CS 360: Machine Learning

Sara Mathieson, Sorelle Friedler

Spring 2024

HAVERFORD COLLEGE

Sit somewhere new!

Admin

- **Lab 2** due Thursday
  - Ideally you should be well past the naïve algorithm
- **OAR** (peer tutoring)
- 300-level class
  - Jump up from CS260 in terms of creative solutions on your own
  - More important to work with others and talk through ideas and algorithms

Outline for Feb 6

- Machine Learning pipeline
- Learning problem so far + terminology
- Sources of error
- Bias-variance tradeoff
- Cross Validation
- Model Cards

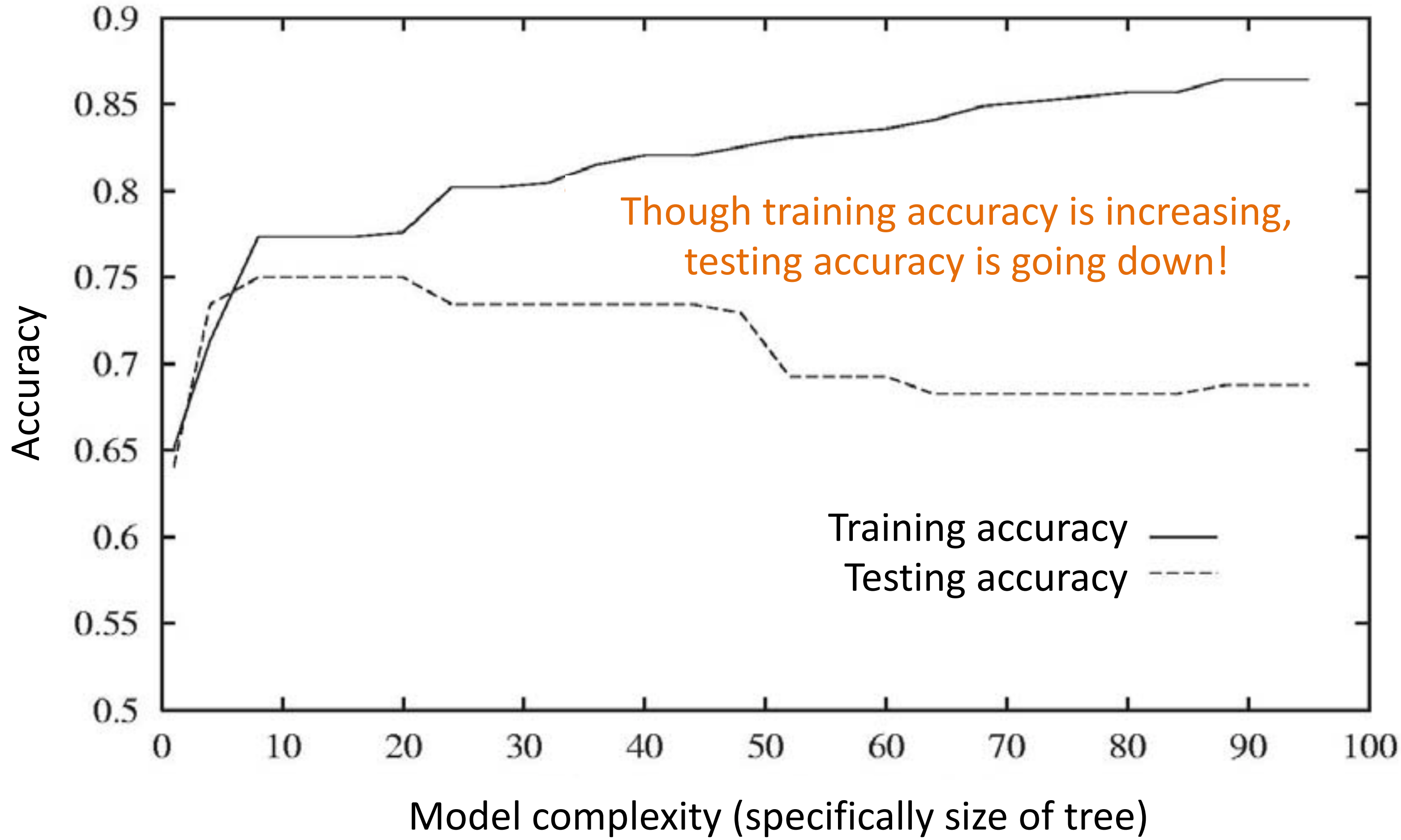
Outline for Feb 6

- Machine Learning pipeline
- Learning problem so far + terminology
- Sources of error
- Bias-variance tradeoff
- Cross Validation
- Model Cards

Learning Problem so far

- Performance on training data overestimates accuracy
- We must use a held aside 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 (ideally)

Training data overestimates accuracy



Slide: modified from Jessica Wu
Based on slide by David Kauchak
(originally by Pedro Domingos)

Overfitting more concretely

- Consider a hypothesis (i.e. model): *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
  - $error_D(h) > error_D(h')$

Loss Functions

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

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

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: modified from 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

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

- Absolute loss (also for regression)

$$\ell(y, \hat{y}) = |y - \hat{y}|$$

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

Generalization Error

- "A sample of data D from an $\boxed{\text{unknown distribution}}$ 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

- 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 Feb 6

- Machine Learning pipeline
- Learning problem so far + terminology
- Sources of error
- Bias-variance tradeoff
- Cross Validation
- Model Cards

Why might learning fail?

