CS 66: Machine Learning

Prof. Sara Mathieson

Spring 2019

Admin

- Office hours: TODAY 12:30-2pm (Sci Center 249)
- Lab 2 check-in this week in lab
- Still looking for one person to switch from Lab A to Lab B (yes this is the opposite of before… :)

Outline for February 4

- Recap decision trees (esp. continuous features)
- Lab 2 implementation suggestions
- Learning problem so far + terminology
- Begin: linear regression

Outline for February 4

- Recap decision trees (esp. continuous features)
- Lab 2 implementation suggestions
- Learning problem so far + terminology
- Begin: linear regression

Continuous Feature Example

temp

	t	$t \le 44$	$t \le 54$	$t \le 85$	Play Tennis
x_1	80	F	F	T	Y
x_2	48	F	T	T	Y
x_3	60				Y
x_4	48				Y
x_5	40				N
x_6	48				N
x_7	90				N

compute entropy for all

① Sort on feature

40	48	48	48	60	80	90
N	Y	Y	N	Y	Y	N

② collapse equal features (unique)

40	48	60	80	90
N	None	Y	Y	N

③ any change in label, create a feature

$t \le 44$ $t \le 54$ $t \le 85$

N N N N Y N Y N N Y Y Y Y Y N Y N Y

min entropy

Outline for February 4

- Recap decision trees (esp. continuous features)
- Lab 2 implementation suggestions
- Learning problem so far + terminology
- Begin: linear regression

Implementation Suggestions

- Think back to trees in CS35 (data structures)
- Distinguish between data (X,y) and options for data (values for each feature, classes for y)
- Start slow with entropy! Build up function by function

Real-World Examples

- Medical diagnostics

- Credit risk analysis

- Modeling calendar scheduling preferences

Decision Trees in Chemistry reactions

- Example of decision trees in practice
- Use decision trees to interpret another ML algorithm (SVMs)

Machine-learning-assisted materials discovery using failed experiments

Paul Raccuglia, Katherine C. Elbert, Philip D. F. Adler, Casey Falk, Malia B. Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A. Friedler, Joshua Schrier & Alexander J. Norquist

Nature **533**, 73–76 (05 May 2016) | Download Citation

Optional Reading!

Brainstorm advantages and disadvantages of decision trees over k-nearest neighbors

	Pro	Con
k-nearest neighbors	- easy to train - matrix libary => easy to implement - high enough k: minimal overfitting	- long time to evaluate - all features have equal weight - make distance metric for discrete features.
decision tree	- fast to evaluate - shows important features - compact representation + storage - intuitive w/ few features	- longer to train - harder to implement - overfitting - not natural for cont. features

Outline for February 4

- Recap decision trees (esp. continuous features)
- Lab 2 implementation suggestions
- Learning problem so far + terminology
- Begin: linear regression

Recap: overfitting

- Consider a hypothesis (tree): h
  - Training error: $error_{train}(h)$
  - Error over all possible data: $error_D(h)$

- A hypothesis h **overfits** training data if there exists another hypothesis h' s.t.
  - $error_{train}(h) < error_{train}(h')$ AND
  - $error_D(h) > error_D(h')$

Slide: modified from Ameet Soni

Learning Problem so far

- Performance on training data <u>overestimates</u> accuracy
- We must use a <u>held aside</u> test set to evaluate
- Both training and testing data should be drawn from the same distribution
- Training/test data should be drawn from the same distribution as seen in deployment

Formalizing the learning problem

- Given:
  - Loss function, ℓ
  - A sample of data D from an unknown distribution of all data $\mathcal{D}$
  - A hypothesis space $H = \{h | h : X \rightarrow Y\}$

Slide: Ameet Soni

Formalizing the learning problem

- Given:
  - Loss function, ℓ
  - A sample of data D from an unknown distribution of all data $\mathcal{D}$
  - A hypothesis space $H = \{h | h : X \rightarrow Y\}$
- Do:
  - Find a function $f(X) \rightarrow y$ that
  - minimize error over $\mathcal{D}$ with respect to ℓ

Slide: Ameet Soni

Loss Functions

- E.g., zero-one loss
  - Simple accuracy - is prediction right?
  - For binary or multi-class prediction

$$l(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

Slide: Ameet Soni

Loss Functions

- E.g., zero-one loss
  - Simple accuracy - is prediction right?
  - For binary or multi-class prediction

$$l(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

- E.g., squared loss
  - For regression

$$l(y, \hat{y}) = (y - \hat{y})^2$$

Slide: Ameet Soni

Loss Functions

- E.g., zero-one loss
  - Simple accuracy - is prediction right?
  - For binary or multi-class prediction

$$l(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

- E.g., squared loss
  - For regression

$$l(y, \hat{y}) = (y - \hat{y})^2$$

- Many alternatives for specific tasks (e.g., cost-sensitive learning)

