CS 171: Intro to ML and DM

Christian Shelton

UC Riverside

Slide Set 5: Cross Validation

Slides from CS 171

- From UC Riverside
  - CS 171: Introduction to Machine Learning and Data Mining
  - Professor Christian Shelton
- DO NOT REDISTRIBUTE
  - These slides contain copyrighted material (used with permission) from
    - Elements of Statistical Learning (Hastie, et al.)
    - Pattern Recognition and Machine Learning (Bishop)
    - An Introduction to Machine Learning (Kubat)
    - Machine Learning: A Probabilistic Perspective (Murphy)
  - For use only by enrolled students in the course

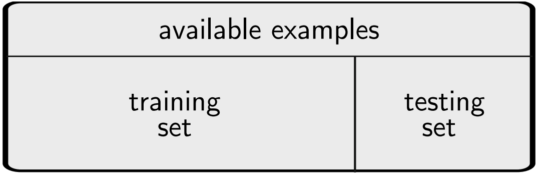
Cross-Validation for Regularization

So, how to pick λ?

Cross-Validation for Regularization

So, how to pick λ?
Cross validation! (or n-fold cross validation)

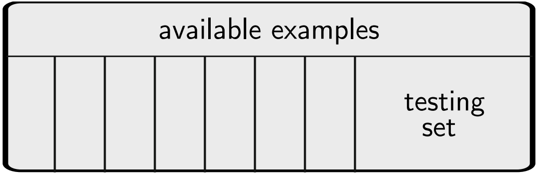
Cross Validation



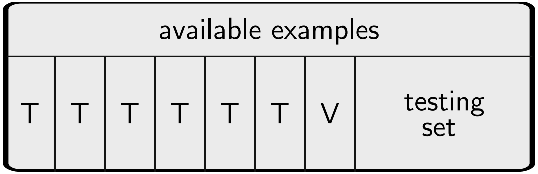
Cross Validation

available examples		
training set	validation set	testing set

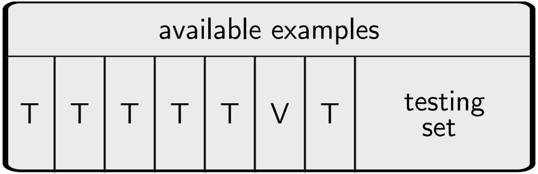
Cross Validation



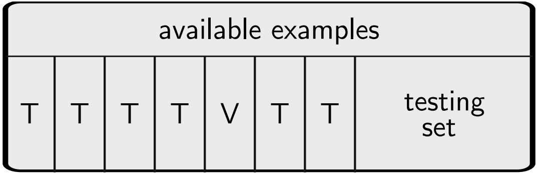
Cross Validation



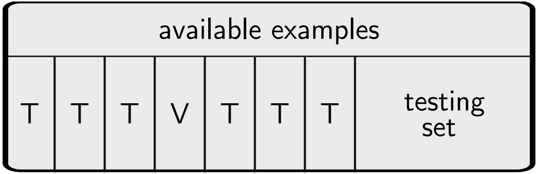
Cross Validation



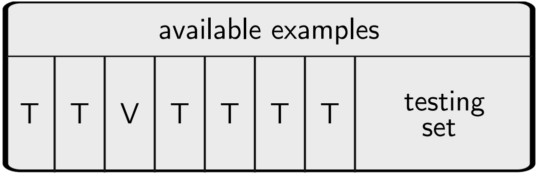
Cross Validation



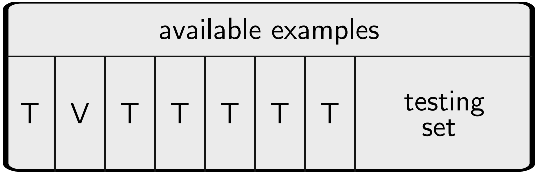
Cross Validation



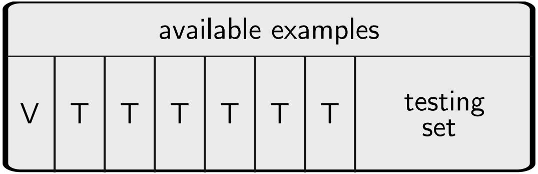
Cross Validation



Cross Validation



Cross Validation



Cross Validation

Cross Validation:

