
Clase 073 — Regularización: Ridge, Lasso, Elastic Net

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 4. Duración estimada: 70 min.

Clase 073 — Regularización: Ridge, Lasso, Elastic Net

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 4. Duración estimada: 70 min.

Objetivo

Aprender a controlar el overfitting en modelos lineales mediante regularización L2 (Ridge), L1 (Lasso) y su combinación (Elastic Net), entendiendo el rol del hiperparámetro alpha, la importancia del scaling, y cuándo conviene cada variante.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Explicar qué es la regularización y por qué reduce la varianza de un modelo lineal.
- Implementar Ridge, Lasso y ElasticNet de scikit-learn sobre un dataset escalado.
- Tunear alpha con RidgeCV / LassoCV / ElasticNetCV y leer los coeficientes resultantes.
- Justificar la elección entre L1, L2 y L1+L2 según el problema (multicolinealidad, sparsity, número de features).
- Diagnosticar por qué Lasso "anula" features y Ridge solo las "encoge".

Temas

- Sesgo–varianza y motivación de la regularización.
- Ridge (L2): penalización $\alpha \cdot \Sigma\beta^2$. Encoge coeficientes hacia cero sin anularlos.
- Lasso (L1): penalización $\alpha \cdot \Sigma|\beta|$. Produce soluciones sparse (selección de features).
- Elastic Net: combinación L1+L2 con l1_ratio.
- Hiperparámetro alpha: efecto en bias/varianza; $\alpha=0 \equiv$ OLS; $\alpha \rightarrow \infty \equiv$ modelo nulo.
- Scaling obligatorio (StandardScaler) antes de regularizar.
- Selección de α con CV: RidgeCV, LassoCV, ElasticNetCV.

Definiciones y características

- Ridge (regresión L2): añade $\alpha \cdot \Sigma\beta^2$ a la función de costo. Tiene solución cerrada. No anula coeficientes; los acerca a cero proporcionalmente. Robusto frente a multicolinealidad.
- Lasso (regresión L1): añade $\alpha \cdot \Sigma|\beta|$. No tiene solución cerrada (se resuelve con coordinate descent). Produce soluciones esparsas: muchos coeficientes quedan exactamente en 0 \rightarrow hace selección automática de features.
- Elastic Net: combina L1 y L2: $\alpha \cdot (l1_ratio \cdot \Sigma|\beta| + (1 - l1_ratio)/2 \cdot \Sigma\beta^2)$. Útil cuando hay más features que muestras o features muy correlacionadas (Lasso puro "elige una al azar" del grupo correlacionado).
- alpha (α): fuerza de la regularización. Más alto \rightarrow más penalización \rightarrow coeficientes más chicos \rightarrow más bias, menos varianza. Se tunea por CV.
- L1 vs L2: L1 promueve sparsity (esquinas del rombo $|\beta|$); L2 promueve coeficientes pequeños y suaves (círculo β^2). L1 hace selección, L2 no.
- Sparsity: propiedad de un vector de coeficientes con muchos ceros. Útil para interpretabilidad y costo computacional en predicción.

- RidgeCV / LassoCV / ElasticNetCV: variantes que internamente buscan alpha (y l1_ratio) por cross-validation sobre una grilla. Más eficiente que GridSearchCV para estos modelos.
- Scaling: como la penalización suma cuadrados/absolutos de coeficientes, una feature con escala grande recibe penalización injustamente alta. Siempre escalar antes.

Dataset / recursos

- Dataset sugerido: `sklearn.datasets.fetch_california_housing` (regresión continua, 8 features con escalas distintas → ideal para mostrar scaling).
- Alternativa sintética: `make_regression(n_features=50, n_informative=10, noise=10)` para ver cómo Lasso anula las 40 no informativas.
- Stack: numpy, pandas, scikit-learn (Ridge, Lasso, ElasticNet, RidgeCV, LassoCV, ElasticNetCV, StandardScaler, Pipeline).

Ejercicios

1. OLS baseline: entrená `LinearRegression` sobre California Housing escalado. Reportá RMSE en train y test. Anotá los coeficientes.
2. Ridge: entrená `Ridge(alpha=1.0)` sobre los mismos datos. Compará RMSE y la magnitud de los coeficientes vs OLS.
3. Lasso: entrená `Lasso(alpha=0.1)`. Contá cuántos coeficientes quedaron exactamente en 0. Subí alpha a 1.0 y a 10.0; observá la sparsity creciente.
4. Elastic Net: entrená `ElasticNet(alpha=0.1, l1_ratio=0.5)`. Compará con Ridge y Lasso puros.
5. Tuning con CV: usá `RidgeCV(alphas=np.logspace(-3, 3, 50))` y `LassoCV(cv=5)` para encontrar el mejor alpha. Graficá el path de coeficientes vs alpha (Lasso path).

Homework verificable

Sobre `make_regression(n_samples=200, n_features=50, n_informative=10, noise=10, random_state=42)`:

1. Entrená OLS, Ridge, Lasso y Elastic Net (todos con `StandardScaler` en Pipeline).
2. Tuneá alpha con la variante *CV correspondiente.
3. Reportá: RMSE en test (split 80/20, `random_state=42`), número de coeficientes $\neq 0$, y top-5 features por `|coef|` en Lasso.

Criterio de verificación: Lasso debe identificar al menos 8 de las 10 features informativas ($\text{coef} \neq 0$) y dejar en 0 al menos 30 de las 40 ruidosas. RMSE de Ridge y Elastic Net dentro de $\pm 10\%$ del de Lasso.

Errores comunes

1. No escalar features antes de Ridge/Lasso/Elastic Net: la penalización castiga a las features de mayor escala. Siempre `StandardScaler` (o equivalente) en un Pipeline.
2. Usar `alpha=0`: equivale a OLS y además dispara warnings/inestabilidad numérica en Lasso. Si querés OLS, usá `LinearRegression`.
3. Confundir alpha de sklearn con λ de la teoría: en sklearn alpha ya incluye el factor $1/(2n)$ o $1/n$ según el modelo. No es directamente comparable entre Ridge y Lasso.
4. Tunear alpha sobre el test set: usá CV (`RidgeCV`, `LassoCV`) o `GridSearchCV` sobre train. El test se toca una sola vez al final.
5. Esperar que Ridge haga selección de features: no las anula, solo las encoge. Si querés sparsity, usá

Lasso o Elastic Net.

Preguntas frecuentes

1. ¿Ridge, Lasso o Elastic Net?
 - Ridge: pocas features, todas potencialmente relevantes, posible multicolinealidad.
 - Lasso: muchas features, sospechas que muchas son irrelevantes, querés un modelo interpretable.
 - Elastic Net: muchas features correlacionadas entre sí, o $p > n$ (más features que muestras).
1. ¿Por qué Lasso anula coeficientes y Ridge no?
1. ¿Qué pasa si alpha es muy grande?
1. ¿Cómo elijo el rango de alpha para buscar?
1. ¿Sirve regularización con árboles o solo con modelos lineales?

Geométrica
restricción
sobre un eje

Todos los
Underfitting

Una grilla lo

Ridge/Lasso
regularización

Referencias

- Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (3.ª ed.), cap. 4 — sección "Regularized Linear Models".
- scikit-learn — Linear Models: Ridge, Lasso, Elastic Net.
- Hastie, Tibshirani, Friedman — The Elements of Statistical Learning, cap. 3.4 (Shrinkage Methods).
- Zou & Hastie (2005), "Regularization and variable selection via the elastic net".

Siguiente clase

Clase 074 — Early stopping

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb