Fundamental Algorithms

Chapter 3: Five Essential Supervised Learning Algorithms

The Five

1. Linear Regression
2. Logistic Regression
3. Decision Tree Learning
4. Support Vector Machine
5. k-Nearest Neighbors

1. Linear Regression

Purpose: Learn a linear combination of features for real-valued predictions

Model:

$$f_{\mathbf{w},b}(x) = \mathbf{w}^{\mathbf{T}}\mathbf{x} + b$$

Goal: Find optimal $\mathbf{w}^*$ and b^* for most accurate predictions

Use Case: Regression tasks (predicting continuous values)

Linear Regression: Problem Setup

Given: Dataset $\{x_i, y_i\}_{i=1}^{N}$

- x_i: D-dimensional feature vector
- y_i: real-valued target

Predict: $y = f_{\mathbf{w},b}(x_{new})$

Key Features:

- Predicts real values, not classes
- Minimizes prediction error
- Simple yet effective

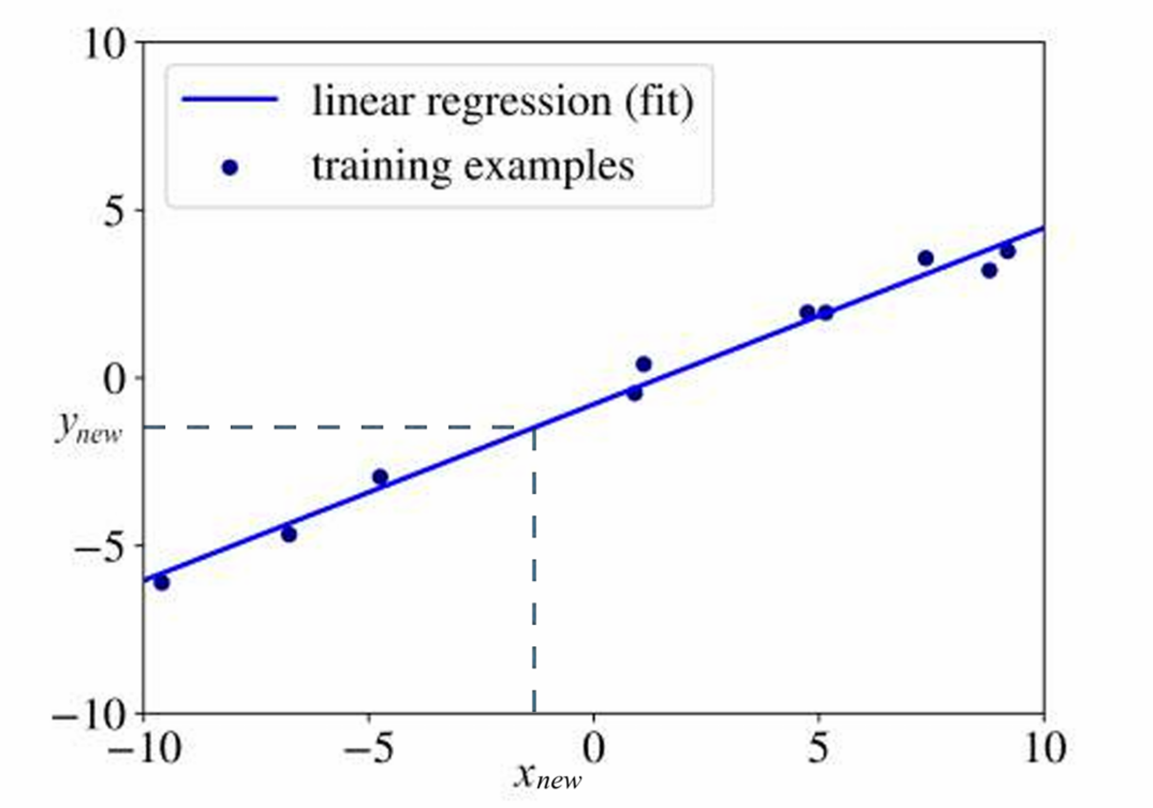

Figure 1: Linear Regression for one-dimensional examples.

Linear Regression: Visualization

Regression Models:

- 1D: A line
- 2D: A plane
- D-dimensions: A hyperplane

Linear Regression: Cost Function

Objective Function (Cost): (Average Loss or Empirical Risk)

$$C(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^{N} (f_{\mathbf{w},b}(x_i) - y_i)^2$$

Loss Function: $(f_{\mathbf{w},b}(x_i) - y_i)^2$ (squared error loss)

Why Squared Loss?

- Penalizes large errors more
- Smooth, continuous derivatives
- Easier optimization

Linear Regression: Optimization

Analytical Solution:

$$\frac{\partial C}{\partial \mathbf{w}} = 0, \quad \frac{\partial C}{\partial b} = 0$$

Gradient Descent Algorithm:

1. Initialize $\mathbf{w}$ and b randomly
2. Compute gradients: $\nabla_{\mathbf{w}} C = \frac{1}{N} \sum_{i=1}^{N} 2(f_{\mathbf{w},b}(x_i) - y_i)x_i$
3. Update: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} C$
4. Repeat until convergence

Linear Regression: Advantages

Advantages:

- Simple and interpretable
- Rarely overfits (low variance)
- Fast to compute
- Closed-form solution available

Linear Regression: Inductive Bias

What does Linear Regression assume?

Core Assumption: The relationship between features and target is **linear**

$$y = w_1 x^{(1)} + w_2 x^{(2)} + \ldots + w_D x^{(D)} + b$$

Inductive Bias Examples:

- **Good for:** House prices (size, location, bedrooms → price)
- **Bad for:** XOR problem (non-linear relationships)
- **Bad for:** Image recognition (pixel intensities aren't linearly related to objects)

Linear Regression: Inductive Bias

When Linear Bias Fails:

```
# XOR problem - linear model fails
X = [[0,0], [0,1], [1,0], [1,1]]
y = [0, 1, 1, 0]  # XOR truth table

# Linear regression will find poor fit
# because no line can separate XOR classes
```

Linear Regression: Implementation

Look at the associated Notebook.