CSC 246 Machine Learning | Lecture 3

Regularization

Lecturer: Jiaming Liang | *September 10, 2026*

1 Overfitting

Linear regression may appear to be a simple model. However, it can become highly flexible when the number of features p is large relative to the number of training samples n.

Recall that $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$. If $p+1 > n$, then $\mathbf{X}$ cannot have full column rank. Consequently, $\mathbf{X}^\top\mathbf{X}$ is singular, and the least-squares solution is generally not unique. More generally, as the number of features increases, the model has more freedom to fit the training data. For example, if $p+1 = n$ and $\mathbf{X}$ is invertible, then

$$\widehat{\boldsymbol{\theta}} = \mathbf{X}^{-1}\mathbf{y}, \qquad \mathbf{X}\widehat{\boldsymbol{\theta}} = \mathbf{y},$$

so the training error is exactly zero. However, a small training error does not necessarily imply good prediction on new data.

A common way to increase the flexibility of linear regression is to introduce nonlinear transformations of the input. For a scalar input x, consider

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_p x^p + \varepsilon.$$

This is called *polynomial regression*. Define

$$\mathbf{x} = \begin{bmatrix} 1 & x & x^2 & \cdots & x^p \end{bmatrix}^\top.$$

Then the model is still linear in the parameters

$$y = \boldsymbol{\theta}^\top\mathbf{x} + \varepsilon.$$

Thus, nonlinear feature transformations can make a linear model much more flexible. If the model is too flexible, it may fit not only the underlying pattern in the training data but also the noise. This is called *overfitting*:

$$\text{low training error} \quad \not\Rightarrow \quad \text{low prediction error on new data.}$$

One way to reduce overfitting is to control model complexity. For polynomial regression, we could use a lower polynomial degree or select only useful features. Another standard approach is *regularization*.

2 Regularization

Regularization modifies the training objective by penalizing overly complex parameter values. The goal is to trade a small increase in training error for better prediction on new data.

2.1 Ridge regression

For linear regression, L_2 regularization gives

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \right\}, \qquad \lambda \geq 0.$$

This is called *ridge regression*.

The parameter λ controls the strength of regularization:

$$\lambda = 0 \quad \Longrightarrow \quad \text{ordinary least squares,}$$

while a larger λ puts a stronger penalty on large coefficients.

Why can smaller parameters help prevent overfitting?

Consider the linear model

$$\widehat{y} = \boldsymbol{\theta}^\top \mathbf{x}.$$

A small change in the input gives

$$\Delta\widehat{y} = \boldsymbol{\theta}^\top \Delta\mathbf{x}.$$

Large coefficients can therefore make the prediction highly sensitive to small variations in the input. This is especially problematic when features are strongly correlated: large positive and negative coefficients may nearly cancel on the training data but become unstable on new data.

Regularization discourages such large coefficients and tends to produce a more stable model:

$$\text{smaller coefficients} \;\Rightarrow\; \text{less sensitivity to data} \;\Rightarrow\; \text{lower variance} \;\Rightarrow\; \text{less overfitting.}$$

Taking the gradient of the ridge objective and setting it to zero gives

$$\left(\mathbf{X}^\top\mathbf{X} + n\lambda\mathbf{I}\right)\widehat{\boldsymbol{\theta}} = \mathbf{X}^\top\mathbf{y}.$$

For $\lambda > 0$,

$$\mathbf{X}^\top\mathbf{X} + n\lambda\mathbf{I}$$

is positive definite and therefore invertible. Hence,

$$\widehat{\boldsymbol{\theta}} = \left(\mathbf{X}^\top\mathbf{X} + n\lambda\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}.$$

Thus, ridge regression has two useful effects:

$$\text{reduces overfitting} \qquad \text{and} \qquad \text{improves conditioning / ensures uniqueness.}$$

Because the size of a coefficient depends on the scale of its feature, features are usually standardized before applying regularization.

In practice, the regularization parameter λ is selected using validation data or cross-validation.

2.2 Lasso regression

Instead of penalizing the squared L_2 norm of the parameters, we can use the L_1 norm:

$$\|\boldsymbol{\theta}\|_1 = \sum_{j=0}^{p} |\theta_j|.$$

This gives

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \right\}, \qquad \lambda \geq 0.$$

This is called *Lasso regression*, where Lasso stands for *Least Absolute Shrinkage and Selection Operator*.

As with ridge regression,

$$\lambda = 0 \quad \Longrightarrow \quad \text{ordinary least squares},$$

while a larger λ imposes stronger regularization.

The important difference from ridge regression is that L_1 regularization can produce *sparse* solutions:

many coefficients can become exactly zero.

If $\widehat{\theta}_j = 0$, then the corresponding feature x_j does not contribute to the prediction. Thus,

$$L_1 \text{ regularization } \Rightarrow \text{ shrinkage} + \text{sparsity } \Rightarrow \text{ automatic feature selection.}$$

Ridge regression typically makes all coefficients smaller, while Lasso can set some of them exactly to zero. Unlike ridge regression, Lasso does not generally have a closed-form solution because the L_1 norm is not differentiable at zero. It is therefore usually solved using numerical optimization methods. In practice, λ is again selected using validation data or cross-validation.

2.3 Elastic net

Ridge and Lasso encourage different types of solutions:

$$\begin{aligned} L_2 \text{ regularization} &\quad \Rightarrow \quad \text{small coefficients,} \\ L_1 \text{ regularization} &\quad \Rightarrow \quad \text{sparse coefficients.} \end{aligned}$$

It is also natural to combine the two regularization terms

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda_1 \|\boldsymbol{\theta}\|_1 + \lambda_2 \|\boldsymbol{\theta}\|_2^2 \right\},$$

where $\lambda_1, \lambda_2 \geq 0$. This is called *elastic net regularization*.

Elastic net combines the effects of ridge and Lasso:

$$\text{Elastic net} = L_1 \text{ regularization} + L_2 \text{ regularization}.$$

The L_1 term encourages sparsity and feature selection, while the L_2 term encourages small and stable coefficients. Special cases include

$$\lambda_1 = 0 \quad \Longrightarrow \quad \text{ridge regression},$$

and

$$\lambda_2 = 0 \quad \Longrightarrow \quad \text{Lasso regression}.$$

Thus, elastic net provides a flexible compromise between coefficient shrinkage and sparsity.

3 Bayesian Perspective

In maximum likelihood estimation, $\boldsymbol{\theta}$ is treated as an unknown but fixed parameter. In the Bayesian perspective, we instead treat $\boldsymbol{\theta}$ as a random variable and place a prior distribution on it.

By Bayes' rule,

$$p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X})}.$$

Since $p(\mathbf{y} \mid \mathbf{X})$ does not depend on $\boldsymbol{\theta}$,

$$p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) \propto \underbrace{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\text{prior}}.$$

Here, the likelihood $p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})$ measures how plausible the observed data are for a given $\boldsymbol{\theta}$, the prior $p(\boldsymbol{\theta})$ represents our belief about $\boldsymbol{\theta}$ before observing the data, and the posterior $p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y})$ is the updated belief after observing the data.

A common point estimate is the *maximum a posteriori* (MAP) estimator:

$$\widehat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}).$$

3.1 Gaussian Prior and Ridge Regression

Recall that under Gaussian noise,

$$p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2\right).$$

Now assume a Gaussian prior $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I})$, so that

$$p(\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2\tau^2} \|\boldsymbol{\theta}\|_2^2\right).$$

Therefore,

$$p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 - \frac{1}{2\tau^2}\|\boldsymbol{\theta}\|_2^2\right).$$

Taking the negative logarithm and multiplying by $2\sigma^2/n$ gives

$$\widehat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\min_{\boldsymbol{\theta}} \left\{\frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2\right\},$$

where $\lambda = \frac{\sigma^2}{n\tau^2}$. Thus,

$$\text{Gaussian likelihood} + \text{Gaussian prior} \quad \Longrightarrow \quad \text{ridge regression.}$$

The prior $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$ expresses a preference for parameters close to zero. A smaller τ^2 corresponds to a stronger prior and therefore stronger regularization.

3.2 Laplace Prior and Lasso

If instead each coefficient has a Laplace prior,

$$p(\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{b}\|\boldsymbol{\theta}\|_1\right),$$

then under the same Gaussian likelihood,

$$\widehat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\min_{\boldsymbol{\theta}} \left\{\frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1\right\},$$

where $\lambda = \frac{2\sigma^2}{nb}$. Thus,

$$\text{Gaussian likelihood} + \text{Laplace prior} \quad \Longrightarrow \quad \text{Lasso regression.}$$

Regularization can therefore be interpreted from a Bayesian perspective as incorporating prior beliefs about the model parameters.

