Machine Learning

CS 165B

Prof. Matthew Turk

Wednesday, March 30, 2016

Today

- Key machine learning concepts (cont.)
- The ingredients of machine learning (Ch. 1)

Notes

- From last lecture:
  - Course web site: http://www.cs.ucsb.edu/~mturk/ML
    - Login: machine
    - Password: learning

- HW#1 (due a week from Friday) will be posted this weekend

- Course registration
  - ~10 new registrations will be added (7 new and 3 dropped)
  - Waitlist is at 33
  - Everyone should now be able to join GauchoSpace site for the course

Notes

CS DISTINGUISHED LECTURE

Michael Jordan, Berkeley

Friday, April 8

11:00am

Corwin Pavilion

“On Computational Thinking, Inferential Thinking, and Data Science”

UNIVERSITY *of* CALIFORNIA, SANTA BARBARA
Education Abroad Program

STUDY ABROAD – Apply NOW!

Application Deadlines

- **Apps for ALL Winter/Spring 2017 programs due May 11, 2016**
- **Need help? Attend an application workshop**
- **It's not too late to start an application!**
- **www.eap.ucsb.edu | South Hall 2431**

Study Abroad with UCEAP to:

- **Earn UC units**
- **Take major/minor/GE courses = graduate on time**
- **Gain internship, volunteer, or research experience**
- **Use your financial aid**

2016 Scholarships & Discounts:

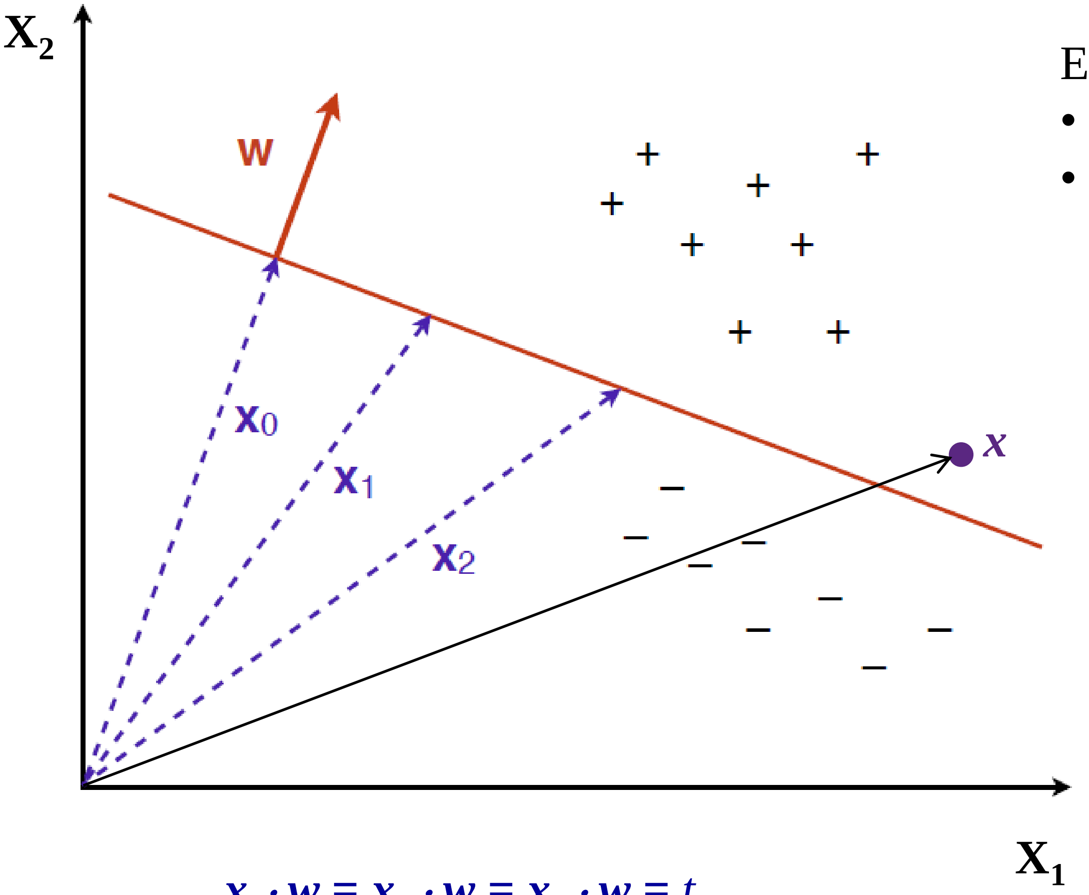
- **Federal Pell Grant Students–**
  Apply for the Gilman Scholarship (up to $8K)
- **UCEAP Promise Awards = giving away $1.5million this year**
- **$3,000 discount for participating in a year-long program**
- **$300 fee reduction on ALL programs**

Graduating seniors & transfer students are eligible!

