CMPSC 448: Machine Learning

Lecture 6. Regularization for Regression: Ridge and Lasso

Rui Zhang

Fall 2021

Toy example model space

- We will fit polynomials of degree 0, 1 , …. ,9 to the data.
- A polynomials of degree p in one dimension can be represented by:

$$f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_p x^p$$

$f(x) = w_0$ Fixed horizontal line

$f(x) = w_0 + w_1 x$ A line with slope w_1

$f(x) = w_0 + w_1 x + w_2 x^2$ A quadratic function

- By increasing p, the model becomes more complex (richer)
- The parameters we have to fit based on training data are:

$$w_0, w_1, w_2, \ldots, w_p$$

- For now, let's not talk about how we fit the parameters

Toy example evaluation (loss function)

- We need to define a loss function to evaluate a model (polynomial)
- Let focus on the single example (x_i, y_i)
- The prediction of a polynomial function, say p = 2 for this point is

$$f(x_i) = w_0 + w_1 x_i + w_2 x_i^2$$

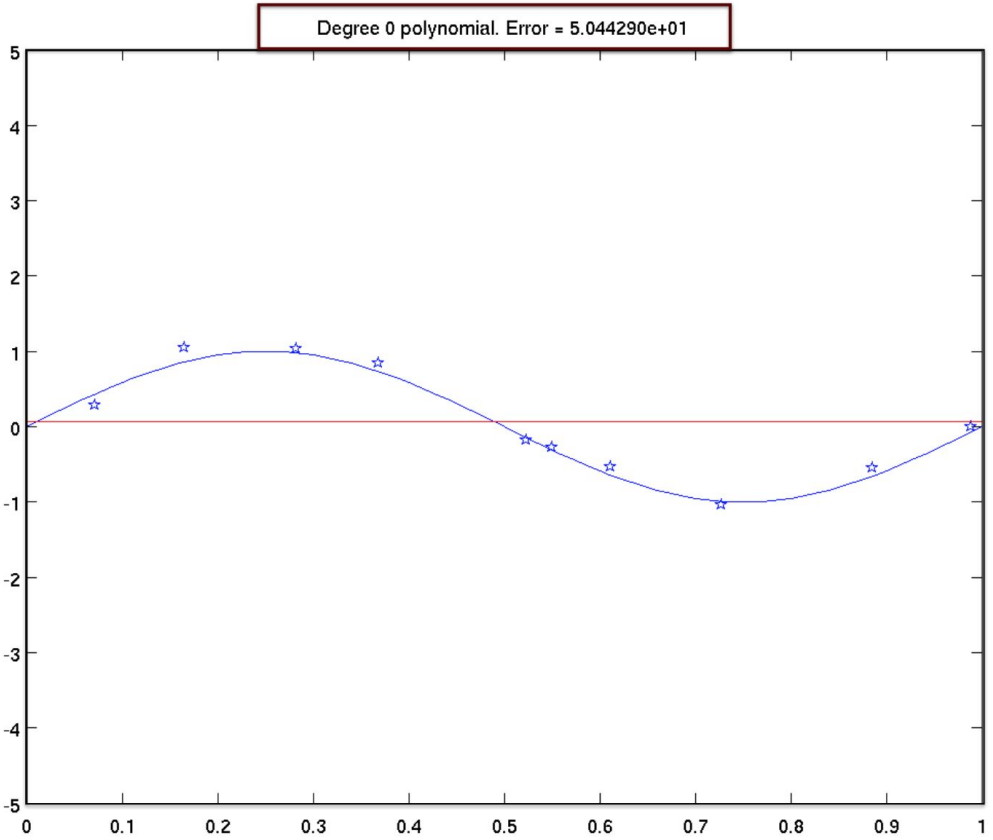
- We use the squared loss to measure the performance of a model:

(actual label - label predicted by model)^2

which is:

