VIRGINIA TECH

Linear Regression

Lifu Huang

Computer Science, Virginia Tech

Feb. 04, 2021

Slides adapted from Luke Zettlemoyer, David Sontag, Bert Huang

Recap - The Naïve Bayes Classifier

- Given:
  - Prior $P(Y)$
  - n conditionally independent features $\boldsymbol{X}$ given the class Y
  - For each X_i, we have likelihood $P(X_i|Y)$
- Decision Rule:

$$P(Y|X) \propto P(X,Y) = P(X|Y)P(Y)$$

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

$$y^* = h_{NB}(\mathbf{x}) = \arg\max_y P(y)P(x_1, \ldots, x_n \mid y)$$
$$= \arg\max_y P(y)\prod_i P(x_i|y)$$

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

Recap - NB for Text Classification

- Bag of Words Model
  - Article at least 1000 words, $\boldsymbol{X} = \{X_1, X_2, \ldots, X_{1000}\}$
  - X_i represents the i^{th} word in document, i.e., the domain of X_i is entire vocabulary
- NB assumption helps a lot!!
  - $P(X_i = x_i | Y = y)$ is just the probability of observing word x_i in a document on topic y

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Recap - What if we have continuous X_i

- Split the continuous variable into several intervals
  - e.g. $X \sim [0, 1] \rightarrow [0, 0.05], (0.05, 0.2], (0.2, 1]$
- Gaussian Naïve Bayes
  - Each feature has a Gaussian distribution given a class

Gaussian Naïve Bayes (GNB):

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \quad e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

mean

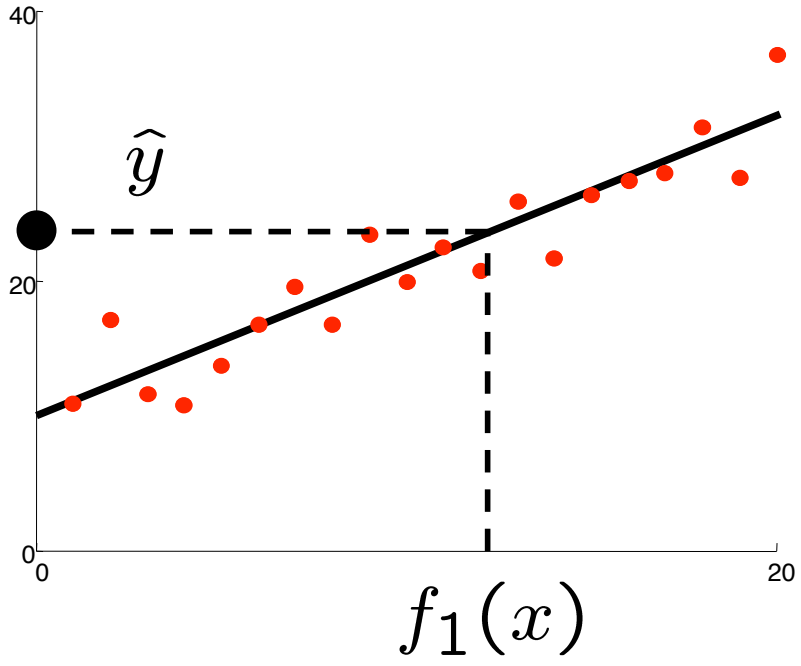
standard deviation

What's regression?

