EMMA ding

emmading.com
info@datainterviewpro.com

Regularization

📌 Interview Questions

- What are L1 and L2 regularizations?
- What are the differences between them?
- Why does L1 cause parameter sparsity whereas L2 does not?

What Is Regularization?

- Regularization is an umbrella term. It is the process of introducing a regularization term to the loss function of a model.
- This technique adds a **penalizing term** for bringing in more features with the objective function, i.e. the penalizing term's value is higher when the model is more complex. Hence, it tries to push the coefficients for many features to zero and reduce the loss function.

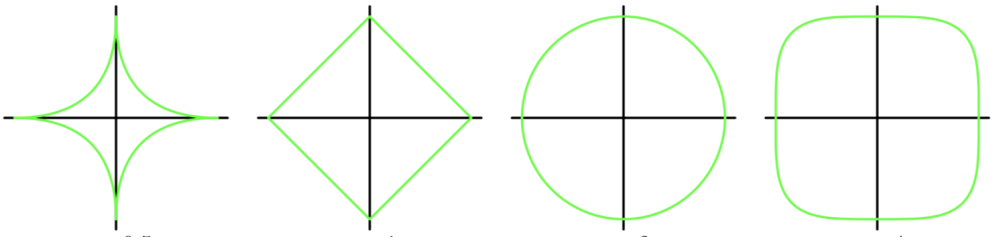
✏️ **Goal**: improve the generalization of a model. It becomes necessary when the model begins to overfit.

▼ When to Use Regularization

- To handle **collinearity** (high correlation among features).
- To filtering out **noise** from data.
- To remove the **complexity** of a model and eventually reduce variance (prevent overfitting).

▼ Regularization Techniques

- Can be any form.
  - E.g. Lp norm. Most widely used methods: **L1** (lasso) and **L2** (ridge) regularizations.



Isosurfaces (the surface on which the norm takes a constant value) of Lp norms (p = 0.5, 1, 2, 4)

- Hybrid of L1 and L2 regularizations: **Elastic net** regularization.
- For neural networks: **dropout** and **batch normalization**.
- Non-mathematical methods that have a regularization effect: **data augmentation** and **early stopping**.

> ✏️ **Feature scaling** is important for regularization - we need to ensure all features are on comparable scales.

L1 and L2 Regularizations

> 📌 L1 and L2 regularizations can be applied to all parametric models including linear regression, logistic regression, SVMs, and neural networks.

▼ L1 Regularization

- L1 regularization adds L1-norm ($\alpha \sum_{j=1}^{m} |w_j|$) to the loss function. L1 regularizes the absolute sum of the weights.
- E.g. Lasso Regression

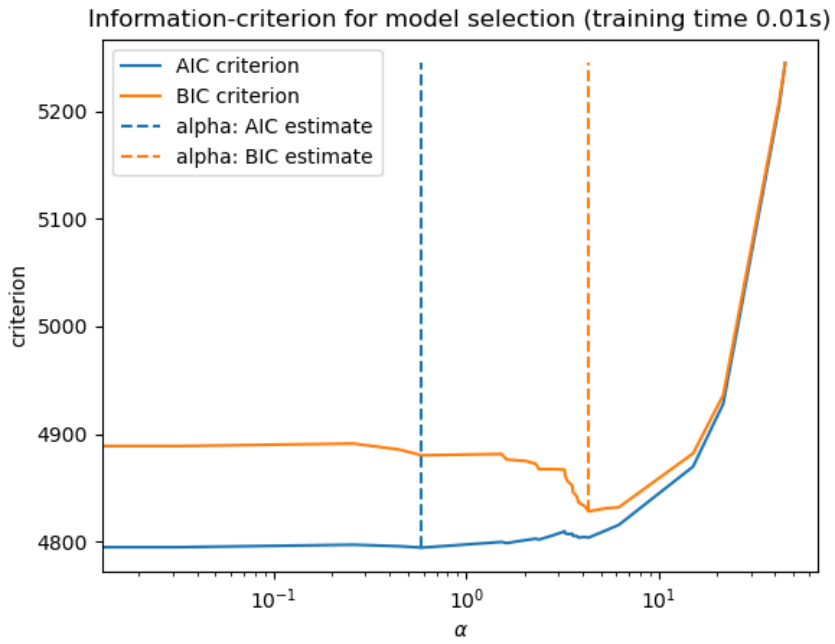
$$\min_{w} \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y_i})^2 + \alpha \sum_{j=1}^{m} |w_j|$$

▼ α: hyperparameter

- Controls the regularization strength

> 💡 We need to be careful when adjusting the regularization strength. If the regularization strength is too high and the weight coefficients shrink to zero, the model can perform poorly due to underfitting.