- Split Training data into two parts: Train and Validation
- For each version (different k), train on train and check performance on validate
- Pick the version that does best on validation set

n-fold Cross Validation:

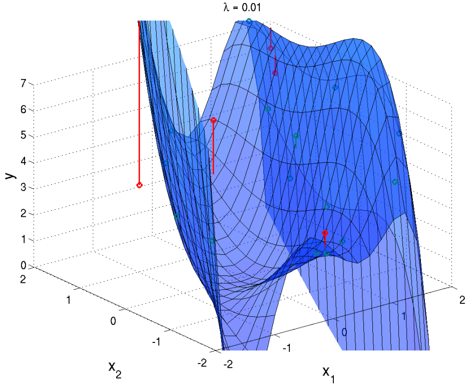
- Do cross validation n times (n different train/validation splits)
- Pick version (value of k) that does best on average across these n different trails
- Most common: partition data into n equal-sized sets and use each one as validation once (and other $n-1$ sets as train)

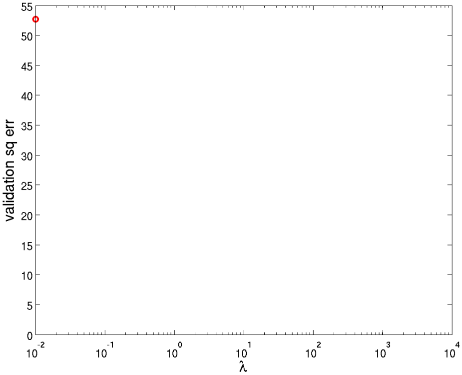
Leave-one-out Cross Validation (LOO CV):