Generation Study Abroad™
IIE | Opening Minds to the World®
COMMITMENT PARTNER

Linear classification

Goal: Find a linear decision boundary



$$x_0 \cdot w = x_1 \cdot w = x_2 \cdot w = t$$

$$x \cdot w = x^T w$$

How to determine if a feature vector ***x*** is on the + or – side of the line?

Evaluate the dot product of ***x*** and ***w***:

- If $x \cdot w > t$, then +
- Otherwise –

2 features means 2D classification and a 1D classification boundary

N features means N dimensional classification and an N-1 dimensional classification boundary

Dimensions	Linear boundary
1	Point
2	Line
3	Plane
>3	Hyperplane

Homogeneous coordinates

- Instead of writing $\boldsymbol{x} \cdot \boldsymbol{w} > t$, let's use homogeneous coordinates to simplify the decision rule to $\boldsymbol{x}^\circ \cdot \boldsymbol{w}^\circ > 0$

$$\boldsymbol{x}^\circ = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

$$\boldsymbol{w}^\circ = \begin{bmatrix} \boldsymbol{w} \\ -t \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ -t \end{bmatrix}$$

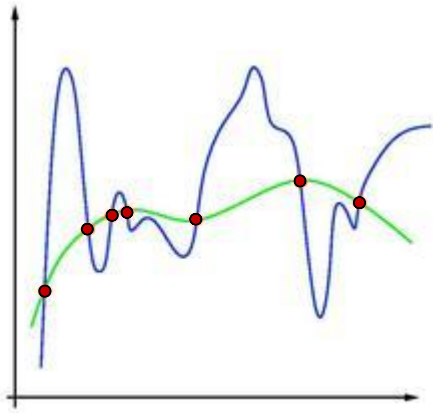
Note: I represent homogeneous coordinates a little differently from the textbook!

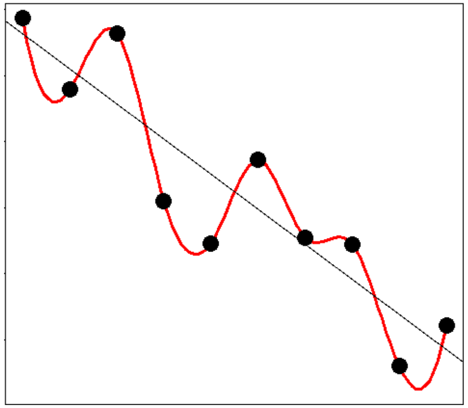
- Homogeneous coordinates embeds an N-dimensional representation in an N+1-dimensional space
- Advantage: The decision boundary passes through the origin of the extended coordinate system
  - Simplifies the math…

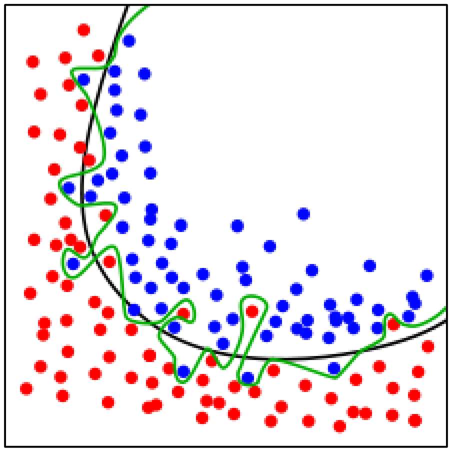
Overfitting

- *Overfitting* and *generalization* are important concepts in machine learning

- Overfitting: Learning that results in good performance on the training data but poor performance on the real task
  - Example: Memorization or lookup table
  - Example: Fitting a model to the data that has more parameters then needed