$$(y_i - f(x_i))^2 = (y_i - w_0 - w_1 x_i - w_2 x_i^2)^2$$

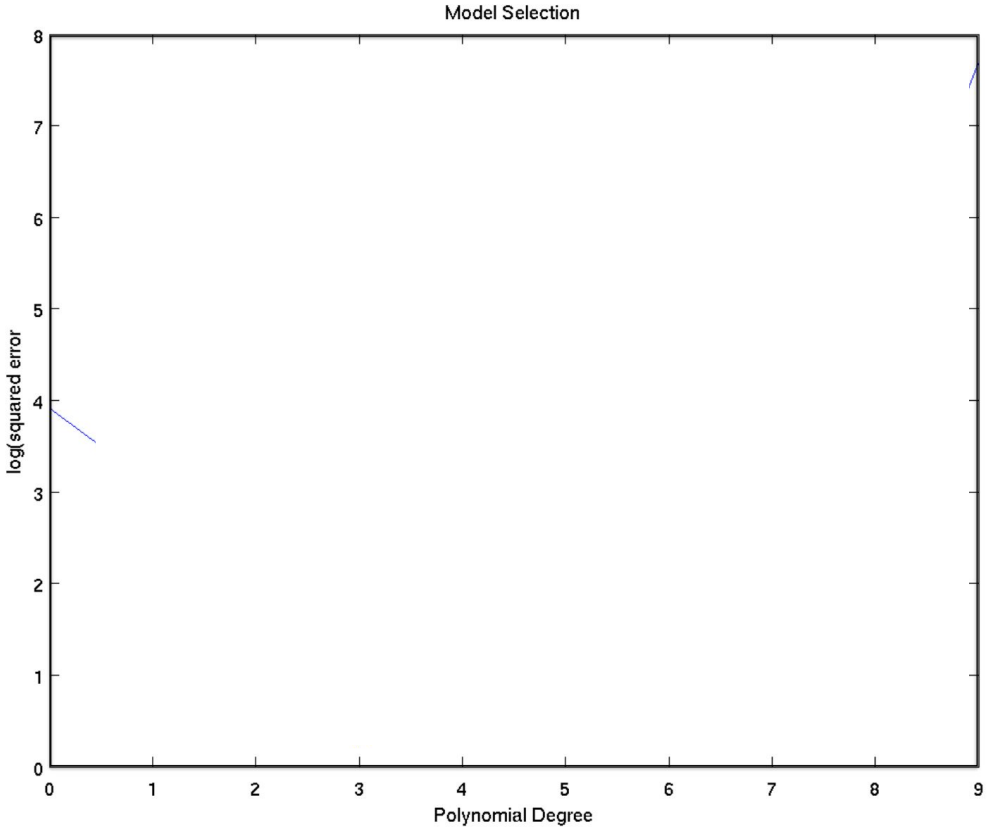
Degree 0 polynomial



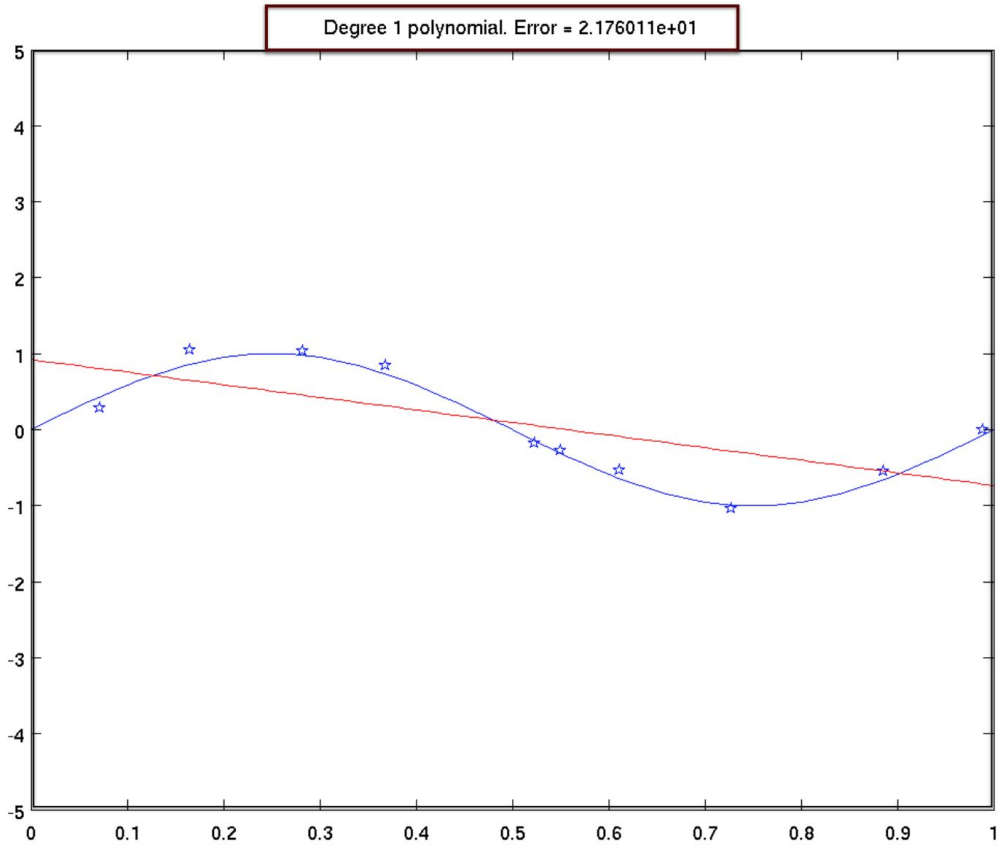
Learned parameters of Deg 0

	Deg 0
w_9	0
w_8	0
w_7	0
w_6	0
w_5	0
w_4	0
w_3	0
w_2	0
w_1	0
w_0	6.6e-2

Generalization Error on Testing Set



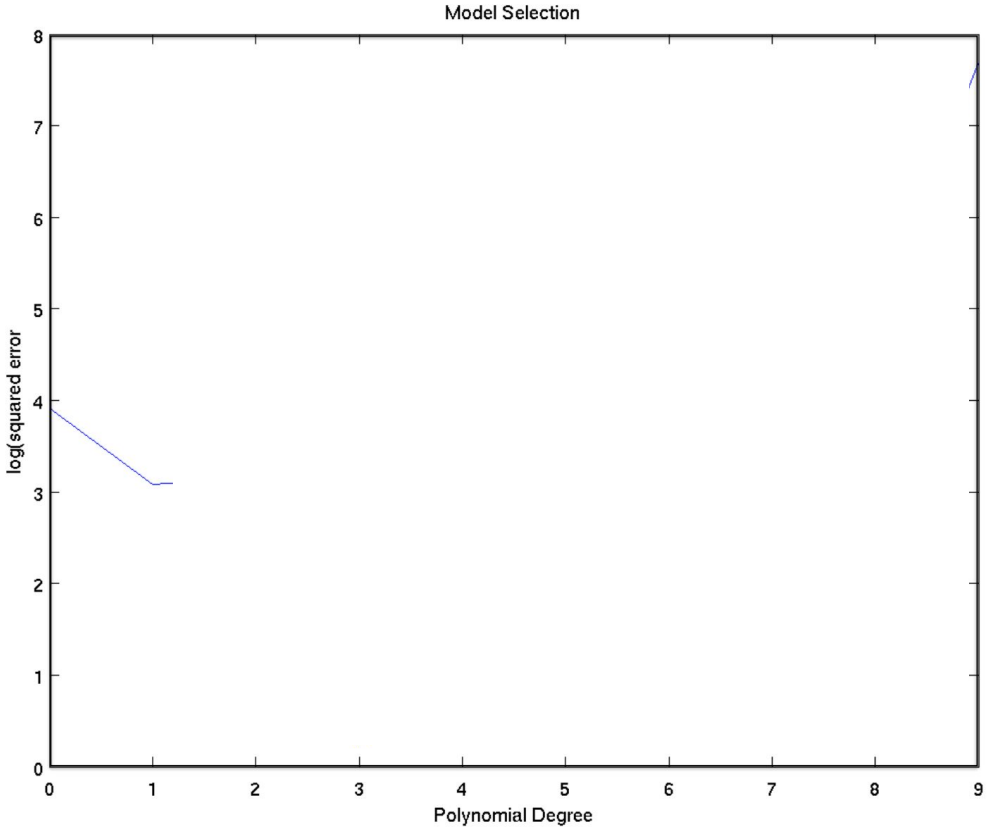
Degree 1 polynomial



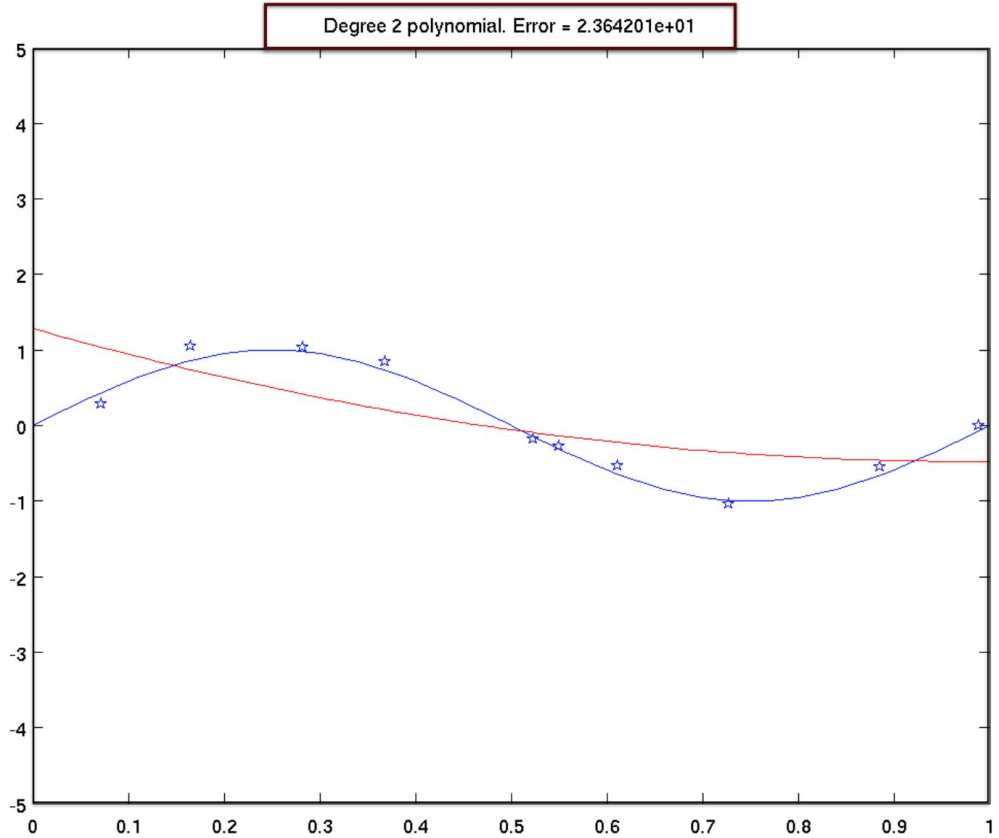
Learned parameters of Deg 1

	Deg 0	Deg 1
w_9	0	0
w_8	0	0
w_7	0	0
w_6	0	0
w_5	0	0
w_4	0	0
w_3	0	0
w_2	0	0
w_1	0	-1.7e0
w_0	6.6e-2	9.2e-1

