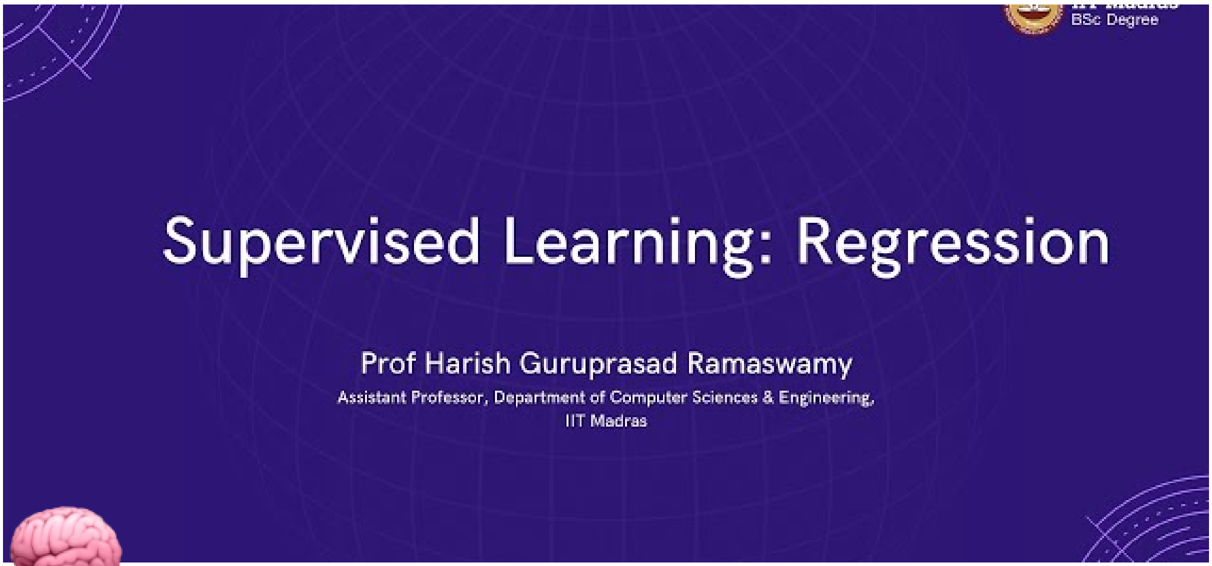


Regularization Techniques - Complete Overview

Regularization Techniques - Complete Overview

> **Regularization** is a technique to control the complexity of machine learning models to reduce **overfitting**, increase **generalization**, and enable **feature selection**.

Learning Objectives

Regularization helps models:

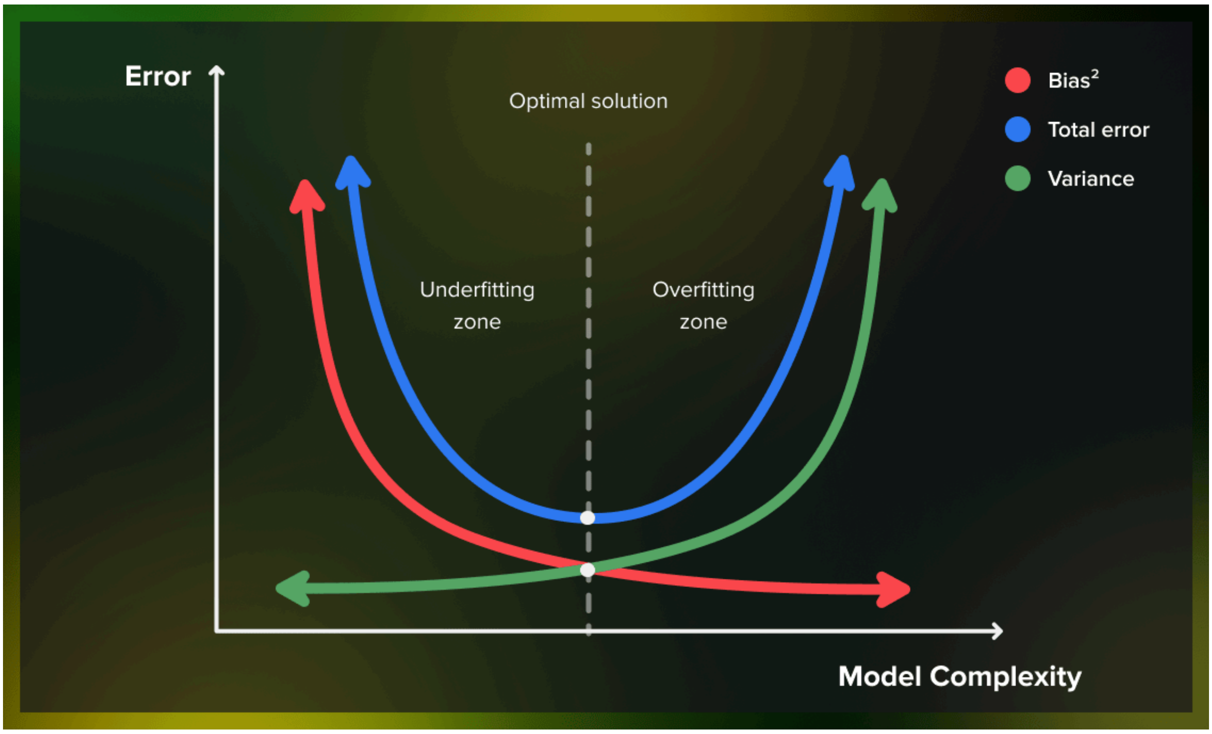
- **Reduce overfitting** (model memorizing training data)
- **Increase generalization** capability
- **Feature selection**