Group discussion

Inductive Bias

Training Data

class A

class B

Testing Data

Figures 2.1 and 2.2 from Duame

Inductive Bias

Training Data

class A

class B

Testing Data

A A

B B

A: “fly”

B: “no fly”

Figures 2.1 and 2.2 from Duame

Inductive Bias

Training Data

class A

class B

Testing Data

A B

B A

A: "bird"

B: "mammal"

Figures 2.1 and 2.2 from Duame

Why might learning fail?

- Noise in the training data
  - Typos in a restaurant review

Why might learning fail?

- Noise in the training data
  - Typos in a restaurant review

- Available features are insufficient
  - x-ray does not capture the medical issue

Adapted from Duame Chap 2

Why might learning fail?

- Noise in the training data
  - Typos in a restaurant review
- Available features are insufficient
  - x-ray does not capture the medical issue
- “Correct” prediction is up to interpretation
  - Parental controls on web content

Why might learning fail?

- Noise in the training data
  - Typos in a restaurant review
- Available features are insufficient
  - x-ray does not capture the medical issue
- “Correct” prediction is up to interpretation
  - Parental controls on web content
- Learning algorithm cannot cope with the data

Adapted from Duame Chap 2

Generalization Error

$$E_{(x,y)}\Big[\ell\big(\underbrace{y}_{\text{truth}},\ \underbrace{\hat{f}(x)}_{\text{prediction}}\big)\Big]$$

minimize

assume regression

$$y = \underbrace{f(x)}_{\text{"true model"}} + \varepsilon$$

y (true) (see it!)

ε error (mean 0)

$$\boxed{\hat{y} = \hat{f}(x)}$$

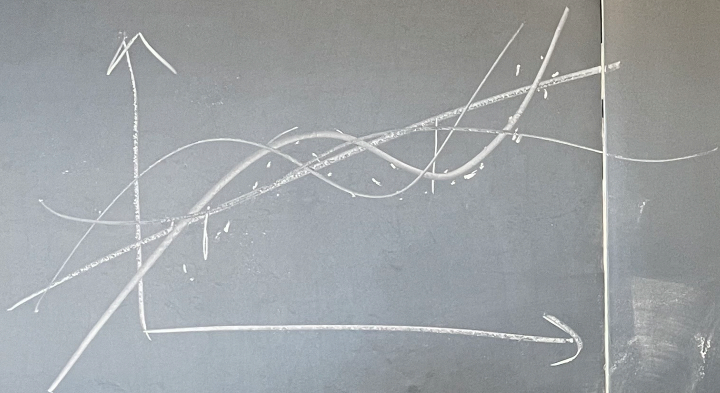
Bias-Variance Tradeoff

expected value of MSE

$\Rightarrow$ mean squared error

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

$$E[(y-\hat{y})^2] = E[f(x)+\varepsilon-\hat{f}(x)]^2$$



$$= \underbrace{E[f(x)-\hat{f}(x)]^2}_{\text{reducible error}} + \underbrace{Var(\varepsilon)}_{\text{irreducible error}}$$

$$= E[\underbrace{f - E[\hat{f}]}_{\text{Bias}} + \underbrace{E[\hat{f}]-\hat{f}}_{\text{Variance}}]^2$$