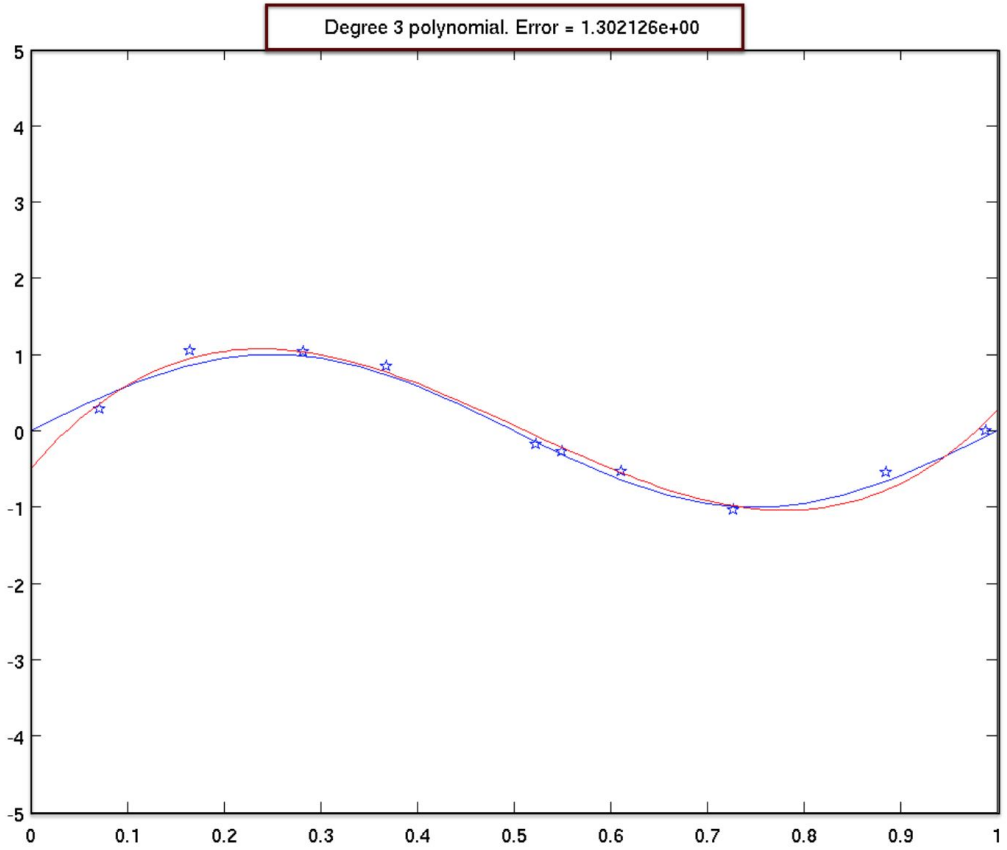
Generalization Error on Testing Set



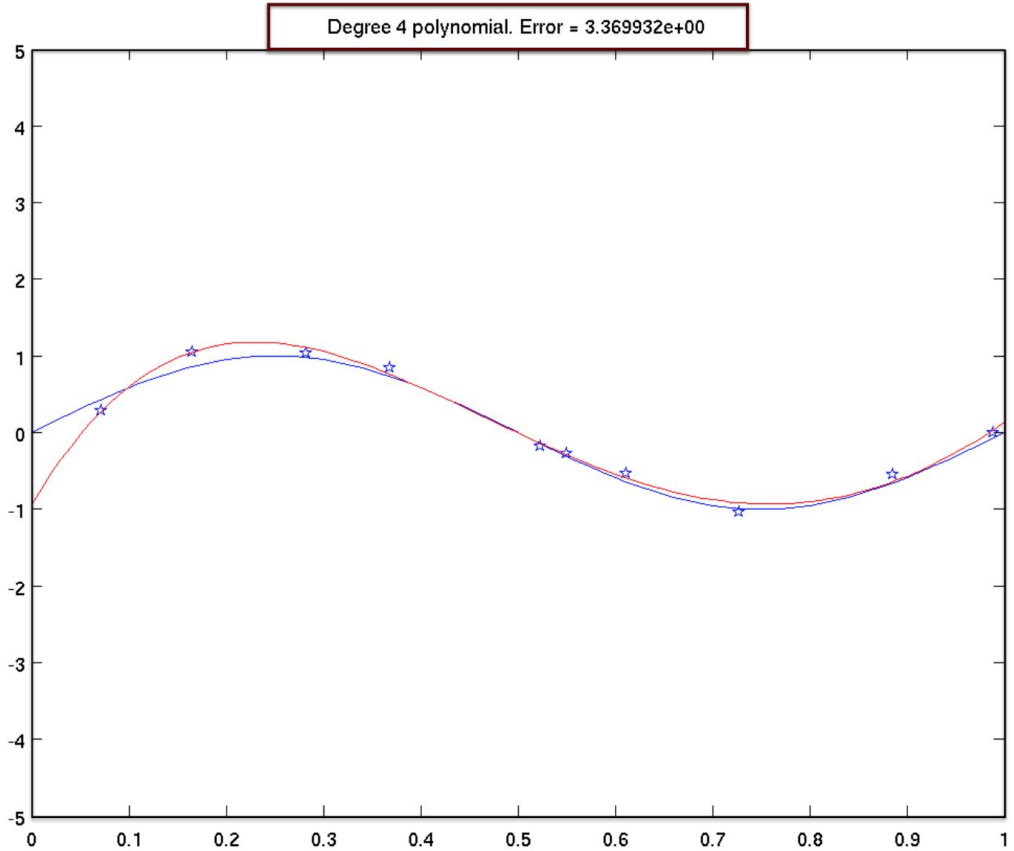
Degree 2 polynomial



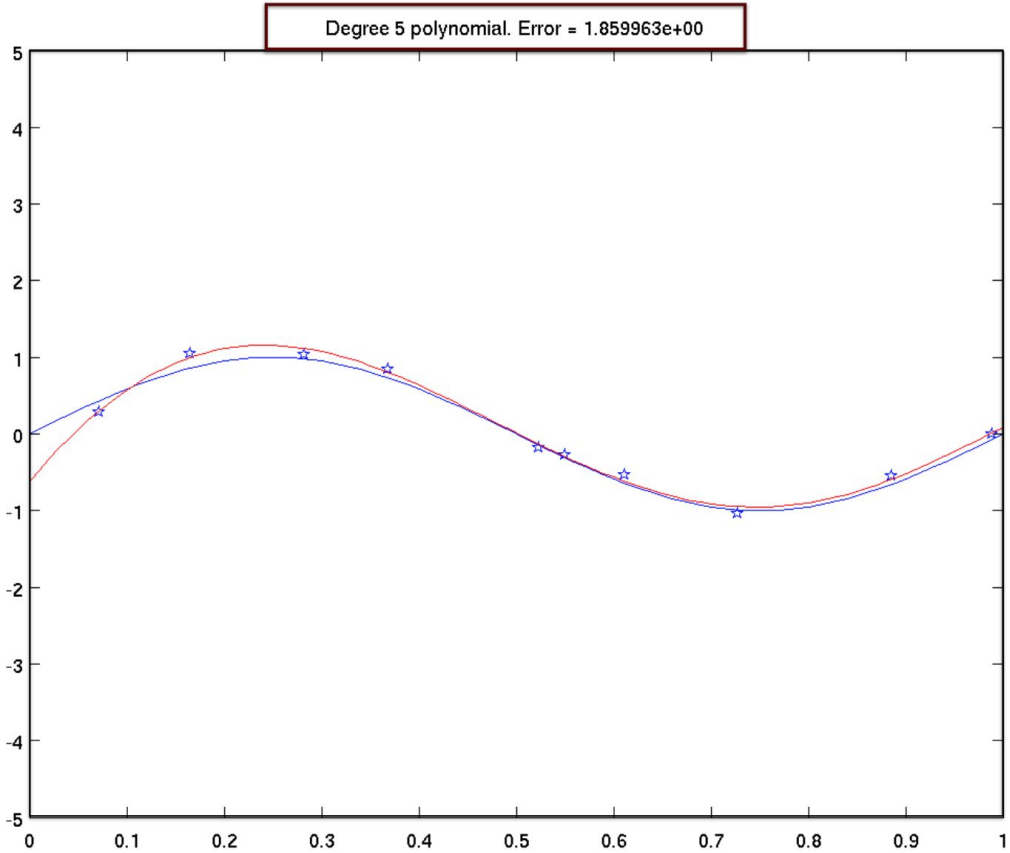
Degree 3 polynomial



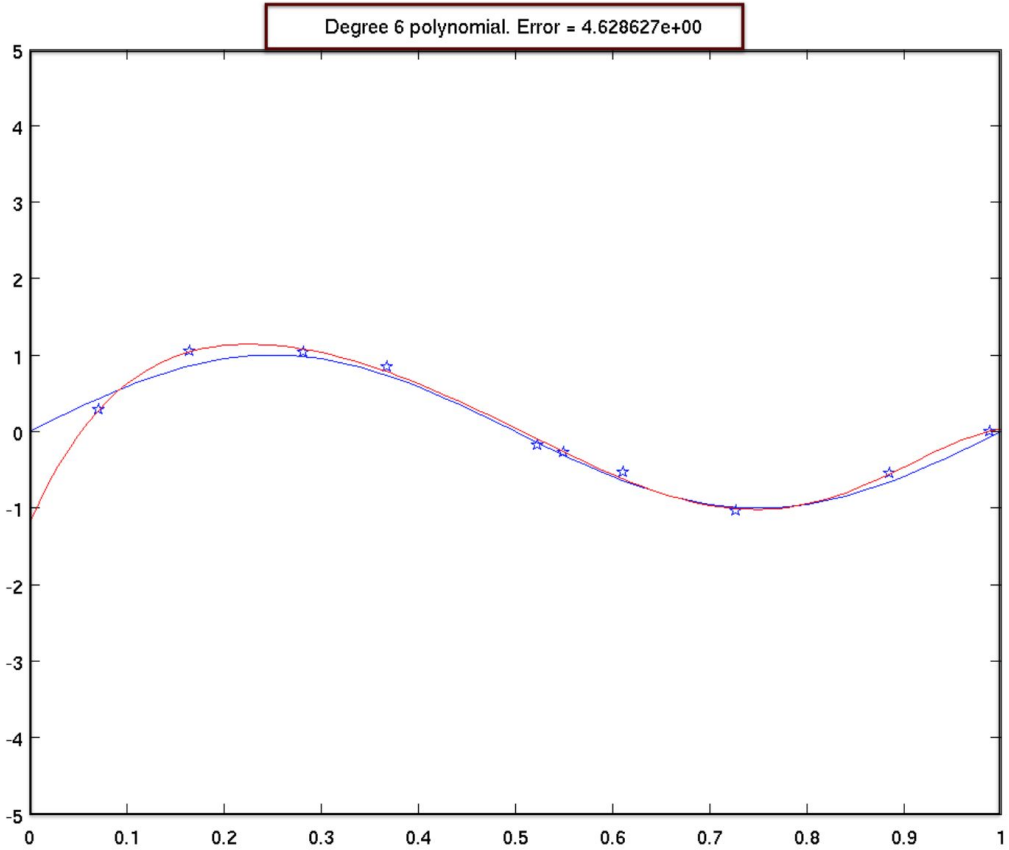
Degree 4 polynomial



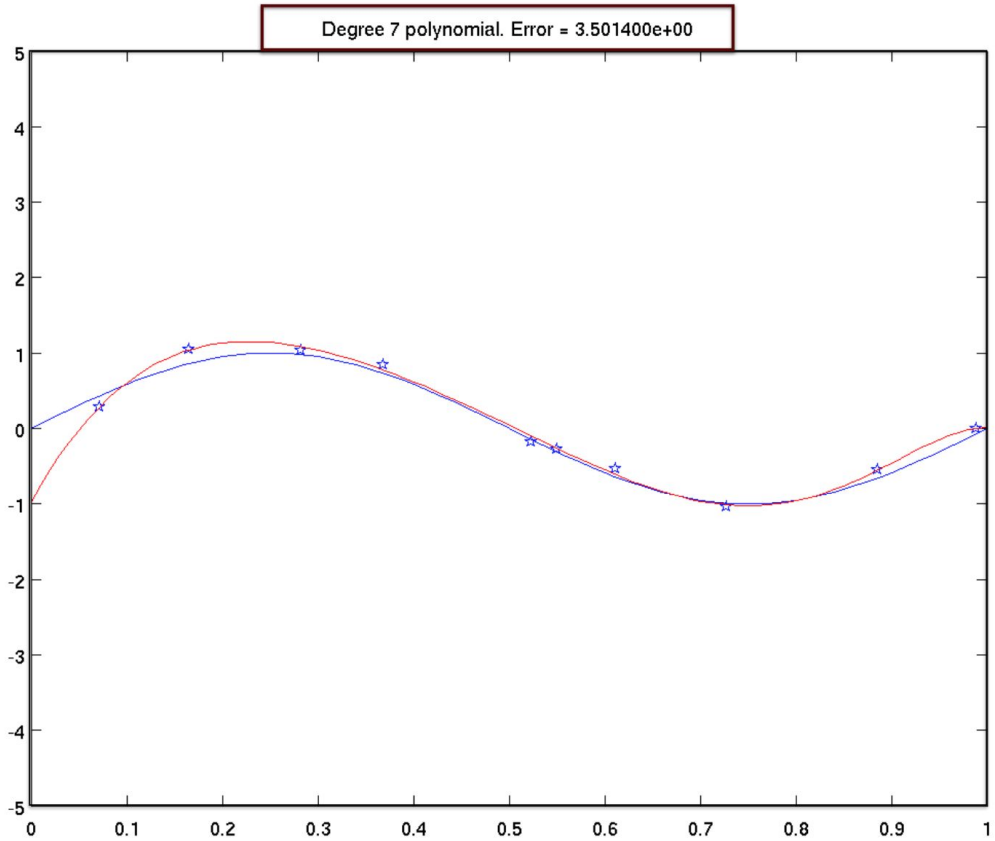
Degree 5 polynomial



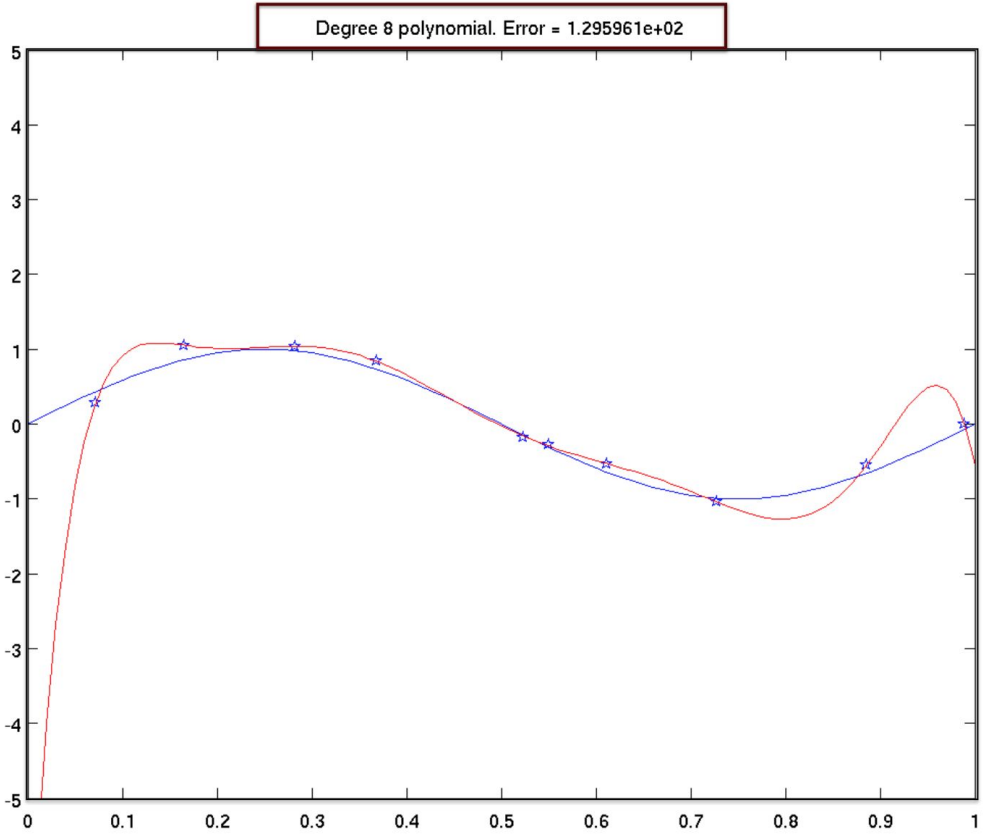
Degree 6 polynomial



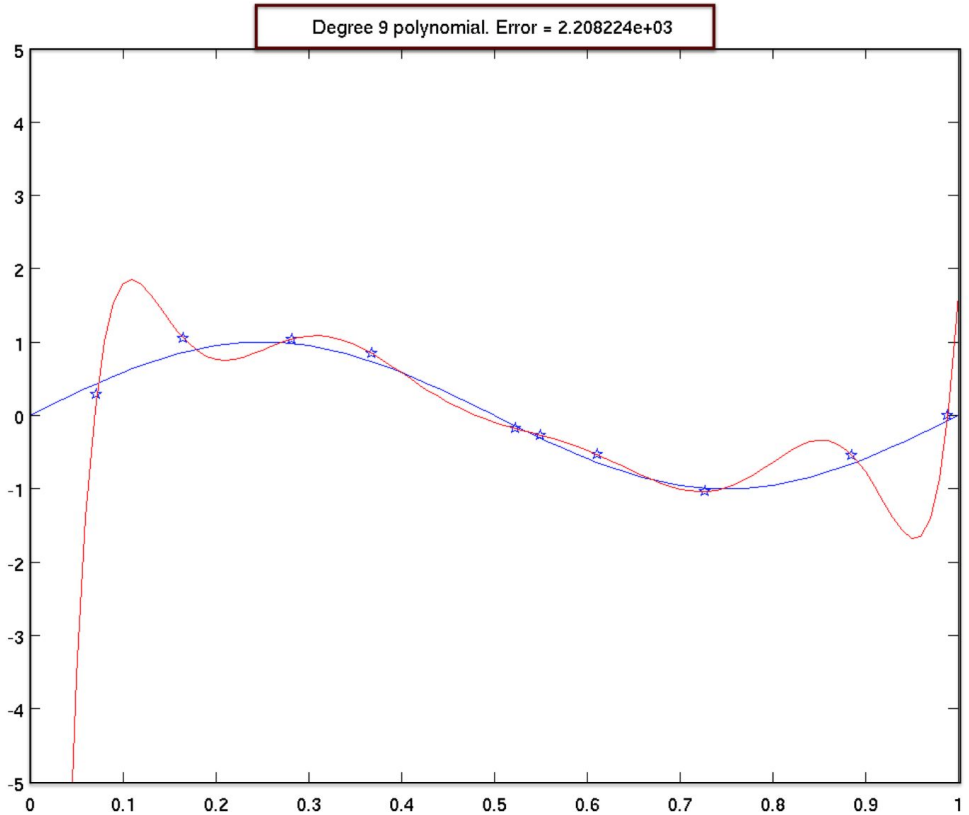
Degree 7 polynomial



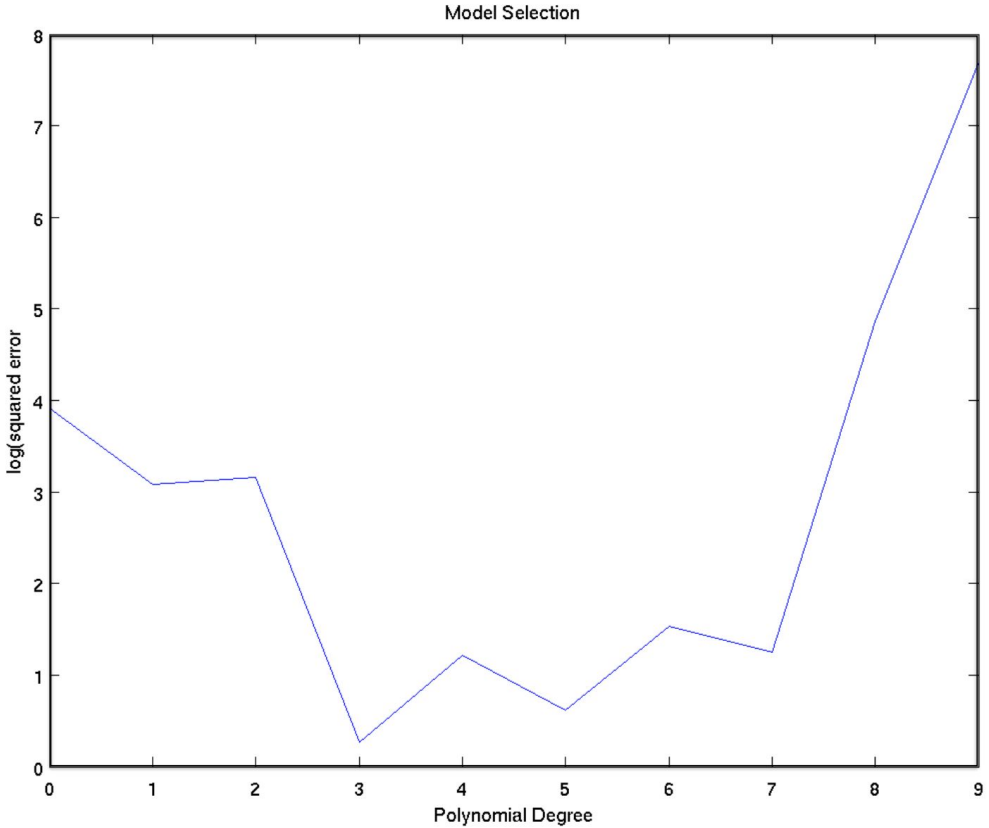
Degree 8 polynomial



Degree 9 polynomial



Generalization Error of all models

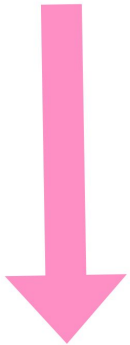


Learned parameters of all models

	Deg 0	Deg 1	Deg 2	Deg 3	Deg 4	Deg 5	Deg 6	Deg 7	Deg 8	Deg 9
w_9	0	0	0	0	0	0	0	0	0	1.0e5
w_8	0	0	0	0	0	0	0	0	-1.2e4	-4.8e5
w_7	0	0	0	0	0	0	0	-1.6e2	5.0e4	9.8e5
w_6	0	0	0	0	0	0	-1.6e2	4.3e2	-8.6e4	-1.1e6
w_5	0	0	0	0	0	-3.3e1	4.8e2	-4.0e2	7.9e4	7.6e5
w_4	0	0	0	0	-1.9e1	6.9e1	-5.6e2	1.2e2	-4.2e4	-3.2e5
w_3	0	0	0	2.6e1	6.6e1	-1.8e1	3.5e2	7.0e1	1.3e4	8.6e4
w_2	0	0	1.8e0	-4.0e1	-6.8e1	-3.2e1	-1.4e2	-7.7e1	-2.4e3	-1.3e4
w_1	0	-1.7e0	-3.6e0	1.5e1	2.1e1	1.5e1	2.9e1	2.3e1	2.3e2	1.1e3
w_0	6.6e-2	9.2e-1	1.3e0	-4.9e-1	-9.4e-1	-6.3e-1	-1.2e0	-9.8e-1	-7.8e0	-3.4e1

Regularization

Intuition: large slopes tend to lead to overfitting



Penalize the coefficients $w_1, w_2, \ldots, w_d$ if they take large values!

How can I regularize?

Penalize the coefficients $w_1, w_2, \ldots, w_d$ if they take large values!

Change the objective function minimize squared error plus penalty on coefficients.

$$\arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

OLS objective
(training error)

How can I regularize?

Penalize the coefficients $w_1, w_2, \ldots, w_d$ if they take large values!

Change the objective function minimize squared error plus penalty on coefficients.

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda\,(\text{penalization of coefficients})$$

OLS objective
(training error)

Regularization
(constraint on coefficients)

Regularization parameter

How can I regularize?

Penalize the coefficients $w_1, w_2, \ldots, w_d$ if they take large values!

Change the objective function minimize squared error plus penalty on coefficients.

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda\,(\text{penalization of coefficients})$$

OLS objective
(training error)

Regularization
(constraint on coefficients)

Regularization parameter

This objective balances getting low training error vs. having small slopes

Nearly-always reduces overfitting

Regularization parameter $\lambda > 0$ controls "strength" of regularization. Large λ puts large penalty on slopes

Equivalent formulation of constrained optimization

An equivalent formulation

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

such that: some function of coefficients $\leq \mu$

- Basically the constrained version of the first formulation!

