

# Lab 4

# LASSO Regularization and Feature Selection

---

CREATED BY PROF. SUNDEEP RANGAN AND PROF. WANG FOR EE-UY 4563/EL-GY 9123

MODIFIED BY ARTIE SHEN

# Outline

---

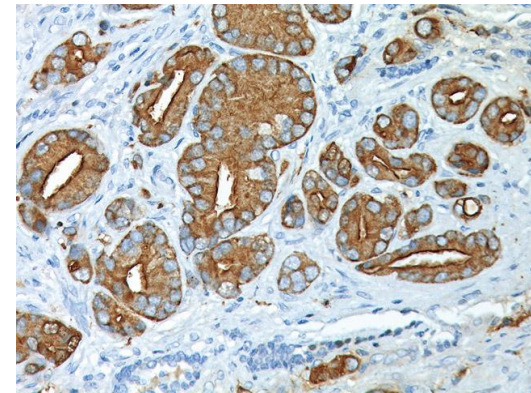


Motivating Example: Predicting prostate cancer from a PSA test

- ☐ Model Selection
- ☐ Model Selection from LASSO regularization
- ☐ Other Model Selection Methods
- ☐ In-Class Exercise: Audio Pitch Detection

# Prostate Specific Antigen Testing

- ❑ PSA levels easily tested
- ❑ **High PSA believed to be associated with prostate cancer**
  - Potential tool for screening
- ❑ Classic 1989 study by Thomas et al:
  - Measured PSA level of 102 men prior to prostate removal
  - Measured characteristics of prostate from samples
  - Characteristics include cancer volume, weight, ...



Stamey, Thomas A., et al. "Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients." The Journal of urology 141.5 (1989): 1076-1083.

# Prostate Specific Antigen Testing

---

Can we use biological and demographical information about the patient to predict PSA level of this patient?

# Data

- ❑ Prostate dataset widely-used in ML classes
- ❑ 97 patients, 8 features
- ❑ Target variable = lpsa (log PSA)

```
# Get data
url = 'https://web.stanford.edu/~hastie/ElemStatLearn/datasets/prostate.data'
df = pd.read_csv(url, sep='\t', header=0)
df = df.drop('Unnamed: 0', axis=1) # skip the column of indices
```

The data frame has the following components:

```
lcavol      log(cancer volume)
lweight     log(prostate weight)
age         age
lbph        log(benign prostatic hyperplasia amount)
svi         seminal vesicle invasion
lcp         log(capsular penetration)
gleason     Gleason score
pgg45       percentage Gleason scores 4 or 5
lpsa        log(prostate specific antigen)
```

# First Try: Linear Model

## □ Simple idea: Use linear regression

$$y \approx \hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d$$

- $y$  = lpsa (target PSA level)
- $x_1, \dots, x_d$  = prostate features ( $d = 8$ )

## □ Why linear regression?

- Easy to compute / interpret
- Coefficients are easy to interpret
- Larger coefficients  $\Rightarrow$  larger influence of feature on PSA

```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
```

```
X_tr, X_ts, y_tr, y_ts = train_test_split(X, y, test_size=0.5, shuffle=True)
ntr = X_tr.shape[0]
nts = X_ts.shape[0]
print("num samples train = %d, test = %d" % (ntr, nts))
```

```
num samples train = 48, test = 49
```

```
regr = linear_model.LinearRegression()
regr.fit(X_tr, y_tr)
```

# Model Fit

## ❑ Evaluate model with cross validation

- Train on 48 samples
- Measure RSS on 49 samples

## ❑ We obtain reasonable fit on test data

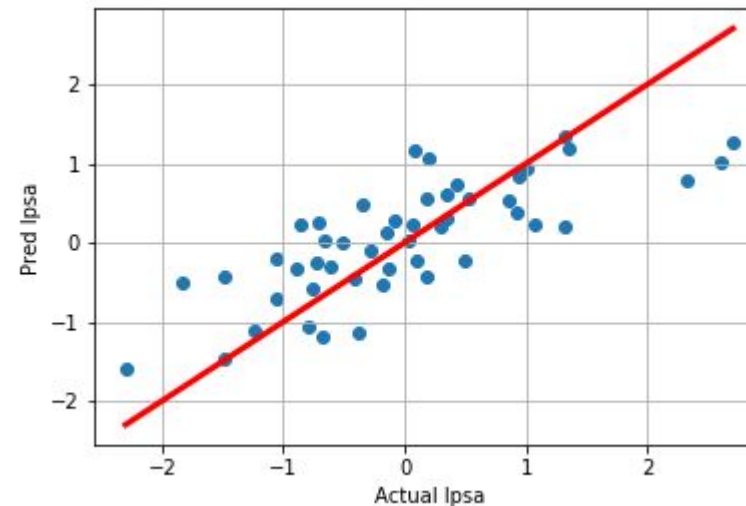
- $R^2 \approx 0.58$

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

```
y_ts_pred = regr.predict(X_ts)
rss_ts = np.mean((y_ts_pred-y_ts)**2)/(np.std(y_ts)**2)
rsq_ts = 1-rss_ts
print("Normalized test RSS = %f" % rss_ts)
print("Normalized test R^2 = %f" % rsq_ts)
```

