Lecture 14: Linear Regression Least Squares and Ridge

CS 169 / CS 268
Introduction to Optimization

Ioannis Panageas

November 19, 2026

Regression: predicting a number

Regression predicts a real-valued output y from observed input features.

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}, \qquad \boldsymbol{x}_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}.$$

n is the number of observations. d is the number of features per observation. The values y_i are the observed targets.

A model is a function that maps an input to a prediction $\widehat{y}$. We choose its parameters using the training data.

One feature and one coefficient

$$\hat{y} = wx, \qquad x, w \in \mathbb{R}.$$

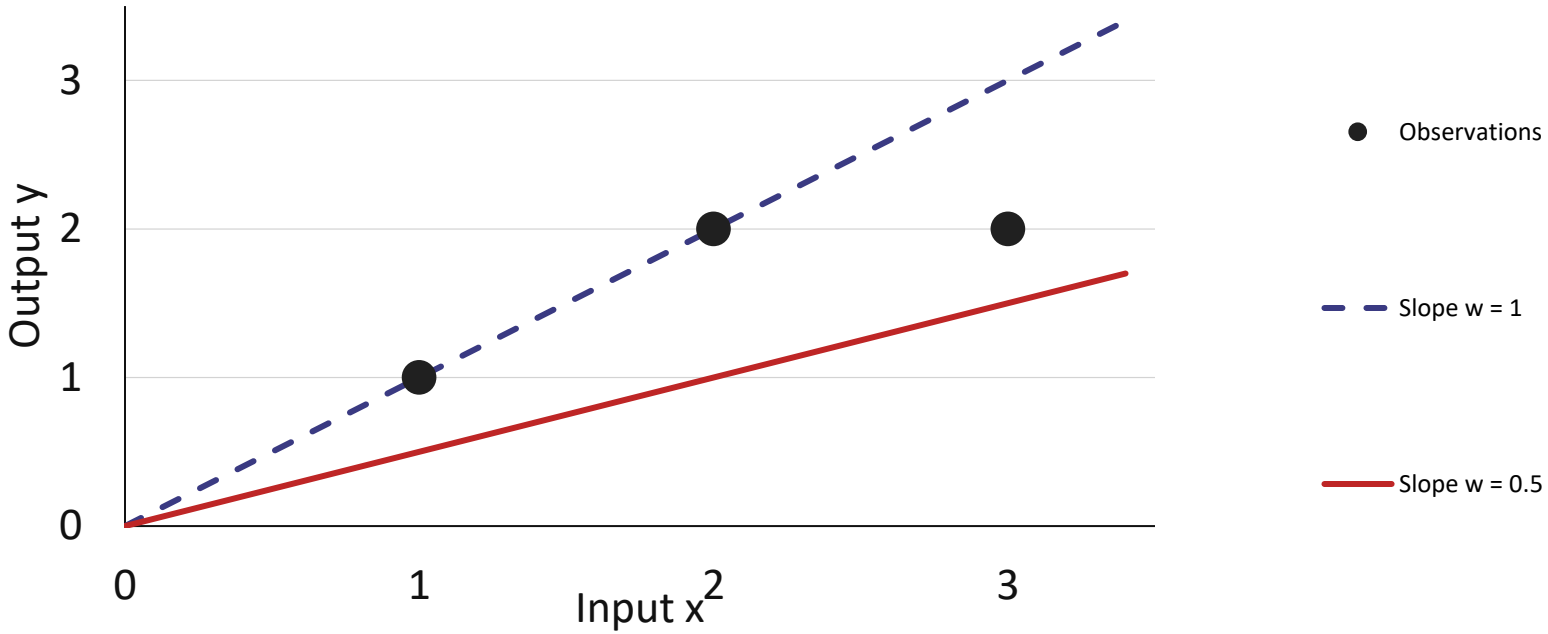
x is the observed input. The coefficient w is the slope we choose.

For this first example, the bias term is fixed at zero. Every candidate line passes through the origin.

The optimization variable is w. The training inputs and targets stay fixed.