0

$$\boxed{\text{flexibility}\uparrow}$$

bias $\downarrow$

Var $\uparrow$

$$= \text{bias}(\hat{f})^2 + Var(\hat{f}) + Var(\varepsilon)$$

Bias-Variance tradeoff

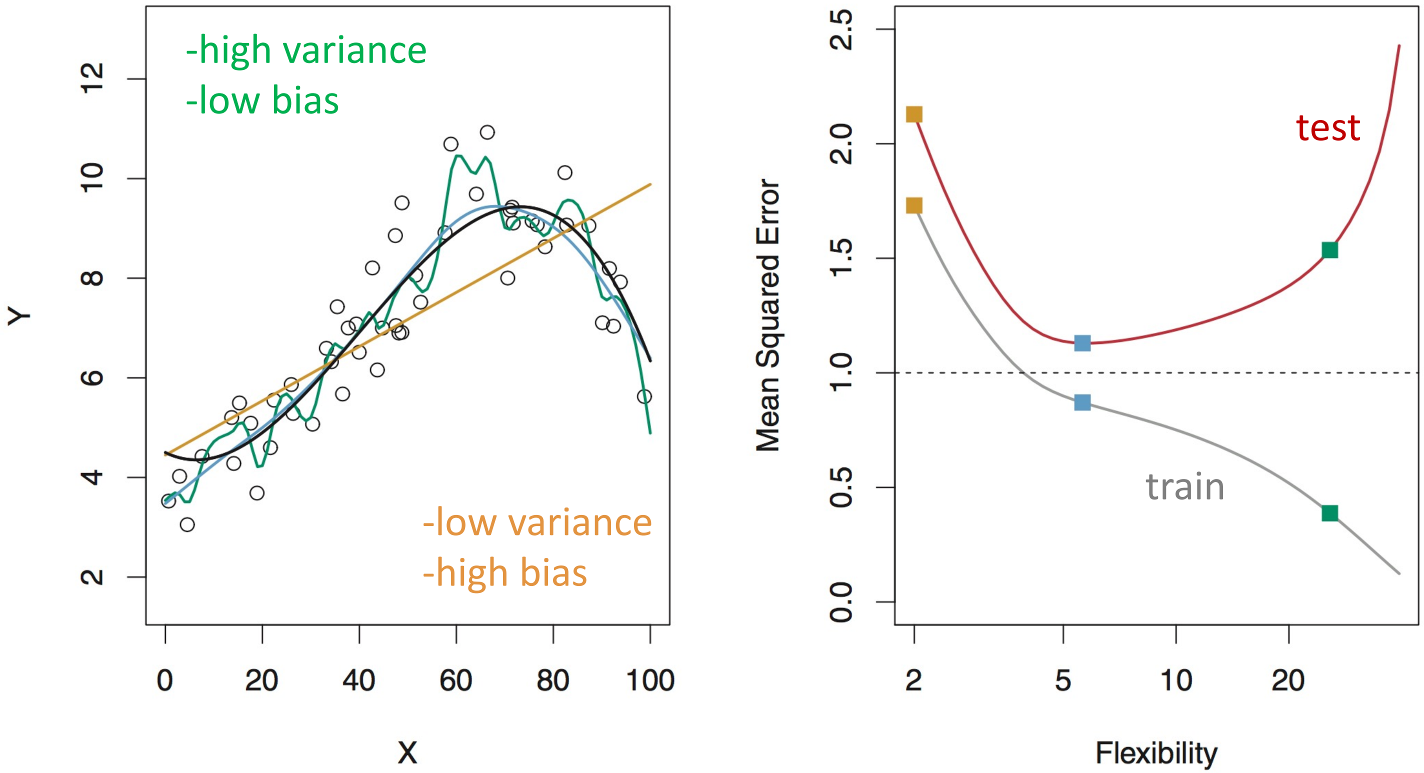


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed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

ISL, Fig 2.9

Outline for Feb 6

- Machine Learning pipeline
- Learning problem so far + terminology
- Sources of error
- Bias-variance tradeoff
- Cross Validation
- Model Cards

General approach to training

1. Split your data into 70% training data, 10% development data and 20% test data. (validation data)

General approach to training

1. Split your data into 70% training data, 10% development data and 20% test data.

(validation data)

2. For each possible setting of your hyperparameters:

   (a) Train a model using that setting of hyperparameters on the training data.

   (b) Compute this model's error rate on the development data.

General approach to training

1. Split your data into 70% training data, 10% development data and 20% test data. (validation data)

2. For each possible setting of your hyperparameters:

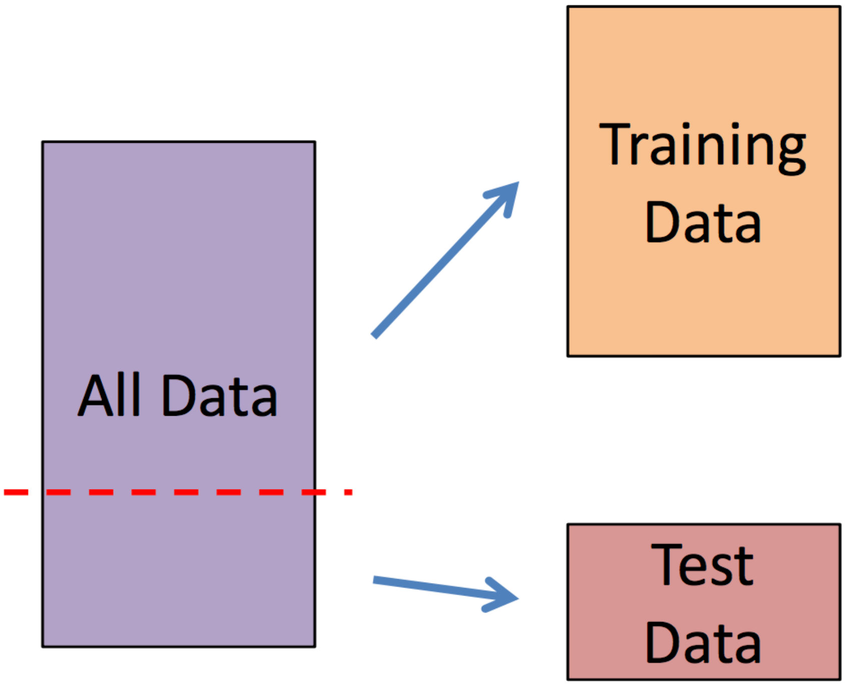
   (a) Train a model using that setting of hyperparameters on the training data.