4 Bias-Variance Tradeoff

The ultimate goal of supervised learning is to perform well on new data drawn from the underlying data distribution. For squared-error regression, the population objective is

$$E_{\text{new}}(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[(f(\mathbf{x}) - y)^2\right].$$

The underlying data distribution is unknown. Given a training dataset $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we replace the population expectation by the empirical average

$$E_{\text{train}}(f; \mathcal{T}) = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2.$$

The learned model $\widehat{f}(\mathbf{x};\mathcal{T})$ is obtained from this finite-sample surrogate, possibly together with a regularization term.

The key point is that the training dataset $\mathcal{T}$ is itself random. If we draw a different training dataset, we may obtain a different learned model. We therefore study the expected population performance of the learning procedure:

$$\overline{E}_{\text{new}} = \mathbb{E}_{\mathcal{T}}\left[E_{\text{new}}\left(\widehat{f}(\cdot;\mathcal{T})\right)\right].$$

Before analyzing this quantity, we first recall a basic bias-variance decomposition.

4.1 Basic Bias-Variance Decomposition

Suppose z is a random estimate of a fixed quantity z_0, and define

$$\bar{z} = \mathbb{E}[z].$$

The bias of z is

$$\text{Bias}(z) = \bar{z} - z_0,$$

and its variance is

$$\text{Var}(z) = \mathbb{E}\left[(z-\bar{z})^2\right].$$

To decompose the expected squared error, write $z - z_0 = (z-\bar{z}) + (\bar{z} - z_0)$. Then

$$\begin{aligned}\mathbb{E}\left[(z-z_0)^2\right] &= \mathbb{E}\left[((z-\bar{z}) + (\bar{z}-z_0))^2\right] \\ &= \mathbb{E}\left[(z-\bar{z})^2\right] + 2(\bar{z}-z_0)\mathbb{E}\left[z-\bar{z}\right] + (\bar{z}-z_0)^2.\end{aligned}$$

Since $\mathbb{E}\left[z-\bar{z}\right] = 0$, we obtain

$$\mathbb{E}\left[(z-z_0)^2\right] = \text{Var}(z) + \text{Bias}(z)^2.$$

Thus, expected squared error = Bias^2 + Variance.

4.2 Bias and Variance in Supervised Learning

Now consider the regression model

$$y = f_0(\mathbf{x}) + \varepsilon, \qquad \mathbb{E}[\varepsilon] = 0, \qquad \text{Var}(\varepsilon) = \sigma^2,$$

where f_0 is the true regression function.

For a fixed test input $\mathbf{x}_\star$, the learned prediction $\widehat{f}(\mathbf{x}_\star;\mathcal{T})$ is random because it depends on the random training dataset $\mathcal{T}$. Define the average learned model

$$\overline{f}(\mathbf{x}) = \mathbb{E}_{\mathcal{T}}\left[\widehat{f}(\mathbf{x};\mathcal{T})\right].$$

At a fixed $\mathbf{x}_\star$, the quantities in the basic bias-variance decomposition correspond to

$$z_0 \longleftrightarrow f_0(\mathbf{x}_\star), \qquad z \longleftrightarrow \widehat{f}(\mathbf{x}_\star;\mathcal{T}), \qquad \bar{z} \longleftrightarrow \overline{f}(\mathbf{x}_\star).$$

Applying the basic decomposition gives

$$\mathbb{E}_{\mathcal{T}}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - f_0(\mathbf{x}_\star)\right)^2\right] = \underbrace{\left(\overline{f}(\mathbf{x}_\star) - f_0(\mathbf{x}_\star)\right)^2}_{\text{Bias}^2 \text{ at } \mathbf{x}_\star} + \underbrace{\mathbb{E}_{\mathcal{T}}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - \overline{f}(\mathbf{x}_\star)\right)^2\right]}_{\text{Variance at } \mathbf{x}_\star}.$$

A new test output satisfies

$$y_\star = f_0(\mathbf{x}_\star) + \varepsilon_\star.$$

For fixed $\mathbf{x}_\star$ and $\mathcal{T}$,

$$\mathbb{E}_{\varepsilon_\star}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - y_\star\right)^2\right] = \mathbb{E}_{\varepsilon_\star}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - f_0(\mathbf{x}_\star) - \varepsilon_\star\right)^2\right] = \left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - f_0(\mathbf{x}_\star)\right)^2 + \sigma^2,$$

because $\mathbb{E}[\varepsilon_\star] = 0$ and $\mathbb{E}[\varepsilon_\star^2] = \sigma^2$. Taking expectation over the training dataset and using the above bias-variance decomposition yield

