Data Mining

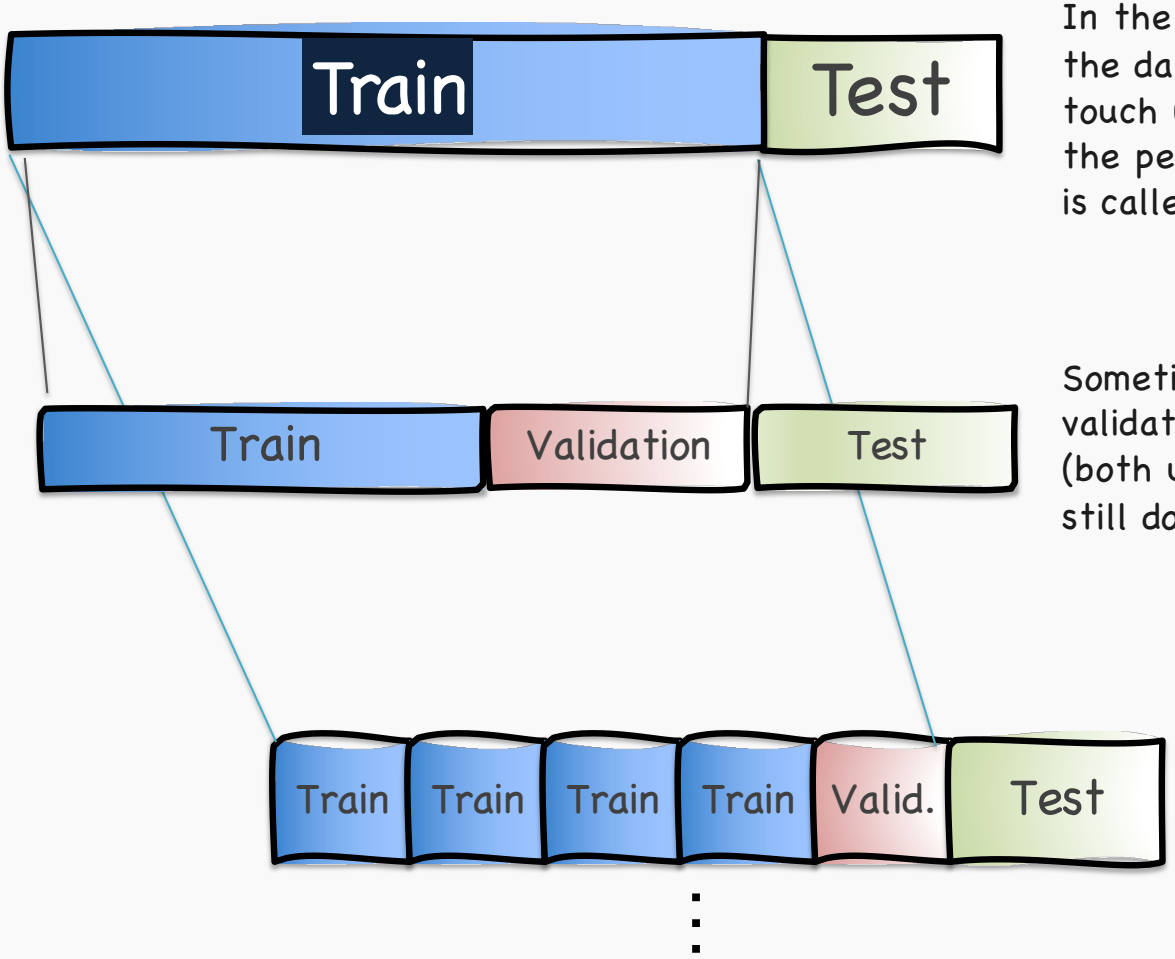
OLS, Lasso and Ridge Regularization

Adapted from slides by Pavlos Protopapas, Natesh Pillai

Part 1: Generalization Error and Bias Variance Tradeoff

Outline

- Generalization Error, Bias Variance Tradeoff
- Regularization
  - Lasso and Ridge
  - Geometric understanding of Lasso and Ridge



In the beginning, we always separate a portion of the data from the main dataset, which we never touch until the very end when we want to evaluate the performance of the final model. Normally, this is called train + test split. *

Sometimes we can further split train data into train + validation, essentially ending up with train + validation (both used to find the best model) **+ test** (which we still don't use until the very end).

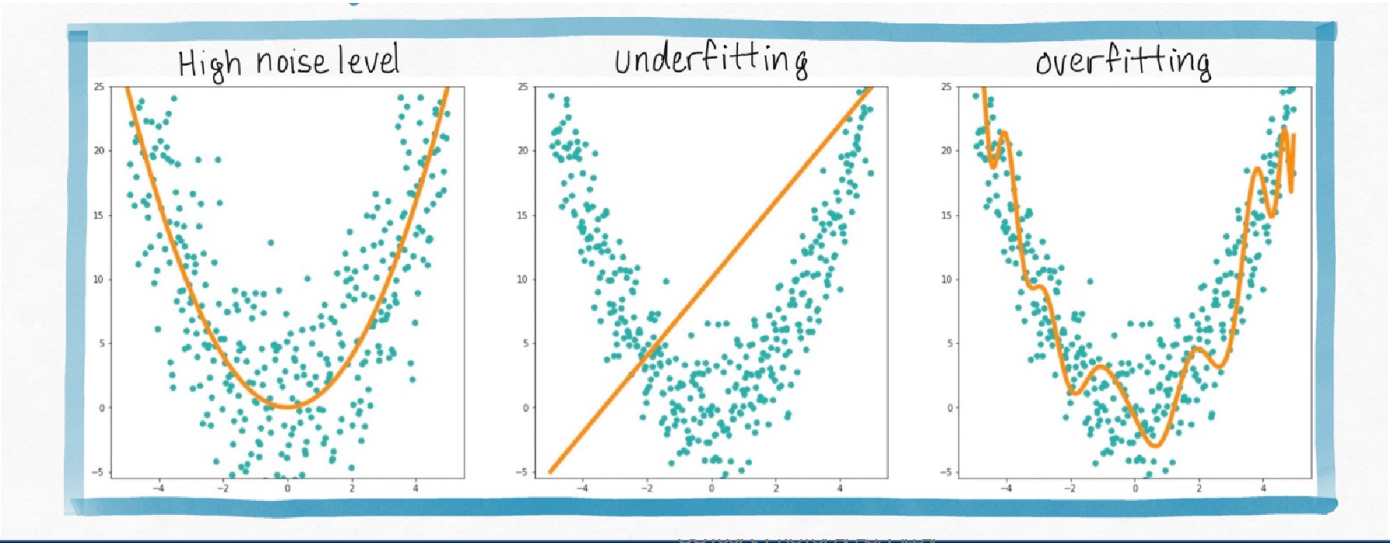
And then, we sometimes also use cross-validation, which has nothing to do with either test or validation splits? Because cross-validation uses the train data to split it into k buckets.

Test Error and Generalization

We know to evaluate models on both train and test data because models can do well on training data but do poorly on new data.

When models do well on new data is called generalization.

There are at least three ways a model can have a high test error.



Irreducible and Reducible Errors

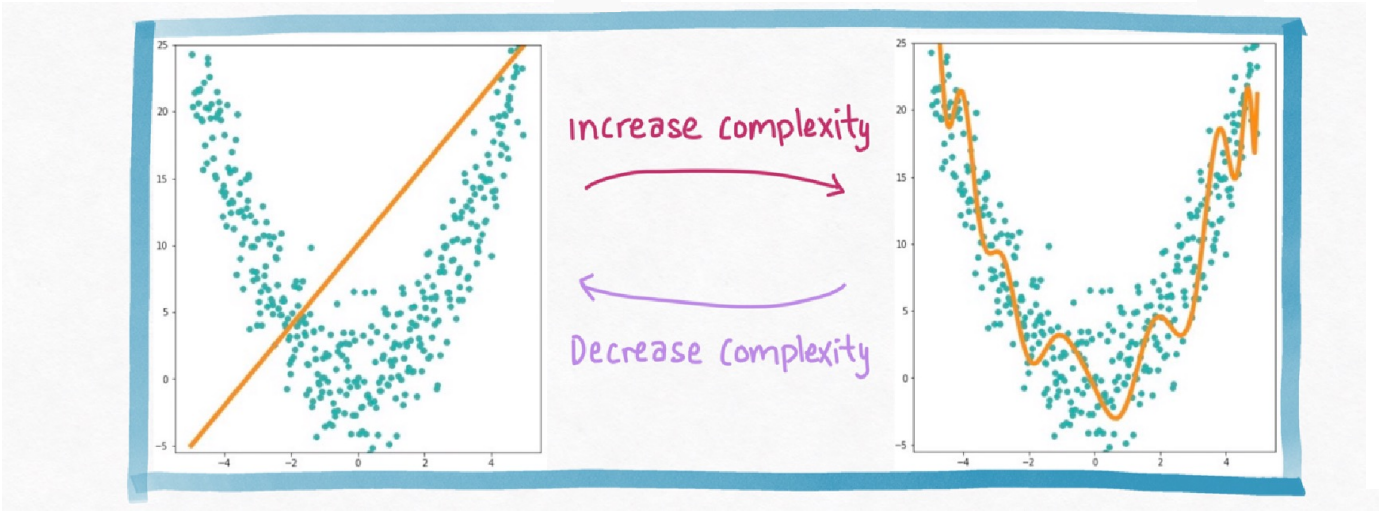
- We distinguished the contributions of noise to the generalization error:

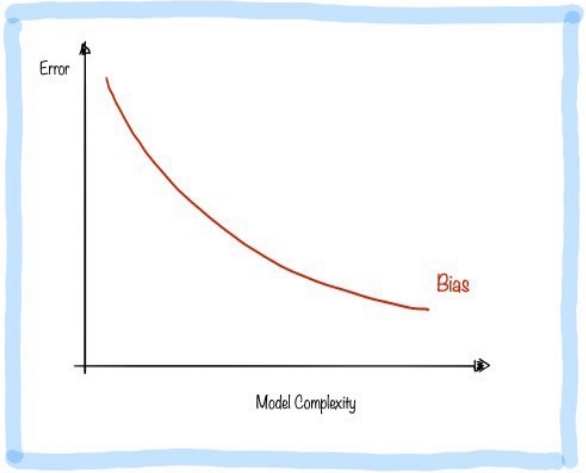
- Irreducible error (or aleatoric error) : we can't do anything to decrease error due to noise.

- Reducible error (or epistemic error): we can decrease error due to overfitting and underfitting by improving the model.