   (b) Compute this model's error rate on the development data.

3. From the above collection of models, choose the one that achieved the lowest error rate on development data.

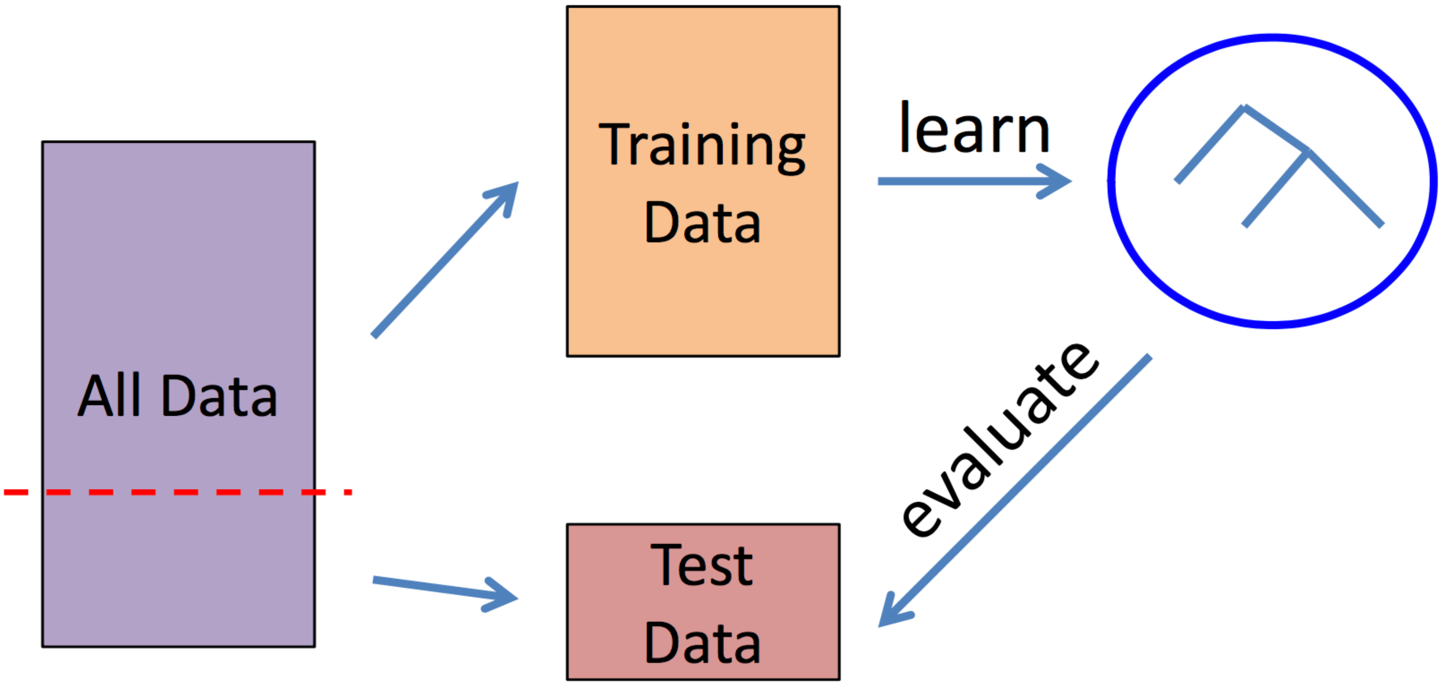
4. Evaluate that model on the test data to estimate future test performance.