Generalization

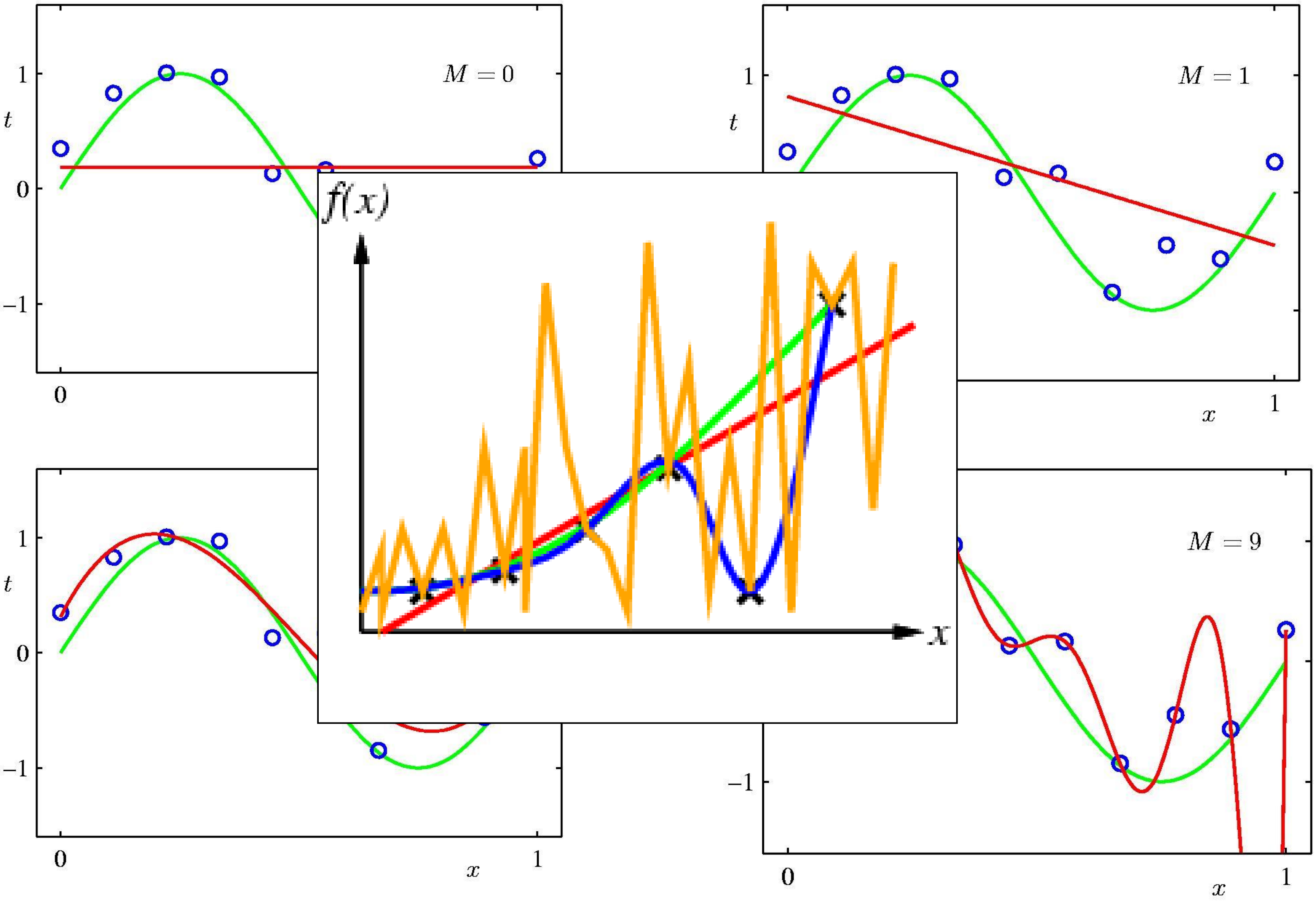
- We want machine learning solutions that ***generalize*** to the range of inputs/data that will be seen – not just solutions that work well on the training data
- The real aim of machine learning is to do well on test data that is not known during learning
- Choosing values for the parameters that minimize the error 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 have! So we have to help….

Which model fits the data best?

Real data = sinusoid
M = degree of polynomial fit

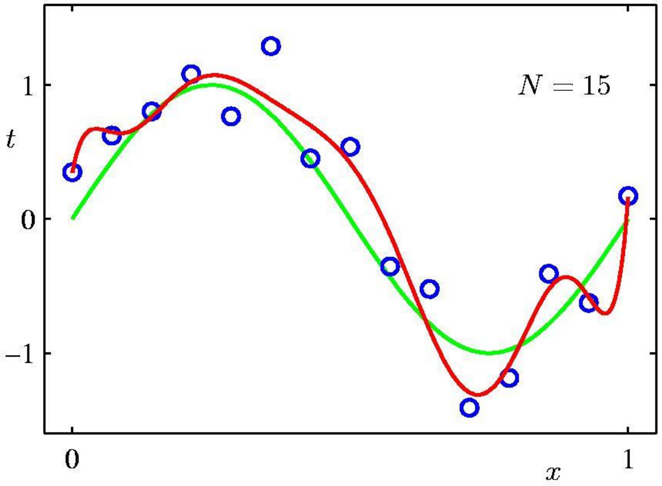
E.g., in a regression task



Reducing model complexity

If we penalize polynomials that have a high number of coefficients, we will get smoother (less wiggly) solutions