A small regression example

$$\mathcal{D} = \{(1,1),(2,2),(3,2)\}, \qquad \hat{y} = wx.$$



Which slope gives the best fit? We need a precise measure of prediction error.

Squared loss and training objective

$$e_i(w) = wx_i - y_i, \qquad \ell_i(w) = \frac{1}{2}(wx_i - y_i)^2.$$

A loss measures error on one observation. The training objective averages the losses.

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(w) = \frac{1}{2n}\sum_{i=1}^{n}(wx_i - y_i)^2.$$

The factor $1/2$ simplifies derivatives. The factor $1/n$ makes this an average. Neither changes the minimizing w.

Solving the one-dimensional example

$$F(w) = \frac{1}{6}\left[(w-1)^2 + (2w-2)^2 + (3w-2)^2\right].$$

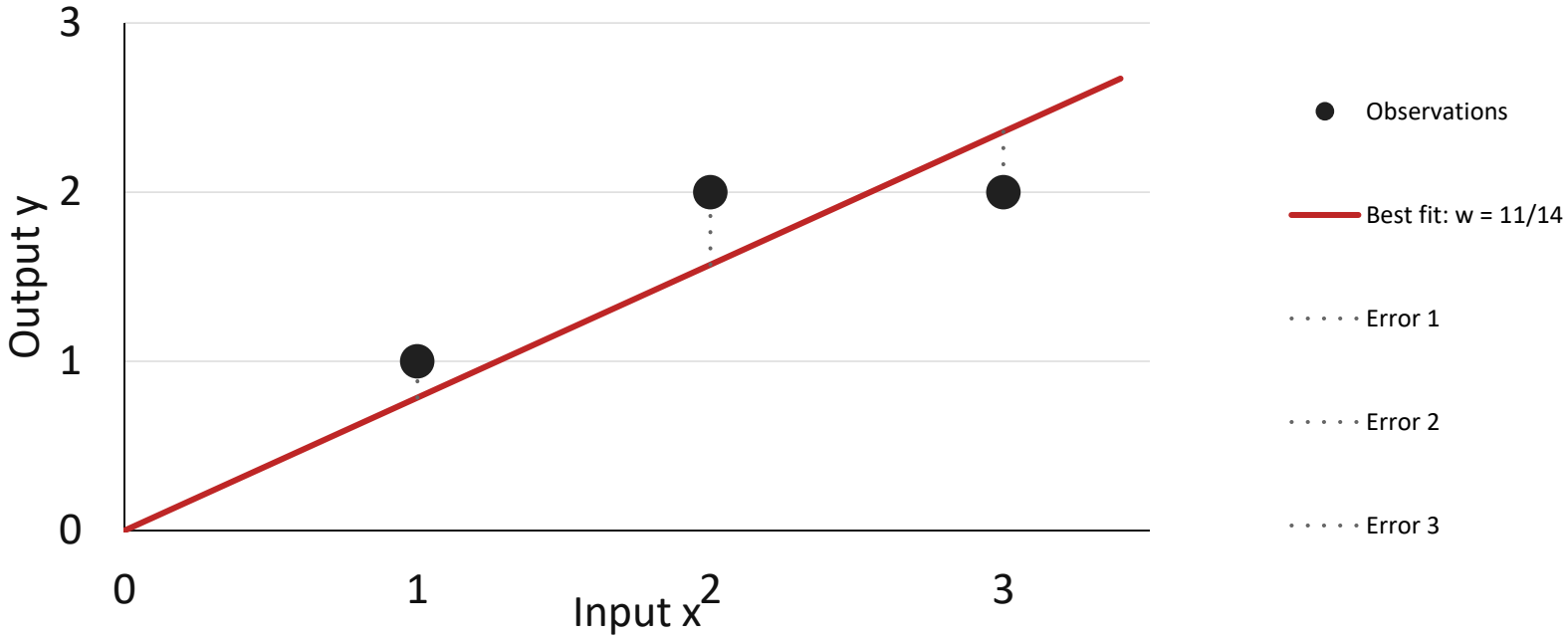
$$F(w) = \frac{14w^2 - 22w + 9}{6}, \qquad F'(w) = \frac{14w - 11}{3}.$$

$$F'(w^*) = 0 \quad \Longrightarrow \quad \boxed{w^* = \frac{11}{14}}.$$

$F''(w) = 14/3 > 0$: the objective is strictly convex, so this stationary point is the unique global minimum.