The Bias-Variance: Bias

- Reducible error comes from either underfitting or overfitting. There is a trade-off between the two sources of errors:





Bias

- Given training data D: $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Subsets of D: D_1, D_2, … D_k, are derived to train the model during N-fold training. The model trained on each D_i: $\hat{f}_D$

  If D_1: you get model $\hat{f}_{D_1}$

  If D_2: you get model $\hat{f}_{D_2}$

  If D_3: you get model $\hat{f}_{D_3}$

- Apply these models to predict input x: $\hat{f}_{D_1}(x), \hat{f}_{D_2}(x), \hat{f}_{D_3}(x), \ldots$
- The *expected* prediction on x with all the models: $\mathbb{E}_D[\hat{f}_D(x)]$
  - What does my learning algorithm *typically* predict at x? (on average across all possible datasets)

Bias

- Bias is the expected prediction of the model compared to the truth: $Bias(x) = \mathbb{E}_D[\hat{f}_D(x)] - f(x)$

What will be Bias if the true value f(0.5)=1?

Suppose at $x = 0.5$:

After training on many datasets you get:

Dataset	Prediction
D1	0.48
D2	0.52
D3	0.46
D4	0.50

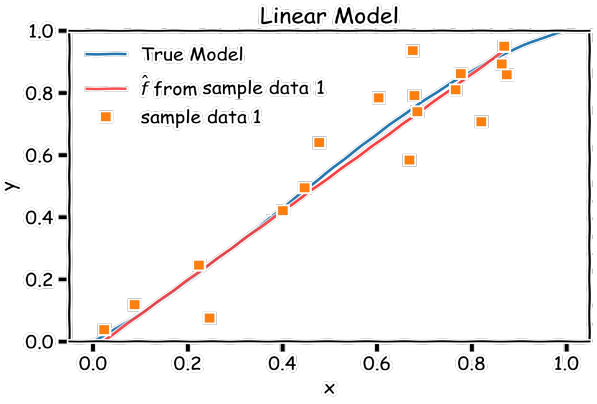
Then:

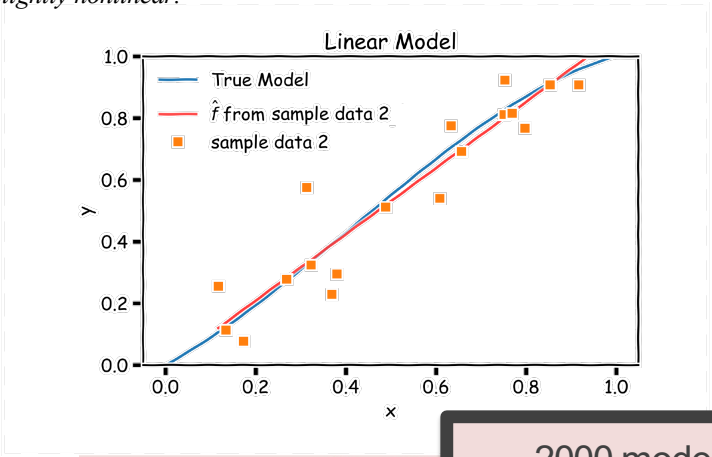
$$\mathbb{E}_D[\hat{f}_D(0.5)] \approx 0.49$$

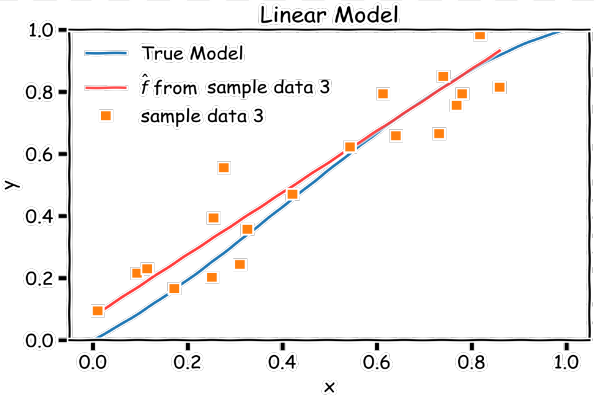
- *How far predictions are from truth* → bias

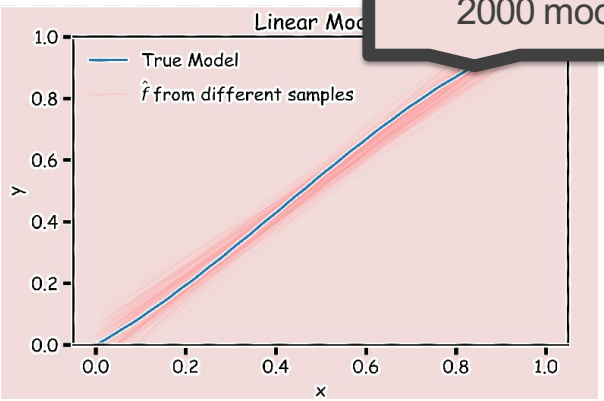
Bias vs Variance: Variance of a SIMPLE model

Note: the true function is slightly nonlinear.



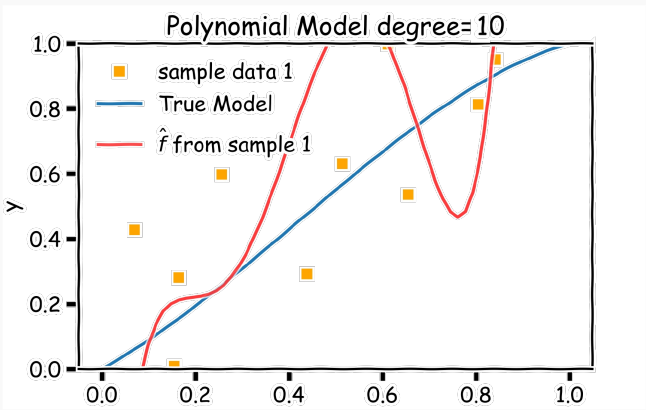


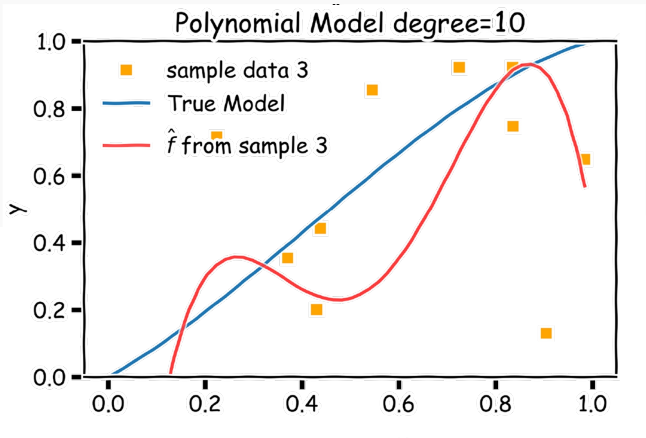


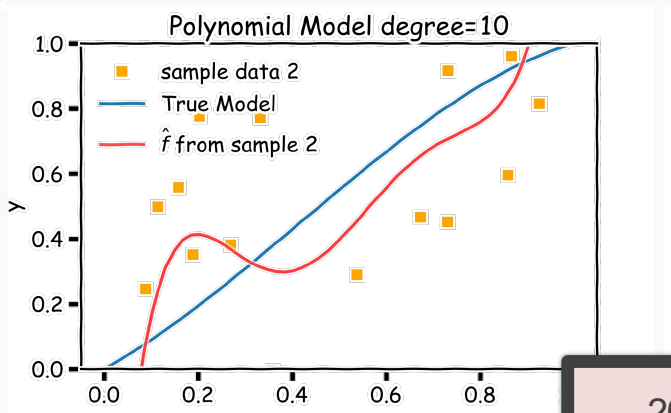


2000 models

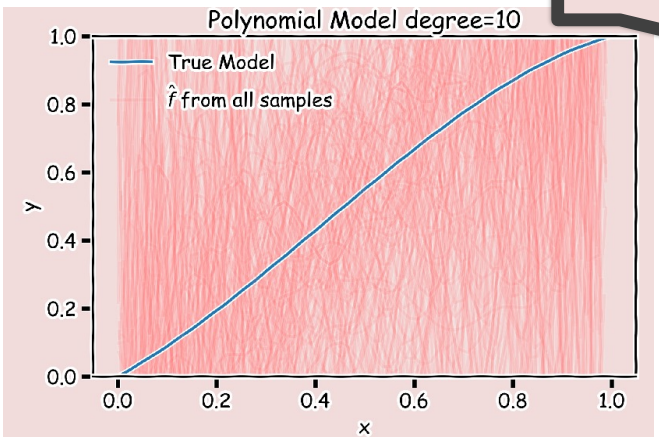
Bias vs Variance: Variance of a COMPLEX model







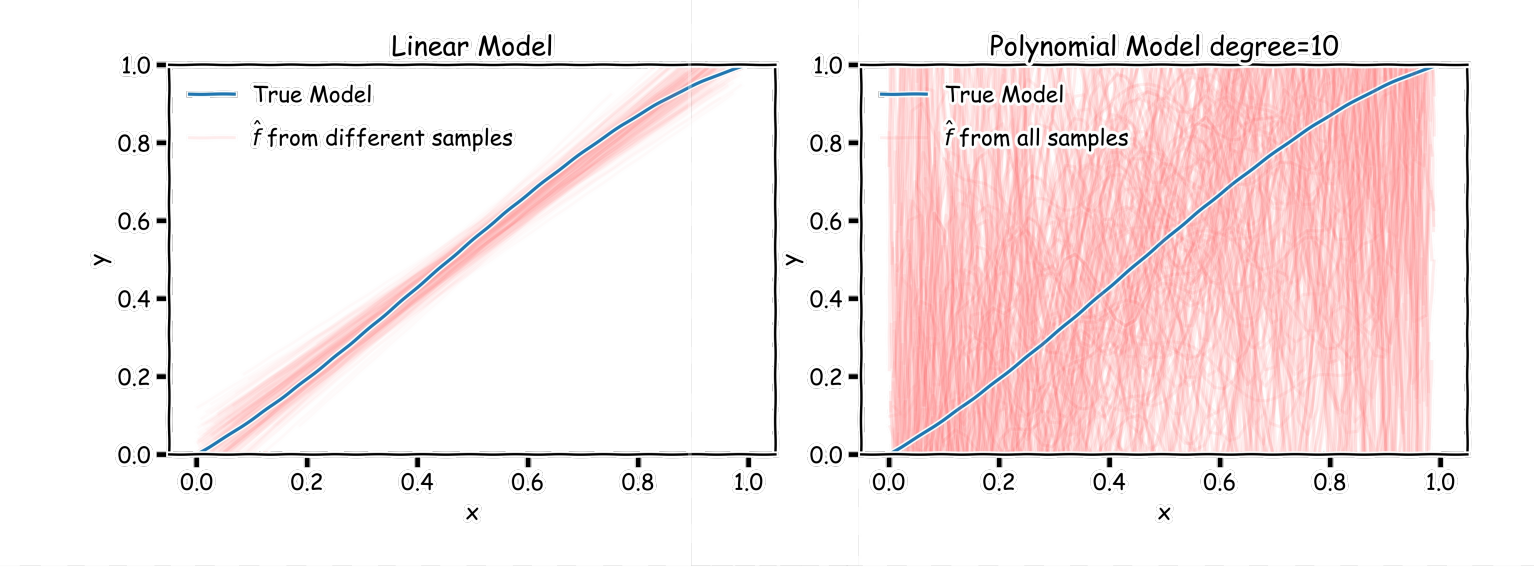
2000 models



Bias vs Variance

Left: 2000 best fit linear models, each fitted on a different 20-points training set.