- n-fold Cross Validation where $n = m$

Cross-Validation for Regularization

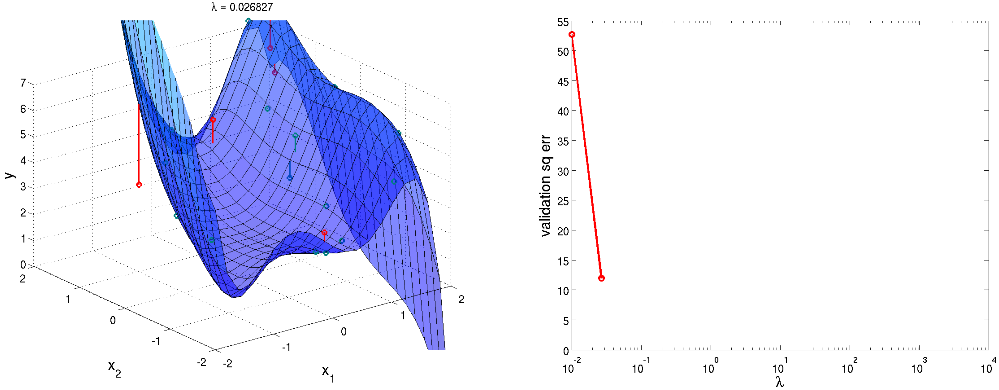
5th order polynomial





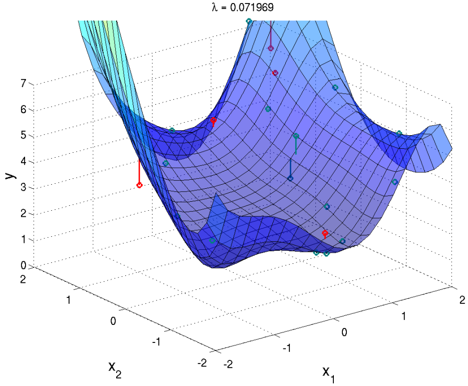
Cross-Validation for Regularization

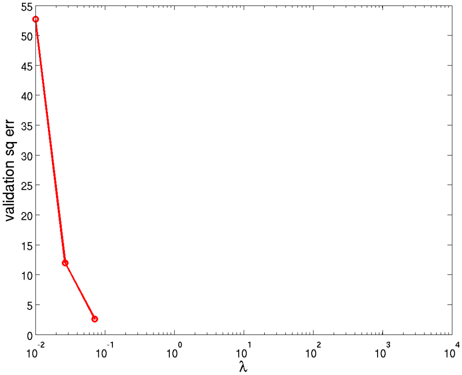
5th order polynomial



Cross-Validation for Regularization

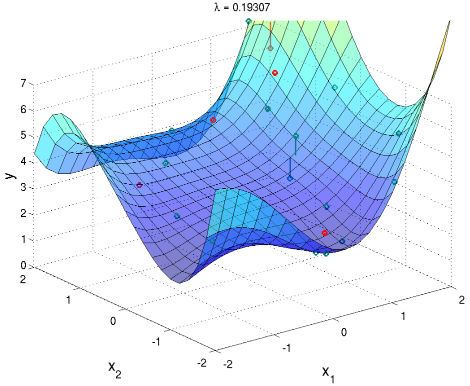
5th order polynomial

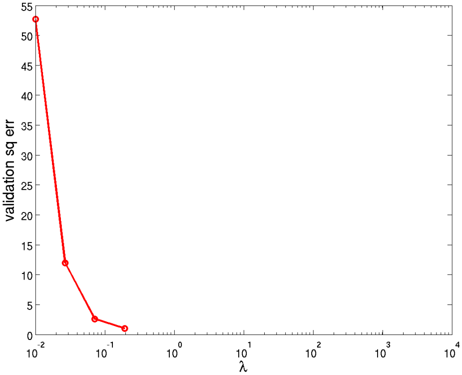




Cross-Validation for Regularization

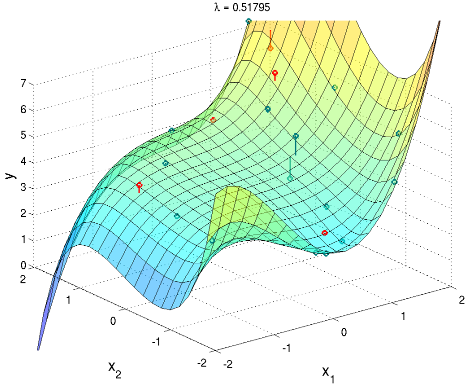
5th order polynomial

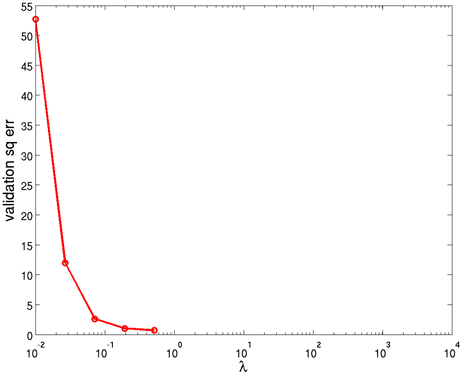




Cross-Validation for Regularization

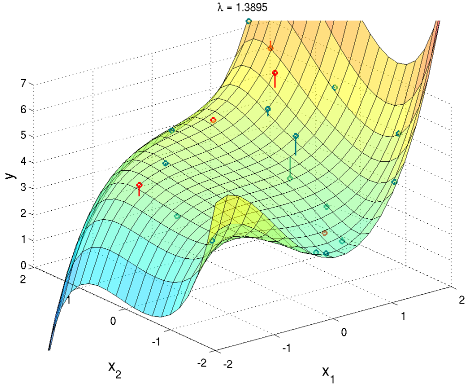
5th order polynomial

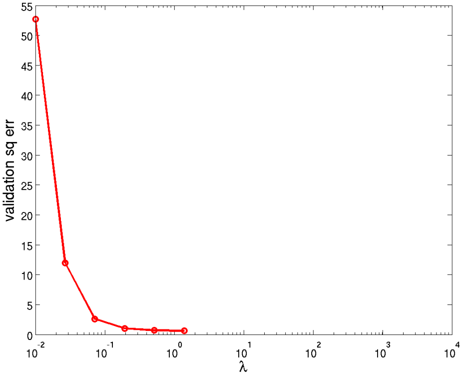




Cross-Validation for Regularization

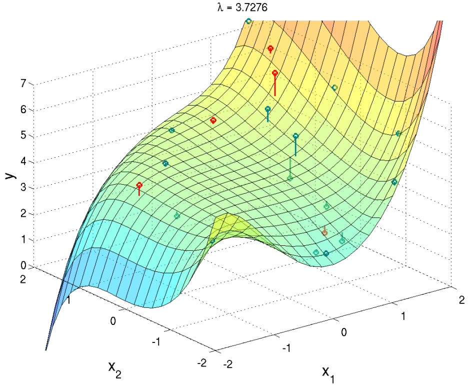
5th order polynomial

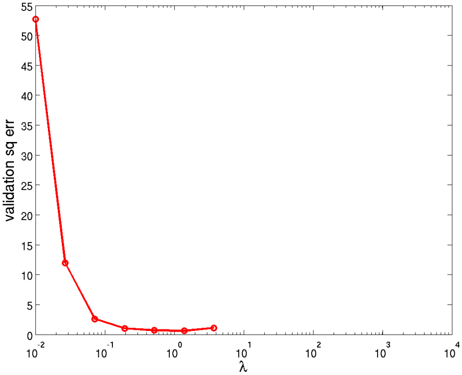




Cross-Validation for Regularization

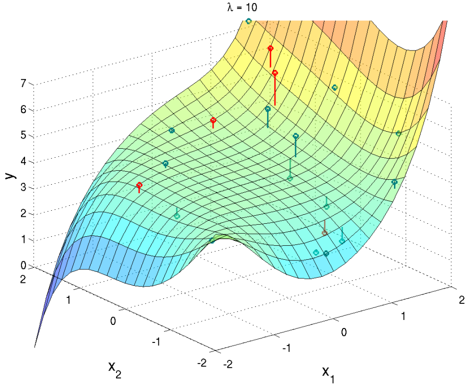
5th order polynomial