Equivalent formulation of constrained optimization

An equivalent formulation

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

such that: some function of coefficients $\leq \mu$

- Basically the constrained version of the first formulation!
- For any choice of μ there is a unique value for λ such that both formulations result in the same optimal solution!
- The standard formulation you usually see is the unconstrained version, but there is no difference (we will use the constrained version for illustration purposes only)

Vector norms

Definition 3.1 (Norm). A *norm* on a vector space V is a function

$$\|\cdot\| : V \to \mathbb{R}\,, \tag{3.1}$$

$$\boldsymbol{x} \mapsto \|\boldsymbol{x}\|\,, \tag{3.2}$$

which assigns each vector $\boldsymbol{x}$ its *length* $\|\boldsymbol{x}\| \in \mathbb{R}$, such that for all $\lambda \in \mathbb{R}$ and $\boldsymbol{x}, \boldsymbol{y} \in V$ the following hold:

- *Absolutely homogeneous:* $\|\lambda\boldsymbol{x}\| = |\lambda|\|\boldsymbol{x}\|$
- *Triangle inequality:* $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$
- *Positive definite:* $\|\boldsymbol{x}\| \geqslant 0$ and $\|\boldsymbol{x}\| = 0 \iff \boldsymbol{x} = \boldsymbol{0}$

There is a family of ℓ_p-norms parametrized by $p \in [1, \infty)$ defined by:

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$$

Examples

(1) ℓ_1-norm: $\|\boldsymbol{x}\|_1 = \left(\sum_{i=1}^{d} |x_i|\right)$

(2) ℓ_2-norm: $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$ (Euclidean norm)

(3) ℓ_∞-norm: $\|\boldsymbol{x}\|_\infty = \max_{i=1,\ldots,d} |x_i|$

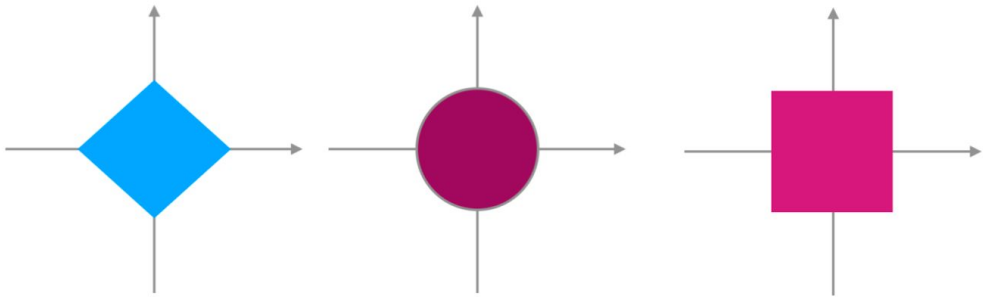
19

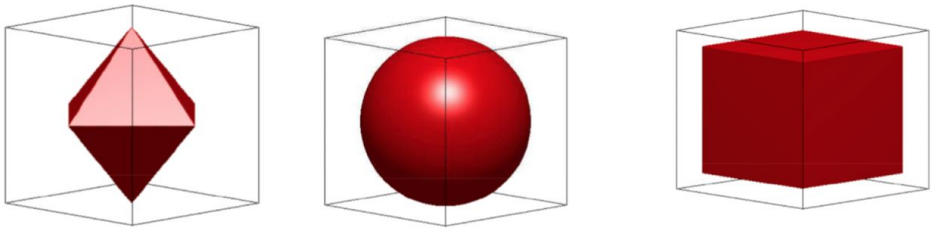
Norm balls

For every ℓ_p norm, we can define its corresponding ball as:

$$\mathbb{B}_p^d = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_p \leq r\}$$

which is the set of all vectors with ℓ_p norm less than equal to a constant.

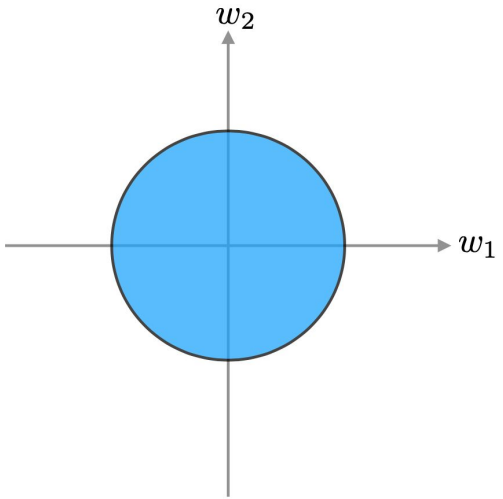




ℓ_1 ℓ_2 ℓ_∞

Two Choices for function of coefficients

such that: some function of coefficients $\leq \mu$

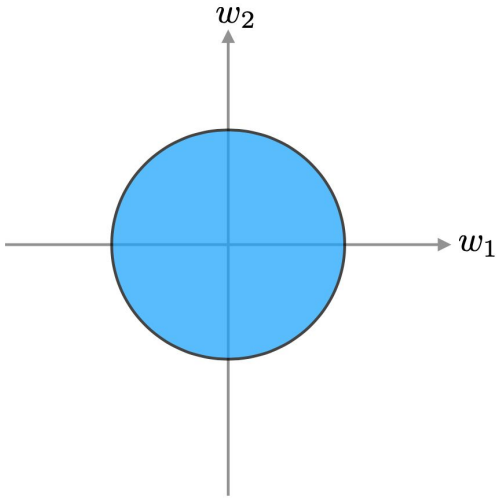


$$\|\boldsymbol{w}\|_2^2 = w_1^2 + w_2^2 + \ldots + w_d^2 \leq \mu$$

Ridge regression: OLS + L2 regularization
(L2 norm of coefficients)

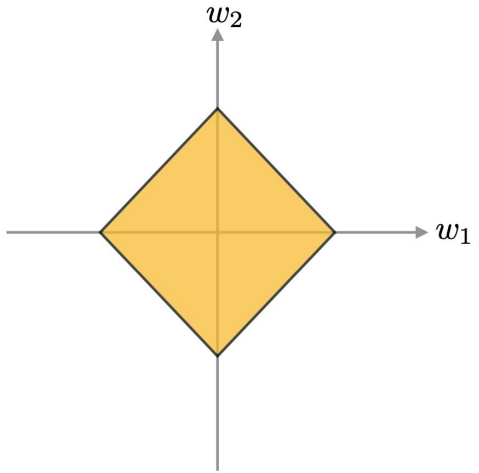
Two Choices for function of coefficients

such that: some function of coefficients $\leq \mu$



$$\|\boldsymbol{w}\|_2^2 = w_1^2 + w_2^2 + \ldots + w_d^2 \leq \mu$$

Ridge regression: OLS + L2 regularization
(L2 norm of coefficients)



$$\|\boldsymbol{w}\|_1 = |w_1| + |w_2| + \ldots + |w_d| \leq \mu$$

Lasso regression: OLS + L1 regularization
(L1 norm of coefficients)

Ridge Regression

The Ridge Regression is a regularization technique that uses L2 regularization (length of vector of parameters) to impose a penalty on the size of coefficients.

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda \left\|\boldsymbol{w}\right\|_2^2$$

Ridge Regression

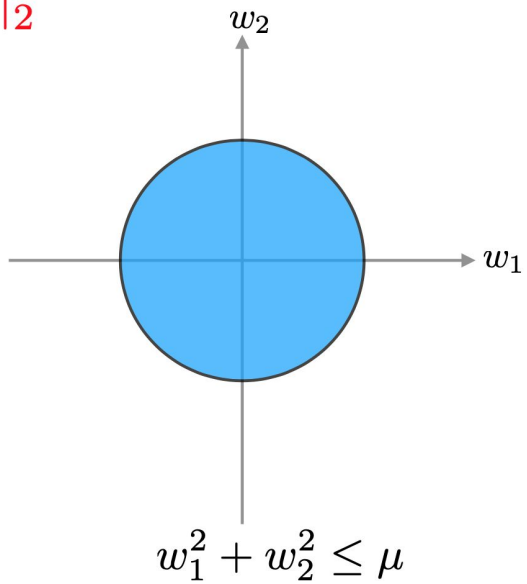
The Ridge Regression is a regularization technique that uses L2 regularization (length of vector of parameters) to impose a penalty on the size of coefficients.

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda \|\boldsymbol{w}\|_2^2$$

OR

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

$$\text{such that} \quad \|\boldsymbol{w}\|_2^2 \leq \mu$$



Ridge solution

We would like to find the optimal solution of following convex problem:

$$\boldsymbol{w}^*_{\text{RR}} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda\|\boldsymbol{w}\|_2^2$$

Let's take the derivative of function and set it to zero:

$$\frac{f_{\text{OLS}}(\boldsymbol{w})}{\partial \boldsymbol{w}} = \sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)\boldsymbol{x}_i + 2\lambda\boldsymbol{w}$$

Ridge solution

We would like to find the optimal solution of following convex problem:

$$\boldsymbol{w}^*_{\mathrm{RR}} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda\|\boldsymbol{w}\|_2^2$$

Let's take the derivative of function and set it to zero:

$$\begin{aligned}
\frac{f_{\text{OLS}}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)\boldsymbol{x}_i + 2\lambda\boldsymbol{w} \\
&= \sum_{i=1}^{n}\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)\boldsymbol{x}_i - \sum_{i=1}^{n} y_i\boldsymbol{x}_i + 2\lambda\boldsymbol{w} \\
&= \sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{w} - \sum_{I=1}^{n} y_i\boldsymbol{x}_i + 2\lambda\boldsymbol{w} \\
&= \left(\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top\right)\boldsymbol{w} - \sum_{i=1}^{n} y_i\boldsymbol{x}_i + 2\lambda\boldsymbol{w} = 0
\end{aligned}$$

Ridge solution

$$\left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top\right) \boldsymbol{w} - \sum_{i=1}^{n} y_i \boldsymbol{x}_i + 2\lambda \boldsymbol{w} = 0$$

$$\rightarrow \quad \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top + 2\lambda \boldsymbol{I}\right) \boldsymbol{w} - \sum_{i=1}^{n} y_i \boldsymbol{x}_i = 0$$

$$\rightarrow \quad \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top + 2\lambda \boldsymbol{I}\right) \boldsymbol{w} = \sum_{i=1}^{n} y_i \boldsymbol{x}_i$$

$$\rightarrow \quad \boldsymbol{w}_{\mathrm{RR}}^* = \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top + 2\lambda \boldsymbol{I}\right)^{-1} \sum_{i=1}^{n} y_i \boldsymbol{x}_i$$

Red terms are those different from OLS!