Right: 2000 best fit models using degree 10 polynomials.



Variance

- Variance computes the expectation of the squared difference of the prediction from individual models learn vs the expected prediction of the models learned ($\mathbb{E}_D[\hat{f}_D(x)]$):

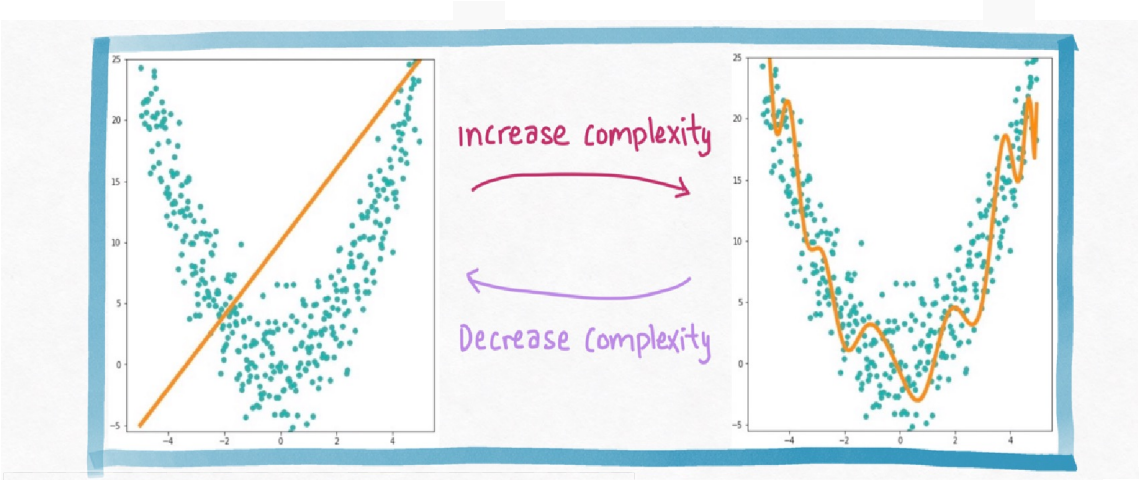
$$Var(x) = \mathbb{E}_D\left[(\hat{f}_{D_i}(x) - \mathbb{E}_D[\hat{f}_D(x)])^2\right]$$

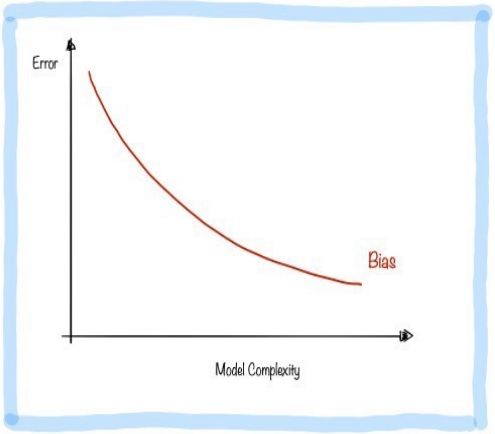
- Example continued:

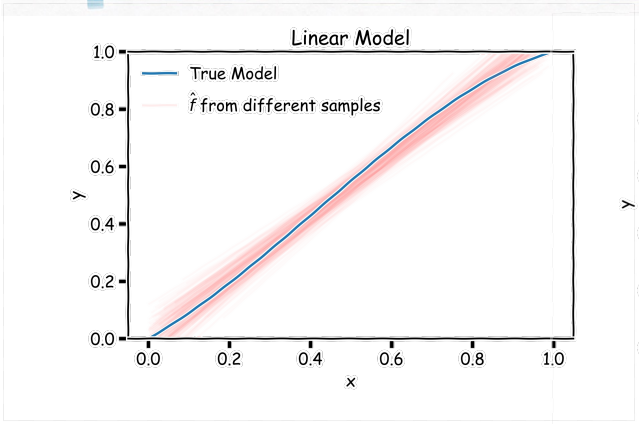
$$\text{Var} = \frac{(0.48 - 0.49)^2 + (0.52 - 0.49)^2 + (0.46 - 0.49)^2 + (0.50 - 0.49)^2}{4}$$

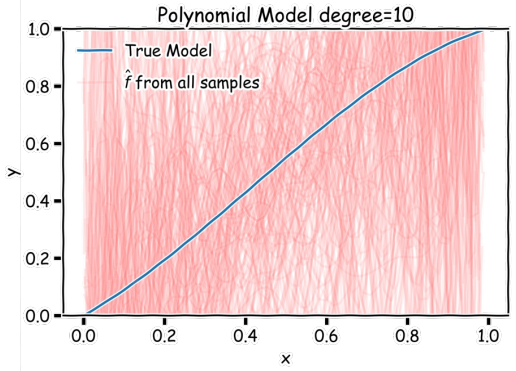
- *How much predictions change across datasets* → variance

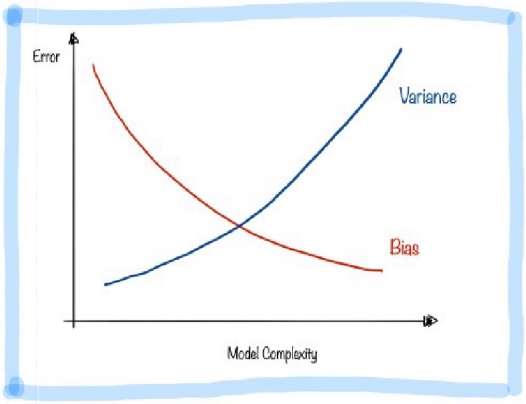
The Bias-Variance Trade Off





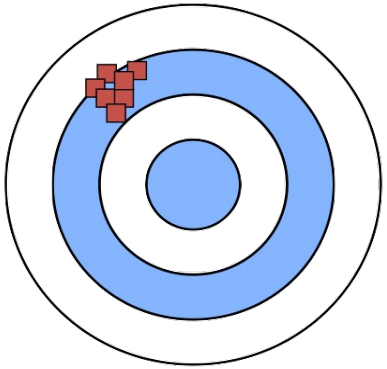


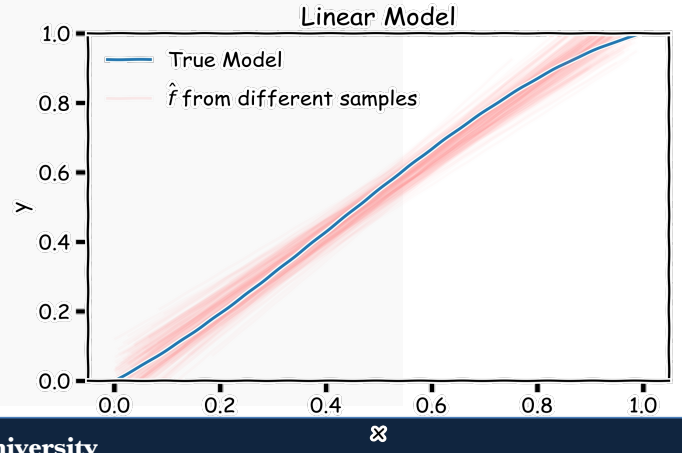




Low Variance
(Precise)

High Bias
(Not Accurate)

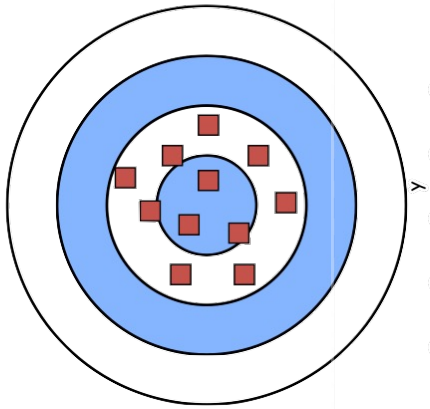


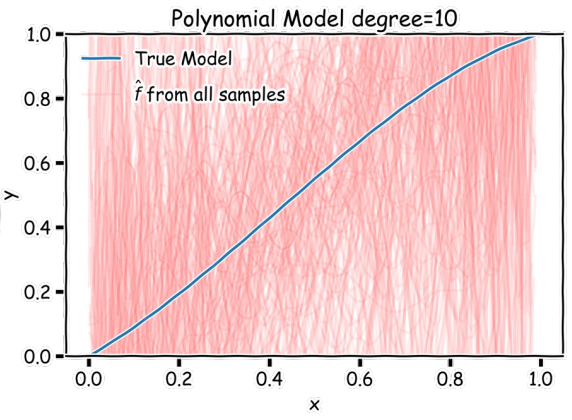


Low Variance
(Precise)

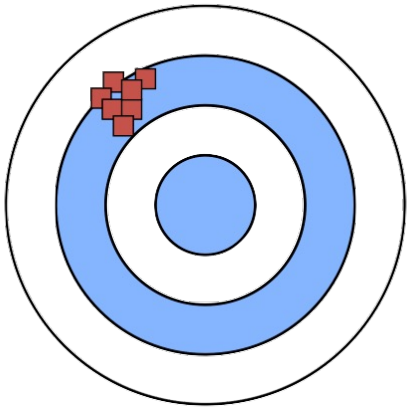
High Variance
(Not Precise)

Low Bias
(Accurate)