Normalized test RSS = 0.419799

Normalized test R^2 = 0.580201



# Looking at the Coefficients

□ Recall that model is:  $\hat{y} = b + w_1x_1 + \dots + w_dx_d$

□ Weights  $w_i$  indicates dependence of feature  $i$  on target  $y$

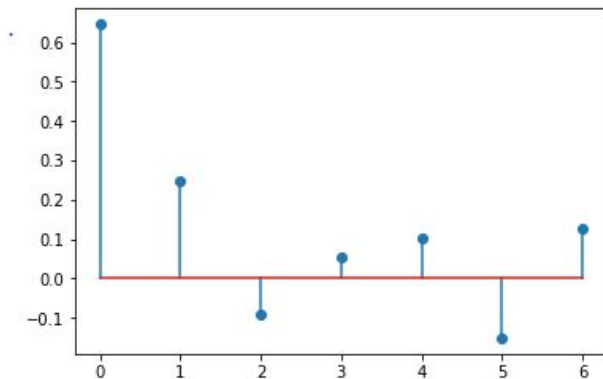
□ For PSA test:

- Highest weight on lcavol (log cancer volume)
- But, weights on all features are non-zero
- Hard to eliminate features

□ How can we tell if some features are not significant?

```
w = regr.coef_  
for name, wi in zip(names_x, w):  
    print('%10s %9.4f' % (name, wi))
```

lcavol	0.6457
lweight	0.2466
age	-0.0895
lbph	0.0543
svi	0.1034
lcp	-0.1508
gleason	0.1253






# Outline

---

☐ Motivating Example: Predicting prostate cancer from a PSA test

 ☐ Model Selection

☐ Model Selection from LASSO regularization

☐ Other Model Selection Methods

☐ In-Class Exercise: Audio Pitch Detection

# Model Selection

---

- ❑ Consider linear model:  $y \approx \hat{y} = b + w_1x_1 + \dots + w_dx_d$
- ❑ Models target  $y$  as function of features  $\mathbf{x} = (x_1, \dots, x_d)$
- ❑ In many problems, we know only a few features are likely relevant
- ❑ This means we want  $w_j = 0$  for most features
- ❑ But, we don't know a priori which features are relevant
- ❑ Model selection problem: Fit a model with a small number of features
- ❑ Mathematically:
  - Determine a subset of features  $I \subseteq \{1, \dots, d\}$  with  $|I|$  small
  - Fit a model:  $\hat{y} = b + w_1x_1 + \dots + w_dx_d$  with  $w_j = 0$  for all  $j \notin I$


# Model Selection with Limited Data

---

- ❑ Model selection is particularly valuable when data is limited
- ❑ Ex: Consider linear model:  $\hat{y} = b + w_1x_1 + \dots + w_dx_d$ 
  - Model has  $d + 1$  parameters
- ❑ From previous lecture, we need  $N > d + 1$  data points  $(x_i, y_i)$
- ❑ In many cases we have  $N \ll d$ 
  - Examples below
  - Many few data points than features
  - Classic linear fit will not work
- ❑ But, suppose we can restrict to  $K \ll N$  non-zero parameters
  - Then, we can find a good fit on those parameters
- ❑ Challenge: How do we find a small number  $K$  of relevant features

# Outline

---

- ☐ Motivating Example: Predicting prostate cancer from a PSA test
- ☐ Model Selection
-  ☐ Model Selection from LASSO regularization
- ☐ Other Model Selection Methods
- ☐ In-Class Exercise: Audio Pitch Detection

# Intuition

- ❑ We know from last lecture:
  - Too many parameters  $\Rightarrow$  Large generalization error
- ❑ In this data set, only a few factors are likely significant
- ❑ But, we don't know which one
- ❑ Can we automatically identify them?
  - Use correlation between features and target
    - Do not always work well
  - Exhaustive search can be expansive!
- ❑ **Idea:** Fit model under constraint:
  - Force only a few parameters to be non-zero
- ❑ General idea of **regularization**:
  - Constrain the parameters with prior knowledge

The data frame has the following components:

```
lcavol      log(cancer volume)
lweight     log(prostate weight)
age         age
lbph        log(benign prostatic hyperplasia amount)
svi         seminal vesicle invasion
lcp         log(capsular penetration)
gleason     Gleason score
pgg45       percentage Gleason scores 4 or 5
lpsa        log(prostate specific antigen)
```

# Regularized LS Estimation

---

□ Standard least squares estimation (from Lecture 3):

$$\hat{\beta} = \arg \min_{\beta} RSS(\beta), \quad RSS(\beta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

□ Regularized estimator:

$$\hat{\beta} = \arg \min_{\beta} J(\beta), \quad J(\beta) = RSS(\beta) + \phi(\beta)$$

- $RSS(\beta)$  = prediction error from before
- $\phi(\beta)$  = regularizing function.

□ Concept: Regularizer penalizes  $\beta$  that are “unlikely”

- Constrains estimate to smaller set of parameters

# Two Common Regularizers

## ❑ Ridge regression (called L2)

$$\phi(\beta) = \alpha \sum_{j=1}^d |\beta_j|^2$$

## ❑ LASSO regression (called L1)

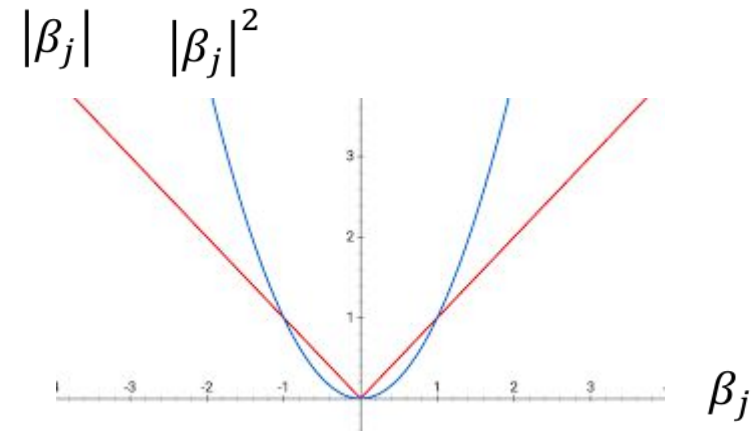
$$\phi(\beta) = \alpha \sum_{j=1}^d |\beta_j|$$

## ❑ Both penalize large $\beta_j$

## ❑ Level of regularization controlled by $\alpha$

## ❑ Note the regularization sum

- Does not include the intercept  $\beta_0$ ,
- This term depends on the mean of the target
- Should not be arbitrarily constrained to be small



Minimize  $|\beta_j|^2$  do not penalize small non-zero coef., overly penalize large coef.

Minimize  $|\beta_j|$  tend to make coefficients either 0 or large (SPARSE!)

# L1 and L2 Norm

---

□ Assuming the data have been scaled to have zero mean and unit variance

□ Ridge cost function:

$$J(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^d |\beta_j|^2 = \|\mathbf{y} - A\boldsymbol{\beta}\|^2 + \alpha \|\boldsymbol{\beta}\|^2$$

□ LASSO cost function:

$$J(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^d |\beta_j| = \|\mathbf{y} - A\boldsymbol{\beta}\|^2 + \alpha \|\boldsymbol{\beta}\|_1$$

◦  $\|\boldsymbol{\beta}\|_1$  = L1 norm (pronounced ell-1)



# Ridge Regression

---

- ❑ Solution for given regularization level
  - Easily obtainable by setting gradient to zero (HW!)

$$J(\boldsymbol{\beta}) = \|\mathbf{y} - A\boldsymbol{\beta}\|^2 + \alpha \|\boldsymbol{\beta}\|^2$$

$$\boldsymbol{\beta}_{ridge} = (A^T A + \alpha I)^{-1} A^T \mathbf{y}$$

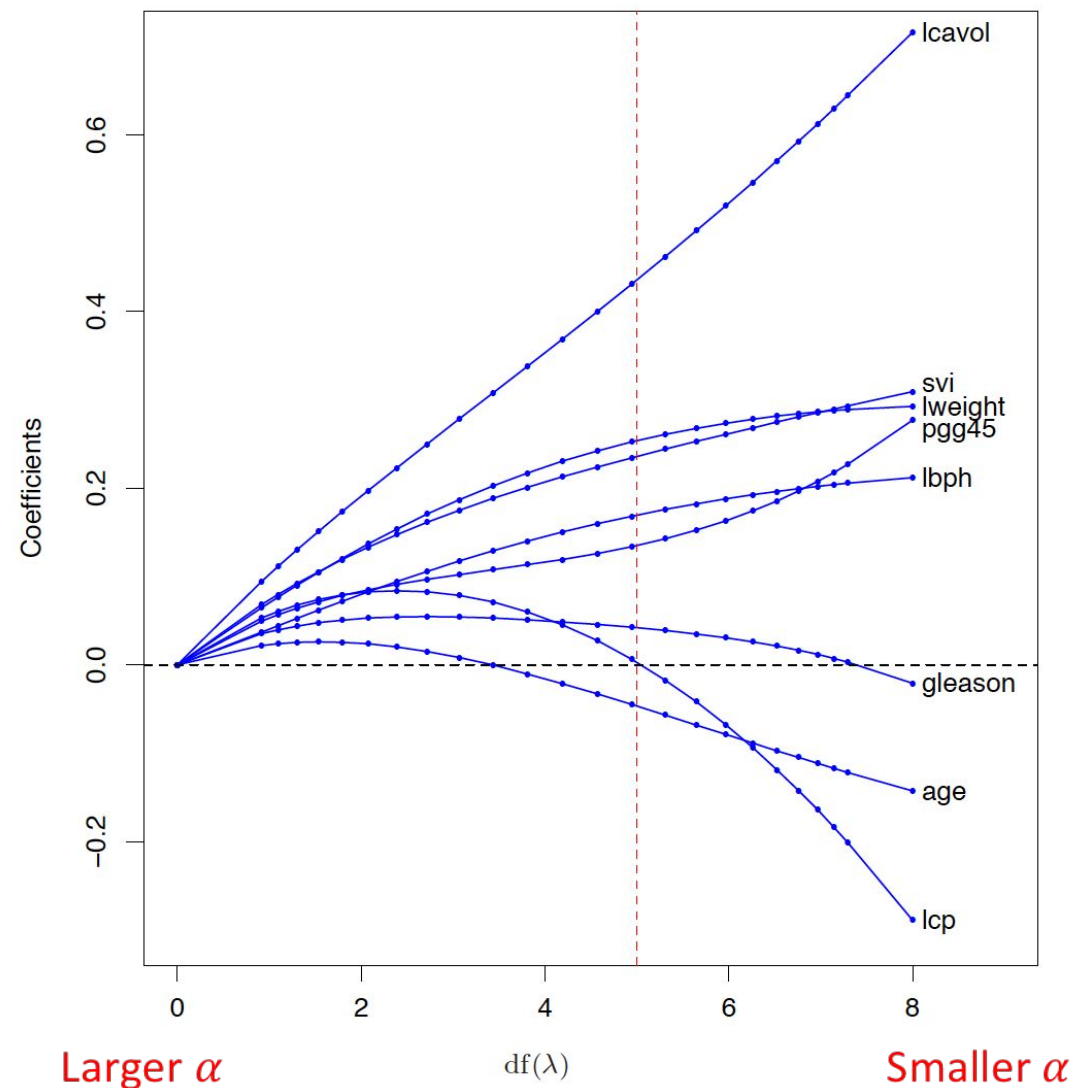
- ❑ How to determine the right regularization level  $\alpha$ ?
  - Through cross validation!
- ❑ Sklearn function for ridge regression:
  - [http://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.Ridge.html](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html)

# Coefficient path with ridge regression

Note that larger  $\alpha$  does not lead to fewer non-zero coefficients, but only smaller (and mostly positive) coefficients!

Figure from [Hastie2008]: Hastie, Tibshirani, Friedman, The elements of statistical learning.

For more on this subject, see Sec. 3.4.1.



**FIGURE 3.8.** Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter  $\lambda$  is varied. Coefficients are plotted versus  $df(\lambda)$ , the effective degrees of freedom. A vertical line is drawn at  $df = 5.0$ , the value chosen by cross-validation.

# LASSO Regression

---

□ LASSO cost function:

$$J(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^d |\beta_j| = \|\mathbf{y} - A\boldsymbol{\beta}\|^2 + \alpha \|\boldsymbol{\beta}\|_1$$

□ Because derivative of  $|\beta_j|$  is not continuous, there is no closed-form solution.

□ However, there is a unique minimum because the cost function is convex.

□ Many methods to solve iteratively

- Least angle regression (LAR), coordinate descent, ADMM
- Beyond the scope of this class
- See textbook [Hastie2008] for LAR method