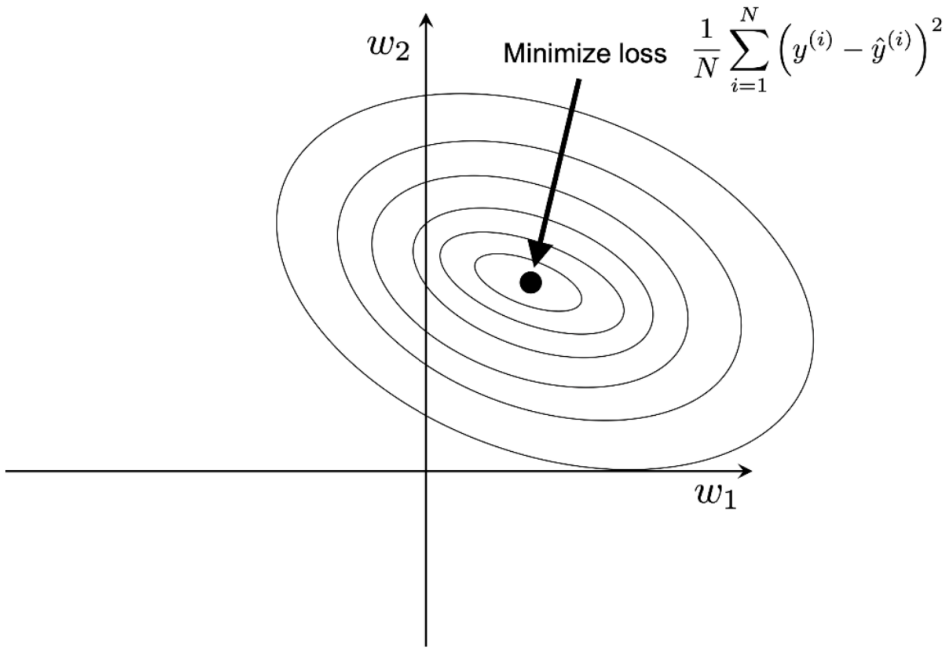
- Evaluated with cross-validation, AIC or BIC.



▼ **Geometric interpretation**

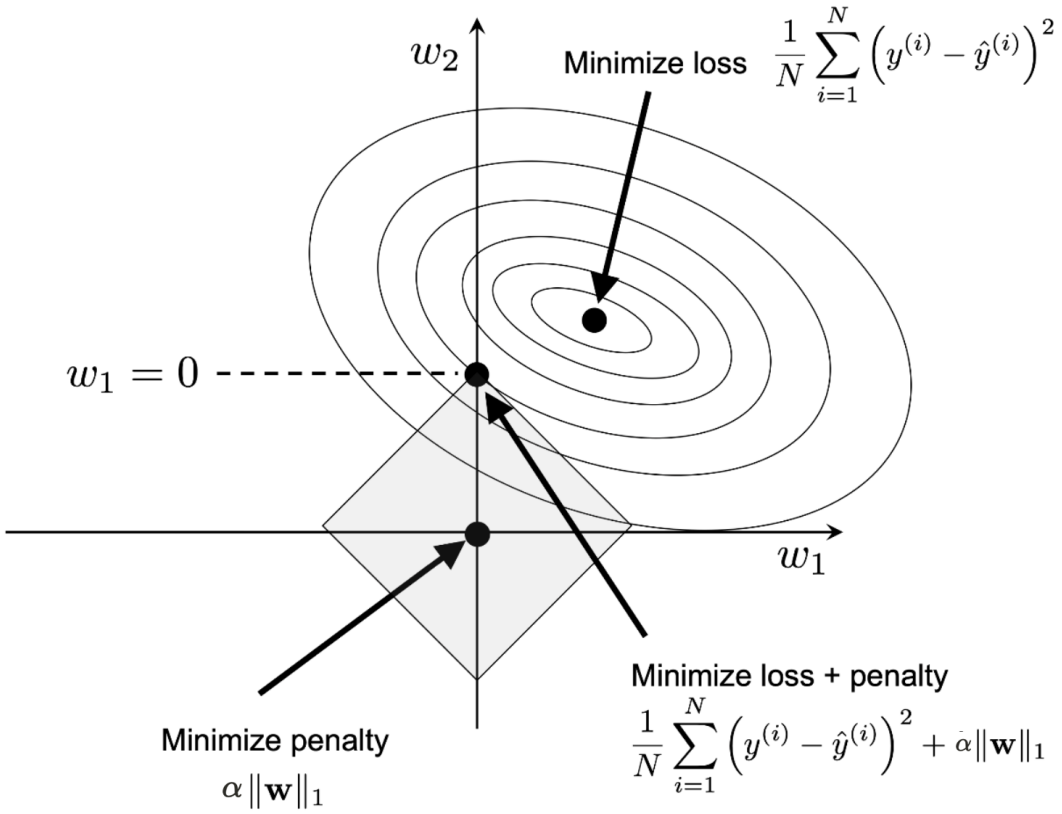
Contours: the **mean squared error** loss function (the squared distance between the true and predicted values) for two weight coefficients, w_1 and w_2.