The fitted line

$$\hat{y} = \frac{11}{14}x, \qquad F(w^*) = \frac{5}{84}.$$



The best fit can have nonzero prediction errors. Least squares minimizes their sum of squares.

Linear regression in many dimensions

$$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T, \qquad \boldsymbol{w} = (w_1, \ldots, w_d)^T.$$

$$\widehat{y}_i = \boldsymbol{x}_i^T \boldsymbol{w} + b = \sum_{j=1}^{d} x_{ij} w_j + b.$$

w_j is the coefficient of feature j. The bias term b is the prediction when all features are zero.

We will analyze arbitrary d, including the case $d > n$ with more coefficients than observations.

Handling the bias term

For fixed $\boldsymbol{w}$, minimizing over the bias term b gives:

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_i \boldsymbol{x}_i, \quad \bar{y} = \frac{1}{n}\sum_i y_i, \quad b = \bar{y} - \bar{\boldsymbol{x}}^T \boldsymbol{w}.$$

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}, \qquad \tilde{y}_i = y_i - \bar{y}.$$

$$\boldsymbol{x}_i^T \boldsymbol{w} + b - y_i = \tilde{\boldsymbol{x}}_i^T \boldsymbol{w} - \tilde{y}_i.$$

When fitting a bias term, use centered data below and omit the tildes. Recover b afterward; penalize only $\boldsymbol{w}$.

The design matrix

$$X = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n.$$

Each row is an observation. Each column is one feature across all observations.

$X\boldsymbol{w} \in \mathbb{R}^n$ contains the predictions. With centered data, add $\bar{y}$ to each entry to recover original-scale predictions.

Least squares in matrix form

$$\min_{w \in \mathbb{R}^d} F(w), \qquad F(w) = \frac{1}{2n}\|Xw - y\|^2.$$

$$\|z\|^2 = \sum_{i=1}^{n} z_i^2 \quad \text{for } z \in \mathbb{R}^n.$$

$$F(w) = \frac{1}{2n}\sum_{i=1}^{n}(x_i^T w - y_i)^2.$$

We now derive the gradient and Hessian to understand the minimizers and how to compute them.

Deriving the gradient

$$\frac{\partial}{\partial w_j}\frac{1}{2}(\boldsymbol{x}_i^T\boldsymbol{w} - y_i)^2 = (\boldsymbol{x}_i^T\boldsymbol{w} - y_i)x_{ij}.$$

$$\frac{\partial F}{\partial w_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}(\boldsymbol{x}_i^T\boldsymbol{w} - y_i).$$

$$\boxed{\nabla F(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i(\boldsymbol{x}_i^T\boldsymbol{w} - y_i) = \frac{1}{n}X^T(X\boldsymbol{w} - \boldsymbol{y}).}$$

The gradient has d coordinates, one derivative for each coefficient.

The Hessian proves convexity

$$H = \nabla^2 F(\boldsymbol{w}) = \frac{1}{n} X^T X = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T.$$

$$\boldsymbol{v}^T H \boldsymbol{v} = \frac{1}{n} \boldsymbol{v}^T X^T X \boldsymbol{v} = \frac{1}{n} \|X\boldsymbol{v}\|^2 \geq 0 \qquad (\boldsymbol{v} \in \mathbb{R}^d).$$

A symmetric matrix is positive semidefinite when its quadratic form is nonnegative in every direction.

The Hessian is positive semidefinite everywhere. Therefore F is convex.

The normal equations

$$\nabla F(\boldsymbol{w}^*) = 0 \quad \Longleftrightarrow \quad \boxed{X^T X \boldsymbol{w}^* = X^T \boldsymbol{y}.}$$

These equations characterize all least-squares minimizers. Convexity makes stationarity sufficient for global optimality.

A least-squares minimizer always exists: the span of the columns of X has a closest point to $\boldsymbol{y}$.