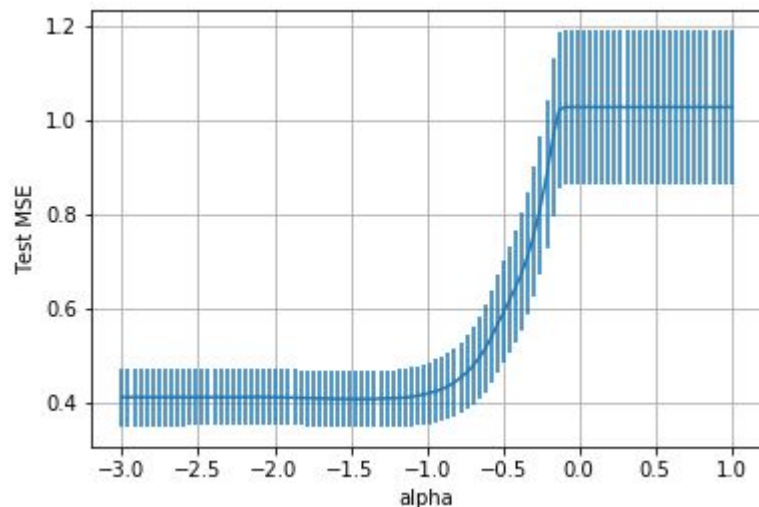
# Computing LASSO in python

## □ Use sklearn Lasso method

- Solve using coordinate descent

## □ Cross validation loop

- Outer loop: Loop over folds
- Inner loop: Loop over  $\alpha$
- Measure mean and std deviation of MSE



```
: # Create a k-fold cross validation object
nfold = 10
kf = sklearn.model_selection.KFold(n_splits=nfold,shuffle=True)

# Create the LASSO model. We use the `warm start` parameter so
# This speeds up the fitting.
model = linear_model.Lasso(warm_start=True)

# Regularization values to test
nalpha = 100
alphas = np.logspace(-3,1,nalpha)

# MSE for each alpha and fold value
mse = np.zeros((nalpha,nfold))
for ifold, ind in enumerate(kf.split(X)):

    # Get the training data in the split
    Itr,Its = ind
    X_tr = X[Itr,:]
    y_tr = y[Itr]
    X_ts = X[Its,:]
    y_ts = y[Its]

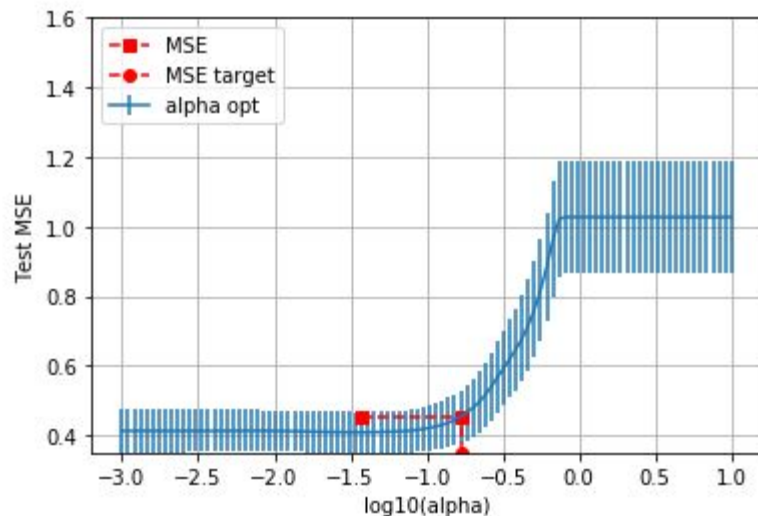
    # Compute the lasso path for the split
    for ia, a in enumerate(alphas):

        # Fit the model on the training data
        model.alpha = a
        model.fit(X_tr,y_tr)

        # Compute the prediction error on the test data
        y_ts_pred = model.predict(X_ts)
        mse[ia,ifold] = np.mean((y_ts_pred-y_ts)**2)
```

# Using One Standard Deviation Rule

- ❑ Use one standard deviation rule from before
  - Find  $\alpha_0$  with minimum mean MSE,  $\text{mean\_mean}$
  - Set  $\text{mse\_tgt} = \text{mse\_mean}[\alpha_0] + \text{mse\_se}[\alpha_0]$
  - Find largest  $\alpha$  where  $\text{mse\_mean}[\alpha] < \text{mse\_tgt}$



```
# Find the minimum MSE and MSE target
imin = np.argmin(mse_mean)
mse_tgt = mse_mean[imin] + mse_se[imin]
alpha_min = alphas[imin]

# Find the least complex model with mse_mean < mse_tgt
I = np.where(mse_mean < mse_tgt)[0]
iopt = I[-1]
alpha_opt = alphas[iopt]
print("Optimal alpha = %f" % alpha_opt)
```

# Coefficients

- ❑ Select  $\alpha$  via cross-validation
- ❑ Then, find coefficients using all training data.
- ❑ Final coefficients are sparse:
  - Only three factors are non-zeros
  - Lcavol: log cancer volume
  - Lweight: Log weight
  - Svi: seminal vesicle invasion
- ❑ Use only features corresponding to non-zero coefficients for linear regression

```
model.alpha = alpha_opt
model.fit(X,y)

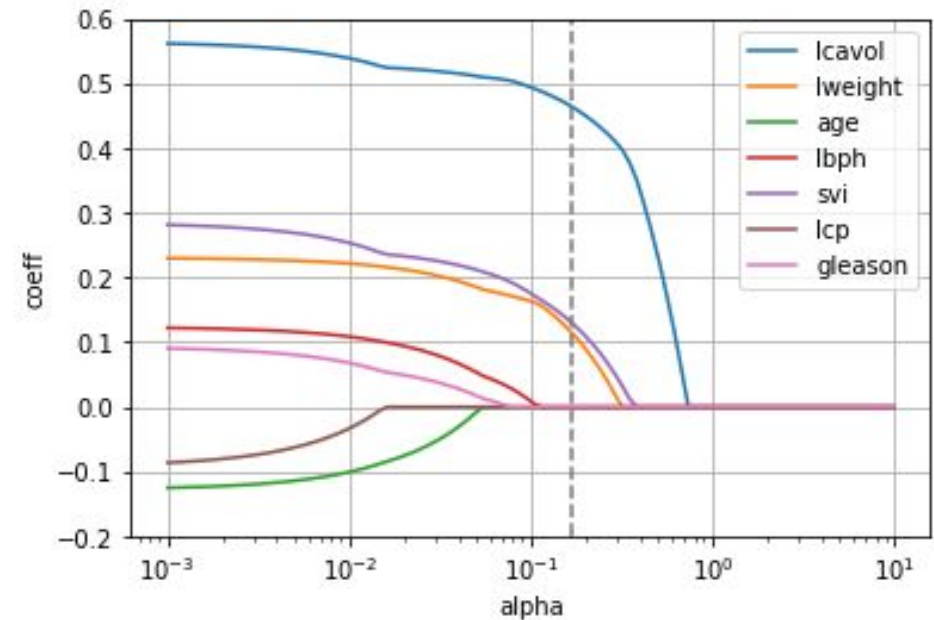
# Print the coefficients
for i, c in enumerate(model.coef_):
    print("%8s %f" % (names_x[i], c))
```

```
lcavol 0.464526
lweight 0.115832
age 0.000000
lbph 0.000000
svi 0.131102
lcp 0.000000
gleason 0.000000
```




# LASSO path

- Useful to plot coefficients as a function of  $\alpha$ .
- Called the LASSO path
- Indicates relative importance of different factors
- For this data set:
  - lcavol most important
- Don't draw medical conclusions
  - Need more detailed significance testing
  - Complex subject for another class...



# Outline

---

- ☐ Motivating Example: Predicting prostate cancer from a PSA test
- ☐ Model Selection
- ☐ Model Selection from LASSO regularization
-  ☐ Other Model Selection Methods
- ☐ In-Class Exercise: Audio Pitch Detection



# Filtering method

---

- ❑ Rank the features based on their correlation with the target
  - Can use other metrics: Correlation, F-test, mutual information, ...
- ❑ Also should consider the redundancy (correlation) among chosen features
  - **Minimal Redundancy Maximum Relevance (mRMR)**
  - Peng, H.C., Long, F., and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp. 1226–1238, 2005.
  - <http://home.penglab.com/proj/mRMR/>
  - <https://www.mathworks.com/matlabcentral/fileexchange/14916-minimum-redundancy-maximum-relevance-feature-selection>

# Ranking metrics

---

❑ Correlation coefficient between a feature and the target

❑ F-test: test the significance of using one feature vs. not using any (use the mean of y only). Essentially measure the difference in the MSE when using only the mean value of y vs. using a single feature.