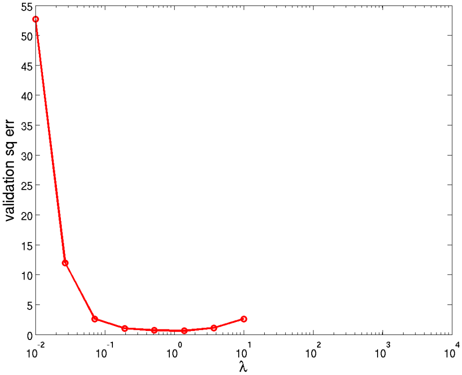




Cross-Validation for Regularization

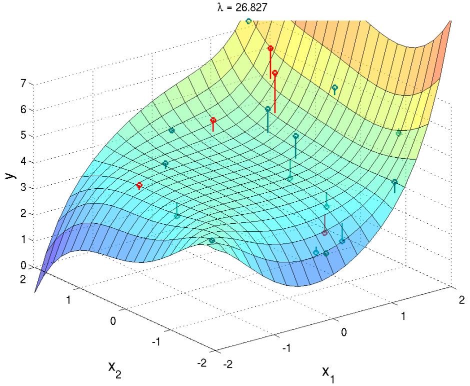
5th order polynomial

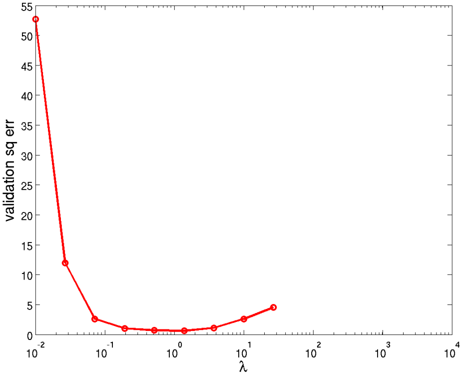




Cross-Validation for Regularization

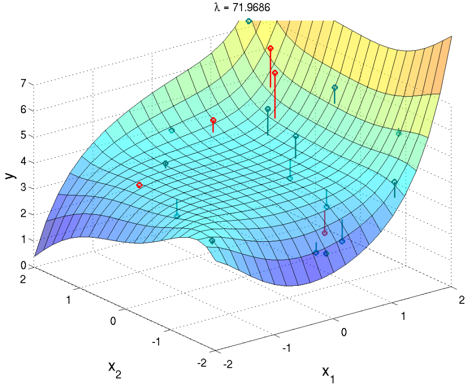
5th order polynomial

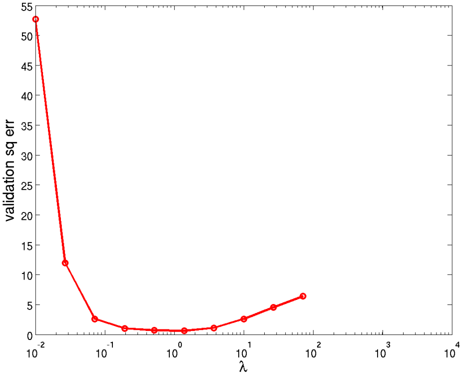




Cross-Validation for Regularization

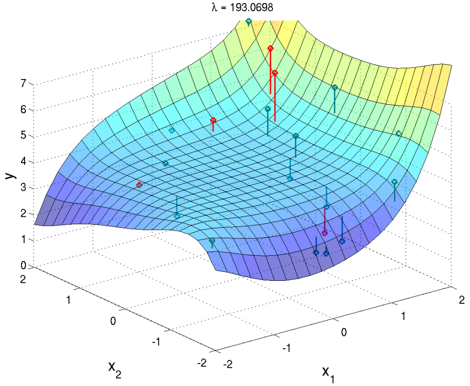
5th order polynomial

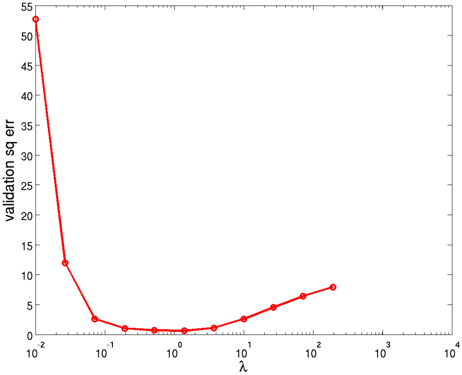




Cross-Validation for Regularization

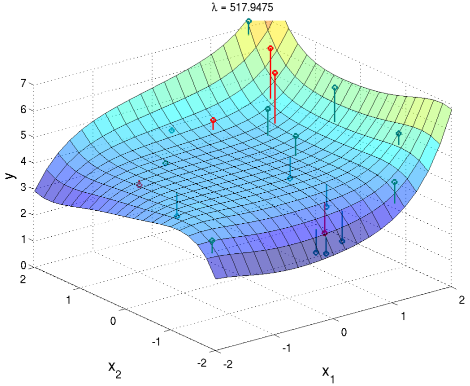
5th order polynomial

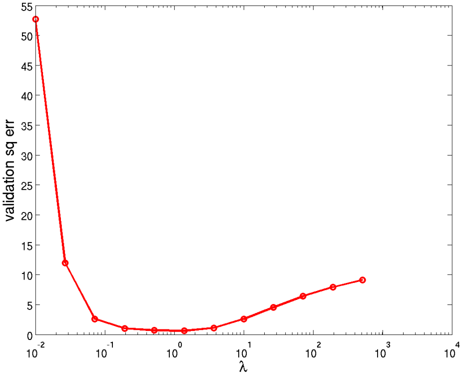




Cross-Validation for Regularization

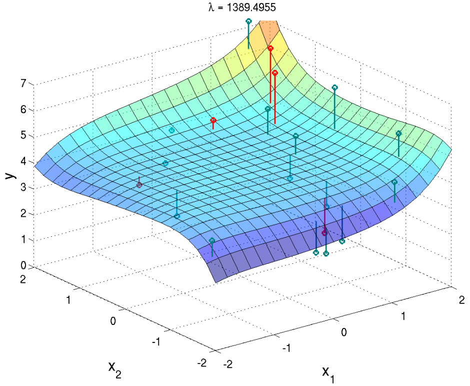
5th order polynomial

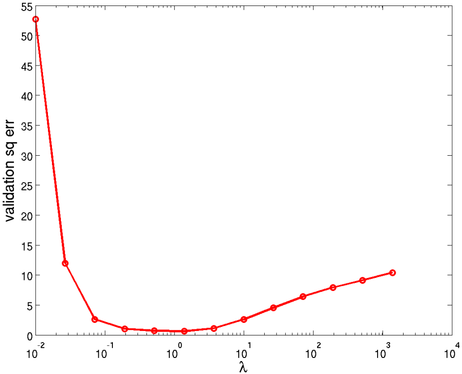




Cross-Validation for Regularization

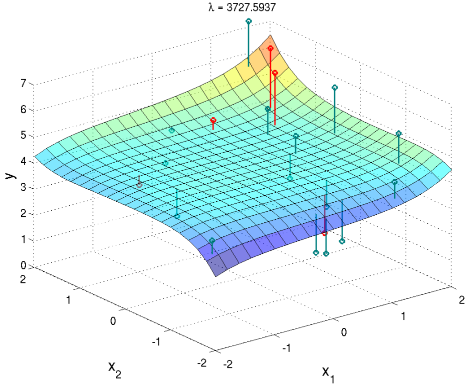
5th order polynomial

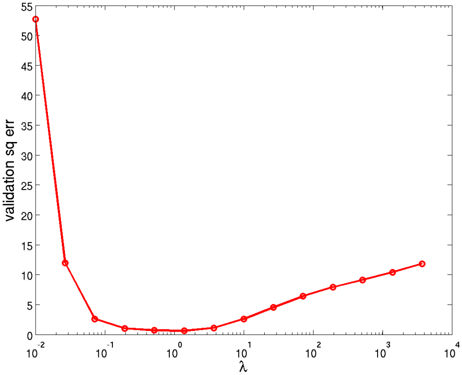




Cross-Validation for Regularization

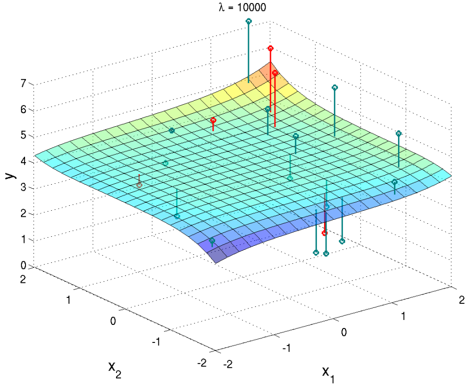