$$\begin{aligned}
\overline{E}_{\text{new}} &= \mathbb{E}_{\mathcal{T}}\left[E_{\text{new}}\left(\widehat{f}(\cdot;\mathcal{T})\right)\right] = \mathbb{E}_{\mathcal{T}}\left[\mathbb{E}_{(\mathbf{x}_\star,y_\star)\sim\mathcal{D}}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - y_\star\right)^2\right]\right] \\
&= \mathbb{E}_{\mathcal{T}}\left[\mathbb{E}_{\varepsilon_\star}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - y_\star\right)^2\right]\right] \\
&= \underbrace{\left(\overline{f}(\mathbf{x}_\star) - f_0(\mathbf{x}_\star)\right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{\mathcal{T}}\left[\left(\widehat{f}(\mathbf{x}_\star;\mathcal{T}) - \overline{f}(\mathbf{x}_\star)\right)^2\right]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{irreducible error}}.
\end{aligned}$$

Hence,

$$\overline{E}_{\text{new}} = \text{Bias}^2 + \text{Variance} + \text{irreducible error}.$$

The three terms have different meanings:

- **Bias:** how far the average learned model $\overline{f}$ is from the true function f_0.
- **Variance:** how much the learned model changes when the training dataset $\mathcal{T}$ changes.
- **Irreducible error:** the noise σ^2 in the data, which cannot be removed by choosing a better model.

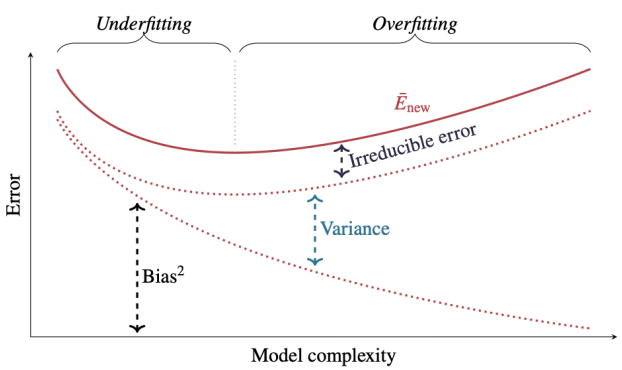

Figure 1: Bias, variance, and irreducible error as model complexity changes.

4.3 Model Complexity and the Bias-Variance Tradeoff

Model complexity affects bias and variance in opposite ways:

	Bias	Variance
simple model	high	low
complex model	low	high

A model that is too simple tends to *underfit*: it cannot represent the underlying relationship well and therefore has high bias. A model that is too flexible tends to *overfit*: it adapts strongly to the particular training dataset, including its noise, and therefore has high variance.

Good generalization therefore requires balancing bias and variance. For ridge regression,

$$\widehat{\boldsymbol{\theta}}_\lambda = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \right\},$$

the regularization parameter λ controls the effective model complexity. Typically,

$$\lambda \uparrow \implies \text{model complexity} \downarrow \implies \text{bias} \uparrow, \qquad \text{variance} \downarrow .$$

Choosing λ therefore amounts to finding a good bias-variance tradeoff.

The amount of training data also affects the variance. Typically,

$$n \uparrow \implies \text{variance} \downarrow,$$

while the bias is approximately unchanged.

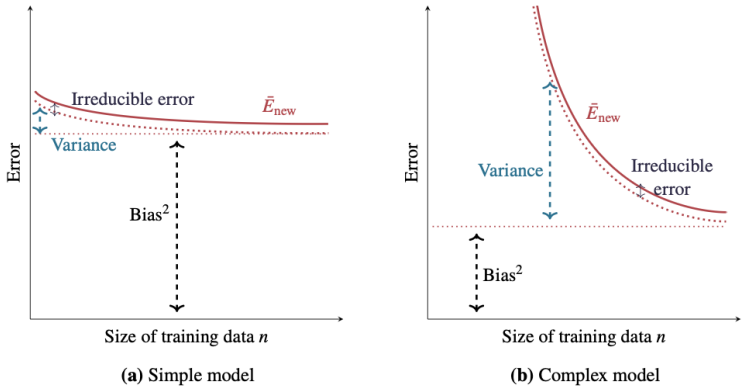

Figure 2: Typical effect of the training-data size on bias and variance.

With more training data, the empirical objective becomes a better approximation of the population objective. As a result, the learned model becomes less sensitive to the particular training sample, and the variance decreases.