Goal: find the combinations of w_1 and w_2 that minimize the loss function.



Source: Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine learning with pytorch and Scikit-Learn: Develop machine learning and deep learning models with python*. Packt Publishing.

- Diamond indicates the sum of the absolute weight coefficients.
  - Larger $\alpha \rightarrow$ smaller diamond.
  - By increasing α, we shrink the weights towards zero.
- Solution:
  - Balance two different losses - cannot decrease either loss without increasing the other.
  - Diamond **intersects** with the contours of the unpenalized loss function. Either w_1 or w_2 becomes zero (sparse vector).



Source: Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine learning with pytorch and Scikit-Learn: Develop machine learning and deep learning models with python*. Packt Publishing.

> 💡 While constraining with L1 regularizer, in order to descent to lower error, some of the parameters tend to **shrink to zero**. L1 produces **sparse** feature vectors, and most feature weights are zero, provided α is large enough.

▼ **Pros and Cons**

- The only norm that introduces sparsity in the solution and remains convex for easy optimization.
- L1 performs **feature selection** by deciding which features are essential for prediction and which are not (will be forced to be exactly zero). It helps increase model interpretability.
- It's useful in cases where we have a high-dimensional dataset with many features that are irrelevant.
- Makes our models more efficient to store and compute.
- The result of L1 may be **inconsistent**, the parameters may differ from each training.

▼ L2 Regularization

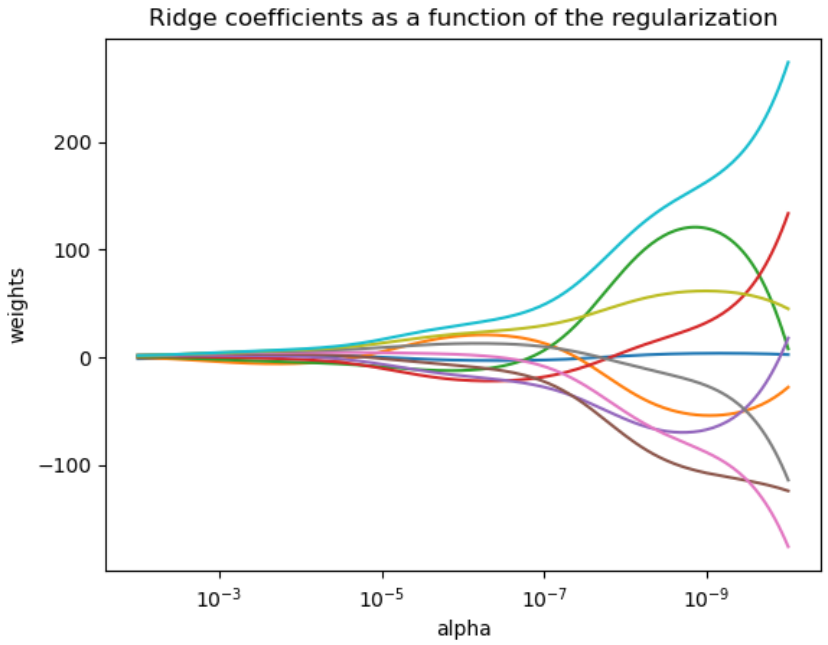
- L2 regularization adds L2-norm ($\alpha \sum_{j=1}^{m} w_j^2$) to the loss function. It regularizes the sum of squares of the weights.