How it works: Add a penalty term to the cost function to constrain the magnitude of weights.

Bias – Variance Tradeoff

Component	Meaning	When High
Bias	Average deviation between predictions and true values	Underfitting
Variance	Model's sensitivity when training data changes	Overfitting
Irreducible Error	Random error that cannot be reduced	—

Goal: Find the balance between bias and variance to make the model "just right" in complexity.



Relationship:

- Complex model → Bias ↓, Variance ↑
- Simple model → Bias ↑, Variance ↓
- Regularization helps adjust complexity using parameter **λ (lambda)**

General Formula of Regularization

$$J(w) = \underbrace{\frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}_{\text{Loss}} + \underbrace{\lambda \cdot \Omega(w)}_{\text{Penalty}}$$

Parameter Explanation

Symbol	Meaning
J(w)	Total cost function to minimize
m	Number of training samples
y_i	True value of sample i
$\hat{y}_i$	Model prediction
w	Weight vector (model coefficients)
λ	Regularization strength parameter
Ω(w)	Penalty function (L1, L2, or combination)

Note: When **λ = 0**, no penalty, prone to overfit. When **λ is too large**, coefficients shrink too much, model underfits.

---

1. Ridge Regression (L2 Regularization)

Formula

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - w^T x_i)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

Characteristics

- Penalizes **squared coefficients** (L2 penalty)
- Shrinks coefficients but **never equals 0**
- Reduces variance → more stable model
- Should normalize data before use

Mathematical Example

Data: (1,1), (2,2), (3,3)

- **Without regularization:** w = 1
- **With λ = 10:** w = 0.5833

✅ Ridge shrinks coefficients → prevents overfitting.

Python Code

```
from sklearn.linear_model import Ridge

ridge = Ridge(alpha=10)
ridge.fit(X, y)
print(ridge.coef_)
```

---

2. LASSO Regression (L1 Regularization)

Formula

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - w^T x_i)^2 + \lambda \sum_{j=1}^{n} |w_j|$$

Characteristics

- Penalizes **absolute value** (L1 penalty)
- Can force **some coefficients = 0** → automatic feature selection
- Suitable when many features are unimportant

Mathematical Example

- **With λ = 3:** w = 0.893
- If λ is larger → w → 0

Python Code

```
from sklearn.linear_model import Lasso

lasso = Lasso(alpha=3)
```

```
lasso.fit(X, y)
print(lasso.coef_)
```

---

3. Elastic Net (L1 + L2 Combination)

Formula

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - w^T x_i)^2 + \lambda\left[\alpha\sum_{j=1}^{n}|w_j| + (1-\alpha)\sum_{j=1}^{n}w_j^2\right]$$

Parameter Explanation

Parameter	Meaning
λ	Overall regularization strength
α	Mixing parameter between L1 and L2
α = 1	Equivalent to LASSO
α = 0	Equivalent to Ridge

Characteristics

- Combines advantages of **Ridge** (stability) and **LASSO** (feature selection)
- Good when features are highly correlated

Python Code

```
from sklearn.linear_model import ElasticNet

elastic = ElasticNet(alpha=3, l1_ratio=0.5)
elastic.fit(X, y)
print(elastic.coef_)
```