5th order polynomial

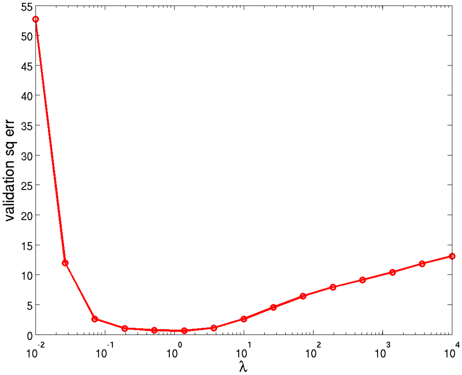




Cross-Validation for Regularization

5th order polynomial





Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:

$$f(x) = w^\top x$$

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:

$$f(x) = w^\top x$$

- Pick per-item loss/cost function:

$$l(y, f) = (y - f)^2$$

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:

$$f(x) = w^\top x$$

- Pick per-item loss/cost function:

$$l(y, f) = (y - f)^2$$

- Pick regularizer:

$$R(w) = \lambda w^\top w$$

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:
$$f(x) = w^\top x$$
- Pick per-item loss/cost function:
$$l(y, f) = (y - f)^2$$
- Pick regularizer:
$$R(w) = \lambda w^\top w$$
- Write down total loss/cost:
$$L = \sum_i l(y_i, f(x_i)) + R(w)$$

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:
$$f(x) = w^\top x$$
- Pick per-item loss/cost function:
$$l(y, f) = (y - f)^2$$
- Pick regularizer:
$$R(w) = \lambda w^\top w$$
- Write down total loss/cost:
$$L = \sum_i l(y_i, f(x_i)) + R(w)$$
- Figure out how to minimize L:
$$w = (X^\top X + \lambda I)^{-1} X^\top Y$$

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:
$$f(x) = w^\top x$$
- Pick per-item loss/cost function:
$$l(y, f) = (y - f)^2$$
- Pick regularizer:
$$R(w) = \lambda w^\top w$$
- Write down total loss/cost:
$$L = \sum_i l(y_i, f(x_i)) + R(w)$$
- Figure out how to minimize L:
$$w = (X^\top X + \lambda I)^{-1} X^\top Y$$

Algorithm Use:

- Pick a set of λs, Λ
- Divide Data into Training and Testing
- Divide Training into Training and Validation

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:
$$f(x) = w^\top x$$
- Pick per-item loss/cost function:
$$l(y, f) = (y - f)^2$$
- Pick regularizer:
$$R(w) = \lambda w^\top w$$
- Write down total loss/cost:
$$L = \sum_i l(y_i, f(x_i)) + R(w)$$
- Figure out how to minimize L:
$$w = (X^\top X + \lambda I)^{-1} X^\top Y$$

Algorithm Use:

- Pick a set of λs, Λ
- Divide Data into Training and Testing
- Divide Training into Training and Validation
- For each $\lambda \in \Lambda$:
  - Train on Training set using λ
  - Check average per-item loss on Validation set
  - If best loss so far, remember w and λ

Regularized Least Squares

Algorithm Development:

- Pick form of function to be estimated:
$$f(x) = w^\top x$$
- Pick per-item loss/cost function:
$$l(y, f) = (y - f)^2$$
- Pick regularizer:
$$R(w) = \lambda w^\top w$$
- Write down total loss/cost:
$$L = \sum_i l(y_i, f(x_i)) + R(w)$$
- Figure out how to minimize L:
$$w = (X^\top X + \lambda I)^{-1} X^\top Y$$

Algorithm Use:

- Pick a set of λs, Λ
- Divide Data into Training and Testing
- Divide Training into Training and Validation
- For each $\lambda \in \Lambda$:
  - Train on Training set using λ
  - Check average per-item loss on Validation set
  - If best loss so far, remember w and λ
- (Optional) Retrain on Training+Validation with best λ
- Report best w
- (If testing) Check average per-item loss on Testing set