Evaluation in Practice



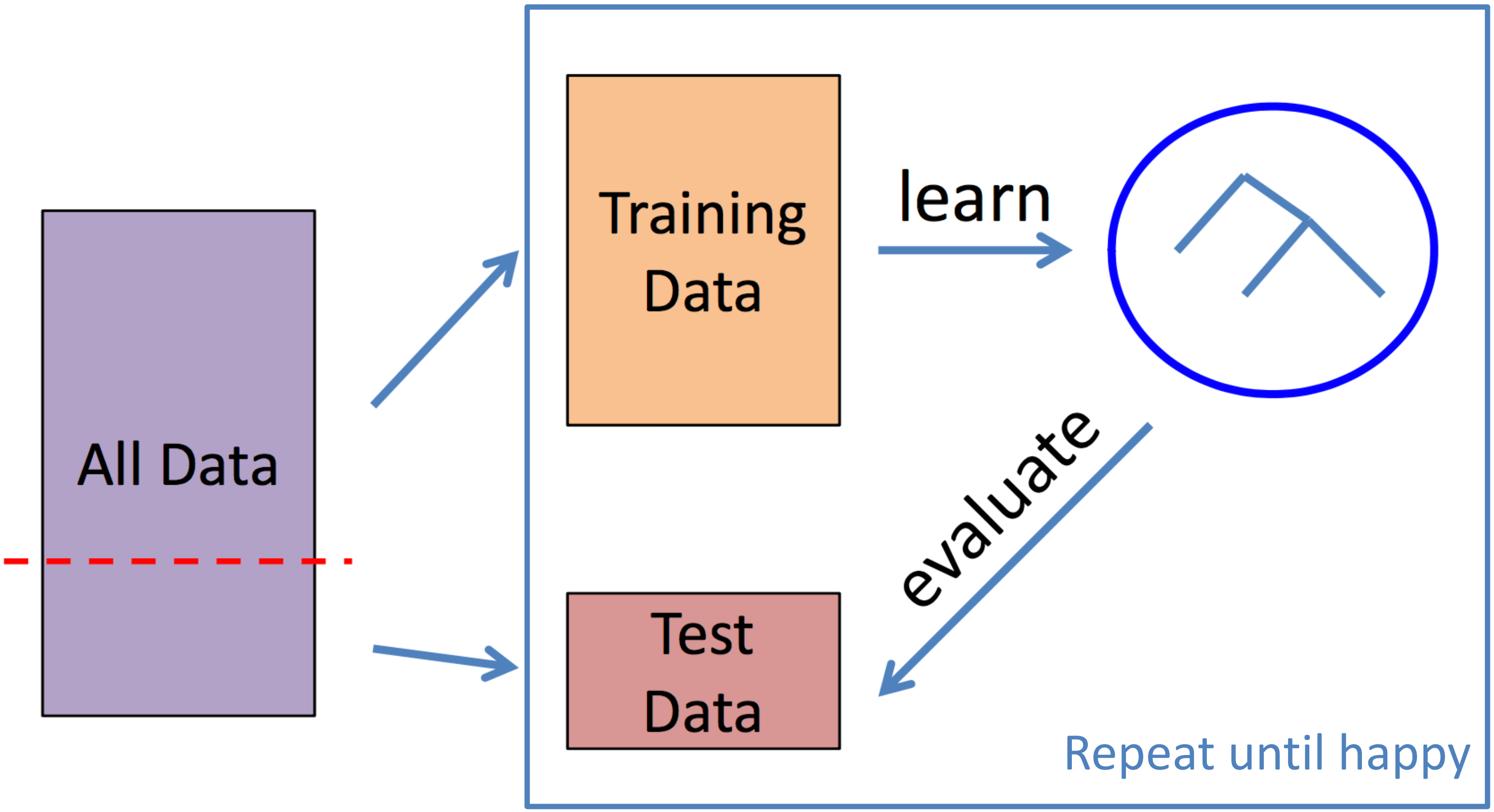
Modified from Jessica Wu

Evaluation in Practice

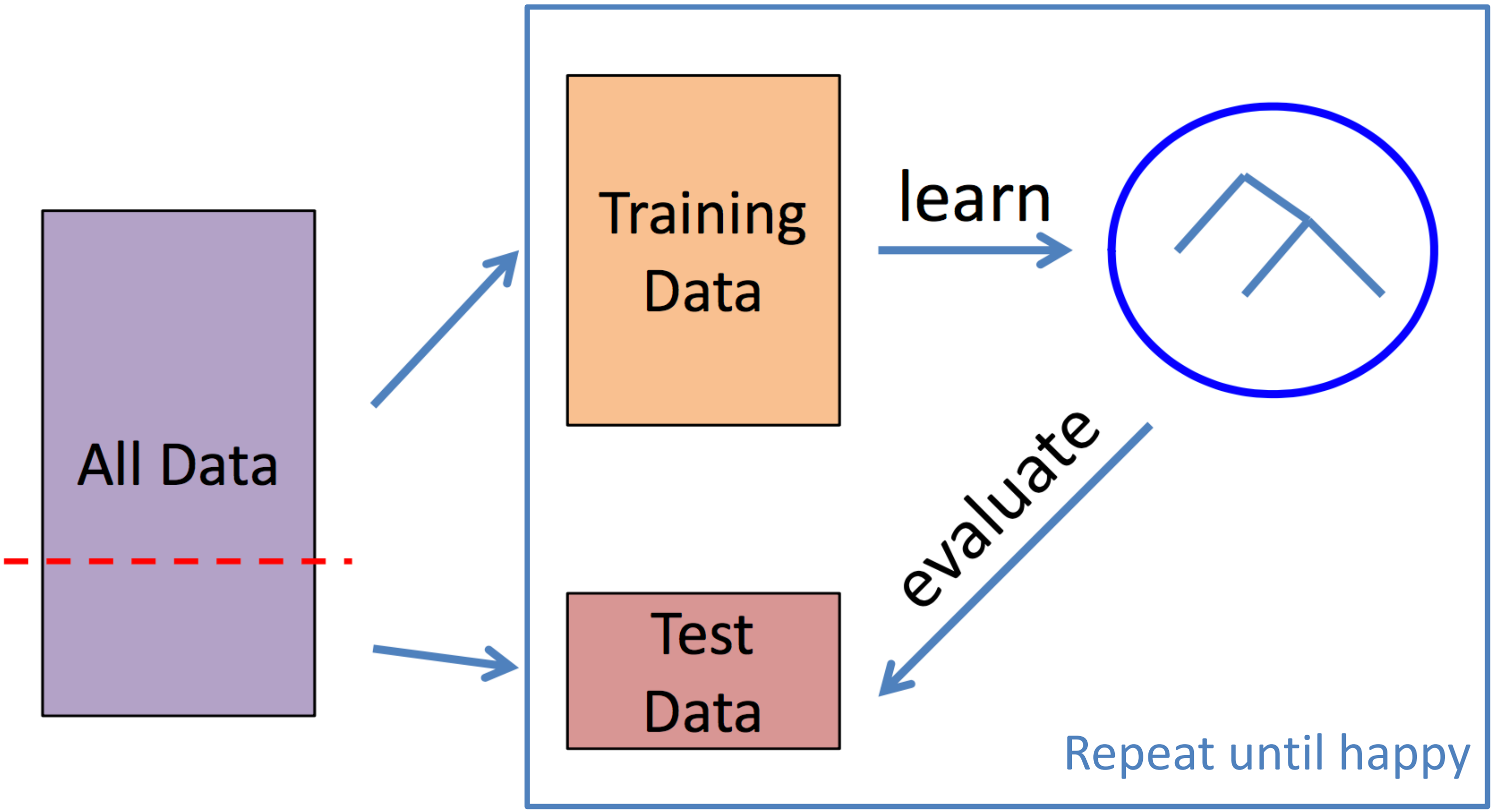


Modified from Jessica Wu

Evaluation in Practice