$$f_{test} = \frac{r^2}{1-r^2}(n_{sample}-2)$$

❑ Mutual information between a feature and the target

$$I(X,Y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dx dy$$


# What You Should Know to Do

---

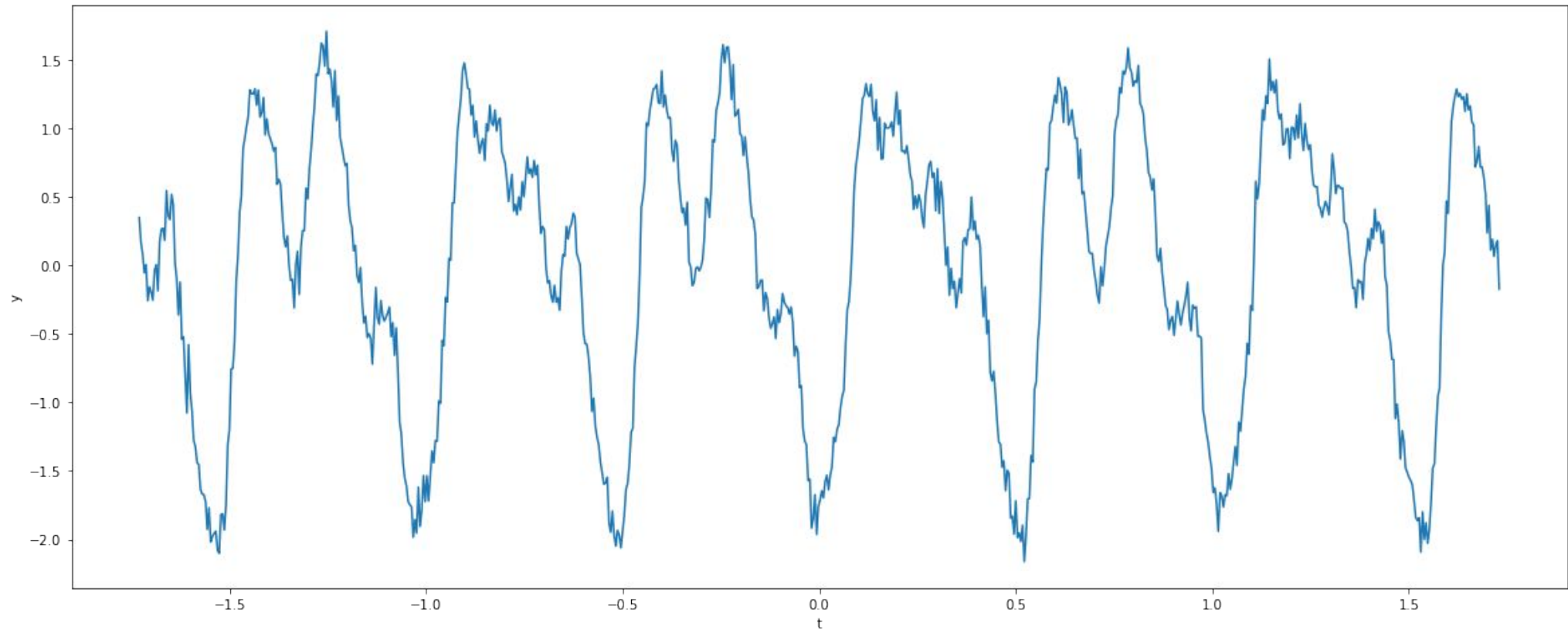
- ☐ Formulate a linear estimation problem with a regularization
- ☐ Compute an L1-regularized estimate (LASSO) using sklearn tools
- ☐ Compute the optimal regularization level using cross validation
- ☐ Interpret results from a LASSO path
- ☐ Determine final regression function from cross validation
- ☐ Perform other feature selection methods

# Outline

---

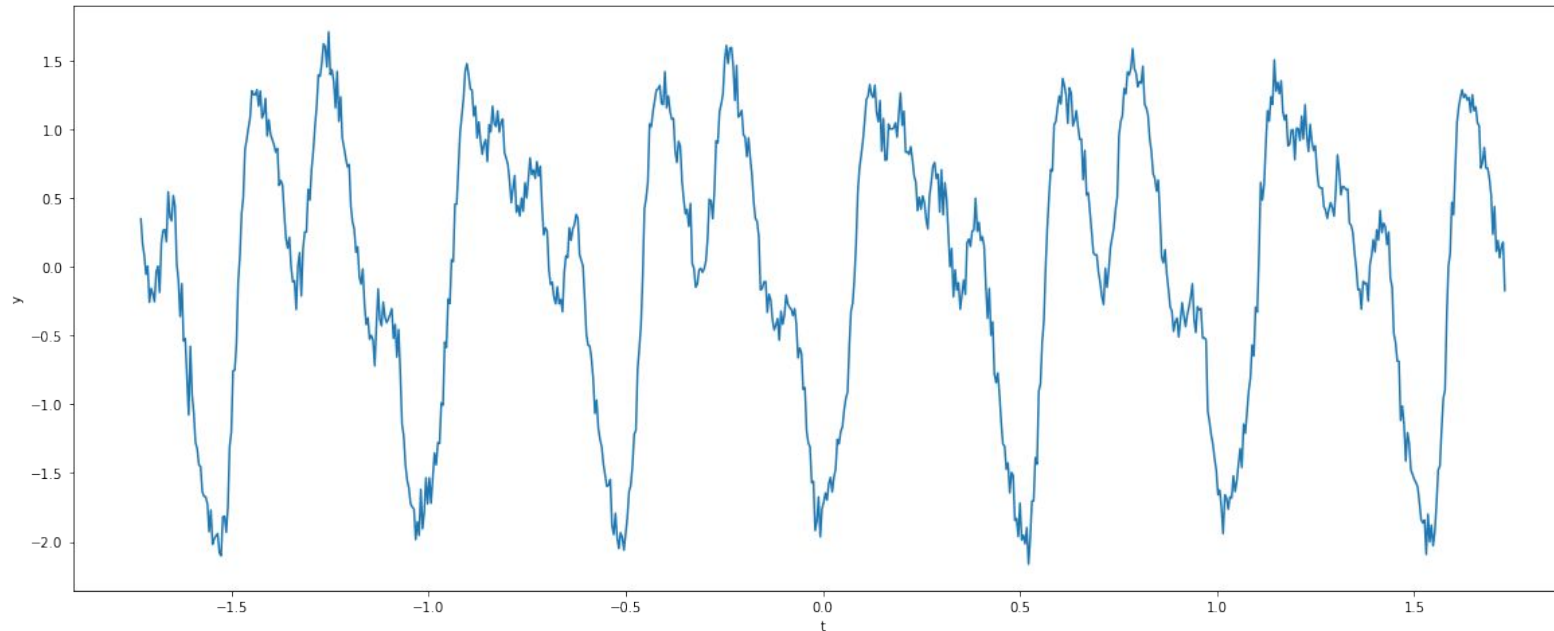
- ☐ Motivating Example: Predicting prostate cancer from a PSA test
- ☐ Model Selection
- ☐ Model Selection from LASSO regularization
- ☐ Other Model Selection Methods
-  ☐ In-Class Exercise: Audio Pitch Detection

