Machine Learning

Regression

Claudio Sartori

DISI
Department of Computer Science and Engineering – University of Bologna, Italy
claudio.sartori@unibo.it

Regression – Forecasting continuous values

- Supervised task
- The target variable is numeric
- Minimize the error of the prediction with respect to the target

This topic was already included in the Statistics and Data Analysis module, it is included here only for completeness

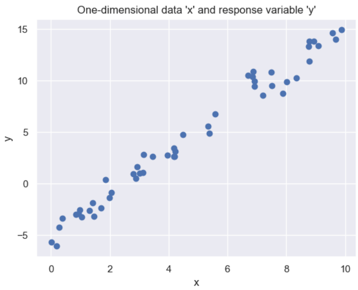
Linear Regression

- data set $\mathcal{X}$ with N rows and D columns
  - x_i is a D dimensional data element
- response vector $\overline{y}$ with N values y_i
- w is a D-dimensional vector of coefficients that needs to be learned
- we model the dependence of each response value y_i from the corresponding independent variables x_i as

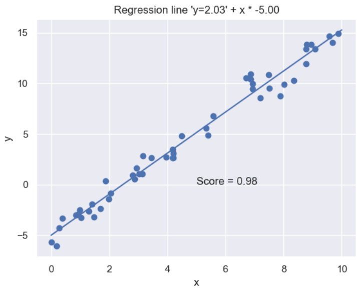
$$y_i \approx w^T \cdot x_i \quad \forall i \in [1 \ldots N]$$

- such that the error of modelling is minimised
- Classical statistic method (1805)

Data and regression line



One–dimensional data and response variable



Regression and score - Score range $(-\inf : 1)$

Objective function and minimisation I

Optional

$$
\begin{aligned}
\mathcal{O} &= \sum_{i=1}^{N} (w^T \cdot x_i - y_i)^2 = ||Xw^T - y||^2 \\
&= (Xw^T - y)^T . (Xw^T - y)
\end{aligned}
$$

Gradient of $\mathcal{O}$ with respect to w

$$
2X^T(Xw^T - y)
$$

Constraining the gradient to 0 we obtain the optimisation condition

$$
X^T X w^T = X^T y
$$

Objective function and minimisation II

Optional

If the symmetric matrix $X^T X$ is invertible the solution can be derived as

$$w = (X^T X)^{-1} X^T y$$

and the forecast is given by

$$y^f = X \cdot w^T$$

Matrix calculus

Optional

- Issues related to matrix calculus if $\overline{x}^T\overline{x}$ is not invertible
- Moore–Penrose pseudoinverse
- Tikonov regularisation (also known as ridge regression)
- Lasso regularisation

Quality of the fitting - R^2

Mean of the observed data $y^{avg} = \frac{1}{N} \sum_i y_i$

Sum of squared residuals $SS_{res} = \sum_i (y_i - y_i^f)^2$

Total sum of squares $SS_{tot} = \sum_i (y_i - y^{avg})^2$

Coefficient of determination $\mathbf{R^2 = 1 - \frac{SS_{res}}{SS_{tot}}}$

Intuition about R^2

- It compares the fit of the chosen model with that of a horizontal straight line
- With perfect fitting the numerator of the second term is zero and $R^2 = 1$
- If the model does not follow the trend of the data the numerator of the second term can reach or exceed the denominator, and R^2 can also be negative
- Despite the name, R^2 isn't the square of anything