Compute a solution by solving a linear system. A unique coefficient vector requires an additional condition on X.

Why a stationary point is global

Let $\boldsymbol{w}^*$ solve the normal equations and let $\boldsymbol{v} \in \mathbb{R}^d$ be any change in coefficients.

$$F(\boldsymbol{w}^* + \boldsymbol{v}) = \frac{1}{2n}\|(X\boldsymbol{w}^* - \boldsymbol{y}) + X\boldsymbol{v}\|^2$$
$$= F(\boldsymbol{w}^*) + \frac{1}{n}\boldsymbol{v}^T X^T(X\boldsymbol{w}^* - \boldsymbol{y}) + \frac{1}{2n}\|X\boldsymbol{v}\|^2.$$

$$X^T(X\boldsymbol{w}^* - \boldsymbol{y}) = 0 \quad \Longrightarrow \quad \boxed{F(\boldsymbol{w}^* + \boldsymbol{v}) = F(\boldsymbol{w}^*) + \frac{1}{2n}\|X\boldsymbol{v}\|^2.}$$

When are the coefficients unique?

X has full column rank when its d columns are linearly independent. Equivalently, $X\boldsymbol{v} = 0$ only for $\boldsymbol{v} = 0$.

$$F(\boldsymbol{w}^* + \boldsymbol{v}) - F(\boldsymbol{w}^*) = \frac{1}{2n}\|X\boldsymbol{v}\|^2.$$

If X has full column rank, every nonzero change strictly increases F. The minimizer is unique.

Otherwise, some $\boldsymbol{v} \neq 0$ satisfies $X\boldsymbol{v} = 0$. Then $\boldsymbol{w}^* + t\boldsymbol{v}$ is a minimizer for every real t.

An example with dependent features

$$X = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \qquad n = d = 2.$$

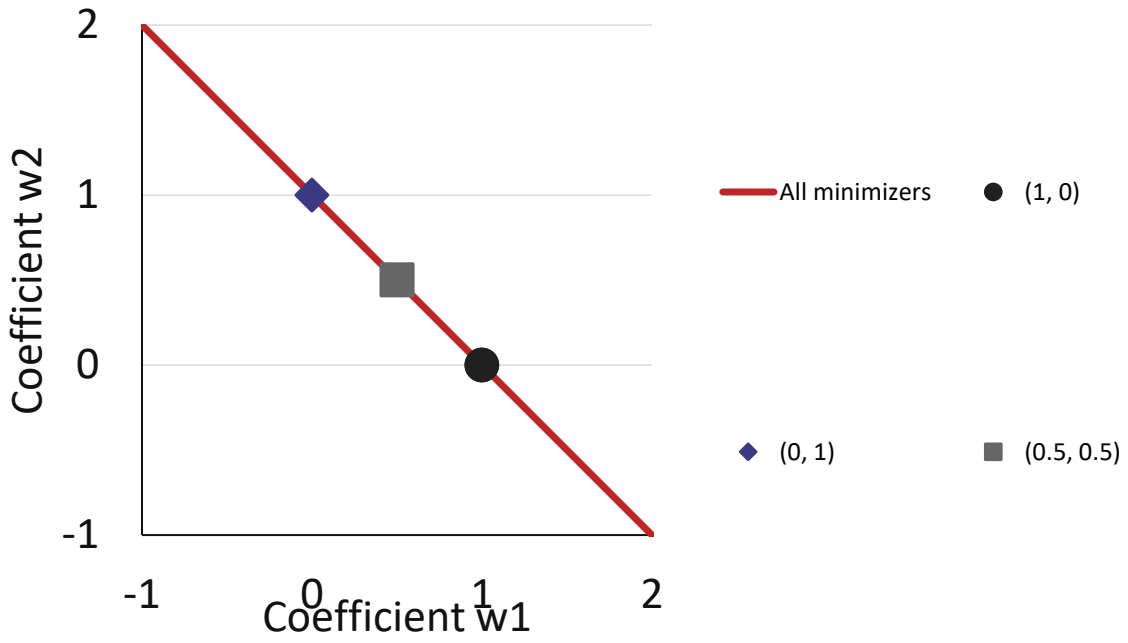
$$X\boldsymbol{w} = \begin{pmatrix} -(w_1 + w_2) \\ w_1 + w_2 \end{pmatrix}, \qquad F(\boldsymbol{w}) = \frac{1}{2}(w_1 + w_2 - 1)^2.$$

$$F(\boldsymbol{w}) = 0 \iff \boxed{w_1 + w_2 = 1.}$$

There are infinitely many coefficient vectors with the same training predictions.

