ECE 469/ECE 568 Machine Learning

Textbook:
Machine Learning: a Probabilistic Perspective by Kevin Patrick Murphy

Southern Illinois University

September 27, 2024

Finding optimal Lambda in Ridge Regularization using Cross Validation

This presentation provides a sample code on finding optimal lambda in Ridge Regularization/Regression using cross-validation.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import Ridge
from sklearn.model_selection import cross_val_score, KFold
```

Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Parameters
variance = 5  # variance value
NoS = 100000  # 100000 data points
x = 10*np.random.rand(NoS, 1) - 4

n = np.random.normal(0, np.sqrt(variance), x.shape)  # Gaussian
noise with mean 0 and variance sigma^2
y = 4 - 3*x + 7*x ** 2 -6*x **3 + n
x1 = np.squeeze(x,1)
y1 = np.squeeze(y,1)

# Create a DataFrame
data = pd.DataFrame({
'x': x1,
'y': y1
})
```

Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Save the DataFrame to a CSV file
filename = f'dataset4_variance_{variance}.csv'
data.to_csv(filename, index=False)

print(f'Dataset saved to {filename}')

plt.figure(figsize=(8,6))
plt.scatter(x, y, s=1, c='blue', alpha=0.5, label="Data points")
plt.title('Plot of x vs y')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()
plt.show()
```

Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Load dataset
data = pd.read_csv(f'dataset2_variance_{variance}.csv')
X = data['x'].values.reshape(-1, 1)
y = data['y'].values

# Polynomial degree
degree = 5

# Range of lambda values for Ridge regression
lambdas = np.logspace(-4, 2, 100)

# Number of folds for K-fold cross-validation
kf = KFold(n_splits=10, shuffle=True, random_state=1)
```

Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Transform the input features to include polynomial features
poly = PolynomialFeatures(degree)
X_poly = poly.fit_transform(X)

# To store the cross-validation results
mse_values = []
```

Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Perform cross-validation for each lambda value
for alpha in lambdas:
ridge = Ridge(alpha=alpha)

# Calculate MSE using cross_val_score
mse = -cross_val_score(ridge, X_poly, y, cv=kf,
 scoring='neg_mean_squared_error').mean()

mse_values.append(mse)
```

Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Find the lambda with the minimum MSE
best_lambda = lambdas[np.argmin(mse_values)]
print(f"Best lambda (with minimum MSE): {best_lambda}")

# Fit the model with the best lambda
ridge_best = Ridge(alpha=best_lambda)
ridge_best.fit(X_poly, y)

# Generate testing X values
x_fit = np.linspace(X.min(), X.max(), 1000).reshape(-1, 1)
x_poly_fit = poly.transform(x_fit)
y_fit = ridge_best.predict(x_poly_fit)
```

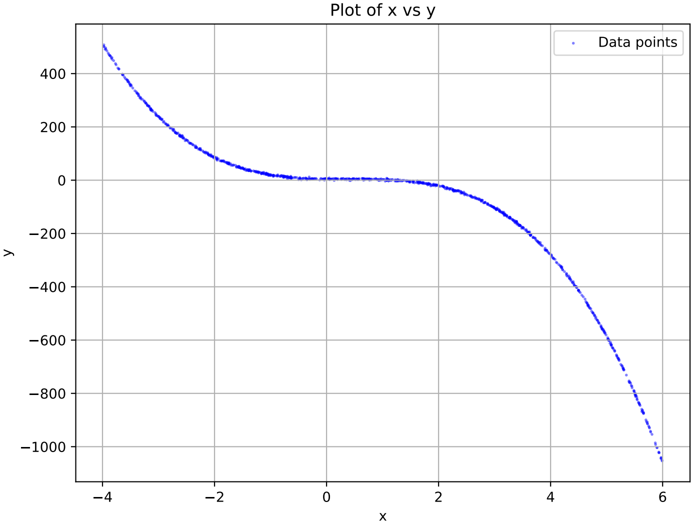
Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Plot MSE vs log(lambda)
plt.figure(figsize=(8, 6))
plt.plot(np.log10(lambdas), mse_values, label='MSE')
plt.xlabel('log10(lambda)')
plt.ylabel('Cross-Validated MSE')
plt.title('Cross-Validated MSE vs log(lambda)')
plt.grid(True)
plt.legend()
plt.show()
```

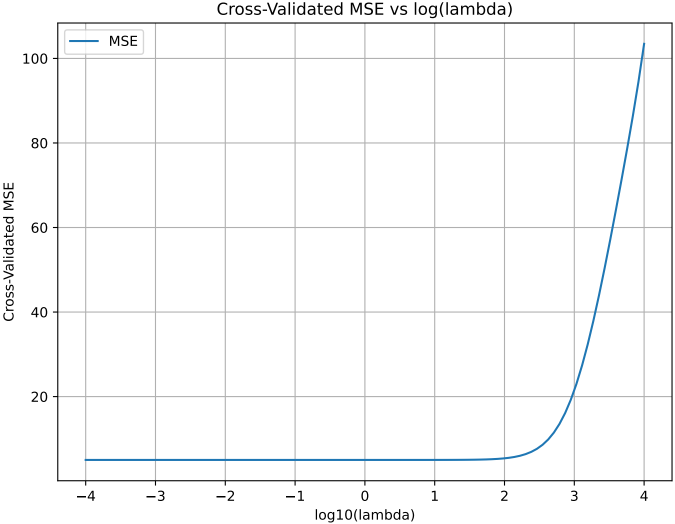
Finding optimal Lambda in Ridge Regression using Cross Validation

```
# Plot the data points and the fitted curve
plt.figure(figsize=(8,6))
plt.scatter(X, y, s=10, color='blue', alpha=0.5,
label='Data points')
plt.plot(x_fit, y_fit, color='red',
label=f'Polynomial Fit (lambda={best_lambda:.4f})')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Polynomial Curve Fit with Best Lambda')
plt.legend()
plt.grid(True)
plt.show()
```

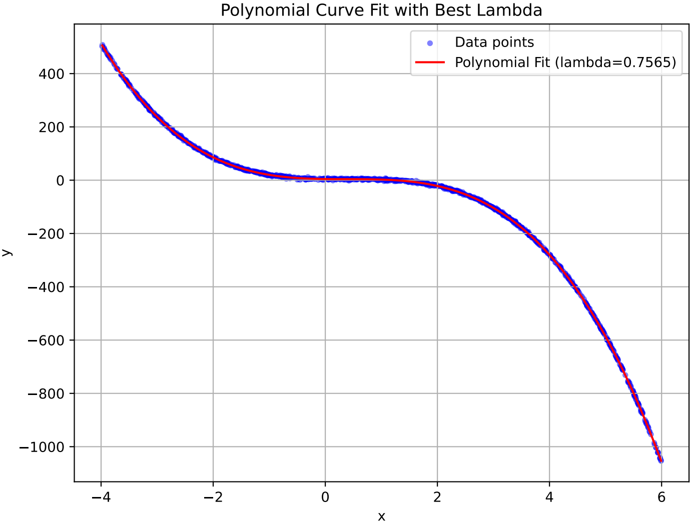
Finding optimal Lambda in Ridge Regression using Cross Validation



Finding optimal Lambda in Ridge Regression using Cross Validation



Finding optimal Lambda in Ridge Regression using Cross Validation



Finding optimal Lambda in Ridge Regression using Cross Validation