---

4. Recursive Feature Elimination (RFE)

Concept

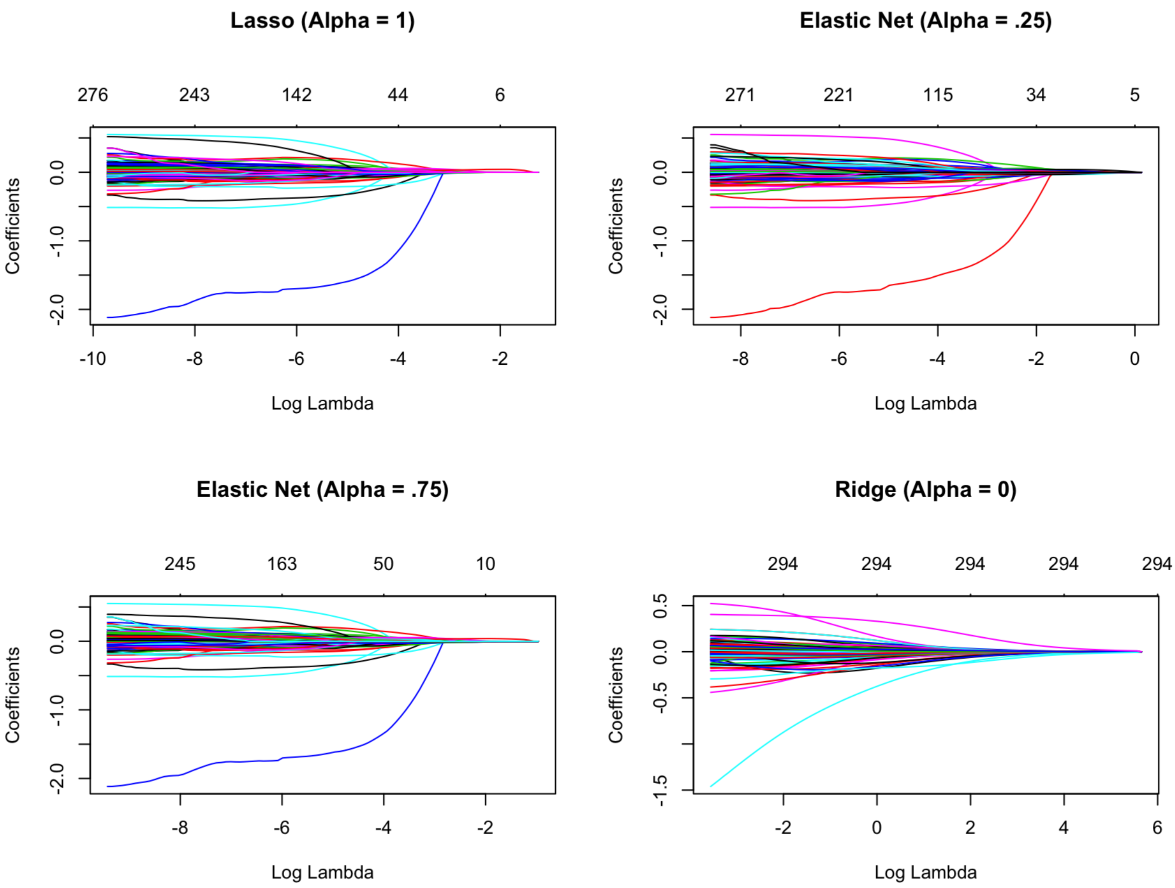
- Train model, evaluate importance of each feature

- Gradually eliminate weakest features until desired number remains

Python Code

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

rfe = RFE(LinearRegression(), n_features_to_select=2)
rfe.fit(X, y)
print("Selected features:", rfe.support_)
```



Calculation Results

Method	λ	Weight w	Notes
No Regularization	0	1.000	Perfect fit
Ridge	10	0.583	Coeff shrinks near 0

Method	λ	Weight w	Notes
LASSO	3	0.893	Mild shrinkage
Elastic Net	3	0.800	Balance between Ridge & LASSO

Comprehensive Comparison Table

Method	Penalty	Effect	Feature Selection	When to Use
Ridge (L2)	L2 norm	Coeff shrinks (≠0)	❌	All features important
LASSO (L1)	L1 norm	Some coeff = 0	✅	Many unimportant features
Elastic Net	L1 + L2	Balanced	✅	Correlated feature groups
RFE	—	Eliminate weak features	✅	Need specific number of features

Impact of λ

- **λ → 0** → no penalty → complex model → **overfit**
- **λ increases** → coefficients shrink → stable model
- **λ too large** → coefficients near 0 → **underfit**

Choose optimal λ using Cross-Validation (GridSearchCV or validation curve).

Conclusion

Method	Advantages	Limitations
Ridge (L2)	Stable, reduces overfitting, easy to optimize	Doesn't remove features

Method	Advantages	Limitations
LASSO (L1)	Automatic feature selection, simplifies model	May miss related features
Elastic Net	Combines selection + stability, good balance	Requires tuning 2 parameters (λ, α)
RFE	Removes specific features, interpretable	Depends on base estimator

Key Takeaways

✅ Regularization prevents **overfitting** by **penalizing large coefficients**

✅ **Ridge:** reduces variance, keeps all variables

✅ **LASSO:** automatic variable selection

✅ **Elastic Net:** combines both advantages

✅ Must normalize data and choose λ via cross-validation

✅ Understand from 3 perspectives: **Analytic – Geometric – Probabilistic**

Regularization is an essential tool that makes models less complex but smarter, preventing overfitting while maintaining accurate learning ability.