Different coefficients, same training fit

$$w^* = (1,0),\ (0,1),\ (1/2,1/2),\ldots \qquad F(w^*) = 0.$$



On a new input $x = (1,0)^T$, the first two models predict 1 and 0. Training agreement does not imply agreement everywhere.

Gradient descent for least squares

$$\boxed{\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{\eta}{n} X^T (X\boldsymbol{w}_k - \boldsymbol{y}).}$$

Each step uses the current prediction errors. We can compute $X\boldsymbol{w}_k$ and then multiply by X^T, without forming $X^T X$.

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \eta_k \boldsymbol{x}_{i_k} (\boldsymbol{x}_{i_k}^T \boldsymbol{w}_k - y_{i_k}).$$

For SGD, choose i_k uniformly from $\{1, \ldots, n\}$. The sampled gradient has expectation $\nabla F(\boldsymbol{w}_k)$.

High-dimensional convergence of GD

Assume X has full column rank. Let $\mu > 0$ and L be the smallest and largest eigenvalues of $H = X^T X/n$.

$$\boldsymbol{e}_k = \boldsymbol{w}_k - \boldsymbol{w}^*, \qquad \boxed{\boldsymbol{e}_{k+1} = (I - \eta H)\boldsymbol{e}_k.}$$

Along an eigenvector of H with eigenvalue s, one step multiplies the error component by $1 - \eta s$.

$$\eta = \frac{1}{L} \quad \Longrightarrow \quad \|\boldsymbol{w}_k - \boldsymbol{w}^*\| \leq \left(1 - \frac{\mu}{L}\right)^k \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|.$$

Ridge regularization

Regularization adds a penalty to the training objective. Ridge uses the squared Euclidean norm of the coefficient vector.

$$F_\lambda(\boldsymbol{w}) = \frac{1}{2n}\|X\boldsymbol{w} - \boldsymbol{y}\|^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2, \qquad \lambda \geq 0.$$

The parameter λ controls the penalty. Setting $\lambda = 0$ gives least squares. Larger λ penalizes large coefficient norms more strongly.

For the centered-data model, we penalize $\boldsymbol{w}$. Recover the unpenalized bias term from $b = \bar{y} - \boldsymbol{w}^T\bar{\boldsymbol{x}}$.

The ridge normal equations

Fix $\lambda > 0$. Differentiate the objective with respect to the coefficient vector:

$$\nabla F_\lambda(\boldsymbol{w}) = \frac{1}{n} X^T (X\boldsymbol{w} - \boldsymbol{y}) + \lambda \boldsymbol{w}.$$

$$\nabla^2 F_\lambda(\boldsymbol{w}) = \frac{1}{n} X^T X + \lambda I = H + \lambda I.$$

$$\nabla F_\lambda(\boldsymbol{w}_\lambda) = 0 \quad \Longleftrightarrow \quad \boxed{(H + \lambda I)\boldsymbol{w}_\lambda = \frac{1}{n} X^T \boldsymbol{y}.}$$

Compute $\boldsymbol{w}_\lambda$ by solving this linear system.