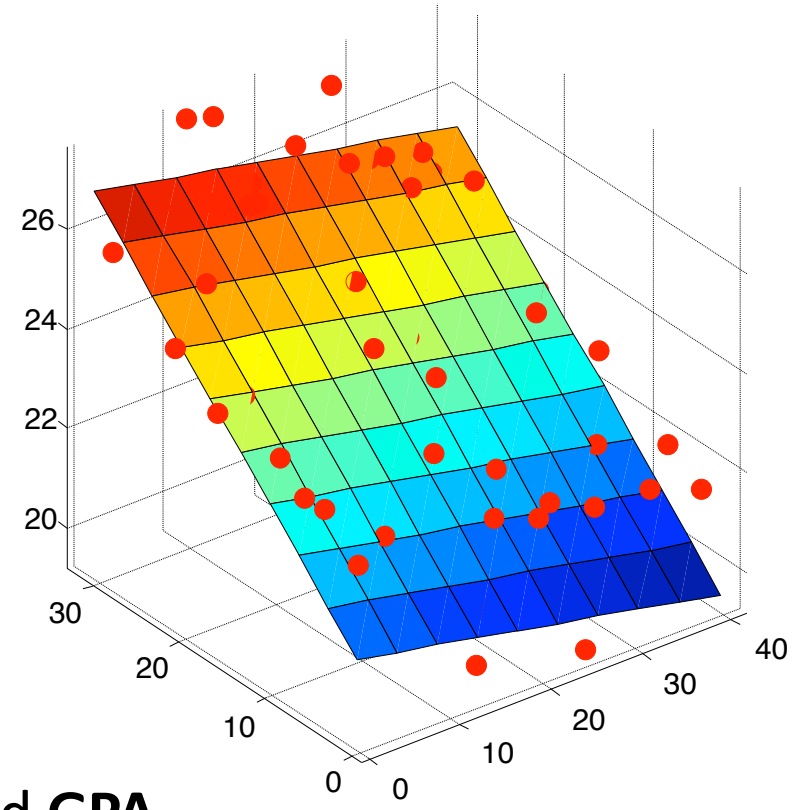
- Regression: a statistical method to determine the relationship between one dependent variable (Y) and a series of other independent variables (X_1, X_2, …, X_n)
  - Y is a continuous variable
- Example
  - Predict the salary based on your major and GPA
  - Predict the food price based on the location and season
  - Predict the stock price based on the market
  - Predict the temperature based on cloud pictures
  - …