Ridge solution in matrix representation

Similar to OLS, we can derive the optimal solution of RR in matrix form:

$$\boldsymbol{w}^*_{\mathrm{RR}} = \left(\boldsymbol{X}^\top \boldsymbol{X} + 2\lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

The name ridge regression alludes to the fact that the $2\lambda \boldsymbol{I}$ term adds positive entries along the diagonal "ridge" of the sample covariance matrix

An apple-to-apple comparison of OLS versus RR solutions:

$$\boldsymbol{w}^*_{\mathrm{OLS}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Gradient descent (GD) for Ridge regression

$$\boldsymbol{w}_{\mathrm{RR}}^{*} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda\|\boldsymbol{w}\|_2^2$$

Let solve the optimization with GD:

Only change compared to OLS:

$\boldsymbol{w}_0 = \mathbf{0}$

for $t = 1, 2, \ldots, T$ **do:**

Update:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \left(\sum_{i=1}^{n} (\boldsymbol{w}_t^\top \boldsymbol{x}_i - y_i)\boldsymbol{x}_i + 2\lambda \boldsymbol{w}_t \right)$$

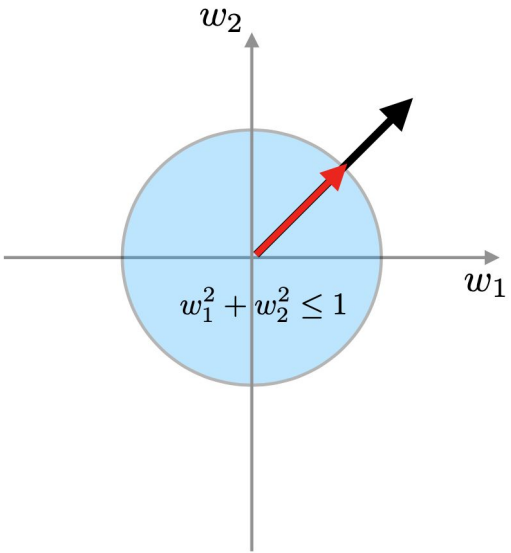
end for

return $\boldsymbol{w}_T$

The time complexity is: $O(Tnd)$

Ridge regression and shrinking effect

The Ridge regression tries to **shrink the parameters**:

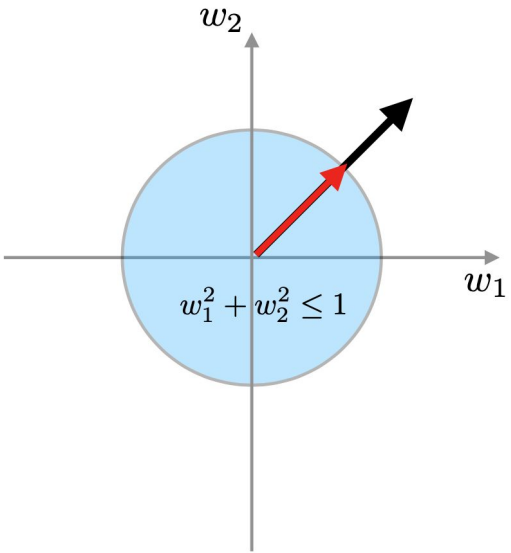


$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

$$\text{such that} \quad \|\boldsymbol{w}\|_2^2 \leq \mu$$

Ridge regression and shrinking effect

The Ridge regression tries to **shrink the parameters**:



$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

$$\text{such that} \quad \|\boldsymbol{w}\|_2^2 \leq \mu$$

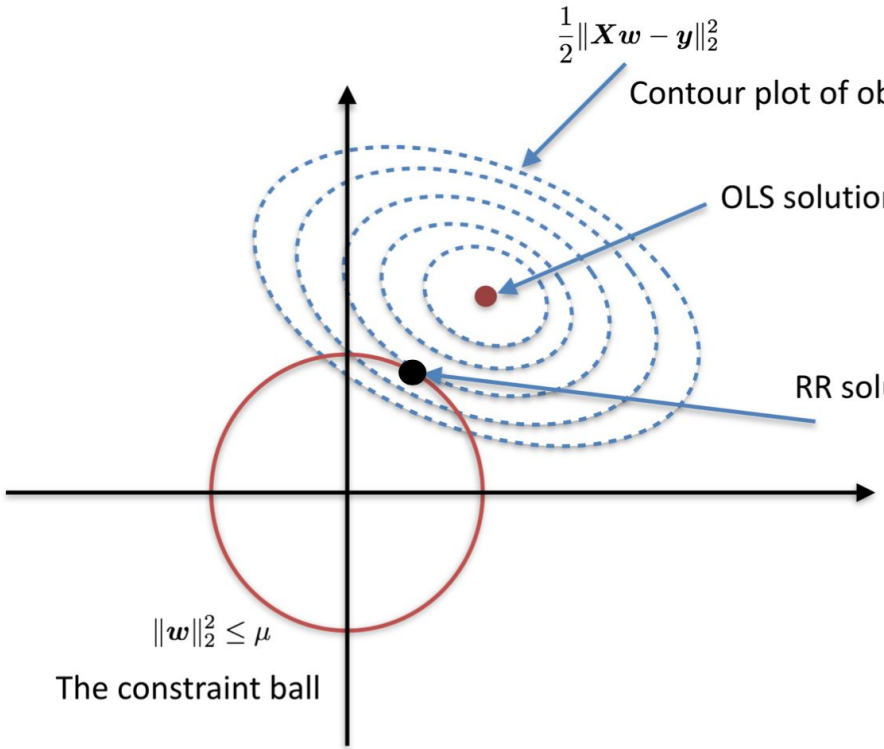
For simplicity assume $\mu = 1$

If the solution lies outside the circle, we just need divide the every coefficient by the length of vector to have it inside the circle.

$[w_1, w_2]$

$$\left[\frac{w_1}{\sqrt{w_1^2 + w_2^2}}, \frac{w_2}{\sqrt{w_1^2 + w_2^2}}\right]$$

A geometric interpretation of Ridge Regression



The concentric ellipse indicates sum of square term. When we move towards smaller ellipse from outer ellipse, the sum of square term will decrease

The RR optimal solution is a point which is common point between ellipse and circle as well as gives a minimum value for RR optimization problem.

Ridge regression reduces the variance of estimate because it limits the space that the solution vector can lie in.

Show that the contours of OLS objective are indeed ellipsoid?

Lasso Regression

Instead of using the squared of the weights to impose the penalty, we take the sum of absolute value of all coefficients as regularization function:

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda \, \|\boldsymbol{w}\|_1$$

Lasso Regression

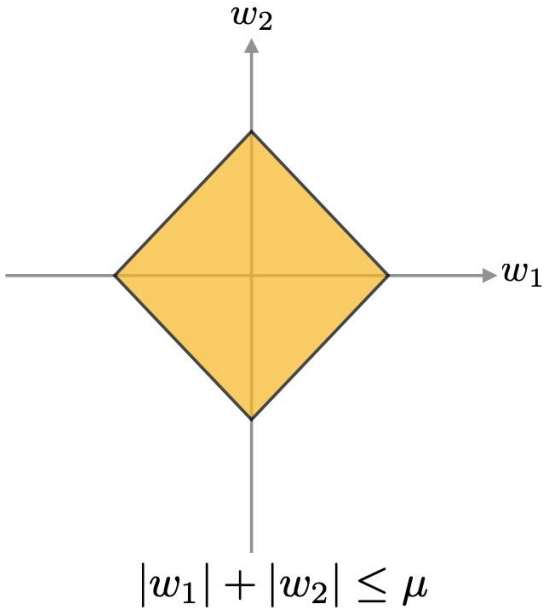
Instead of using the squared of the weights to impose the penalty, we take the sum of absolute value of all coefficients as regularization function:

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda \, \|\boldsymbol{w}\|_1$$

OR

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2$$

$$\text{such that} \quad \|\boldsymbol{w}\|_1 \leq \mu$$

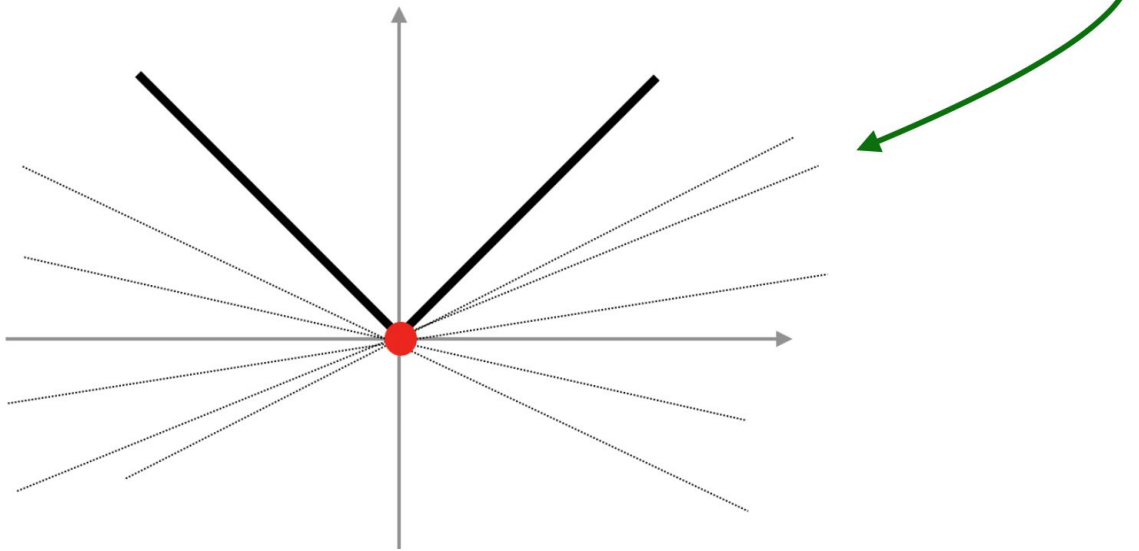


Lasso solution

The loss function is not differentiable, so no analytical (closed-form) solution

Consider the 1 dimensional setting:

$$\arg\min_{w \in \mathbb{R}} \frac{1}{2} \sum_{i=1}^{n} (wx_i - y_i)^2 + \lambda |w|$$