Why ridge has a unique minimizer

For $\lambda > 0$ and every nonzero vector $\boldsymbol{v}$,

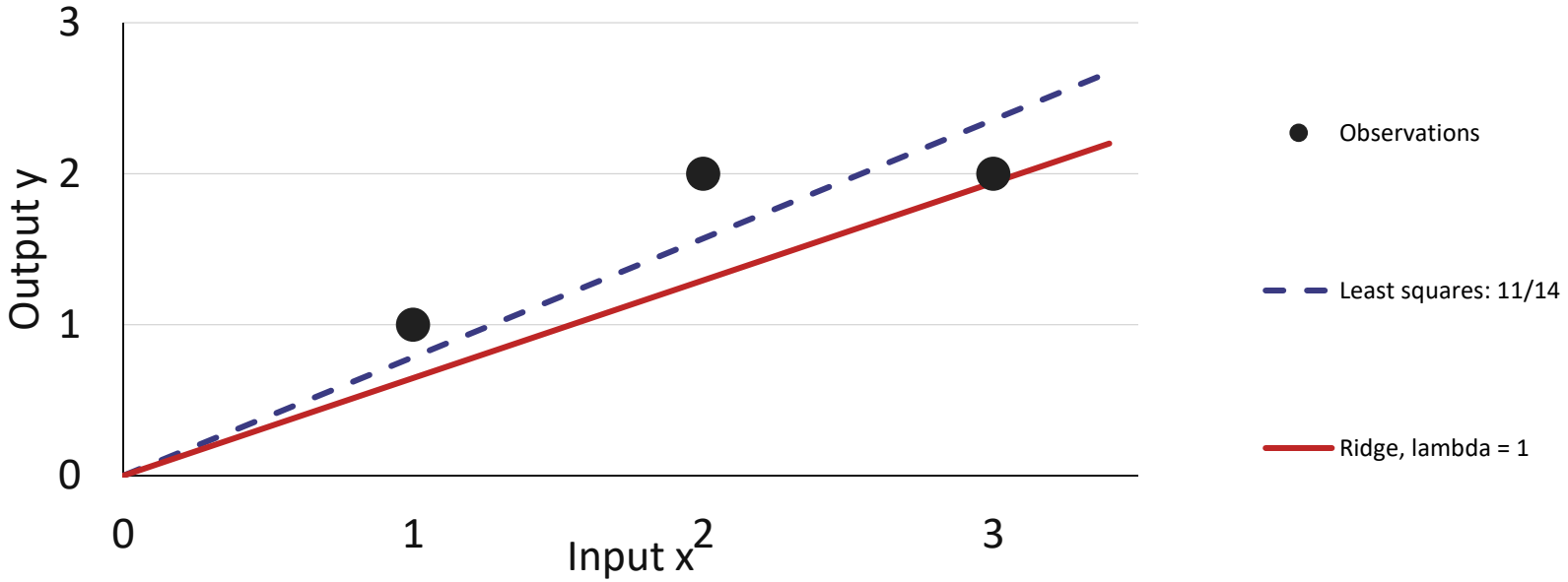
$$\begin{aligned} \boldsymbol{v}^T(H + \lambda I)\boldsymbol{v} &= \frac{1}{n}\|X\boldsymbol{v}\|^2 + \lambda\|\boldsymbol{v}\|^2 \\ &\geq \lambda\|\boldsymbol{v}\|^2 > 0. \end{aligned}$$

Thus $H + \lambda I$ is positive definite. The normal equations have exactly one solution $\boldsymbol{w}_\lambda$, and

$$\begin{aligned} F_\lambda(\boldsymbol{w}_\lambda + \boldsymbol{v}) - F_\lambda(\boldsymbol{w}_\lambda) &= \frac{1}{2}\boldsymbol{v}^T(H + \lambda I)\boldsymbol{v} \\ &\geq \frac{\lambda}{2}\|\boldsymbol{v}\|^2 > 0 \quad (\boldsymbol{v} \neq 0). \end{aligned}$$

