — *target function* $f : \mathcal{X} \mapsto \mathcal{Y}$.
  (The target f is *unknown*.)
- Data on customers — *data set* $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$.
  ($y_n = f(\mathbf{x}_n)$.)

The ML setup



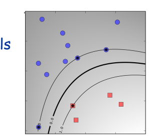
Not every problem is a ML problem

Which of the following problems are best suited for machine learning?

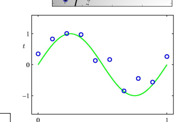
- Classifying numbers into primes and non-primes.
- Determining the time it would take a falling object to hit the ground.
- Determining the optimal cycle for traffic lights in a busy intersection.
- Medical diagnosis

ML tasks

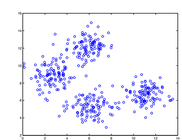
Classification: discrete/categorical labels



Regression: continuous labels



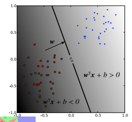
Clustering: no labels



Types of models

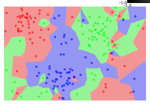
Geometric

- Ridge-regression, SVM, perceptron



Distance-based

- K-nearest-neighbors

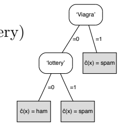


Probabilistic

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

Logical models: Tree/Rule based

- Decision trees



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

- Not covered in this course.

ML in Practice

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

An iterative process

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

Training vs testing

Out-of-sample error: (testing error)

$$\underbrace{}_{\text{hypothesis}} \quad \underbrace{}_{\text{target function}}$$

hypothesis target function

$$E_{\text{out}}(h) = \mathbb{P}_{\mathbf{x}}\left[h(\mathbf{x}) \neq f(\mathbf{x})\right]$$

In-sample error: (training error)

$$E_{\text{in}}(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[h(\mathbf{x}_i) \neq f(\mathbf{x}_i)\right]$$

Training: finding a rule that minimizes E_{in}

Training vs testing

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.

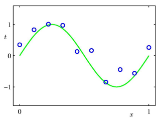
- But the learning machine does not know which regularities are real and which are accidental quirks of the particular set of training examples we happen to pick.

So how can we be sure that the machine will generalize well to new data?

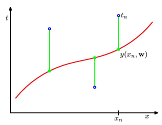
A simple example: Fitting a polynomial

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

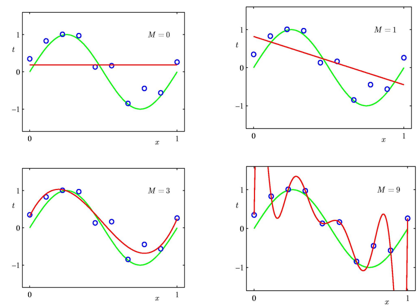
The data points have noise in y.



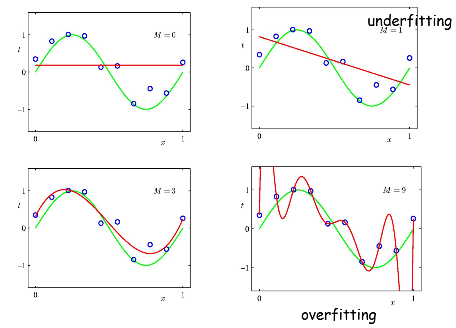
Measure of error (loss function) that measures the squared error in the prediction of y(x) from x. The loss for the red polynomial is the sum of the squared vertical errors.



Which model is best?



Which model is best?

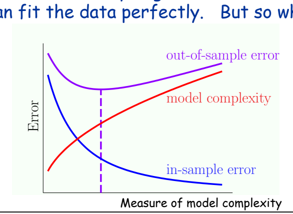


Figures from: "Pattern Recognition and Machine Learning" by Christopher Bishop

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?



What we'll cover

Supervised learning
- Linear classifiers
- Probabilistic classifiers
- Support vector machines
- Neural networks
- Nearest neighbor classifiers
- Briefly mentioned: decision trees and ensemble models

Unsupervised learning
- Clustering
- Dimensionality reduction

Running and interpreting ML experiments