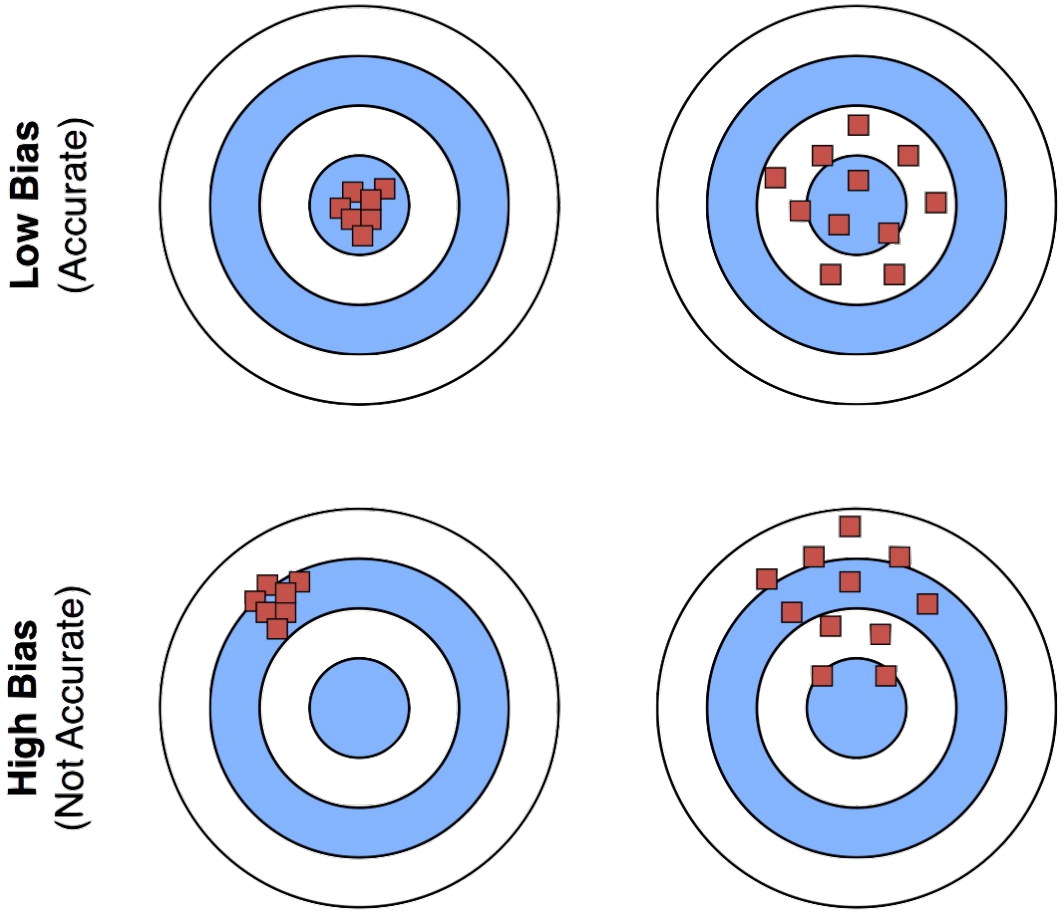
High Bias
(Not Accurate)



Low Variance
(Precise)

High Variance
(Not Precise)

WE WANT THIS



WE WANT TO AVOID THIS

Overfitting

- Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

- Overfitting can happen when:
  - Too many parameters
  - Degree of the polynomial is too large
  - Too many interaction terms
  - others

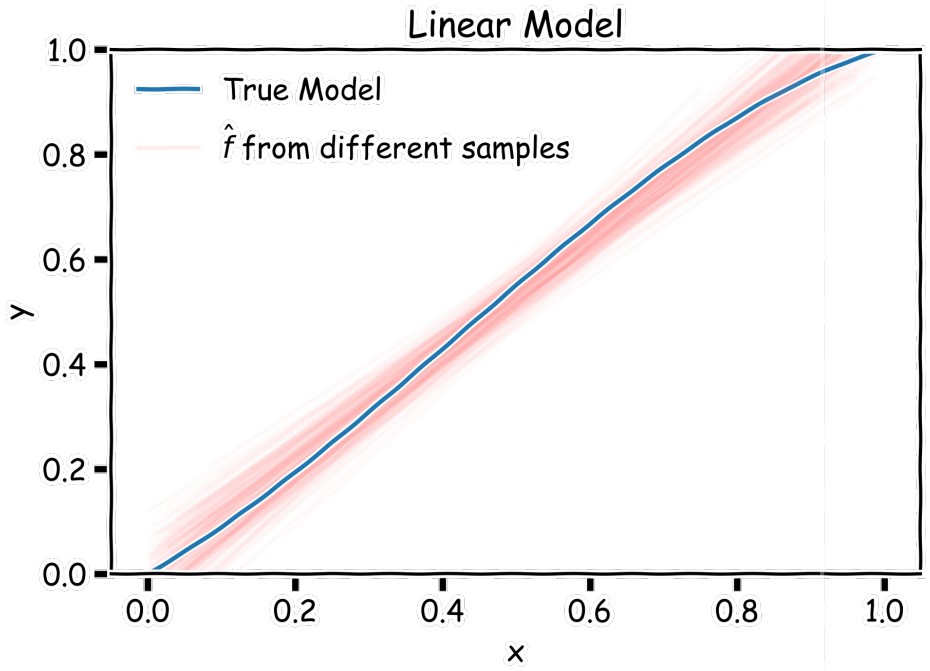
- A way of avoiding overfitting: Ridge and Lasso regressions.

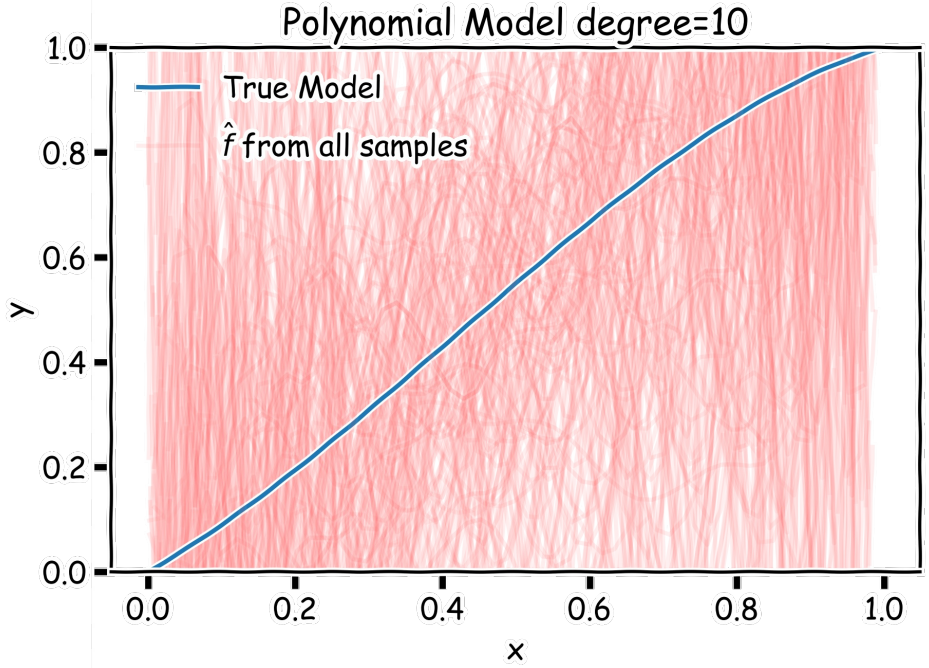
Part 2: Ridge and Lasso – Hyperparameters

Bias vs Variance

Left: 2000 best fit straight lines, each fitted on a different 20 point training set.

Right: Best-fit models using degree 10 polynomial

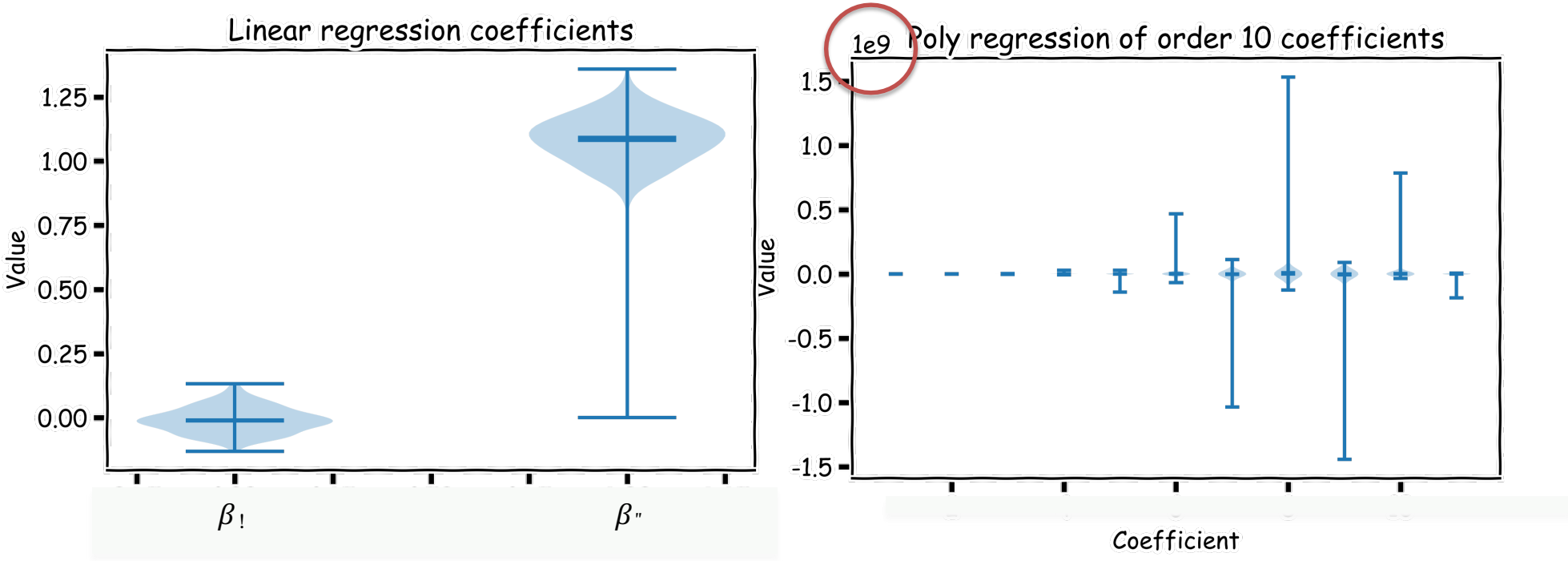




Bias vs Variance

Left: Linear regression coefficients

Right: Poly regression of order 10 coefficients



Model Selection

Model selection is the application of a principled method to determine the complexity of the model, e.g., choosing a subset of predictors, choosing the degree of the polynomial model etc.