# In-class Exercise: Pitch Detection



# In-class Exercise: Pitch Detection

- Pitch is a perceptual property of sounds that allows their ordering on a frequency-related scale. Pitch may be quantified as a **frequency**.
- The task: given a sound wave  $(t, y)$ , determine the **dominating pitch**.



# In-class Exercise: Pitch Detection

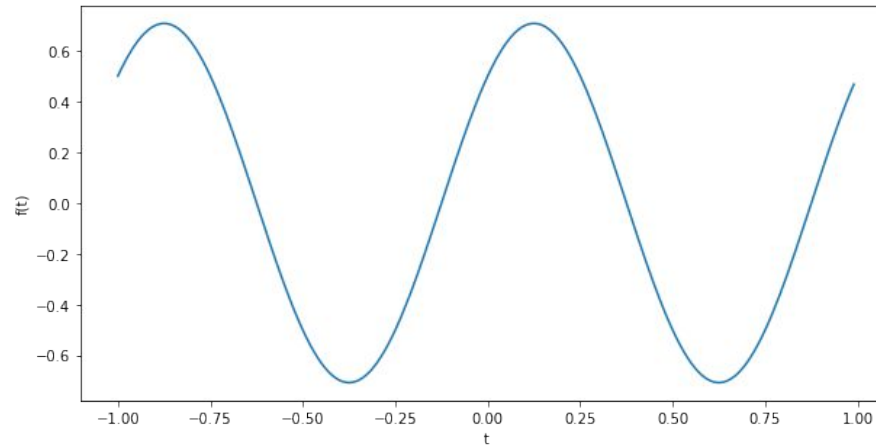
---

- ❑ A **sinusoid** is a mathematical curve that describes a smooth periodic oscillation.
- ❑ For this lab, let's use the following equations to represent sinusoid:

$$a \sin(2\pi \cdot t \cdot f) + b \cos(2\pi \cdot t \cdot f)$$

- ❑ For example, when  $a = b = 0.5$ ,  $f=10$ , the corresponding sinusoid looks like this:

$$f(t) = 0.5 \sin(2\pi t \cdot 10) + 0.5 \cos(2\pi t \cdot 10)$$



# In-class Exercise: Pitch Detection

---

- Waves can be represented as the sum of a series of sinusoids.
- This fact can let us transform our task into a regression problem:

Given a sound wave,  $S = \{(y_i, t_i) | i \in \{1, \dots, n\}\}$ , and a set of frequencies

$F = \{f_1, f_2, \dots, f_d | f_i \in \mathbb{R}\}$ , we would like to find an optimal set of parameters  $a^* \in \mathbb{R}^d, b^* \in \mathbb{R}^d$  that minimizes following loss functions:

$$l(f(t), y) = \frac{1}{n} \sum_{i=1}^n (f(t_i) - y_i)^2$$

where

$$f(t) = \sum_{j=1}^d a_j \sin(2\pi t f_j) + b_j \cos(2\pi t f_j)$$



# In-class Exercise: Pitch Detection

---

- ❑ Our plan:
- ❑ Step 1: Use sinusoids represent pitches.
- ❑ Step 2: A sound wave can be seen as a **linear combination of multiple sinusoids**, each having a unique frequency.
- ❑ Step 3: We can apply **Lasso regression** to find the **optimal weights (a and b)** associated with each sinusoid.
- ❑ Step 4: For those sinusoids whose weights are large, their corresponding frequencies will be the **dominating pitches**.