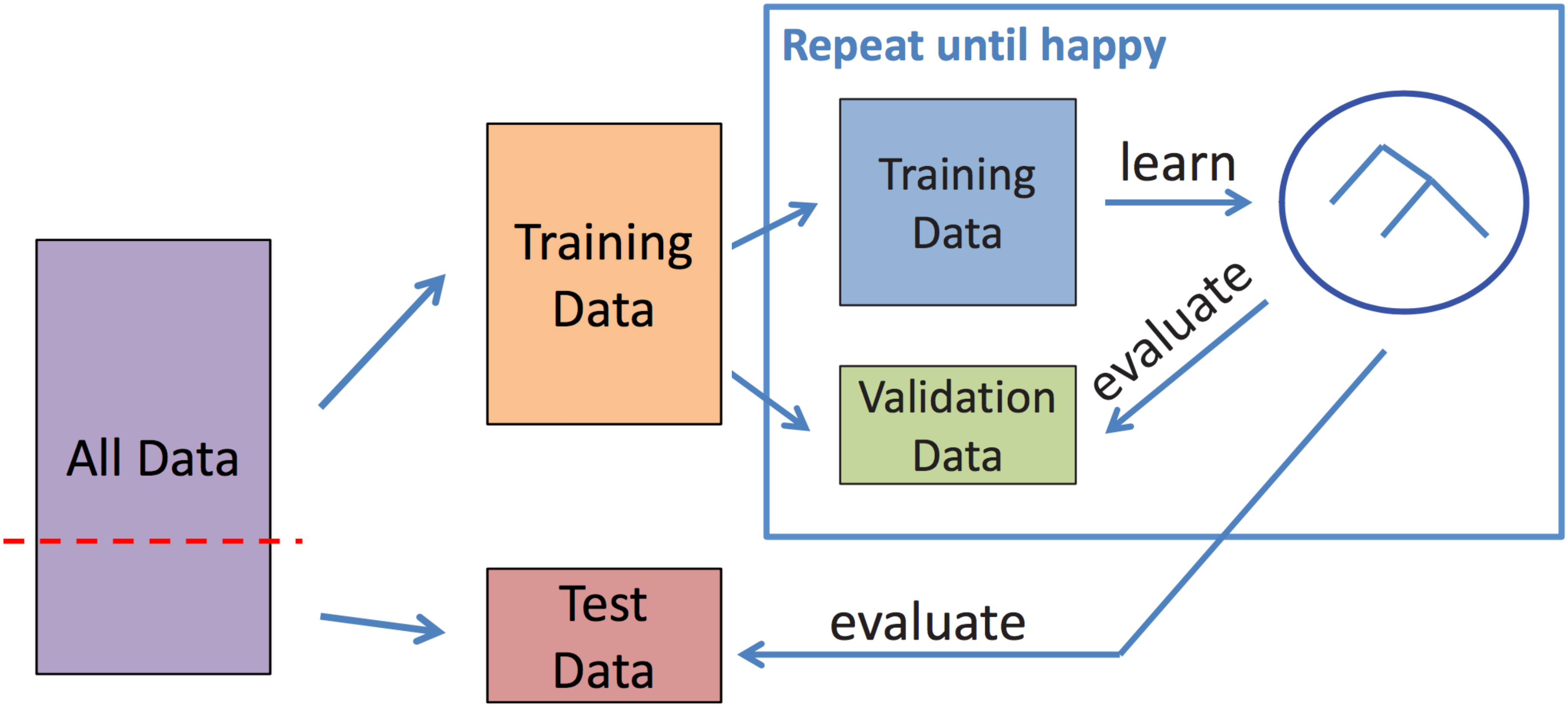
Evaluation in Practice



NO! Using test data as part of the model selection process

Modified from Jessica Wu

Better: use a *validation* dataset



Modified from Jessica Wu

