Scaler Companion beta

Revision Notes on Linear Regression and Gradient Descent

Introduction to Linear Regression

Linear Regression is a fundamental statistical approach used to model and analyze the relationships between a dependent variable, denoted as Y, and one or more independent variables, denoted as $X_1, X_2, \ldots, X_d$. The primary goal is to find the best-fitting linear relationship through:

$$Y = W_1X_1 + W_2X_2 + \ldots + W_dX_d + W_0$$

where W_0 is the intercept, and $W_1, W_2, \ldots, W_d$ are the coefficients representing the weights of the independent variables 【4:6†transcript.txt】.

Simple vs. Multiple Linear Regression

- **Simple Linear Regression**: Involves a single independent variable: $Y = W_1X_1 + W_0$
- **Multiple Linear Regression**: Involves multiple independent variables: $Y = W_1X_1 + W_2X_2 + \ldots + W_dX_d + W_0$ Here, each W represents the impact of its respective X on Y 【4:7†transcript.txt】.

Interpretation of Coefficients

- **Positive Coefficient** ($W > 0$): Indicates a positive impact, meaning as X increases, Y also increases 【4:0†transcript.txt】.
- **Negative Coefficient** ($W < 0$): Indicates a negative impact, suggesting an inverse relationship— as X increases, Y decreases 【4:4†transcript.txt】.

Example

Scaler Companion beta

- W_1 for age might be negative, implying older age reduces lifespan.
- W_2 for weight might be positive, implying higher weight increases lifespan.

These relationships are crucial for understanding the significance of each feature in the model【4:4†transcript.txt】.

Defining a Linear Regression Model in Python

To define and work with a Linear Regression model programmatically:

1. **Initialization**: Set up with parameters like learning rate and iterations.
2. **Prediction Function**: Use the dot product to calculate predicted values: $predict(X) = X \cdot W + b$
3. **Evaluation Metric (R^2 Score)**: Measures how well model predictions approximate real data points【4:2†typed.md】.

Gradient Descent

Gradient Descent is an optimization method to update the parameters of the model in order to minimize a cost function, typically the Mean Squared Error (MSE): $Cost(W) = \frac{1}{m}\sum_{i=1}^{m}(\hat{Y_i} - Y_i)^2$

- **Learning Rate:** Controls step size towards the minimum. A high learning rate might overshoot, while too small can slow convergence.
- **Iterations**: Number of steps taken to update weights【4:1†transcript.txt】【4:5†transcript.txt】.

Gradient Calculation

The gradient provides the direction and magnitude for each step: $W = W - \alpha \cdot \nabla J(W)$

where α is the learning rate, and $\nabla J(W)$ is the derivative of the cost function【4:5†transcript.txt】.

The aim is to converge to the global minima, which represents the best fit for the model to the data:

- **Mean Squared Error (MSE):** Focuses on minimizing the sum of the squared differences between the predicted and actual values 【4:8†transcript.txt】.

Partial Differentiation and Combined Effects

- **Partial Differentiation**: Not directly addressed in the session, but understanding each parameter's contribution to the target variable's change is crucial.
- **Combined Effects**: Analyzing the interactive effect of variables by holding some constant to deduce the effect of others【4:6†transcript.txt】.

R^2 - Coefficient of Determination

R^2 Score is a metric for assessing the goodness-of-fit for regression models: $R^2 = 1 - \left(\frac{Residual\ Sum\ of\ Squares\ (RSS)}{Total\ Sum\ of\ Squares\ (TSS)}\right)$

- Measures the proportion of variance captured by the model vs. the total variance present in the data【4:16†transcript.txt】【4:18†transcript.txt】.

Understanding these elements forms the backbone for developing robust predictive models using linear regression and optimizations with gradient descent in machine learning.