A strong motivation for performing model selection is to avoid **Overfitting** which can happen when:

- there are too many features
- the polynomial degree is too high
- too many cross terms are considered
- the coefficients values are too **extreme**

Regularization

What we want

Low model error.

Minimize:

$$\frac{1}{n}\sum_{i=1}^{n}\left|y_i-\boldsymbol{\beta}^\top \boldsymbol{x}_i\right|^2$$

Discourage extreme values in model parameters.

$$L_{reg}=\begin{cases}\sum_{j=1}^{J}\beta_j^2\\ \sum_{j=1}^{J}|\beta_j|\end{cases}$$

Regularization

What we want

Low model error.

Minimize:

$$\frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i\right|^2$$

Discourage extreme values in model parameters.

Minimize:

$$L_{reg} = \begin{cases} \sum_{j=1}^{J} \beta_j^2 \\ \sum_{j=1}^{J} |\beta_j| \end{cases}$$

How do we combine these two objectives?

Regularization

What we want

Low model error.

Minimize:

Discourage extreme values in model parameters.

Minimize:

$$\mathcal{L}_{REG} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^{\top}\boldsymbol{x}_i\right|^2 + L_{reg}$$

Regularization

What we want

Low model error.

Minimize:

Discourage extreme values in model parameters.

Minimize:

λ is the **regularization parameter**. It controls the relative importance between model error and regularization term

$$\mathcal{L}_{REG} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^{\top}\boldsymbol{x}_i\right|^2 + \lambda L_{reg}$$

Regularization

What we want

Low model error.

Discourage extreme values in model parameters.

nize:

$\lambda = 0$: equivalent to simple linear regression

$\lambda = \infty$: yields a model with $\beta's$ =0

$$\mathcal{L}_{REG} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i\right|^2 + \lambda L_{reg}$$

Regularization

What we want

Low model error.

Minimize:

Discourage extreme values in model parameters.

Minimize:

How do we determine λ?

$$\mathcal{L}_{REG} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i\right|^2 + \lambda L_{reg}$$

Regularization

What we want

Low model error.

Minimize:

Discourage extreme values in model parameters.

Minimize:

Cross Validation!

$$\mathcal{L}_{REG} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^{\top}\boldsymbol{x}_i\right|^2 + \lambda L_{reg}$$

Regularization: LASSO Regression

What we want

Low model error.

Minimize:

Discourage extreme values in model parameters.

Minimize:

Note that $\sum_{j=1}^{J} |\beta_j|$ is the l_1 norm of the vector $\boldsymbol{\beta}$

$$\mathcal{L}_{LASSO} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i\right|^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

Regularization: LASSO Regression

Lasso regression: minimize $\mathcal{L}_{LASSO}$ with respect to $\beta's$

$$\mathcal{L}_{LASSO} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^{\top}\boldsymbol{x}_i\right|^2 + \lambda\sum_{j=1}^{J}|\beta_j|$$

Regularization: Ridge Regression

Ridge regression: minimize $\mathcal{L}_{RIDGE}$ with resp

Note that $\sum_{j=1}^{J} \beta_j^2$ is the l_2 norm of the vector β

$$\mathcal{L}_{LASSO} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i\right|^2 + \lambda \sum_{j=1}^{J} \beta_j^2$$

Ridge regularization with only **validation** : step by step

For ridge regression there exist an analytical solution for the coefficients:

$$\hat{\beta}_{Ridge}(\lambda) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$$

1. split data into $\{\{X,Y\}_{train}, \{X,Y\}_{validation}, \{X,Y\}_{test}\}$
2. for λ in $\{\lambda_{min}, \ldots \lambda_{max}\}$:
    1. determine the β that minimizes the L_{ridge}, $\beta_{Ridge}(\lambda) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$, using the train data.
    2. record $L_{MSE}(\lambda)$ using validation data.
3. select the λ that minimizes the MSE loss on the validation data,

$$\lambda_{ridge} = \mathrm{argmin}_{\lambda}\, L_{MSE}(\lambda)$$

4. Refit the model using both train and validation data, $\{\{X,Y\}_{train}, \{X,Y\}_{validation}\}$, now using λ_{ridge}, resulting to $\hat{\beta}_{ridge}(\lambda_{ridge})$
5. Report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{ridge}(\lambda_{ridge})$

Ridge regularization with only **validation** : step by step

For ridge regression there exist an analytical solution for the coefficients:

$$\hat{\beta}_{Ridge}(\lambda) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$$

1. split data into $\{\{X,Y\}_{train}, \{X,Y\}_{validation}, \{X,Y\}_{test}\}$
2. for λ in $\{\lambda_{min}, \ldots \lambda_{max}\}$:
    1. determine the β that minimizes the L_{ridge}, $\beta_{Ridge}(\lambda) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$, using the train data.
    2. record $L_{MSE}(\lambda)$ using validation data.
3. select the λ that minimizes the MSE loss on the validation data,

$$\lambda_{ridge} = \mathrm{argmin}_{\lambda}\, L_{MSE}(\lambda)$$

4. Refit the model using both train and validation data, $\{\{X,Y\}_{train}, \{X,Y\}_{validation}\}$, now using λ_{ridge}, resulting to $\hat{\beta}_{ridge}(\lambda_{ridge})$
5. Report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{ridge}(\lambda_{ridge})$

Ridge regularization with only **validation** : step by step

For ridge regression there exist an analytical solution for the coefficients:

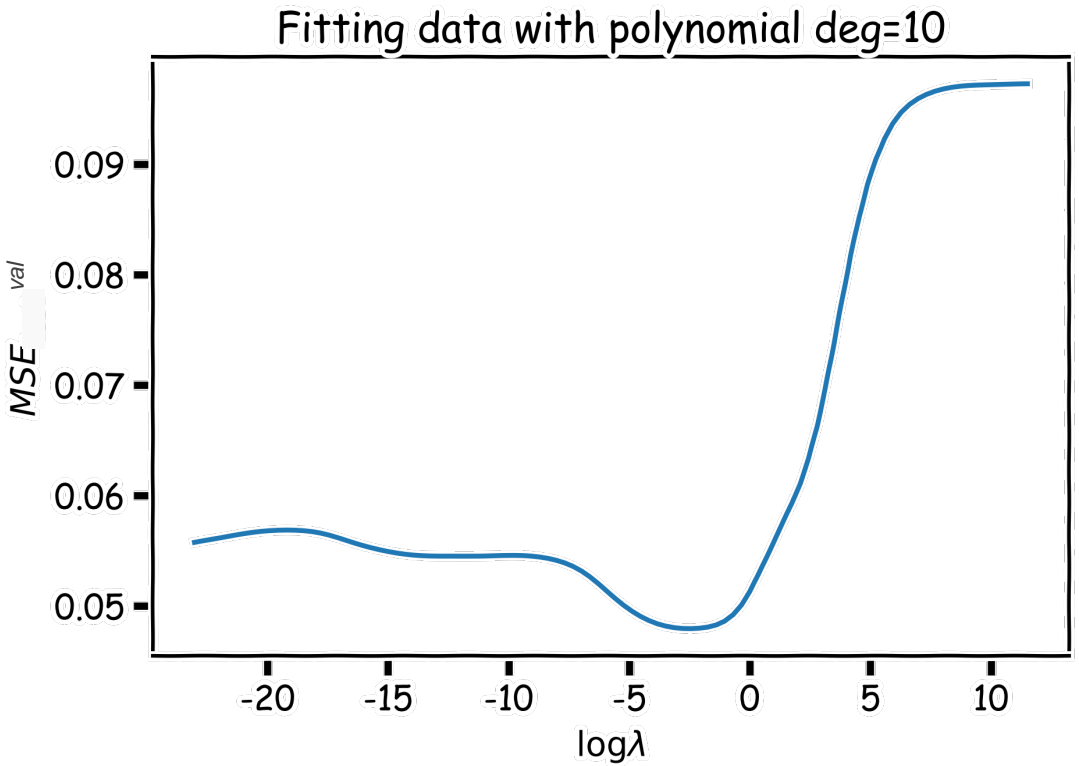
$$\hat{\beta}_{Ridge}(\lambda) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$$

1. split data into $\{\{X,Y\}_{train}, \{X,Y\}_{validation}, \{X,Y\}_{test}\}$
2. for λ in $\{\lambda_{min}, \ldots \lambda_{max}\}$:
    1. determine the β that minimizes the L_{ridge}, $\beta_{Ridge}(\lambda) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$, using the train data.
    2. record $L_{MSE}(\lambda)$ using validation data.
3. select the λ that minimizes the MSE loss on the validation data,

$$\lambda_{ridge} = \mathrm{argmin}_{\lambda}\, L_{MSE}(\lambda)$$

4. Refit the model using both train and validation data, $\{\{X,Y\}_{train}, \{X,Y\}_{validation}\}$, now using λ_{ridge}, resulting to $\hat{\beta}_{ridge}(\lambda_{ridge})$
5. Report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{ridge}(\lambda_{ridge})$

Ridge regularization with **validation** only



Lasso regularization with **validation** only: step by step

For Lasso regression, there is **no** analytical solution for the coefficients, so we use a **solver**.

1. split data into $\{\{X,Y\}_{train}, \{X,Y\}_{validation}, \{X,Y\}_{test}\}$
2. for λ in $\{\lambda_{min}, \ldots \lambda_{max}\}$:
   A. determine the β that minimizes the L_{lasso}, $\beta_{lasso}(\lambda)$, using the train data. **This is done using a solver.**
   B. record $L_{MSE}(\lambda)$ using the validation data.
3. select the λ that minimizes the ***MSE* loss** on the validation data,

$$\lambda_{lasso} = \text{argmin}_{\lambda}\, L_{MSE}(\lambda)$$

4. Refit the model using both train and validation data, $\{\{X,Y\}_{train}, \{X,Y\}_{validation}\}$, now using λ_{Lasso}, resulting to $\hat{\beta}_{lasso}(\lambda_{lasso})$
5. Report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{lasso}(\lambda_{lasso})$

Lasso regularization with **validation** only: step by step

For Lasso regression, there is **no** analytical solution for the coefficients, so we use a **solver**.

1. split data into $\{\{X,Y\}_{train}, \{X,Y\}_{validation}, \{X,Y\}_{test}\}$
2. for λ in $\{\lambda_{min}, \ldots \lambda_{max}\}$:
   A. determine the β that minimizes the L_{lasso}, $\beta_{lasso}(\lambda)$, using the train data. **This is done using a solver.**
   B. record $L_{MSE}(\lambda)$ using the validation data.
3. select the λ that minimizes the ***MSE*** **loss** on the validation data,

$$\lambda_{\ldots} = \text{argmin}_{\lambda} L_{\ldots}(\lambda)$$

4. Refit the model using both train and validation data, $\{\{X,Y\}_{train}, \{X,Y\}_{validation}\}$, now using λ_{Lasso}, resulting to $\hat{\beta}_{lasso}(\lambda_{lasso})$
5. Report MSE or R^2 on $\{X,Y\}_{test}$ given the $\beta_{lasso}(\lambda_{lasso})$

Ridge regularization with **CV**: step by step

	λ_1	λ_2	...	λ_n
k_1	L_{11}	L_{12}	..	...
k_2	L_{21}	...	..	...
...	..	...	..	...
k_n	...	...	...	...
E[]	$\bar{L}_1$	$\bar{L}_2$	...	$\bar{L}_n$

1. remove $\{X,Y\}_{test}$ from data
2. split the rest of data into K folds, $\{\{X,Y\}_{train}^{-k}, \{X,Y\}_{val}^{k}\}$
3. for k in $\{1,\dots,K\}$

   for λ in $\{\lambda_0, \dots, \lambda_n\}$:

   A. determine the β that minimizes the L_{ridge}, $\beta_{ridge}(\lambda,k) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$, using the train data of the fold, $\{X,Y\}_{train}^{-k}$.

   B. record $L_{MSE}(\lambda,k)$ using the validation data of the fold $\{X,Y\}_{val}^{k}$

At this point we have a 2-D matrix, rows are for different k, and columns are for different λ values.

1. Average the $L_{MSE}(\lambda,k)$ for each λ, $\bar{L}_{MSE}(\lambda)$.
2. Find the λ that minimizes the $\bar{L}_{MSE}(\lambda)$, resulting to λ_{ridge}.

Ridge regularization with **CV**: step by step

1. remove $\{X,Y\}_{test}$ from data
2. split the rest of data into K folds, $\{\{X,Y\}_{train}^{-k}, \{X,Y\}_{val}^{k}\}$
3. for k in $\{1,\dots,K\}$

   for λ in $\{\lambda_0,\dots,\lambda_n\}$:

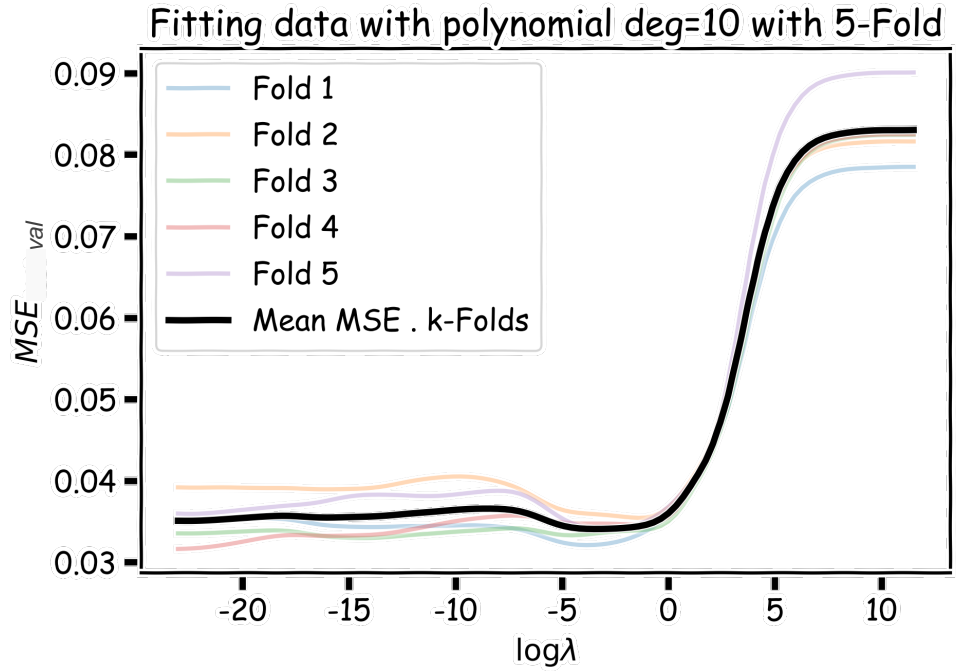
   A. determine the β that minimizes the L_{ridge}, $\beta_{ridge}(\lambda,k) = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda I\right)^{-1} X^T Y$, using the train data of the fold, $\{X,Y\}_{train}^{-k}$.

   B. record $L_{MSE}(\lambda,k)$ using the validation data of the fold $\{X,Y\}_{val}^{k}$

At this point we have a 2-D matrix, rows are for different k, and columns are for different λ values.

4. Average the $L_{MSE}(\lambda,k)$ for each λ, $\bar{L}_{MSE}(\lambda)$.
5. Find the λ that minimizes the $\bar{L}_{MSE}(\lambda)$, resulting to λ_{ridge}.
6. Refit the model using the full training data, $\{\{X,Y\}_{train}, \{X,Y\}_{val}\}$, resulting to $\hat{\beta}_{ridge}(\lambda_{ridge})$
7. report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{ridge}(\lambda_{ridge})$

Ridge regularization with **cross-validation** only: step by step



Part 3: Comparison of Ridge and Lasso

Ridge, LASSO - Computational Complexity

- Solution to ridge regression:

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

- The solution to the LASSO regression:

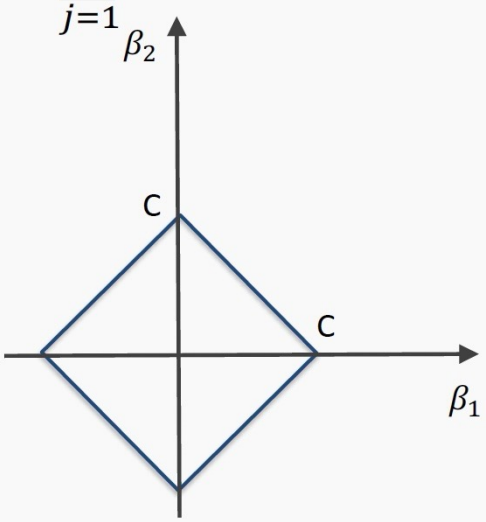
LASSO has no conventional analytical solution, as the L1 norm has no derivative at 0. We can, however, use the concept of subdifferential or subgradient to find a manageable expression.

The Geometry of Regularization (LASSO)

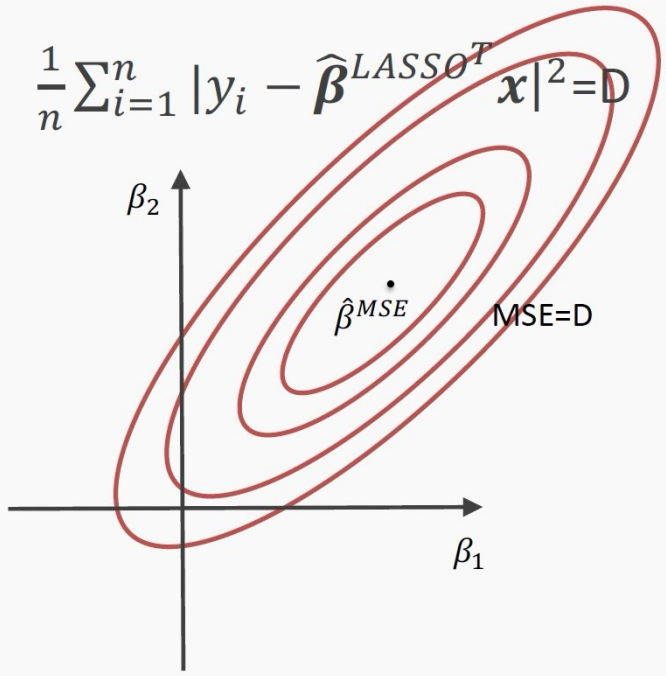
$$L_{LASSO}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^T \boldsymbol{x}|^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

$$\widehat{\boldsymbol{\beta}}^{LASSO} = \operatorname{argmin} L_{LASSO}(\boldsymbol{\beta})$$