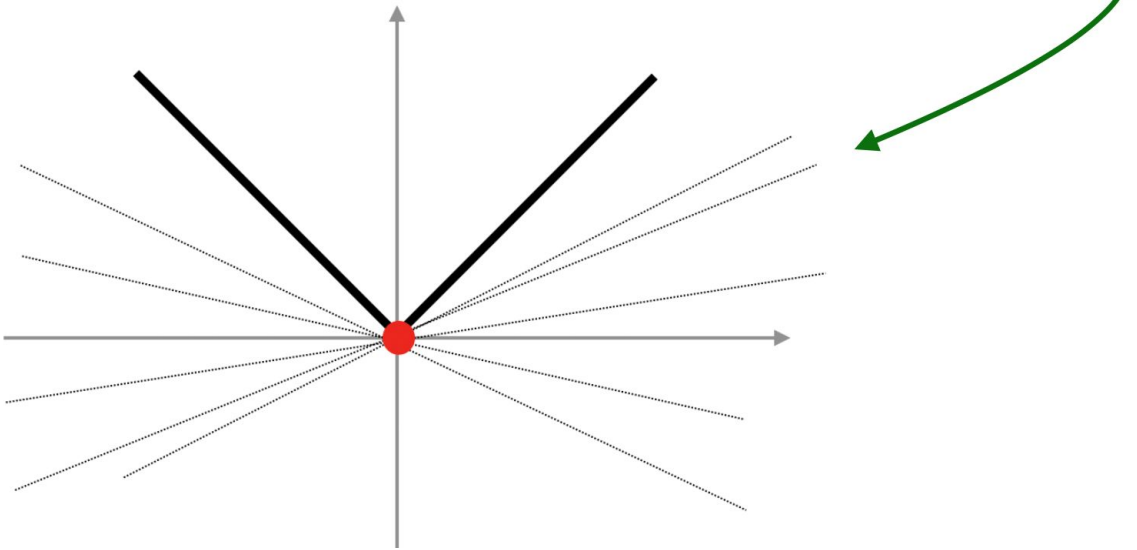


Lasso solution

The loss function is not differentiable, so no analytical (closed-form) solution

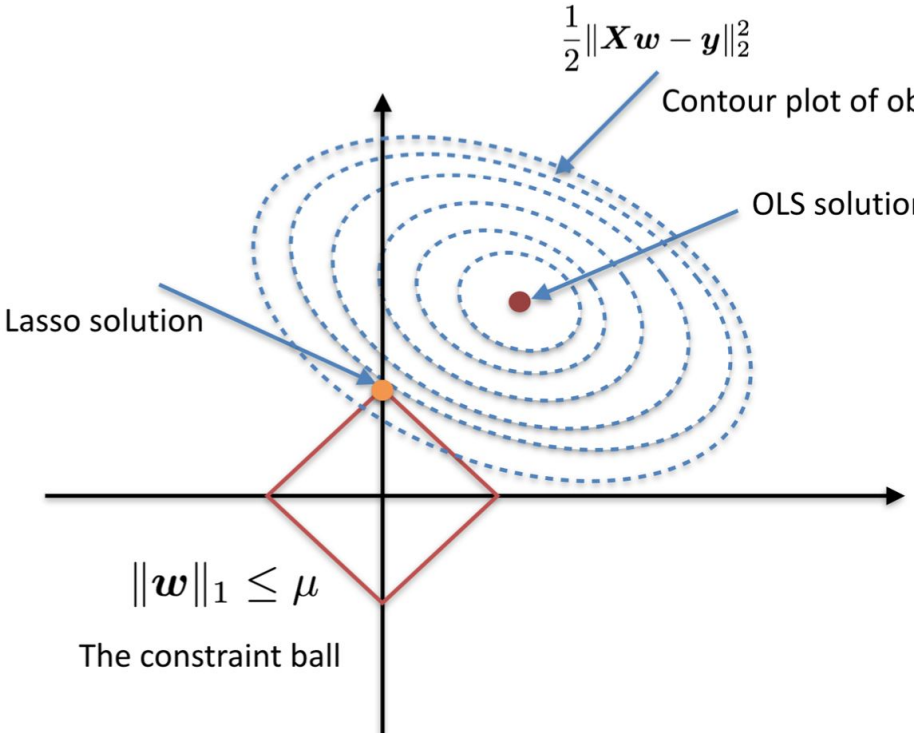
Consider the 1 dimensional setting:

$$\arg\min_{w \in \mathbb{R}} \frac{1}{2} \sum_{i=1}^{n} (wx_i - y_i)^2 + \lambda |w|$$



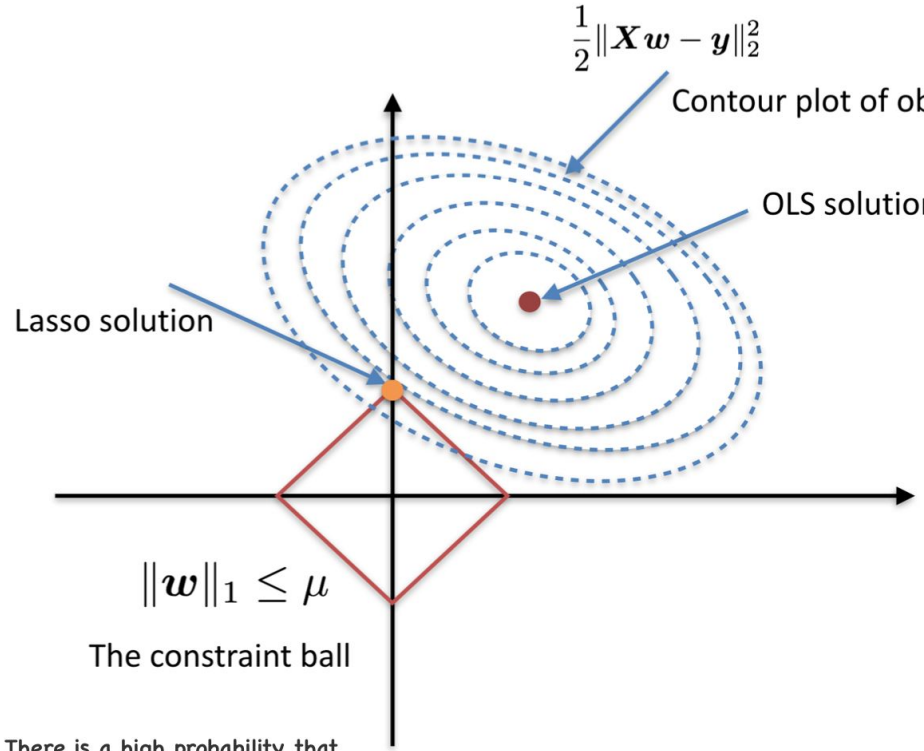
Can be optimized using **Subgradient Descent** algorithm that will be covered later in the course!

A geometric interpretation of Lasso Regression



The Lasso optimal solution is a point which is common point between ellipse and diamond as well as gives a minimum value for the optimization problem.

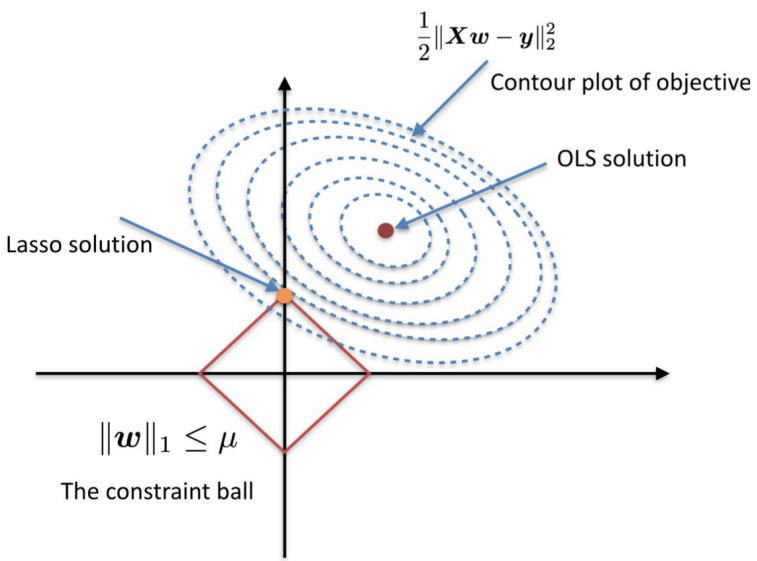
A geometric interpretation of Lasso Regression

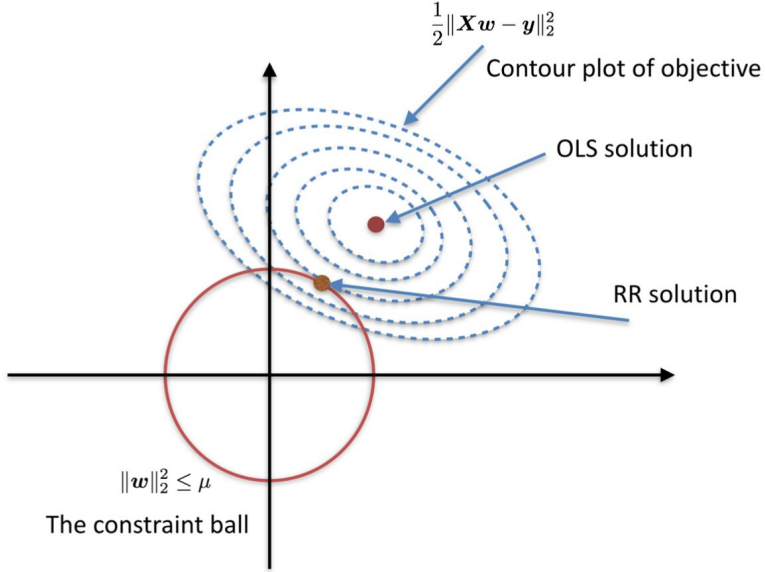


The Lasso optimal solution is a point which is common point between ellipse and diamond as well as gives a minimum value for the optimization problem.

There is a high probability that optimum point falls in the corner point of diamond, leading to a solution with zero value for subset of coefficients (sparsity)

Lasso vs Ridge





Lasso results in sparser solutions (most of coefficients in optimal solution will be zero) **compared to Ridge**!

If you have prior knowledge that only a few variables affect the prediction (i.e., the optimal solution is sparse) use Lasso!