Linear Regression

$$\hat{y} = \hat{f}(X) = w_0 + w_1 h_1(X)$$



$$\hat{y} = \hat{f}(X) = w_0 + w_1 h_1(X) + w_2 h_2(X)$$



e.g., Predict the salary based on your **major** and **GPA**

Linear Regression

- Generalized Denotation

$$\hat{y} = \hat{f}(X) = w_0 + w_1 h_1(X) + w_2 h_2(X) + \cdots + w_M h_M(X)$$

$$= w_0 + \sum_{k=1}^{K} w_k h_k(X)$$

bias

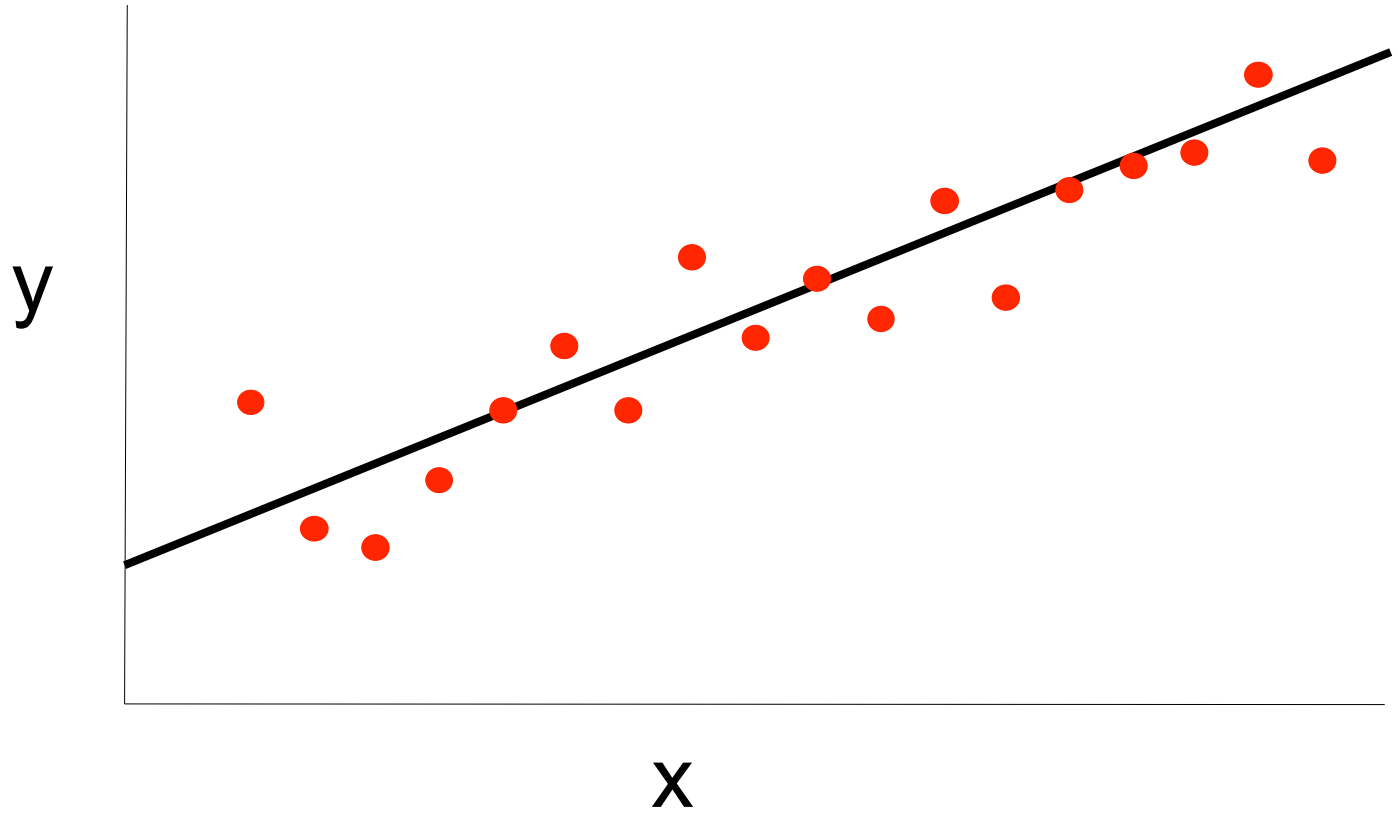
basis functions or feature functions

Assume: a dummy basic function $h_0(X) = 1$

$$\hat{y} = \hat{f}(X) = \sum_{k=0}^{K} w_k h_k(X)$$

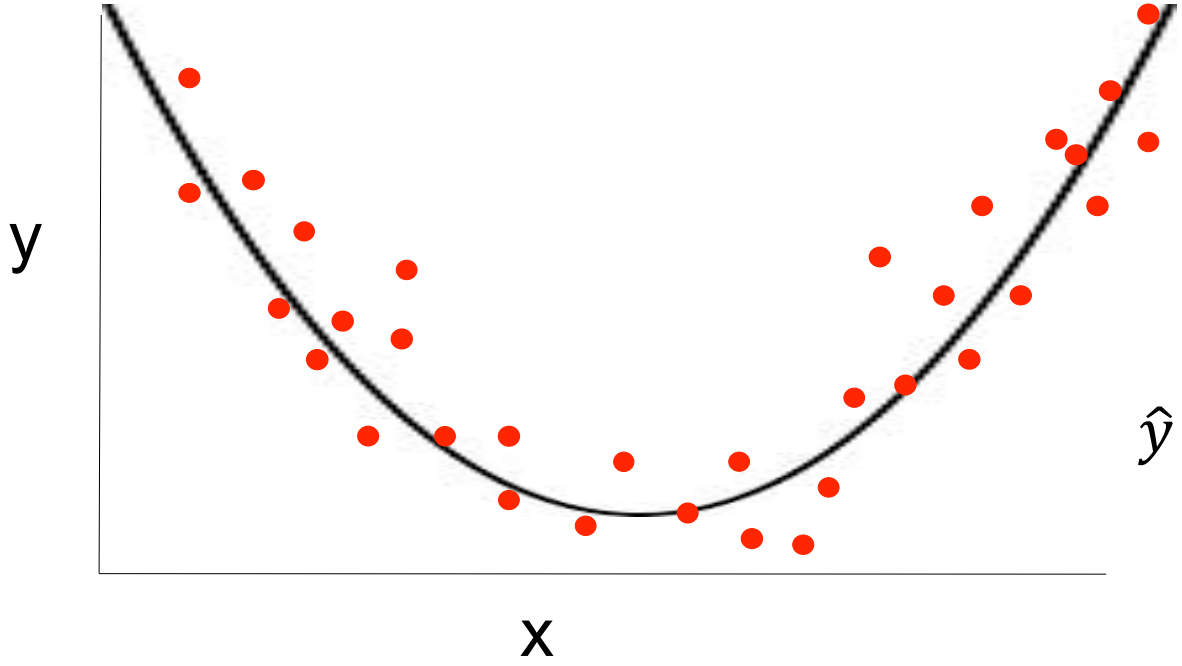
Linear Basis: 1D input

Need a bias term: $\{h_1(x) = x, h_2(x)=1\}$



Linear Basis

- Parabola: $\{h_1(x) = x^2, h_2(x)=x, h_3(x)=1\}$

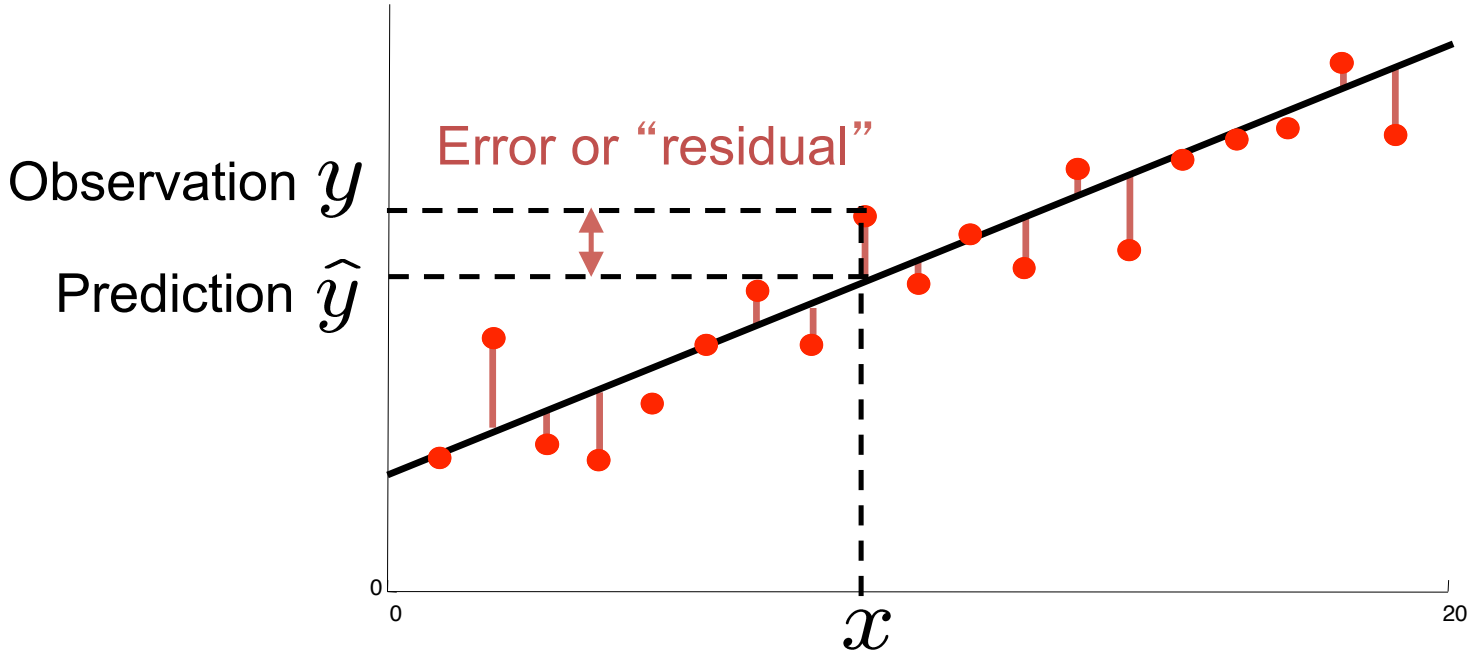


$$\hat{y} = \hat{f}(X) = w_0 + w_1 h_1(X) + w_2 h_2(X)$$

- 2D: $\{h_1(\mathbf{x}) = x_1^2, h_2(\mathbf{x})= x_2^2, h_3(\mathbf{x})=x_1x_2,\ldots\}$
- Can define any basis functions $h_i(\mathbf{x})$ for n-dimensional input $\mathbf{x}=\langle x_1,\ldots,x_n\rangle$

Ordinary Least Squares (OLS)

$$total\ error = \sum_i (y_i - \hat{y}_i)^2 = \sum_i (y_i - \sum_k w_k h_k(x_i))^2$$



The regression problem

- Instances: $< X_j, t_j >$
- Learn: Mapping from X to $t(X)$
- Hypothesis space:
  - Given basic functions $\{h_1, h_2, \ldots, h_k\}$
  - $h_i(X) \in R$
  - define coefficients $W = \{w_1, w_2, \ldots, w_k\}$
  - linear regression: model is linear in the parameters
- Minimize the residual squared error:

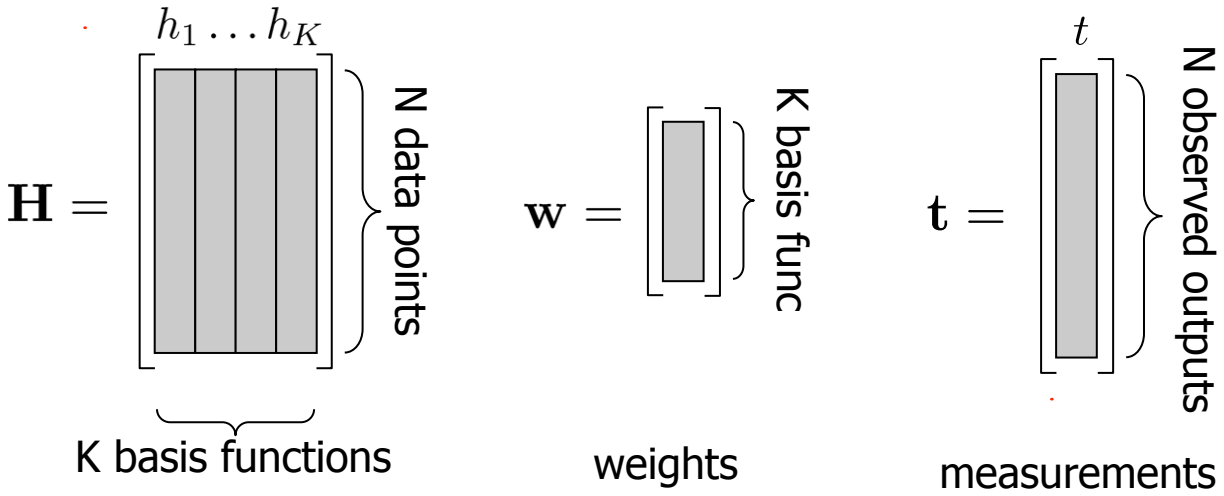
$$\underbrace{t(\mathbf{x})}_{\text{data}} \approx \widehat{f}(\mathbf{x}) = \sum_i w_i h_i(\mathbf{x})$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

Regression: matrix notation

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \underbrace{(\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})}_{\text{residual error}}$$



Regression: closed form solution

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} (\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})$$

$$\mathbf{F}(\mathbf{w}) = (\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})$$

$$\nabla_{\mathbf{w}} \mathbf{F}(\mathbf{w}) = \mathbf{0}$$

$$2\mathbf{H}^T(\mathbf{Hw} - \mathbf{t}) = \mathbf{0}$$

$$\mathbf{H}^T\mathbf{Hw} - \mathbf{H}^T\mathbf{t} = \mathbf{0}$$

$$\mathbf{w}^* = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{t}$$

$$\frac{\partial X^T X}{\partial X} = 2X, \qquad \frac{\partial \beta^T X}{\partial X} = \beta$$

Normal Equations for the least squares problem

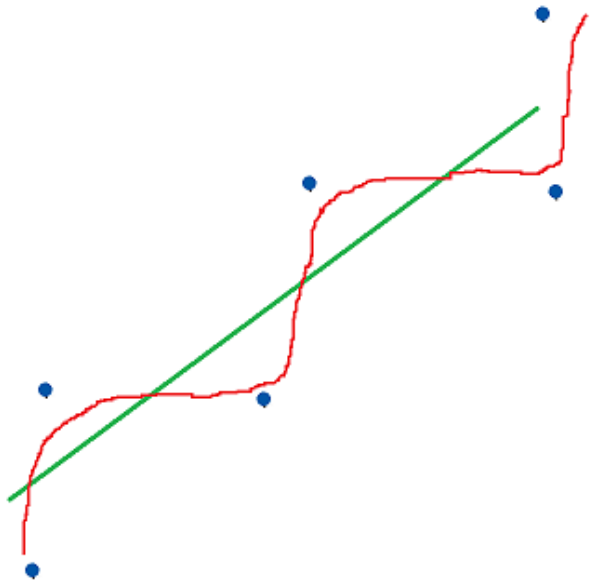
$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \underbrace{(\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})}_{\text{residual error}}$$