- These tend to generalize best



Ockham's Razor (aka **Occam's Razor**): *Lex parsimoniae*

Prefer the simplest hypothesis that is consistent with the data

Note: This is a heuristic, not a logical principle or a scientific result

The curse of dimensionality

- Machine learning often involves very high-dimensional data
  - In general, the required amount of training data (and computational resources) scales exponentially with the dimensionality
- Sometimes the *intrinsic dimensionality* is lower and the problem is feasible if the relevant dimensions can be identified (and irrelevant dimensions ignored)
  - E.g., through dimensionality reduction methods
- How much training data is enough?
  - This is a difficult question in machine learning
- With a fixed number of training samples, the predictive power reduces as the dimensionality increases
  - This is known as the Hughes Effect
- Distance measures lose their usefulness in high dimensionality
  - Thus affecting clustering, classification, and other ML measures

Distance measures

- How similar are two faces? Two chess board configurations? Two countries' economies? Two DNA sequences?
  - We need ways to measure such things
- General assumption in ML: Similarity is a function of distance
  - But how to measure distance?
  - In what space? (What are the features?)
  - What's relevant and what's irrelevant in the data?
- Distance measures
  - Compute N features, resulting in a feature vector of N elements
  - The feature vector is then the only information the systems knows about the data sample
  - Define a distance measure between two feature vectors

How do we typically measure distance?

Some common distance measures

- Manhattan (L1) distance: $d(x,y) = \sum_{i=1}^{d} |x_i - y_i|$

  aka Cityblock distance

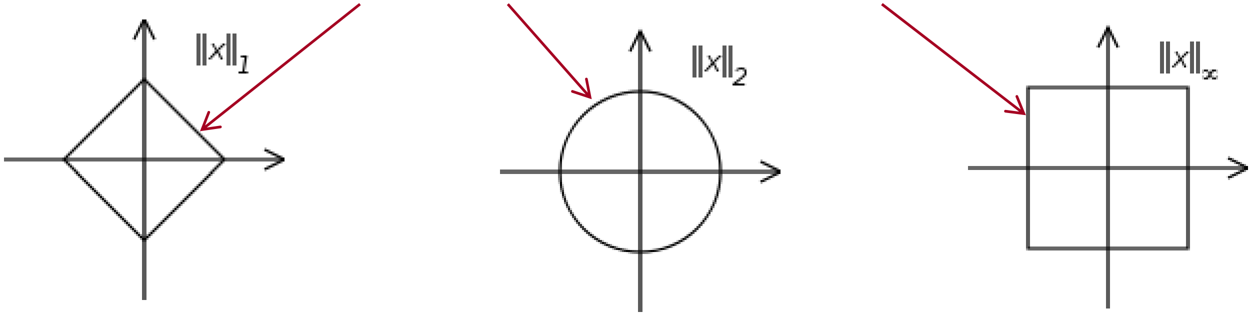
- Euclidian (L2) distance: $d(x,y) = \|x - y\| = \left(\sum_{i=1}^{d} (x_i - y_i)^2 \right)^{1/2}$

- Minkowski (L_p) distance: $d(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$

Also:

- Mahalanobis distance
- Chebyshev distance
- Hamming distance
- Edit distance
- etc.

These points are equidistant from the origin



L1 L2 L_∞

Bayes' Rule

The chain rule of probability states that

$$p(x, y) = p(y|x)p(x) = p(x|y)p(y)$$

Thus

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

This is called **Bayes' Rule**

This simple equation is the foundation of statistical machine learning, probabilistic reasoning, and much else!

Bayes’ Rule

- This simple equation is very useful in practice
  - Usually framed in terms of hypotheses (*H*) and data (*D*)
    - Which of the hypotheses is best supported by the data?

Likelihood
(causal knowledge)

Prior probability

$$P(H_i \mid D) = \frac{P(D \mid H_i)\, P(H_i)}{P(D)}$$

Posterior probability
(diagnostic knowledge)

Normalizing constant

$$\underbrace{P(H_i \mid D)}_{\text{Posterior}} = k\, P(D \mid H_i)\, \underbrace{P(H_i)}_{\text{Prior}}$$

Remember…

Machine learning is concerned with using the right features to build the right models that achieve the right tasks.

Training data is used to build the model

- E.g., to determine the parameters of the classification boundary or the regression function

Learning is concerned with accurate prediction of future (unseen) data, *not* accurate prediction of training data!