Slide: Ameet Soni

Generalization Error

- "A sample of data D from an unknown distribution of all data $\mathcal{D}$"
- What are D and $\mathcal{D}$?
- i.i.d. assumption - training data should be drawn **i**ndependently and **i**dentically **d**istributed from all data
  - Exceptions: time-series data, structured data, active learning

Slide: modified from Ameet Soni

Generalization Error

$$\text{error}_{\mathcal{D}} = \underbrace{\sum_{(x,y)\in\mathcal{D}}}_{\text{all samples}} P(x,y)\cdot \underbrace{\ell(y, f(x))}_{\text{loss, or error of model}}$$

not realistic

Prob of data point (x,y)

unknown

model

Generalization Error

- Problem: we (usually) don't know $\mathcal{D}$ (distribution of data)
- We do have training data D
- Key dilemma: want to minimize generalization error but all we can guarantee is training error

Slide: Ameet Soni

Outline for February 4

- Recap decision trees (esp. continuous features)
- Lab 2 implementation suggestions
- Learning problem so far + terminology
- Begin: linear regression

Assessing Model Accuracy

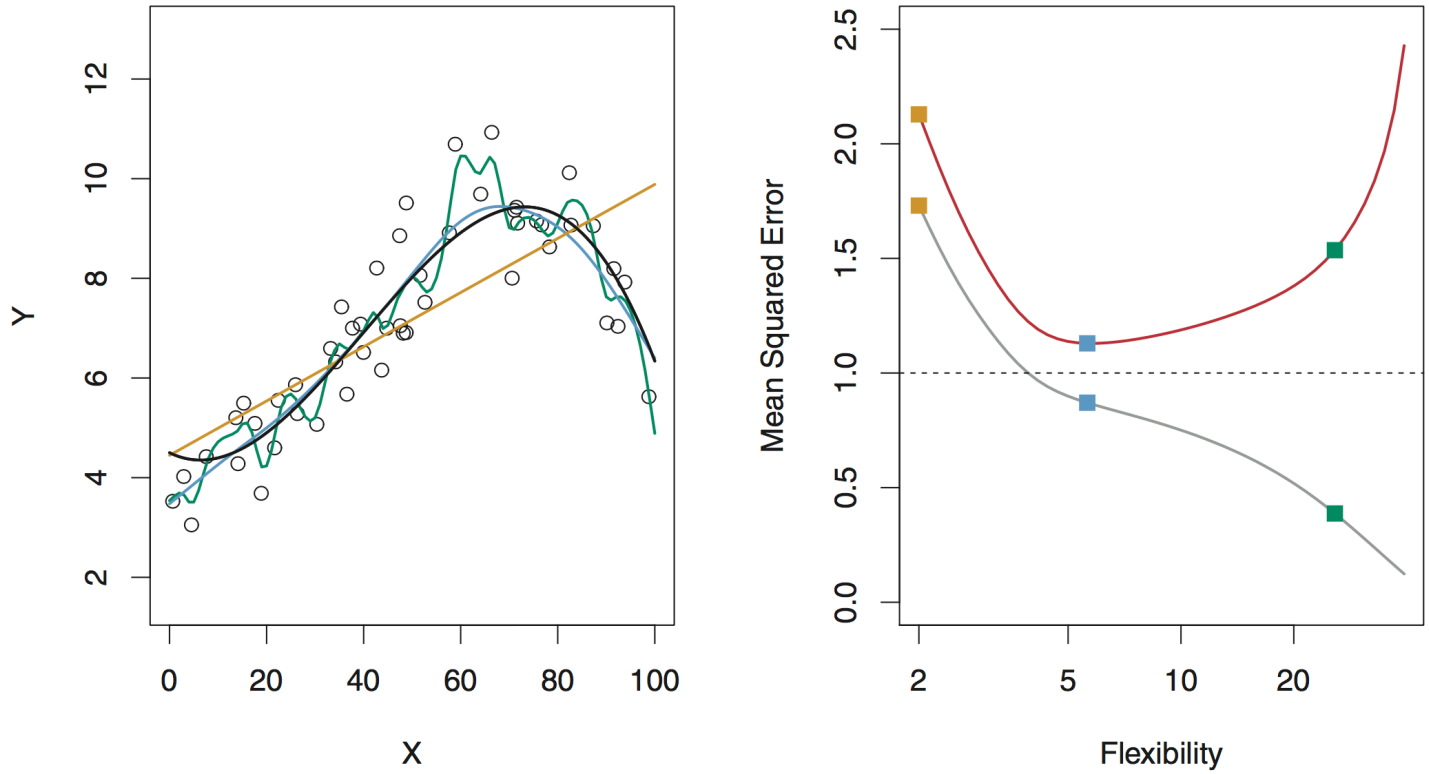

FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

ISL, Fig 2.9