One feature: ridge changes the slope

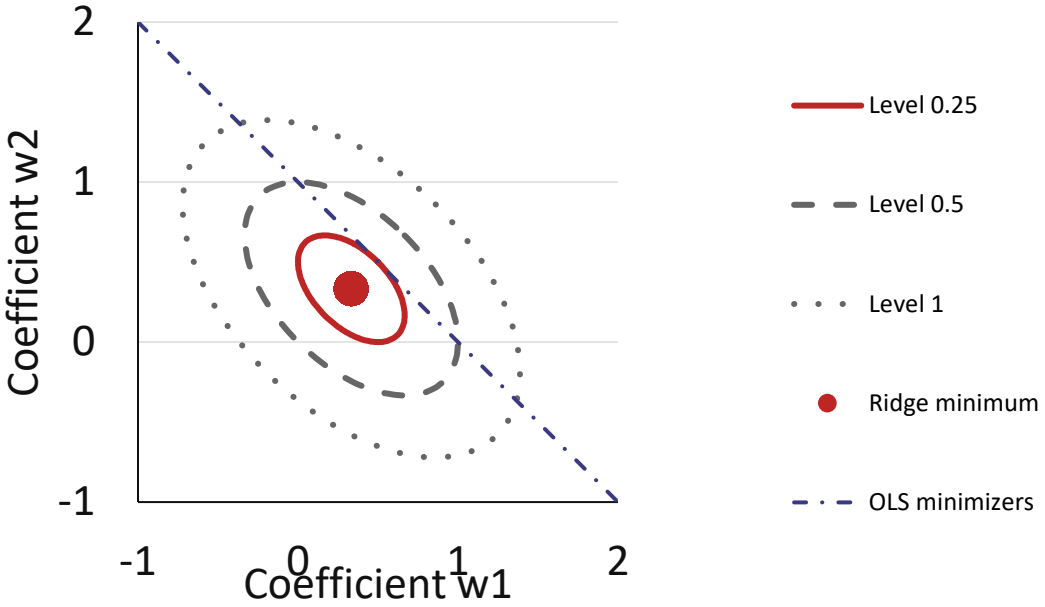
Bias term $b = 0$: $\quad w_\lambda = \dfrac{11}{14 + 3\lambda}, \quad w_1 = \dfrac{11}{17}.$



Ridge trades training fit for a smaller coefficient. Use separate data to compare prediction accuracy.

Ridge with dependent features

$$F_1(\boldsymbol{w}) = \frac{1}{2}(w_1 + w_2 - 1)^2 + \frac{1}{2}(w_1^2 + w_2^2).$$



$$\boldsymbol{w}_1 = (1/3, 1/3)^T, \qquad F_1(\boldsymbol{w}_1) = 1/6.$$

Gradient descent for ridge

Fix $\lambda > 0$. Let $\mu = \lambda_{\min}(H) \geq 0$ and $L = \lambda_{\max}(H)$. Adding λI shifts every eigenvalue by λ.

$$\mu_\lambda = \mu + \lambda > 0, \qquad L_\lambda = L + \lambda, \qquad \kappa_\lambda = \frac{L + \lambda}{\mu + \lambda}.$$

The condition number is κ_λ. With $\eta = 1/(L + \lambda)$, the eigenvector analysis gives

$$\boxed{\|\boldsymbol{w}_k - \boldsymbol{w}_\lambda\| \leq \left(1 - \frac{\mu + \lambda}{L + \lambda}\right)^k \|\boldsymbol{w}_0 - \boldsymbol{w}_\lambda\|.}$$

Ridge removes flat directions and gives geometric convergence.

A norm constraint and its Lagrangian

A coefficient budget $R > 0$ gives the constrained problem

$$\min_{w} F(w) \qquad \text{s.t.} \qquad \frac{1}{2}\|w\|^2 \le \frac{1}{2}R^2.$$

$$\mathcal{L}(w, \lambda) = F(w) + \frac{\lambda}{2}(\|w\|^2 - R^2) = F_\lambda(w) - \frac{\lambda}{2}R^2.$$

$$\nabla F(w^*) + \lambda w^* = 0, \qquad \lambda \ge 0, \qquad \lambda(\|w^*\|^2 - R^2) = 0.$$

For fixed λ, the last term in $\mathcal{L}$ is constant in w. The KKT stationarity equation is exactly the ridge equation.

Reading

Chong, Lu, and Żak, *An Introduction to Optimization*, fifth edition.

Section 28.1: least-squares linear regression.

Section 28.3: model selection by regularization.

Optional: Section 28.2, model selection by cross-validation. Use these slides for our normalization and the gradient-descent derivation.