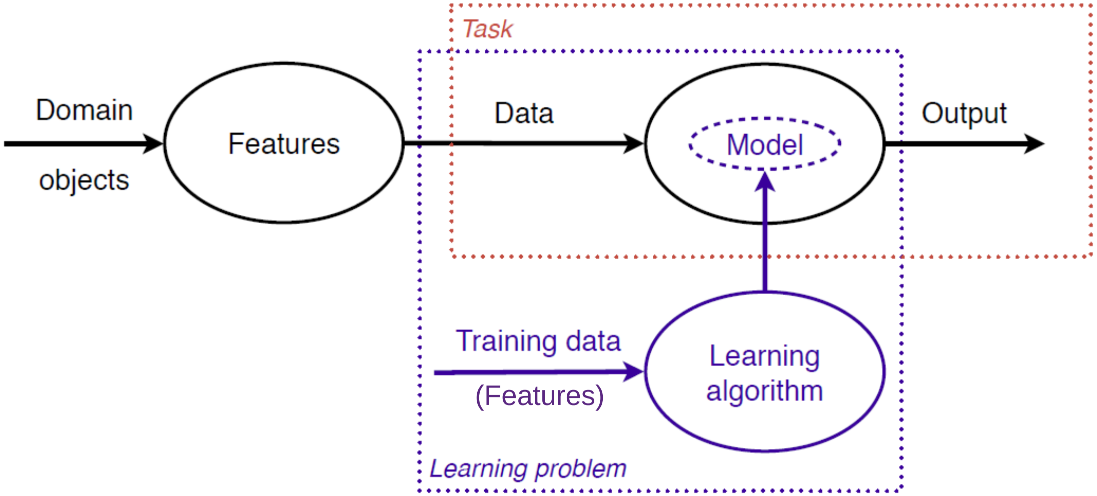
The ingredients of machine learning

Tasks, models, and features

Chapter 1 in the textbook

Machine learning

Machine learning is about using the right features to build the right models that achieve the right tasks



Features – how we describe our data objects

Model – a mapping from data points to outputs:

Output = *f* (*Data*)

This is what machine learning produces!

Task – an abstract representation of the problem we want to solve

Common ML tasks

- Classification – assign the target variable to one of N states
  - Binary: face or non-face, fraud or not fraud, malignant or benign…
  - Non-binary: person identity, correctly spelled word, movie genre…
- Regression – assign the target variable to a real-valued (scalar or vector) function of the input
  - For estimation or prediction
  - Learn the functional relationship, *output* = *f* (*input*)
  - Linear regression: Fit a line to the data
  - Non-linear regression: Fit a higher-dimensional function to the data
  - E.g.,
    - A trend line (stock prices, GDP, weight)
    - Epidemiology (e.g., the relationship between smoking and morbidity)
    - Economics – predict consumer spending, labor demand, imports

Common ML tasks (cont.)

- Clustering – grouping data without prior information (unlabeled data)
  - Objects in the same group (a cluster) are more similar to one another than to objects in other groups (clusters)
  - I.e., the within-class (intra-class) variance is smaller than the between-class (inter-class) variance
  - Why cluster?
    - To make apparent the natural groupings/structure in the data (perhaps for further processing)
    - To discover previously unknown relationships
    - To provide generic labels for the data

Common ML tasks (cont.)

- Association rule learning
  - Discovering interesting relations between variables in large databases (data mining)
- Subgroup discovery
  - Identifying subgroups of the data that behave differently with respect to the target variable. E.g., groups with higher rates of a disease.
- Anomaly detection
  - Detect outliers – items, events or observations that do not conform to an expected pattern
  - Anomalies, outliers, novelties, noise, deviations, exceptions…

Tasks: predictive and descriptive

- The most common ML tasks are predictive, aiming to predict/estimate a target variable from features:
  - Binary and multi-class classification: categorical target
    - Learn decision boundaries
  - Regression: numerical target
    - Learn relationship (a real-valued function) between input and output spaces

- Descriptive tasks are concerned with exploiting underlying structure in the data, finding patterns:
  - No specific problem to solve per data element
  - Goal: discover “interesting things” about the data
  - E.g., (descriptive) clustering
    - Grouping data without prior information

Models

- Machine learning models can be distinguished according to their main intuition:
  - Geometric models use intuitions from geometry such as separating (hyper-)planes, linear transformations and distance metrics
  - Probabilistic models view learning as a process of reducing uncertainty, modelled by means of probability distributions
  - Logical models are defined in terms of easily interpretable logical expressions

- Alternatively, they can be characterized by their *modus operandi*:
  - Grouping models divide the instance space into segments; in each segment a very simple (e.g., constant) model is learned
  - Grading models learning a single, global model over the instance space