$$\text{solution: } \mathbf{w}^* = \underbrace{(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}}_{\mathbf{A}^{-1}} \underbrace{\mathbf{H}^{\mathrm{T}}\mathbf{t}}_{\mathbf{b}} = \mathbf{A}^{-1}\mathbf{b}$$

$$\text{where } \mathbf{A} = \mathbf{H}^{\mathrm{T}}\mathbf{H} = \underbrace{\left[\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline\end{array}\right]}_{\text{k×k matrix for k basis functions}} \quad \mathbf{b} = \mathbf{H}^{\mathrm{T}}\mathbf{t} = \underbrace{\left[\begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \\ \hline\end{array}\right]}_{\text{k×1 vector}}$$

Overfitting

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$



Green: true relationship between variables
Red: overfit model

- When the model is too complex, e.g., larger set of parameters, large parameter values, the model will be overfitted to the training samples, and perform poorly on new and unseen test instances
- Regularized or penalized regression: modify learning objective to penalize large parameters

Regularization

- Regularization: adding constraint on the coefficients *W*
- LASSO (Least Absolute Shrinkage and Selection Operator)
  - L1 norm: add the sum of magnitudes of the coefficients

$$\hat{w}_{LASSO} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

- Ridge Regression
  - L2 norm: add a squared penalize for large weights
  - Explicitly writing out bias feature ($h_0 = 1$), which is not penalized

$$\hat{w}_{ridge} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

λ is hyperparameter that balances tradeoffs

Ridge Regression: matrix notation

$$\hat{w}_{ridge} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

$$= \arg\min_{\mathbf{w}} \underbrace{(\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})}_{\text{residual error}} + \lambda \, \mathbf{w}^T I_{0+k} \mathbf{w}$$

$$\mathbf{H} = \begin{bmatrix} 1 & h_1 \ldots h_K \\ \cdot & \\ \cdot & \\ \cdot & \\ 1 & \end{bmatrix}$$

N data points

bias column and k basis functions

$$\mathbf{w} = \begin{bmatrix} \\ \\ \end{bmatrix}$$

k basis functs plus bias

weights

$$\mathbf{t} = \begin{bmatrix} t \\ \\ \end{bmatrix}$$

N observed outputs

measurements

$$I_{0+k} = \begin{bmatrix} 0 & 0 & 0 & \ldots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \ldots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

k+1 (columns) and k+1 (rows)

k+1 x k+1 identity matrix, but with 0 in upper left

Ridge Regression: Closed form solution

$$\hat{w}_{ridge} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

$$= \arg\min_{\mathbf{w}} \underbrace{(\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})}_{\text{residual error}} + \lambda \, \mathbf{w}^T I_{0+k} \mathbf{w}$$

$$\mathbf{F}(\mathbf{w}) = (\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t}) + \lambda \mathbf{w}^T I_{0+\mathrm{k}} \mathbf{w}$$

$$\nabla_{\mathbf{w}} \mathbf{F}(\mathbf{w}) = \mathbf{0}$$

$$\mathbf{2H}^T (\mathbf{Hw} - \mathbf{t}) + 2\lambda I_{0+\mathrm{k}} \mathbf{w} = \mathbf{0}$$

$$w^*_{ridge} = (\mathbf{H}^T \mathbf{H} + \lambda I_{0+\mathrm{k}})^{-1} \mathbf{H}^T \mathbf{t}$$

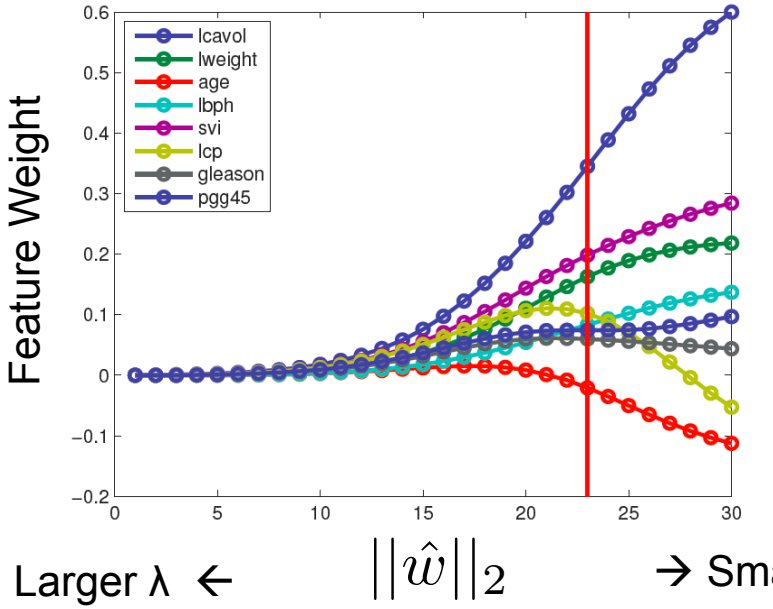
Compare to un-regularized regression:

$$\mathbf{w}^* = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{t}$$

Ridge Regression

- How does varying λ change W?
  - Larger λ or Smaller λ?
  - As $\lambda \to 0$?
    - Becomes the same as unregularized
  - As $\lambda \to \infty$?
    - All weights will be 0!
- How to pick λ?
  - Tune it on help-out set
  - If no held-out set, then k-fold cross validation

Ridge Coefficient Path



From Kevin Murphy textbook

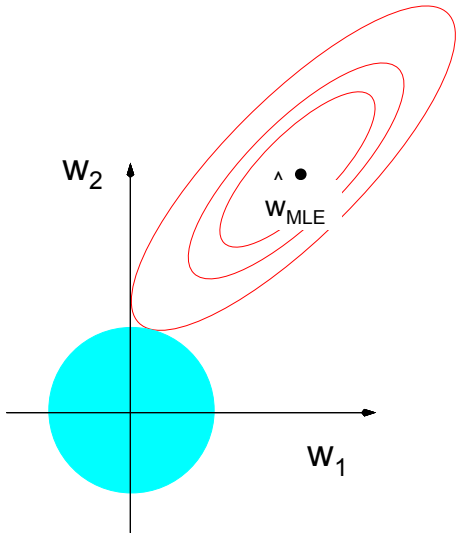
From Kevin Murphy's textbook

$$\hat{w}_{ridge} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

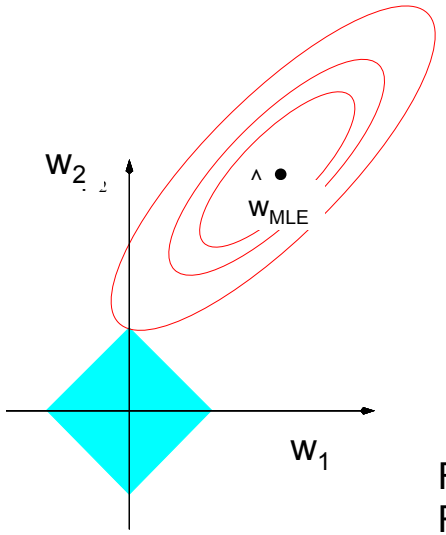
LASSO v.s. Ridge

LASSO: $$\hat{w}_{LASSO} = \arg\min_w \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

Ridge: $$\hat{w}_{ridge} = \arg\min_w \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$



Ridge Regression



Lasso

Geometric Intuition (when K=2):

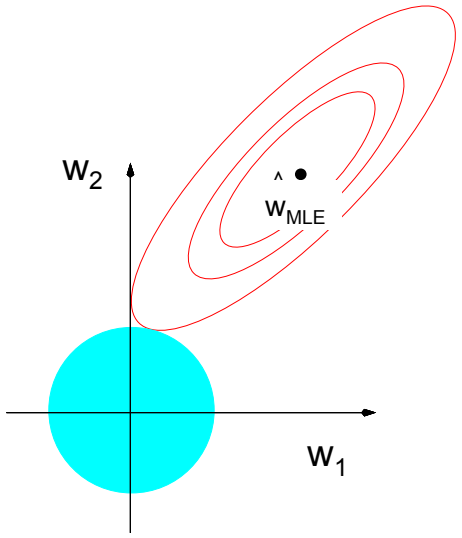
- ***Ellipse plot***: sum of square error term
- ***Circle shape***: constraint region of Ridge
- ***Diamond shape***: constraint region of LASSO
- ***Common points***: optimal points

From Rob Tibshirani slides

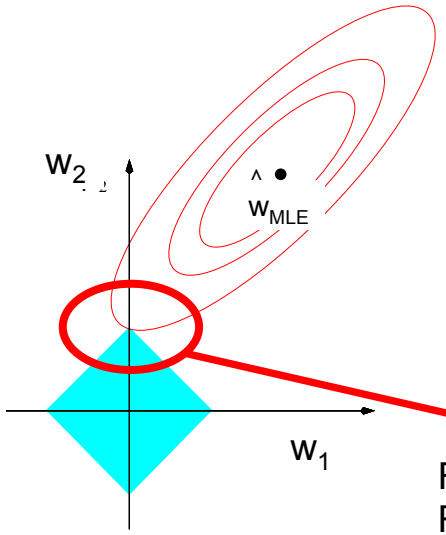
LASSO v.s. Ridge

LASSO: $$\hat{w}_{LASSO} = \arg\min_w \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

Ridge: $$\hat{w}_{ridge} = \arg\min_w \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$



Ridge Regression



Lasso

Geometric Intuition (when K=2):

- ***Ellipse plot***: sum of square error term
- ***Circle shape***: constraint region of Ridge
- ***Diamond shape***: constraint region of LASSO
- ***Common points***: optimal points

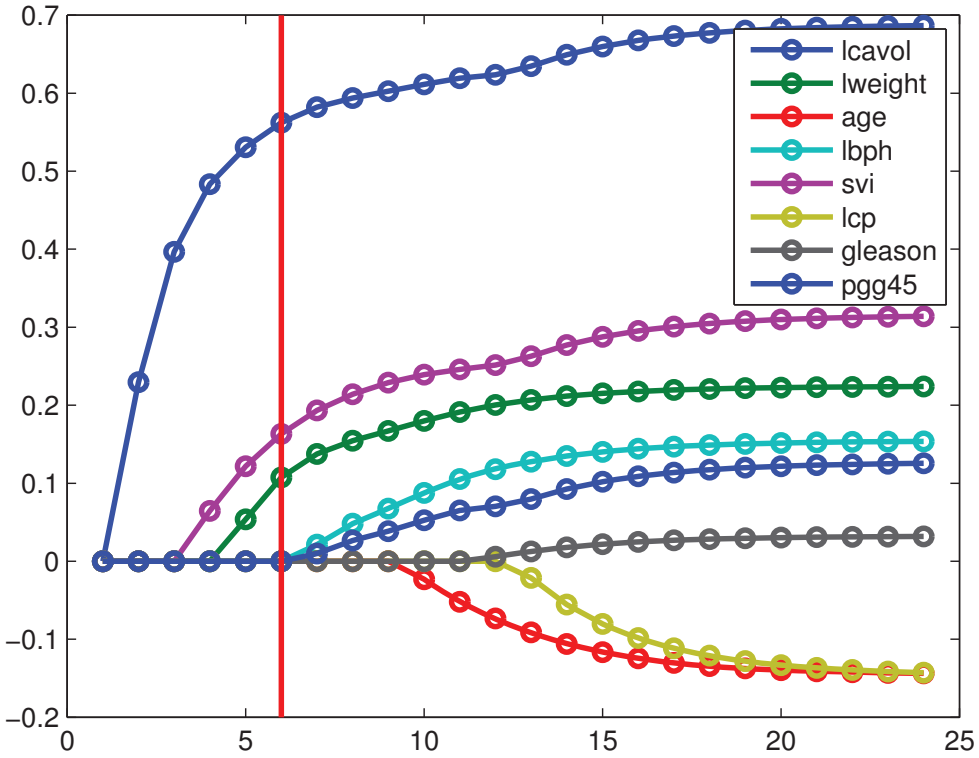
From Rob Tibshirani slides

$w_1 = 0$

LASSO

- Linear penalty pushes more weights to zero
- Allows for a type of feature selection
- Not differentiable and no closed form solution

LASSO Coefficient Path



From Kevin Murphy's textbook

Summary: Linear Regression

- Mainly be used for regression problems
- Predicting the continuous dependent variable with the help of independent variables and a deterministic function
- Goal: find the best fit line that can actually predict the output continuous dependent variable
- By finding the best fit line, algorithm establish the relationship between output dependent variable and input independent variables
- The relationship should be of linear nature
- Least square estimation to measure the accuracy
- Regularization:
  - Ridge, LASSO, How to set lambda