- E.g. Ridge regression

$$\min_w \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{m} w_j^2$$

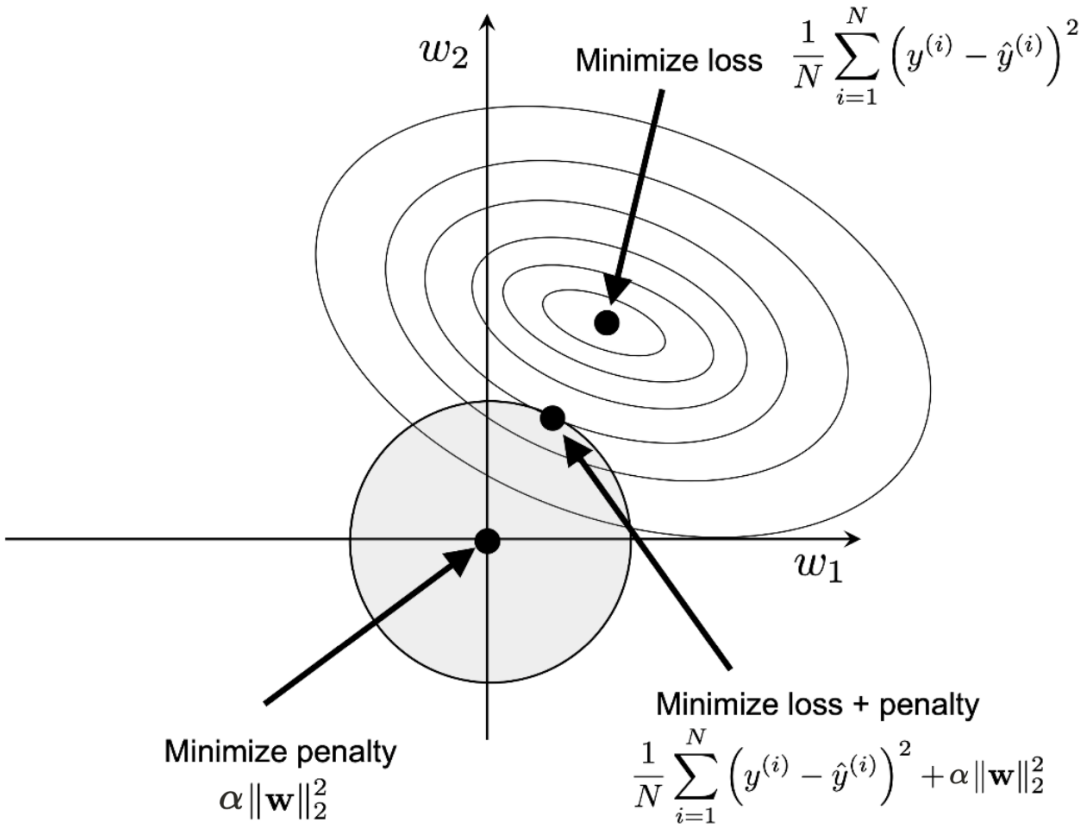
▼ $\alpha \geq 0$**: hyperparameter**

Controls the amount of shrinkage - the larger the value of α, the greater the amount of shrinkage.



▼ **Geometric interpretation**

- Disk indicates the quadratic L2 regularization term. The combination of w_1 and w_2 cannot fall outside the disk.
  - Larger $\alpha \rightarrow$ narrower disk.
- Solution:
  - Balance two different losses - cannot decrease either loss without increasing the other.
  - Disk **intersects** with the contours of the unpenalized loss function. Both w_1 and w_2 will be penalized, but neither is zero.



Source: Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine learning with pytorch and Scikit-Learn: Develop machine learning and deep learning models with python*. Packt Publishing.

▼ **Pros and Cons**

- L2 shrinks the parameters and reduces influence of unimportant features.
- It is more stable than L1 regularization.

> ✏️ L2 regularization is **differentiable**, so gradient descent can be used for optimizing the objective function.

- It does not shrink parameters to zero, therefore L2 can not be used to do feature selection. It's helpful when the only goal is to improve the performance of the model (prevent overfitting).

▼ L1 vs. L2 Regularization

- L1 and L2 regularizations are generally used to add constraints to optimization problems to prevent models from being overfitting.
- In L1, features are penalized more than L2 which results in **sparsity** while L2 regularization tends to spread error among all the terms, so L1 does **feature selection** while L2 does not.

- L1 is more **sparse/binary** (increases in one value of one parameter must be exactly offset by decreases in the other), with many features either being assigned a 0 or 1 in weighting.
- For correlated features, L1 selects the best one while L2 spreads out the weights.
- Errors are squared in L2, so the model sees higher error and tries to minimize that squared error.