Lasso vs Ridge

Sparsity feature of Lasso can also be used as **feature selection** (finding the most important features among all features), an advantage compared to Ridge

Lasso vs Ridge

Sparsity feature of Lasso can also be used as **feature selection** (finding the most important features among all features), an advantage compared to Ridge

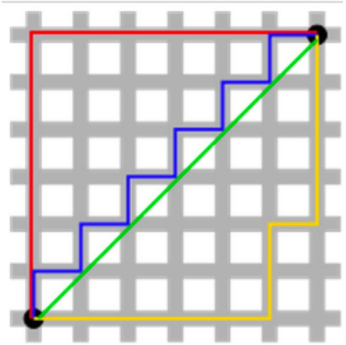
Optimization Ridge objective can be done more efficiently (differentiable vs non-differentiable)

Lasso vs Ridge

Sparsity feature of Lasso can also be used as **feature selection** (finding the most important features among all features), an advantage compared to Ridge

Optimization Ridge objective can be done more efficiently (differentiable vs non-differentiable)

Ridge has a unique solution but Lasso might have multiple optimal solutions!



A toy example: we wanna find the shortest path between two points, the green is based on squared loss (unique) and red, blue, and yellow are based on absolute loss (same cost but different, so not unique)

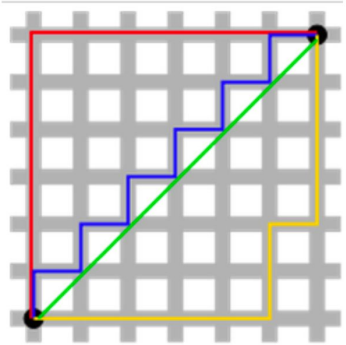
Image: http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/

Lasso vs Ridge

Sparsity feature of Lasso can also be used as **feature selection** (finding the most important features among all features), an advantage compared to Ridge

Optimization Ridge objective can be done more efficiently (differentiable vs non-differentiable)

Ridge has a unique solution but Lasso might have multiple optimal solutions!



A toy example: we wanna find the shortest path between two points, the green is based on squared loss (unique) and red, blue, and yellow are based on absolute loss (same cost but different, so not unique)

Image: http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/

The best of both worlds (**shrinkage of Ridge + sparsity of Lasso**): the Elastic Net Regression which sets the regularization as $\alpha\|\boldsymbol{w}\|_1 + (1-\alpha)\|\boldsymbol{w}\|_2^2$

Regression in scikit-learn

```
>>> from sklearn import linear_model
>>> reg = linear_model.LinearRegression()
>>> reg.fit([[0, 0], [1, 1], [2, 2]], [0, 1, 2])
...
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                 normalize=False)
>>> reg.coef_
array([0.5, 0.5])
```

```
>>> from sklearn import linear_model
>>> reg = linear_model.Ridge(alpha=.5)
>>> reg.fit([[0, 0], [0, 0], [1, 1]], [0, .1, 1])
Ridge(alpha=0.5, copy_X=True, fit_intercept=True, max_iter=None,
      normalize=False, random_state=None, solver='auto', tol=0.001)
>>> reg.coef_
array([0.34545455, 0.34545455])
>>> reg.intercept_
0.13636...
```

```
>>> from sklearn import linear_model
>>> reg = linear_model.Lasso(alpha=0.1)
>>> reg.fit([[0, 0], [1, 1]], [0, 1])
Lasso(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=1000,
   normalize=False, positive=False, precompute=False, random_state=None,
   selection='cyclic', tol=0.0001, warm_start=False)
>>> reg.predict([[1, 1]])
array([0.8])
```

Regularization parameter

Regularization methods **introduce bias** into the regression solution that can **reduce variance** considerably relative to the ordinary least squares (OLS) solution.

In terms of fundamental trade-off:

- Regularization increases training error (why?).
- Regularization decreases variance.

Regularization parameter

Regularization methods **introduce bias** into the regression solution that can **reduce variance** considerably relative to the ordinary least squares (OLS) solution.

In terms of fundamental trade-off:

- Regularization increases training error (why?).
- Regularization decreases variance.

How should you choose λ ?

- **Theory**: as n grows, λ should grow
- **Practice**: optimize validation set or cross-validation error.

This almost always decreases the TEST error.

I am not convinced yet...

It’s a reasonable thing to do, but can we “always use regularization”?
Here are more reasons:

I am not convinced yet...

It's a reasonable thing to do, but can we "always use regularization"?
Here are more reasons:

$$\boldsymbol{w}^*_{\text{RR}} = \left(\boldsymbol{X}^\top \boldsymbol{X} + 2\lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

- The optimal solution is unique (why ?).
- $\boldsymbol{X}^\top \boldsymbol{X}$ does not need to be invertible. (no collinearity issues).

$$\boldsymbol{w}^*_{\text{OLS}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

I am not convinced yet...

It’s a reasonable thing to do, but can we “always use regularization”?
Here are more reasons:

- The optimal solution is unique (why ?).
- $\boldsymbol{X}^\top \boldsymbol{X}$ does not need to be invertible. (no collinearity issues).
- Less sensitive to changes in $\boldsymbol{X}$ or $\boldsymbol{y}$

I am not convinced yet...

It's a reasonable thing to do, but can we "always use regularization"?
Here are more reasons:

- The optimal solution is unique (why ?).
- $\boldsymbol{X}^\top \boldsymbol{X}$ does not need to be invertible. (no collinearity issues).
- Less sensitive to changes in $\boldsymbol{X}$ or $\boldsymbol{y}$
- Gradient descent converge faster (bigger λ means fewer iterations).

I am not convinced yet...

It's a reasonable thing to do, but can we "always use regularization"?
Here are more reasons:

- The optimal solution is unique (why ?).
- $\boldsymbol{X}^\top \boldsymbol{X}$ does not need to be invertible. (no collinearity issues).
- Less sensitive to changes in $\boldsymbol{X}$ or $\boldsymbol{y}$
- Gradient descent converge faster (bigger λ means fewer iterations).
- Stein's paradox: if $d \geq 3$, 'shrinking' moves us closer to 'true' $\boldsymbol{w}$

I am not convinced yet...

It's a reasonable thing to do, but can we "always use regularization"?
Here are more reasons:

- The optimal solution is unique (why ?).
- $\boldsymbol{X}^\top \boldsymbol{X}$ does not need to be invertible. (no collinearity issues).
- Less sensitive to changes in $\boldsymbol{X}$ or $\boldsymbol{y}$
- Gradient descent converge faster (bigger λ means fewer iterations).
- Stein's paradox: if $d \geq 3$, 'shrinking' moves us closer to 'true' $\boldsymbol{w}$
- Worst case: just set small λ and get the same performance. [let's delegate it to cross-validation to decide, if regularization is not needed, we will get tiny or even zero value for regularization parameter]

Appendix

Ridge Regression as Eigenvalue Shrinkage

Let’s look at the effect of regularization from the linear algebraic viewpoint

We saw that the eigenvalues of covariance matrix $\boldsymbol{X}^\top \boldsymbol{X}$ has a lot to say about the solution:

- The optimal solution exists if this matrix is full rank or All the eigenvalues are non-zero

We now ask how regularization affects the optimal solution, in particular transformation of eigenvalues?

Ridge Regression as Eigenvalue Shrinkage

Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the SVD decomposition of the data matrix

We can compute the optimal OLS solution in terms of above decomposition:

$$
\begin{aligned}
\boldsymbol{w}^*_{\text{OLS}} &= \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} \\
&= \left(\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)^{-1} \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)^\top \boldsymbol{y} \\
&= \left(\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)^{-1} \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{y} \\
&= \left(\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top\right)^{-1} \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{y} \\
&= (\boldsymbol{V}^\top)^{-1}(\boldsymbol{\Sigma}^2)^{-1} \underbrace{\boldsymbol{V}^{-1}\boldsymbol{V}}_{=\boldsymbol{I}} \boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{y} \\
&= \boldsymbol{V}\boldsymbol{\Sigma}^{-2}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{y} = \left(\sum_{i=1}^{d} \frac{\sigma_i}{\sigma_i^2} \boldsymbol{v}_i \boldsymbol{u}_i^\top\right) \boldsymbol{y}
\end{aligned}
$$

Ridge Regression as Eigenvalue Shrinkage

Similarly, we can compute the optimal Ridge solution in terms of SVD decomposition of data matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$

$$
\begin{aligned}
\boldsymbol{w}_{\text{RR}}^* &= (\boldsymbol{X}^\top\boldsymbol{X} + 2\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y} \\
&= ((\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top + 2\lambda\boldsymbol{I})^{-1}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top\boldsymbol{y} \\
&= (\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top + 2\lambda\boldsymbol{I})^{-1}\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top\boldsymbol{y} \\
&= (\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top + 2\lambda\boldsymbol{I})^{-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{y} \\
&= (\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top + 2\lambda\boldsymbol{V}\boldsymbol{V}^\top)^{-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{y} \\
&= (\boldsymbol{V}(\boldsymbol{\Sigma}^2 + 2\lambda\boldsymbol{I})\boldsymbol{V}^\top)^{-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{y} \\
&= (\boldsymbol{V}^\top)^{-1}(\boldsymbol{\Sigma}^2 + 2\lambda\boldsymbol{I})^{-1}\boldsymbol{V}^{-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{y} \\
&= \boldsymbol{V}(\boldsymbol{\Sigma}^2 + 2\lambda\boldsymbol{I})^{-1}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{y} \\
&= \boldsymbol{V}\,\text{diag}\left(\frac{\sigma_1}{\sigma_1^2 + 2\lambda}, \ldots, \frac{\sigma_d}{\sigma_d^2 + 2\lambda}\right)\boldsymbol{U}^\top\boldsymbol{y} \\
&= \left(\sum_{i=1}^{d} \frac{\sigma_i}{\sigma_i^2 + 2\lambda}\boldsymbol{v}_i\boldsymbol{u}_i^\top\right)\boldsymbol{y}
\end{aligned}
$$

Ridge Regression as Eigenvalue Shrinkage

Let's compare the solutions:

$$\boldsymbol{w}^*_{\texttt{OLS}} = \left(\sum_{i=1}^{d} \frac{\sigma_i}{\sigma_i^2} \boldsymbol{v}_i \boldsymbol{u}_i^\top \right) \boldsymbol{y}$$

$$\boldsymbol{w}^*_{\texttt{RR}} = \left(\sum_{i=1}^{d} \frac{\sigma_i}{\sigma_i^2 + 2\lambda} \boldsymbol{v}_i \boldsymbol{u}_i^\top \right) \boldsymbol{y}$$

Regularization makes the solution not sensitive to very small eigenvalues (coming from noise)