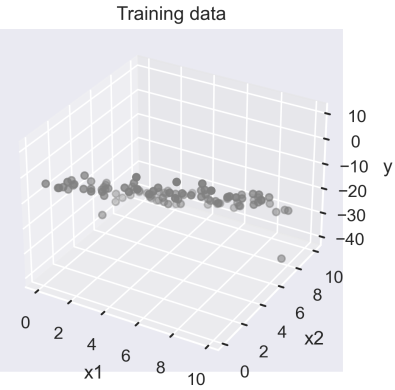
R^2 and Mean Squared Error

Optional

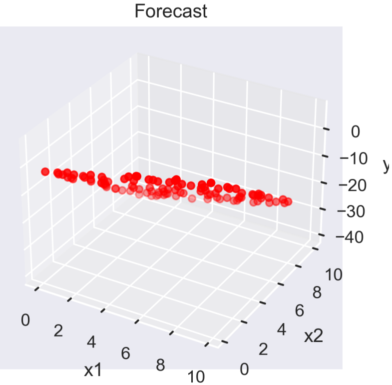
- Both refer to the error of the predictions
- R^2 is a standardised index,
- *RMSE* measures the mean error, this it is influenced by the order of magnitude of the data,
- Both *RMSE* and R^2 quantifies how well a linear regression model fits a dataset
- The RMSE tells how well a regression model can predict the value of a response variable in absolute terms
- R^2 tells how well the predictor variables can explain the variation in the response variable
- For comparing the accuracy among different linear regression models, RMSE is a better choice than R Squared
- R^2 is not meaningful for non–linear or non–algebraic regression models

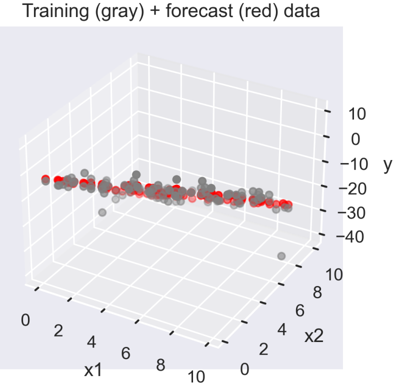
Multiple regression

- The response variable depends by two or more features
- The regression technique is quite similar to that of simple regression
- In `scikit-learn` the estimator is the same



Multiple regression - forecast



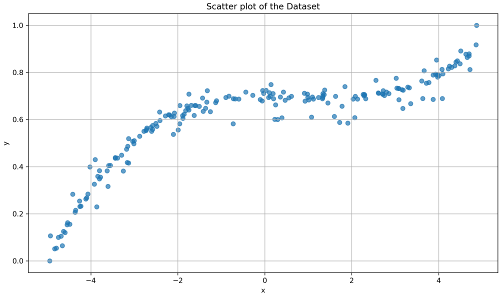


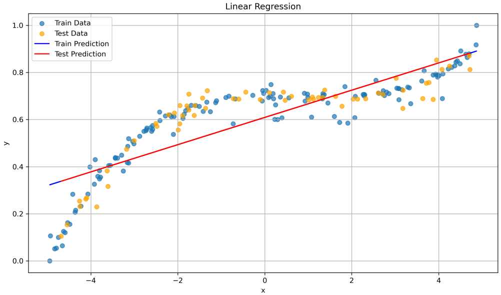
Overfitting and Regularisation

- In presence of high number of features overfitting is possible
  - performance on test data becomes much worse
- Regularisation reduces the influence of less interesting attributes and therefore reduces overfitting
  - see section 3

Polynomial regression (univariate)

What if the relationship between the independent variable and the target is not linear at all?





Univariate Polynomial Regression

- It is an extension of linear regression that models the relationship between the independent variable x and the dependent variable $y(x)$ as an n-degree polynomial.
- It fits nonlinear relationships between the input and output variables with a polynomial.
- The general equation for polynomial regression is:

$$y(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon$$

- Here, $\beta_0, \beta_1, \ldots, \beta_n$ are the model parameters, and ϵ represents the error term.

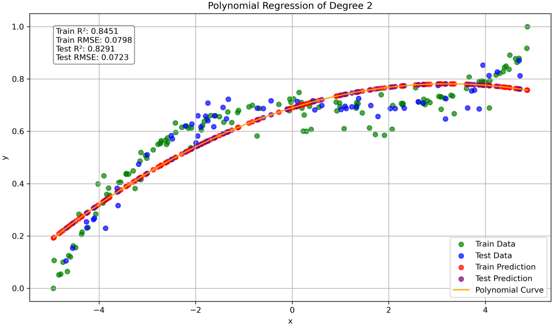
Steps in Polynomial Regression

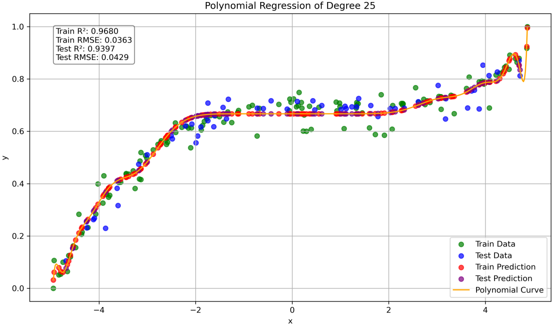
- Step 1: Generate the Polynomial Features
  - Transform the original input variable x into higher-order polynomial terms.
  - For example, if the degree $n = 2$, the polynomial features would be:

$$\mathbf{X} = [1, x, x^2]$$

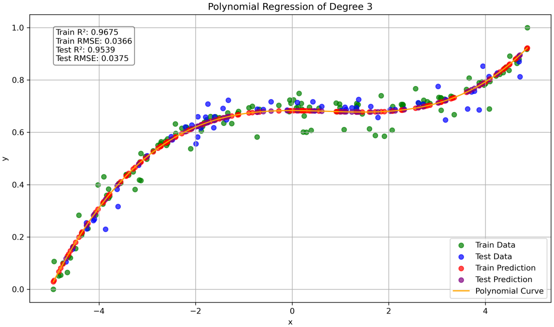
  - The transformation can be extended for higher degrees, $n = 3$, $n = 4$, etc.
- Step 2: Fit a Linear Regression Model
  - Despite being polynomial, the problem is treated as a linear regression problem in terms of the parameters $\beta_0, \beta_1, \ldots, \beta_n$.
  - The model is fit using least squares estimation to minimize the sum of squared residuals.
- Step 3: Evaluate the Model
  - Use standard regression metrics such as Root Mean Squared Error (RMSE).
  - Overfitting must be controlled using cross-validation to assess the optimal degree.

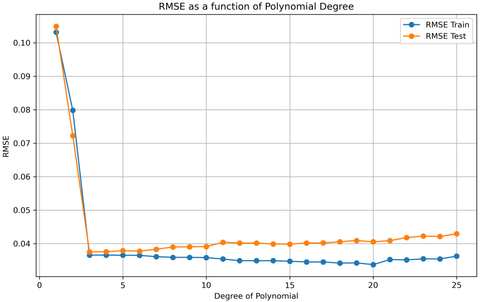
Underfitting and Overfitting





Good fitting and RMSE versus degree





Regularised regression

- The standard multivariate linear regression does not have hyperparameters for controlling the fitting quality, in particular to guarantee good performance on the test set
- A general way for controlling overfitting is to simplify the model
- How can we simplify a linear multivariate (and possibly polynomial)?

Regularised regression

- The standard multivariate linear regression does not have hyperparameters for controlling the fitting quality, in particular to guarantee good performance on the test set
- A general way for controlling overfitting is to simplify the model
- How can we simplify a linear multivariate (and possibly polynomial)?

Using a loss function

Loss in Regression

- The loss function quantifies the error between the model's prediction and the actual value
- In regression, the most common loss is the Root Mean Squared Error (RMSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Other loss functions:
  - Mean Absolute Error (MAE)
  - Log-loss (for probabilistic classifiers)

OLS - Ordinary Least Squares

- Cost function:

$$L(\mathbf{w}) = \sum_{i=1}^{N} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

- OLS regression simply determines the coefficient vector **w** that minimizes the loss of predictions with respect to the ground truth

Regularisation

- Ordinary Least Squares (OLS) regression minimizes the prediction error on the training set
- Risk: overfitting, especially with many variables or noisy data
- Regularization: technique to penalize model complexity
  - a way to reduce the complexity is to reduce, in several ways, the values of the coefficients
- Goal: find a good trade-off between accuracy and model simplicity

Lasso Regression

Least Absolute Shrinkage and Selection Operator [Tibshirani(1996)]

- A linear regression method that adds $L1$-regularization to the cost function
- Encourages sparse models by shrinking some coefficients to exactly zero
- Useful for feature selection and regularization in high-dimensional data

Cost Function [1]

$$L(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{D} x_{ij} w_j\right)^2 + \alpha \sum_{j=1}^{D} |w_j|$$

- Components
    - Residual sum of squares: Measures prediction error
    - $L1$-norm penalization $= \sum_{j=1}^{D} |w_j| = ||\mathbf{w}||_1$
        - penalizes the sum of absolute values of coefficients

1 For simplicity, here we do not consider the intercept

L1 Regularization: Penalizes Coefficients with an Absolute Value Constraint

- The Lasso penalty, $\alpha \sum_{j=1}^{D} |w_j|$, grows linearly with the magnitude of the coefficients
- This penalty creates a strong incentive to make some coefficients exactly zero due to:
  - Equal contribution to the penalty:
    - Small changes in the magnitude of a coefficient contribute equally to the penalty, whether the coefficient is large or small
  - Efficient penalty reduction:
    - When coefficients are near zero, shrinking them to zero entirely results in a significant penalty reduction with minimal cost to the residual sum of squares (RSS)

Lasso Regression: Compact Coordinate Descent

OPTIONAL

Input : $\mathbf{X} \in \mathbb{R}^{N \times D}$, $\mathbf{y} \in \mathbb{R}^N$, α, ϵ
Output: $\mathbf{w} \in \mathbb{R}^D$
Initialize $w \leftarrow \mathbf{0}$;
repeat
  for $j = 1$ **to** D **do**
    $r \leftarrow \mathbf{y} - \mathbf{X}\mathbf{w} + X_j w_j$;
    $\rho \leftarrow \frac{1}{N} \sum_i X_{ij} r_i$;
    $z \leftarrow \frac{1}{N} \sum_i X_{ij}^2$;
    $w_j \leftarrow \text{sign}(\rho) \cdot \max(|\rho| - \alpha, 0)/z$;
until $\|\Delta \mathbf{w}\|_\infty < \epsilon$;

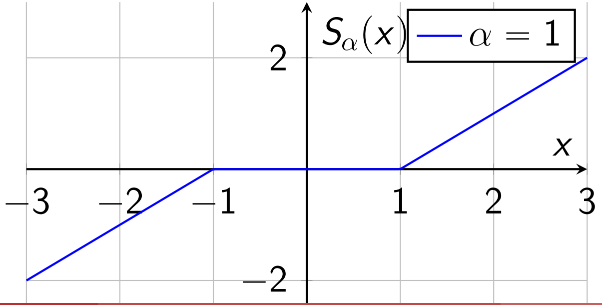
Soft-Thresholding Function in Lasso

Optional

- Lasso updates coefficients using the **soft-thresholding function**:

$$S_\alpha(x) = \text{sign}(x) \cdot \max(|x| - \alpha, 0)$$

- Promotes sparsity by shrinking small values to zero.



Computational Complexity

OPTIONAL

- Training Complexity
  - Depends on the number of features D, samples N, and iterations T
  - for coordinate descent:

$$\mathcal{O}(TND)$$

  - for large datasets, this is linear in N and D
- Convergence
  - Faster convergence if many coefficients are sparse
  - Slower for high-dimensional dense datasets
- Prediction Complexity
  - Linear in $\bar{D}$ (number of nonzero coefficients):

Understanding **w** in Lasso Regression

Optional

- w represents the coefficients (or weights) of the linear regression model
- Structure of **w**:
  - $\mathbf{w} = [w_0, w_1, \ldots, w_p]$
    - w_0: The intercept term of the model
    - w_j: The weight for the j-th feature, where $j = 1, \ldots, D$
- Predicted value $\hat{y}_i$ for a sample x_i:

$$\hat{y}_i = w_0 + \sum_{j=1}^{D} w_j x_{ij}$$

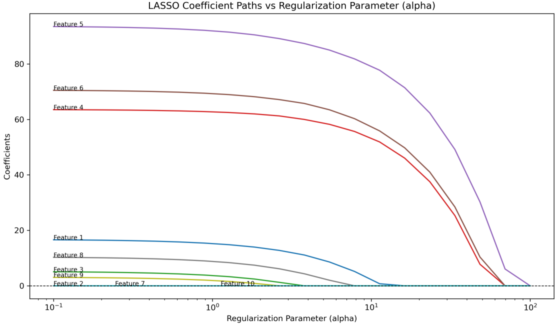
  - $\hat{y}_i$: Predicted output for the i-th sample
  - x_{ij}: Value of the j-th feature for the i-th sample

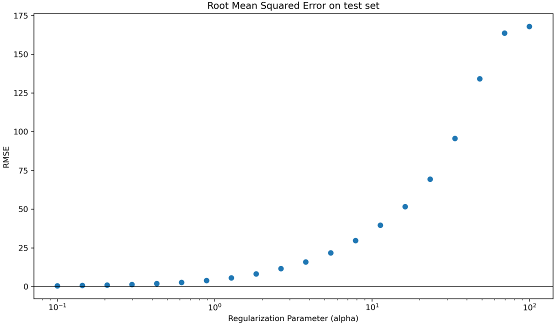
Role of **w** in Lasso Regression

OPTIONAL

- The optimization process adjusts **w** to:
  - Minimize residual error: $\frac{1}{2N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
  - Penalize large values of w_j using $L1$-norm regularization: $\alpha \sum_{j=1}^{D} |w_j|$
    - α is a multiplying factor: a hyper parameter allowing to calibrate the penalty
    - try several values for α together with cross–validation
- Effect of $L1$-regularization:
  - Encourages sparsity in **w**
    - Many coefficients w_j are set to exactly zero
- **w** embodies the importance of each feature in the regression model, while ensuring simplicity and robustness

LASSO effect and RMSE





Lasso: Summary

- Advantages
  - Produces sparse models for feature selection
  - Scales linearly with the size of the dataset
- Limitations
  - Struggles with collinearity among features
  - Computationally expensive for very large Ddue to iterative updates
- Applications
  - High-dimensional datasets where feature selection is essential

Ridge Regression[2]

- Ridge Regression is a type of linear regression
  - It adds a penalty term to the cost function to prevent overfitting
- Key Features:
  - Reduces model complexity
  - Improves generalization performance

2 The name derives from the matrix representation of the solution, where the α value adds a ridge to the main diagonal

The Ridge Regression Cost Function

- Ridge Regression modifies OLS by adding a penalty:

$$L(\mathbf{w}) = \sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \alpha\|\mathbf{w}\|$$

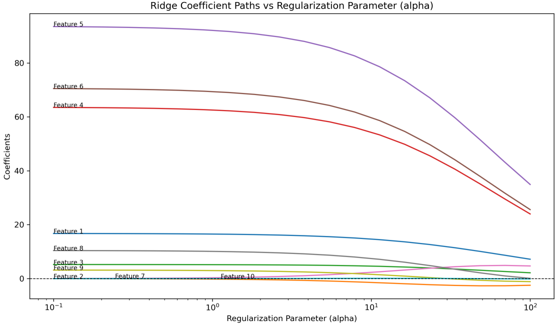
  - α: Regularization parameter controlling penalty strength
  - $\|\mathbf{w}\|$: $L2$ norm of the weight vector

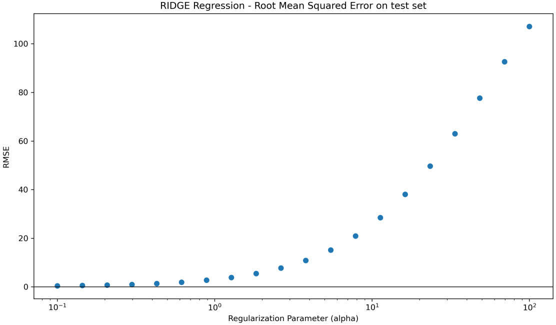
$$\beta_j \leftarrow \frac{1}{1+\alpha} \cdot \left(\frac{1}{N}\sum_{i=1}^{N} X_{ij}\left(y_i - \sum_{k\neq j} X_{ik}\beta_k\right)\right)$$

Effects of Regularization

- High α:
  - More penalty, leading to smaller weights
  - Reduces variance but increases bias
- Low α:
  - Less penalty, resembling OLS regression
  - Retains variance but may overfit the data
- Choosing α:
  - Cross-validation is commonly used to find the optimal α

RIDGE effect and RMSE





Ridge Regression: Coordinate Descent[3]

Optional

Input : $\mathbf{X} \in \mathbb{R}^{N \times D}$, $\mathbf{y} \in \mathbb{R}^{N}$, α, ϵ
Output: $\mathbf{w} \in \mathbb{R}^{D}$

Initialize $\mathbf{w} \leftarrow \mathbf{0}$;
$z \leftarrow 1 + \alpha$;
repeat
  for $j = 1$ **to** D **do**
    $r \leftarrow \mathbf{y} - \mathbf{X}\mathbf{w} + X_j w_j$;
    $\rho \leftarrow \frac{1}{N} \sum_i X_{ij} r_i$;
    $w_j \leftarrow \rho / z$;
until $\|\Delta \mathbf{w}\|_\infty < \epsilon$;

Coefficient update

$$w_j \leftarrow \frac{1}{1+\alpha} \cdot \left(\frac{1}{N} \sum_{i=1}^{N} X_{ij} \left(y_i - \sum_{k \neq j} X_{ik} w_k \right) \right)$$

[3] Assume that **X** is [illegible]

Ridge - Applications and Summary

- Applications:
  - Multicollinear data where features are highly correlated
  - Scenarios requiring reduced overfitting
- Summary:
  - Ridge Regression introduces a regularization term
  - Balances bias and variance for better generalization
  - Cross-validation helps in optimal parameter selection

Elastic Net Regression

- Elastic Net Regression is a linear regression method
  - Combines penalties from Ridge Regression and Lasso Regression
- Why Elastic Net?
  - Addresses limitations of Ridge and Lasso:
    - Ridge cannot perform feature selection
    - Lasso struggles when features are highly correlated
  - Offers a balance between these methods

The Elastic Net Cost Function

- Ordinary Least Squares (OLS) cost function:

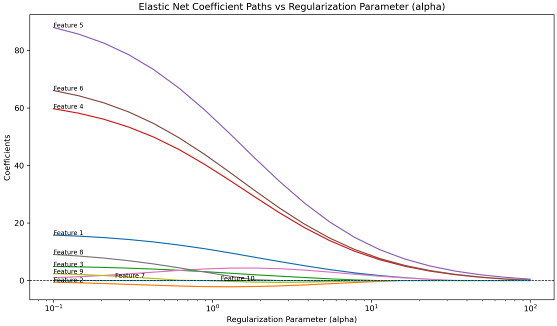
$$L(\mathbf{w}) = \sum_{i=1}^{N} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

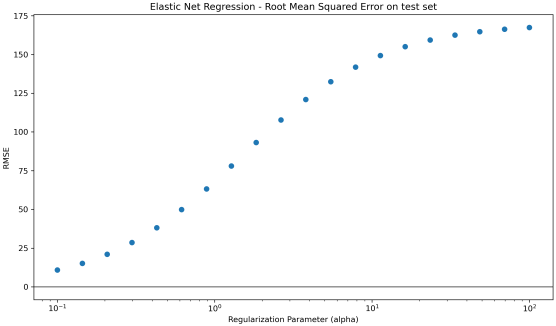
- Elastic Net modifies OLS with two penalties:

$$L(\mathbf{w}) = \sum_{i=1}^{N} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \alpha_1 \|\mathbf{w}\|_1 + \alpha_2 \|\mathbf{w}\|^2$$

  - α_1: Controls the Lasso penalty (L1 norm)
  - α_2: Controls the Ridge penalty (L2 norm)

Elastic Net effect and RMSE





Elastic Net: Coordinate Descent

Input : $\mathbf{X} \in \mathbb{R}^{N \times D}$, $\mathbf{y} \in \mathbb{R}^N$, α, η, ϵ
Output: $w \in \mathbb{R}^D$

```
Initialize w ← 0;
repeat
    for j = 1 to D do
        r ← y − Xw + X_j w_j;
        ρ ← (1/N) Σ_i X_ij r_i;
        z ← (1/N) Σ_i X_ij^2 + α(1 − η);
        w_j ← sign(ρ) · max(|ρ| − αη, 0)/z;
until ‖Δw‖_∞ < ε;
```

Initialize $w \leftarrow \mathbf{0}$;
repeat
  for $j = 1$ **to** D **do**
    $r \leftarrow \mathbf{y} - \mathbf{X}w + X_j w_j$;
    $\rho \leftarrow \frac{1}{N} \sum_i X_{ij} r_i$;
    $z \leftarrow \frac{1}{N} \sum_i X_{ij}^2 + \alpha(1 - \eta)$;
    $w_j \leftarrow \text{sign}(\rho) \cdot \max(|\rho| - \alpha\eta, 0)/z$;
until $\|\Delta w\|_\infty < \epsilon$;

Properties and Advantages

- Properties:
  - Encourages sparsity in coefficients (like Lasso)
  - Groups correlated features (like Ridge)
- Advantages:
  - Handles multicollinear data effectively
  - Can select relevant features while maintaining stability
  - Useful in high-dimensional data scenarios

Applications and Summary

- Applications:
  - Genomics (e.g., selecting gene expressions)
  - Financial modeling with highly correlated features
  - High-dimensional datasets with potential multicollinearity
- Summary:
  - Elastic Net combines Lasso and Ridge penalties
  - Effective in handling multicollinear data and sparse solutions
  - Requires hyperparameter tuning (α_1, α_2)

Comparison of regularized regression techniques

- Lasso, Ridge, and Elastic Net are regularization techniques used in regression
- They address overfitting and multicollinearity by introducing penalties in the cost function
- This presentation compares their real-world use cases, strengths, and limitations

Lasso Regression

- Strengths:
  - Performs feature selection, producing sparse models by setting some coefficients to zero
  - Useful for high-dimensional datasets with many irrelevant features
- Limitations:
  - Struggles with datasets where predictors are highly correlated
- Use Cases:
  - Genomics: Identifying relevant genes influencing a disease
  - Text Processing: Selecting keywords or n-grams in sentiment analysis
  - Sparse Sensor Networks: Identifying critical sensors in IoT or environmental monitoring

Ridge Regression

- Strengths:
  - Handles multicollinearity by shrinking coefficients
  - Retains all predictors, avoiding the elimination of variables
- Limitations:
  - Does not perform feature selection
- Use Cases:
  - Finance: Predicting stock prices using correlated economic indicators
  - Marketing: Modeling customer demand influenced by correlated factors
  - Engineering: Calibration of multivariate systems like chemical processes
  - Medical Imaging: Predicting outcomes from high-dimensional MRI or CT data

Elastic Net Regression

- Strengths:
  - Combines Lasso and Ridge penalties, balancing sparsity and multicollinearity handling
  - Selects groups of correlated features, unlike Lasso alone
- Limitations:
  - Requires careful tuning of two parameters (α_1 and α_2)
- Use Cases:
  - Genomics: Selecting groups of genes associated with traits
  - Healthcare Analytics: Modeling patient outcomes from clinical predictors
  - Customer Segmentation: Identifying clusters of customer behaviors in retail
  - Climate Science: Modeling climate variables with correlated predictors
  - Social Media Analysis: Predicting trends from sparse and correlated features

Comments

- Lasso, Ridge, and Elastic Net offer distinct strengths tailored to different data characteristics
- Choosing the right method depends on:
  - Presence of multicollinearity
  - Sparsity of the solution required
  - Dimensionality of the dataset
- Elastic Net is often a robust choice when both sparsity and correlation must be addressed

Comparison of Lasso, Ridge, and Elastic Net Regression

Feature	**Lasso**	**Ridge**	**Elastic Net**
Feature Selection	Yes	No	Yes groups correlated features
Handles Multicollinearity	Weak	Strong	Strong
Model Interpretability	High (sparse coefficients)	Moderate	Moderate (sparse, but groups features)
Dataset Characteristics	High-dimensional, sparse predictors	Correlated predictors	Sparse and correlated predictors

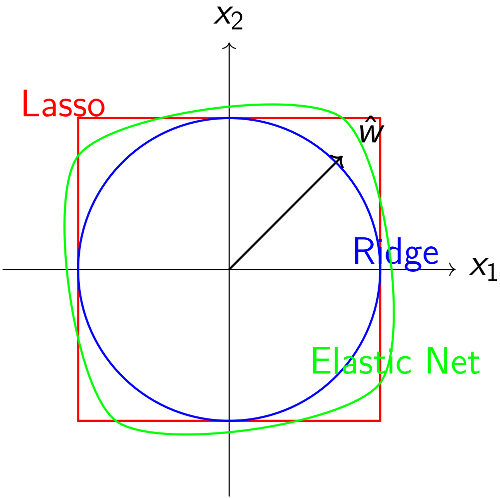
Explanation

- Sparsity:
  - Elastic Net can set some coefficients to zero, removing irrelevant predictors
  - This results in a simpler and more interpretable model, similar to Lasso
- Groups Features:
  - When predictors are highly correlated, Elastic Net:
    - Tends to select them together rather than choosing one arbitrarily
    - Shrinks their coefficients toward each other using the Ridge-like penalty
  - This behavior arises because Elastic Net combines:
    - L1 penalty (Lasso) for sparsity
    - L2 penalty (Ridge) for handling multicollinearity

Practical Example: Correlated Predictors

- Suppose two predictors, x_1 and x_2, are highly correlated:
  - Lasso:
    - May select only x_1 or x_2, ignoring the other entirely
  - Ridge:
    - Keeps both x_1 and x_2, but shrinks their coefficients
  - Elastic Net:
    - Selects both x_1 and x_2, but their coefficients may be reduced (shrunk) in different proportions
    - Balances between sparsity and correlation handling

Visualization of Sparse and Grouping Behavior



Description of the figure I

- Visual representation of the constraints applied by Lasso, Ridge, and Elastic Net regression
- The axes represent the coefficient of two predictors
- Shapes of Constraints
  - Lasso: A diamond-shaped constraint indicating L1 penalty, which promotes sparsity (coefficients set to zero)
  - Ridge: A circular constraint indicating L2 penalty, which shrinks coefficients uniformly but does not set them to zero
  - Elastic Net: A combination of Lasso and Ridge constraints, allowing both sparsity and handling of correlated groups

Description of the figure II

- Interpretation of Coefficient Paths
  - In Lasso, coefficients are pushed to the edges, setting some to zero
  - In Ridge, coefficients shrink but remain non-zero, resulting in a smoother path
  - Elastic Net provides a balance, with paths that follow the L1 and L2 constraints, enabling feature selection and correlation handling

Key Takeaways

- Elastic Net combines the best of Lasso and Ridge:
  - Sparsity: Sets some coefficients to zero for simpler models
  - Handles Correlated Predictors: Selects groups of features rather than one arbitrarily
- Ideal for:
  - High-dimensional datasets with multicollinearity
  - Applications requiring both feature selection and robust performance

Selection of Regression Models

Method	Library	Model Name
Linear Regression	sklearn.linear_model	LinearRegression
Elastic Net Regression	sklearn.linear_model	ElasticNet
Stochastic Gradient Descent Regression	sklearn.linear_model	SGDRegressor
Bayesian Ridge Regression	sklearn.linear_model	BayesianRidge
Lasso Regression	sklearn.linear_model	Lasso
Support Vector Machine	sklearn.svm	SVR
Kernel Ridge Regression	sklearn.kernel_ridge	KernelRidge
Gradient Boosting Regression	sklearn.ensemble	GradientBoostingRegressor
XGBoost Regressor	xgboost	XGBRegressor
CatBoost Regressor	catboost	CatBoostRegressor
LGBM Regressor	lightgbm	LGBMRegressor

Bibliography I

- Robert Tibshirani.
  Regression shrinkage and selection via the lasso.
  Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.