k-fold Cross Validation

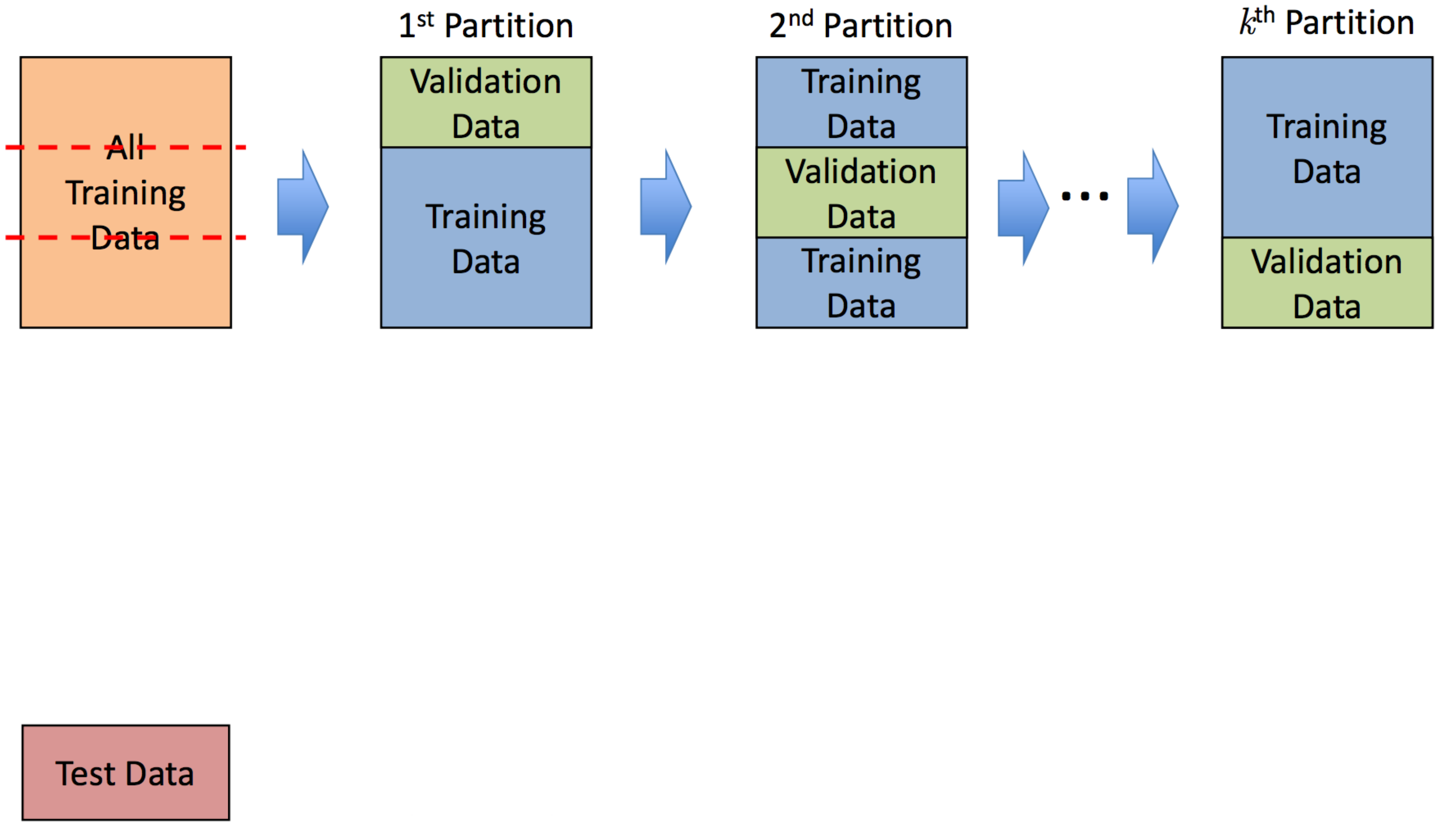
- Why just choose one particular "split" of data?
  - in principle, we should do this multiple times since performance may be different for each split

k-fold Cross Validation

- Why just choose one particular "split" of data?
  - in principle, we should do this multiple times since performance may be different for each split

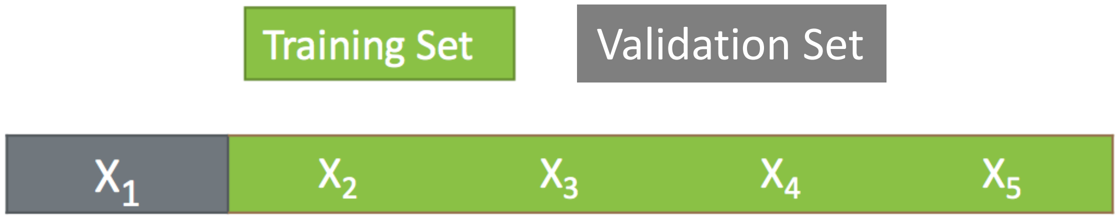
- k-Fold Cross-Validation (e.g., $k = 10$)
  - randomly partition full data set of n instances into $\boldsymbol{k}$ **disjoint subsets** (each roughly of size n/k)
  - choose each fold in turn as validation set; train model on the other $k - 1$ folds and evaluate
  - compute statistics over k test performances, or choose best of k models

k-fold Cross Validation



Slide: Jessica Wu

k-fold Cross Validation



Fold	Evaluation
1	*Accuracy*(P_1) *= 11/20*

k-fold Cross Validation

Evaluation

Accuracy$(P_1) = 11/20$

Accuracy$(P_2) = 17/20$

Accuracy$(P_3) = 16/20$

Accuracy$(P_4) = 13/20$

Accuracy$(P_5) = 16/20$

Slide: Ameet Soni

k-fold Cross Validation

Evaluation

Accuracy(P_1) = *11/20*

Accuracy(P_2) = *17/20*

Accuracy(P_3) = *16/20*

Accuracy(P_4) = *13/20*

Accuracy(P_5) = *16/20*

Generalization: average accuracy across all folds = 73/100 = 73%

Slide: Ameet Soni

sklearn example of cross-validation

```python
from sklearn.model_selection import cross_val_score

tree_rmses = -cross_val_score(tree_reg, housing, housing_labels,
                              scoring="neg_root_mean_squared_error", cv=10)
```

count	10.000000
mean	66868.027288
std	2060.966425
min	63649.536493
25%	65338.078316
50%	66801.953094
75%	68229.934454
max	70094.778246

count	10.000000
mean	47019.561281
std	1033.957120
min	45458.112527
25%	46464.031184
50%	46967.596354
75%	47325.694987
max	49243.765795

```python
from sklearn.ensemble import RandomForestRegressor

forest_reg = make_pipeline(preprocessing,
                           RandomForestRegressor(random_state=42))
forest_rmses = -cross_val_score(forest_reg, housing, housing_labels,
                                scoring="neg_root_mean_squared_error", cv=10)
```

Discussion

1) What are the costs of *k*-fold cross validation?

2) Pros and cons of no longer having one model?

3) How to choose *k*?

Discussion

1) What are the costs of *k*-fold cross validation?

    - Computational, especially if training takes a long time

2) Pros and cons of no longer having one model?

    - Con: might be hard to interpret
    - Pro: might be able to average results

3) How to choose *k*?

    - Large *k* can be good for small datasets (i.e. where *n* is small)
    - Tradeoff between computation and reducing variance
    - Many choose *k*=10 in practice :)

Cross Validation: other considerations

- Can use cross-validation to choose hyperparameters

- Leave-one-out cross validation (LOOCV)
  - Special case of $k=n$
  - Train using $n-1$ examples, evaluate on remaining
  - Repeat n times

- Can do multiple trials of CV

Outline for Feb 6

- Machine Learning pipeline
- Learning problem so far + terminology
- Sources of error
- Bias-variance tradeoff
- Cross Validation
- Model Cards