$$\lambda \sum_{j=1}^{J} |\hat{\beta}_j^{LASSO}| = C$$

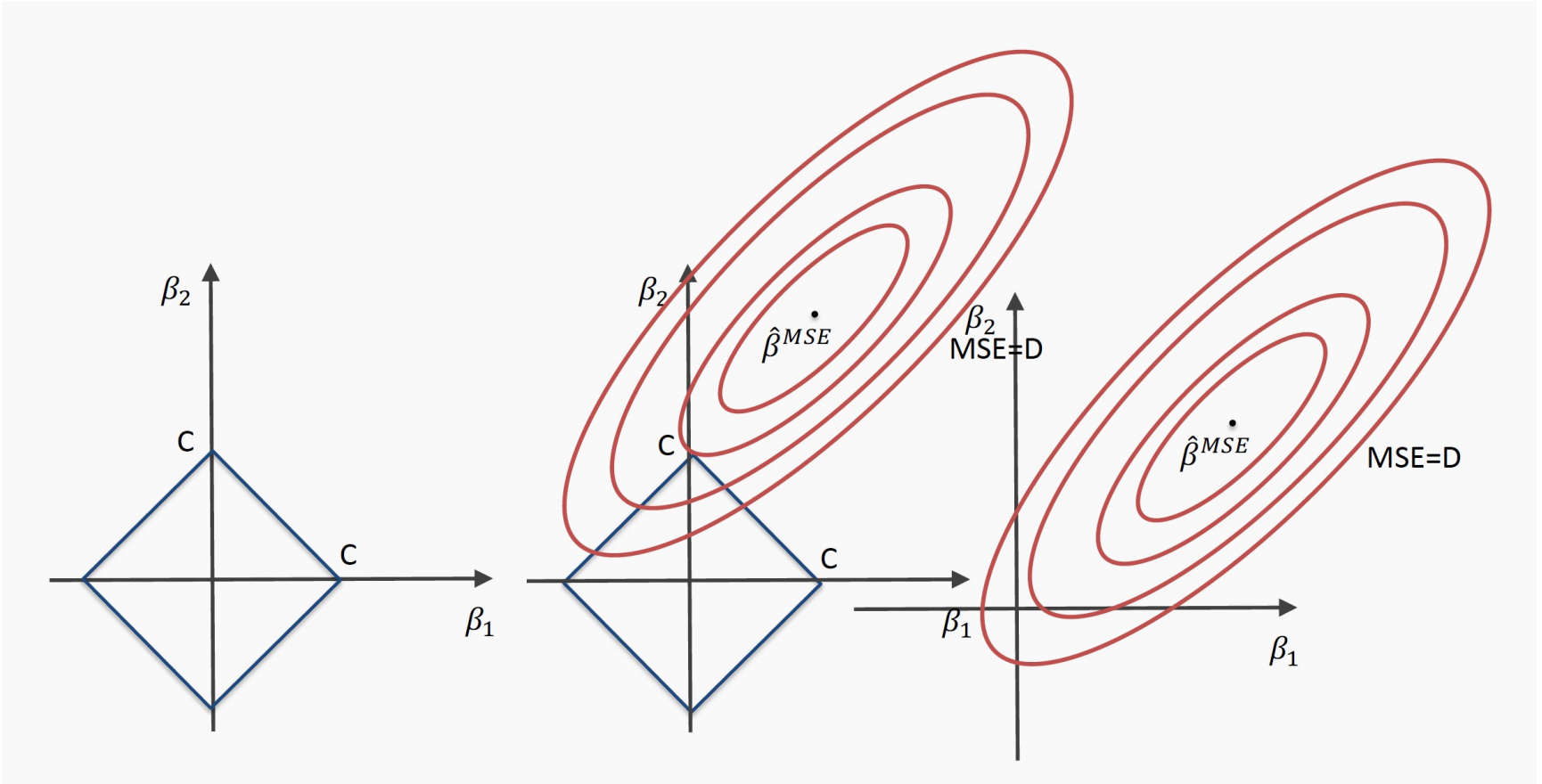


$$\frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{\boldsymbol{\beta}}^{LASSO^T} \boldsymbol{x}|^2 = D$$



Assuming two features with β_1 and β_2

The Geometry of Regularization (LASSO)

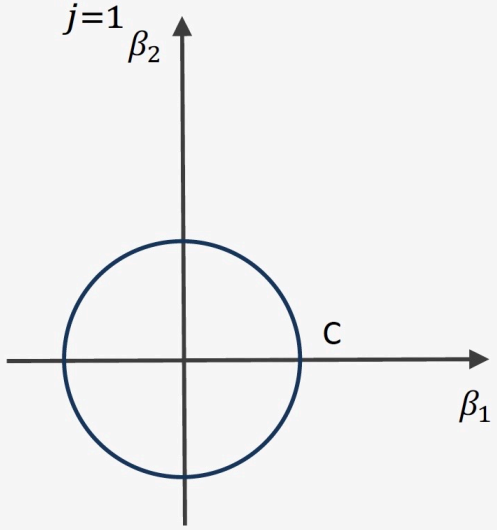


The Geometry of Regularization (Ridge)

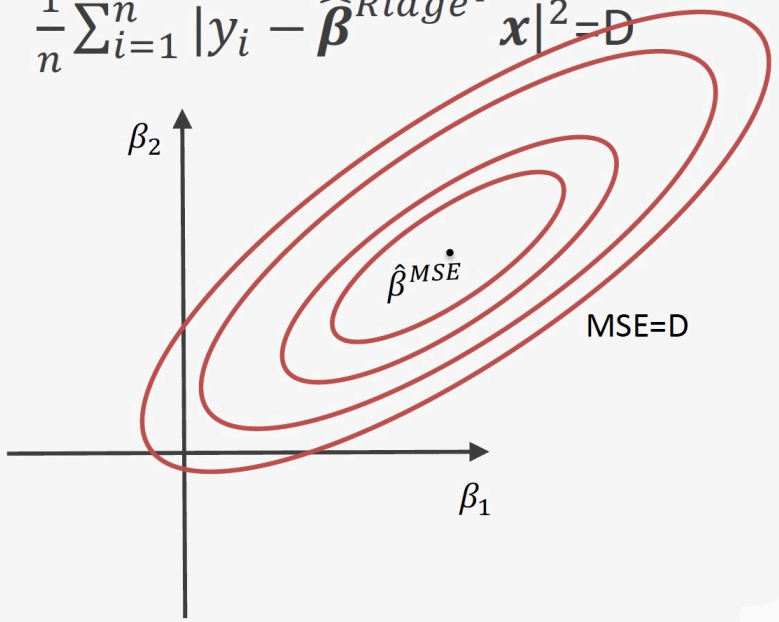
$$L_{Ridge}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \boldsymbol{\beta}^T\boldsymbol{x}\right|^2 + \lambda\sum_{j=1}^{J}(\beta_j)^2$$

$$\widehat{\boldsymbol{\beta}}^{Ridge} = \operatorname{argmin} L_{Ridge}(\boldsymbol{\beta})$$

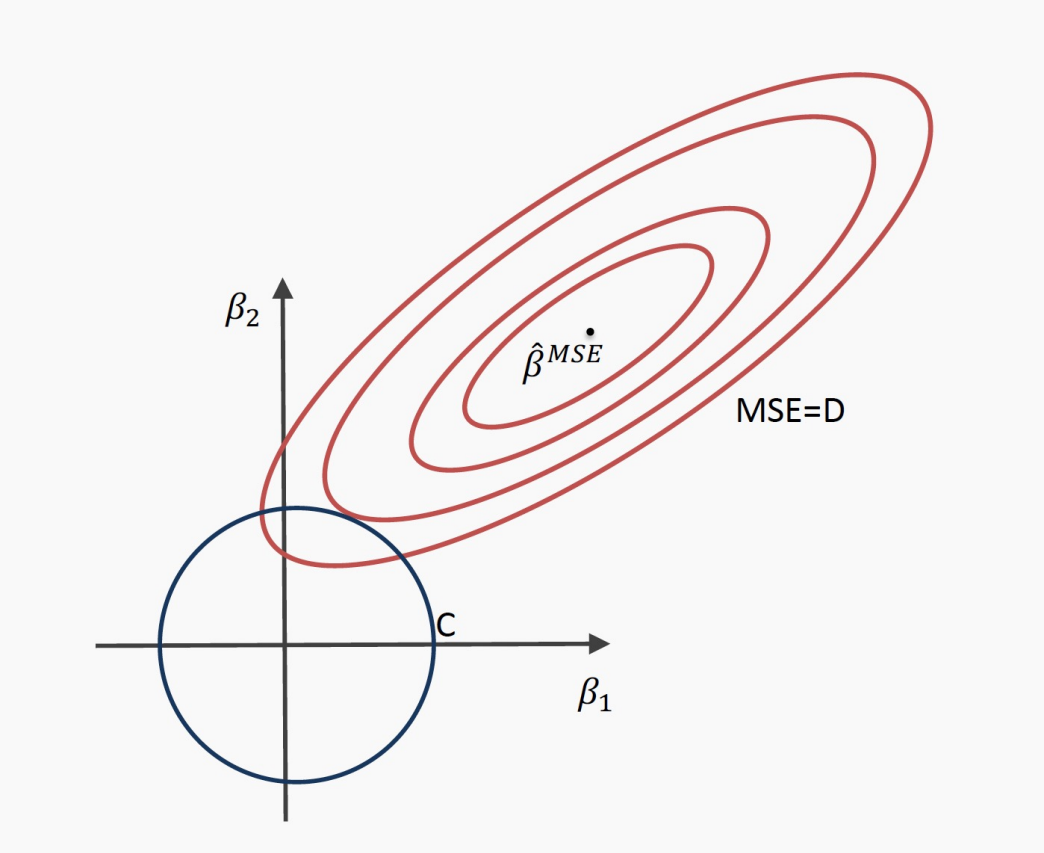
$$\lambda\sum_{j=1}^{J}\left|\hat{\beta}_j^{Ridge}\right|^2 = C$$



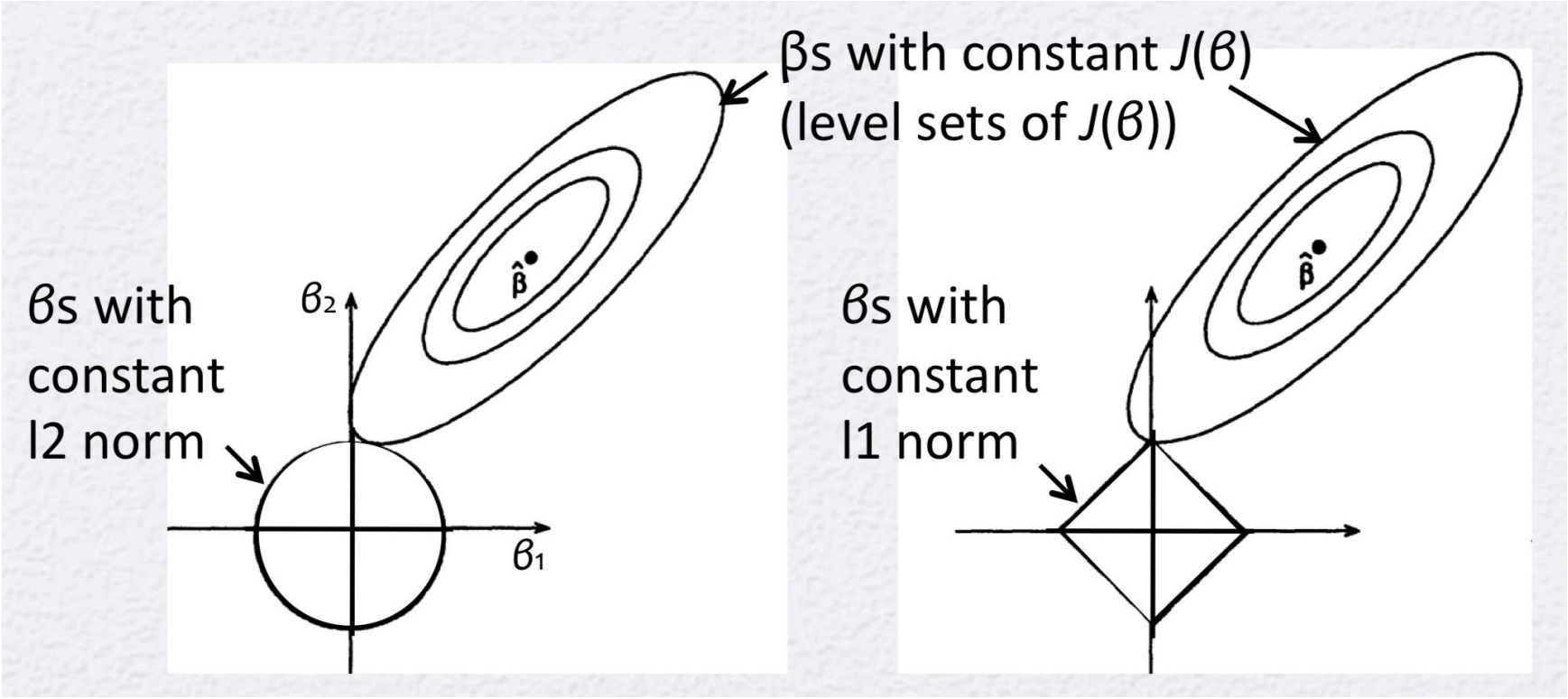
$$\frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{\boldsymbol{\beta}}^{Ridge^T}\boldsymbol{x}\right|^2 = D$$



