Linear Regression

Linear Regression: Detailed Study Notes

1. Introduction to Linear Regression

Linear Regression is a fundamental supervised machine learning algorithm used for predicting a continuous target variable based on one or more predictor variables. It models the relationship between variables using a linear equation.

- **Simple Linear Regression:** Involves one predictor variable. Example: `life_satisfaction = θ₀ + θ₁ * GDP_per_capita` where θ_0 and θ_1 are model parameters.
- **Multiple Linear Regression:** Involves two or more predictor variables. The general equation is: $\hat{y} = \theta_0 + \theta_1x_1 + \theta_2x_2 + ... + \theta_nx_n$ where $\hat{y}$ is the predicted value, x_i are the feature values, and θ_i are the model parameters (including the bias term θ_0).

2. Vectorized Representation

The linear regression model can be concisely represented in vectorized form:

$\hat{y} = h_\theta(x) = \theta \cdot x$

where:

- θ is the model's parameter vector (including the bias term).
- x is the instance's feature vector (with $x_0 = 1$ for the bias term).
- $\theta \cdot x$ represents the dot product.

3. Model Training

The goal of training a linear regression model is to find the optimal values for the parameters θ that minimize the difference between the model's predictions and the actual target values in the training data. This is typically done by minimizing a cost function, such as the Mean Squared Error (MSE).

- **Methods for finding optimal θ:**
  - **Normal Equation:** A closed-form solution that directly calculates the optimal θ. This is computationally expensive for large datasets.
  - **Gradient Descent:** An iterative optimization algorithm that iteratively updates θ to minimize the cost function. Variants include batch, mini-batch, and stochastic gradient descent.

```
# Using scikit-learn for Linear Regression
```

```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
```

4. Model Evaluation

After training, the model's performance is evaluated using metrics such as:

- **Mean Squared Error (MSE):** The average squared difference between predicted and actual values.
- **Root Mean Squared Error (RMSE):** The square root of MSE, providing a more interpretable measure in the original units.
- (R^2) **score:** Represents the proportion of variance in the dependent variable explained by the model.

```
# Model Evaluation in scikit-learn
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

5. Regularization

Regularization techniques, such as Ridge Regression (L2) and Lasso Regression (L1), are used to prevent overfitting by adding a penalty term to the cost function. This penalty discourages large parameter values. Elastic Net is a combination of both L1 and L2 regularization.

```
# Elastic Net Regularization
from sklearn.linear_model import ElasticNet
elastic_net = ElasticNet(l1_ratio=0.3) # l1_ratio controls the mix of L1 a
elastic_net.fit(X_train, y_train)
```

6. Model Selection and Hyperparameter Tuning

- **Model Selection:** Choosing the appropriate type of linear regression model (simple vs. multiple, with or without regularization).
- **Hyperparameter Tuning:** Optimizing hyperparameters (e.g., regularization strength, learning rate in gradient descent) using techniques like grid search or randomized search with cross-validation.

7. Baseline Models

A baseline model, such as `DummyRegressor` in scikit-learn, provides a simple prediction (e.g., using the mean of the target variable) to compare against more complex models.

```
from sklearn.dummy import DummyRegressor
dummy_regr = DummyRegressor(strategy="mean")
dummy_regr.fit(X_train, y_train)
```

8. Making Predictions

Once the model is trained, predictions can be made on new, unseen data using the `predict()` method.

```
y_pred = lin_reg.predict(X_test)
```

Linear Regression is a crucial algorithm in machine learning due to its simplicity, interpretability, and wide applicability. Its ability to model linear relationships between variables makes it a valuable tool for prediction and understanding data, forming a strong foundation for more advanced techniques.