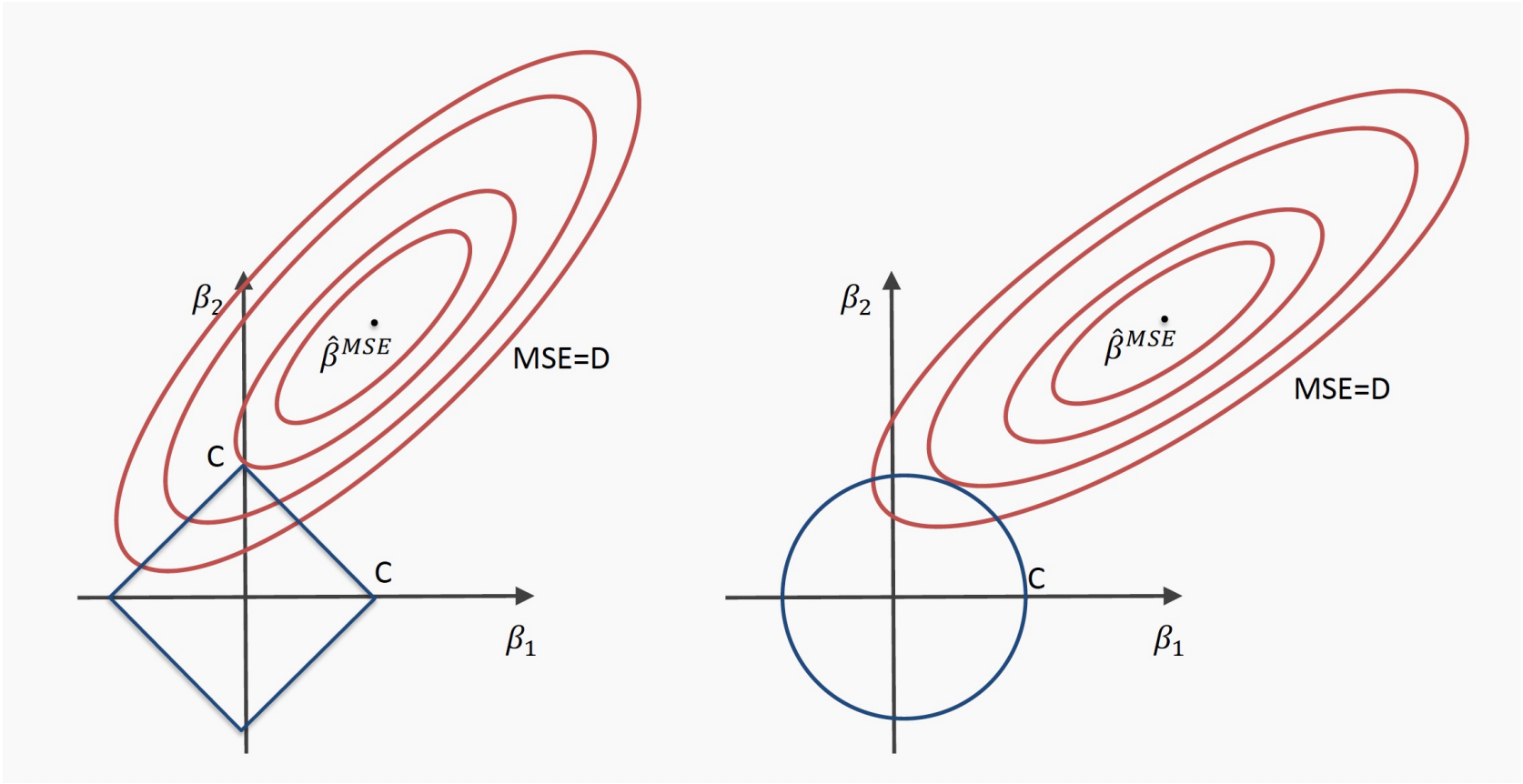
The Geometry of Regularization (Ridge)



At the Corner

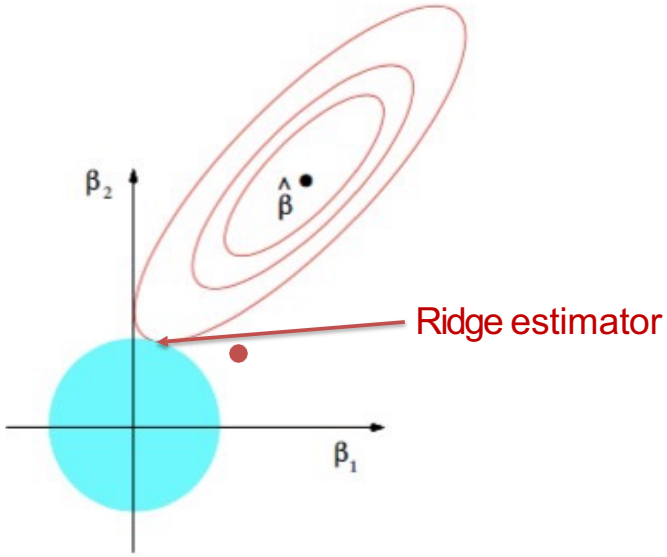


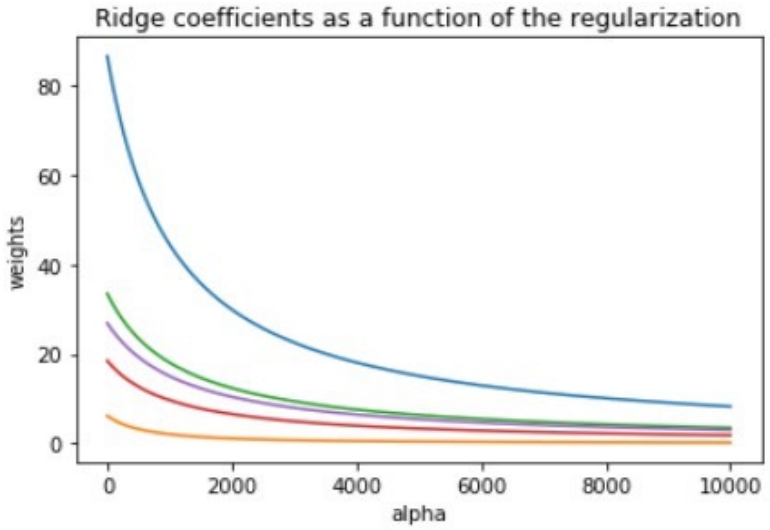
The Geometry of Regularization



Ridge visualized

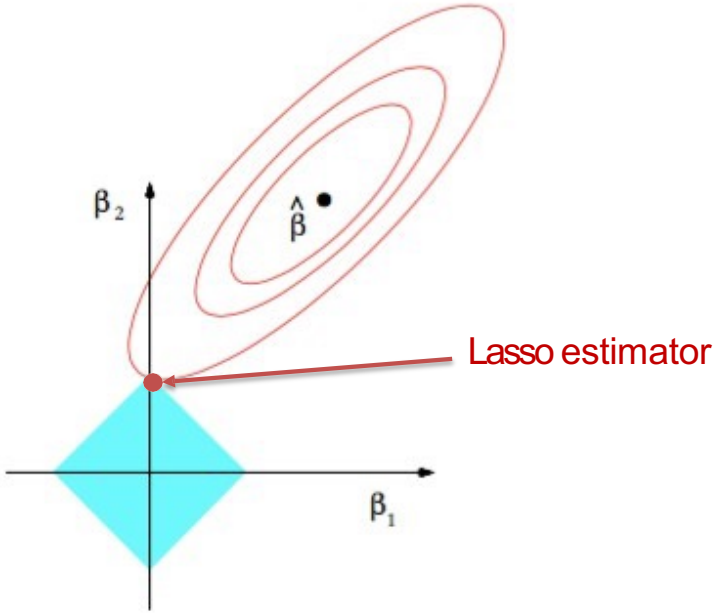
The ridge estimator is where the constraint and the loss intersect.



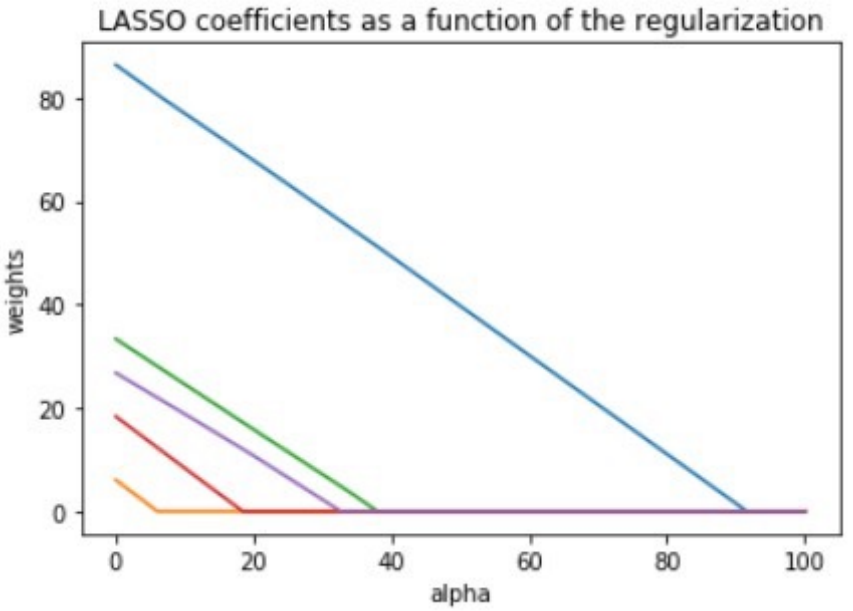


- The values of the coefficients decrease as λ (alpha) increases, but they are not nullified.
- Ridge shrinks continuously.

LASSO visualized



The Lasso estimator tends to zero out parameters as the OLS loss can easily intersect with the constraint on one of the axis.



- The values of the coefficients decrease as λ increases and are nullified fast.
- Lasso kills coefficients.