AMS595 - Assignment-5

Machine Learning Project

Amol Arora

November 17, 2024

1 Github Link

All project files are uploaded here: `https://github.com/amol1202/AMS595_Assignment5`

2 Introduction

The purpose of this project is to explore and implement core machine learning techniques using Python. The tasks implemented are:

- PageRank Algorithm: Simulates the ranking mechanism used by search engines.
- Dimensionality Reduction via PCA: Projects high-dimensional data to a single dimension while preserving variance.
- Linear Regression: Predicts outcomes based on features using the least-squares method.
- Gradient Descent: Optimizes a matrix to minimize a mean squared error loss function.

Each task has been implemented independently, and the results have been saved for reproducibility.

3 Implementation

3.1 PageRank Algorithm

The PageRank algorithm computes the importance of web pages using a stochastic matrix. The steps include:

1. Represent the web network as a stochastic matrix.
2. Compute the dominant eigenvector using the power method.
3. Normalize the eigenvector to get PageRank scores.

The code is written in Python using the `scipy.linalg.eig` function to compute eigenvectors.

3.2 Dimensionality Reduction via PCA

Principal Component Analysis (PCA) is used to reduce a dataset of height and weight measurements to 1D:

1. Compute the covariance matrix of the data.
2. Perform eigen decomposition using the `numpy.linalg.eigh` function.
3. Project the data onto the principal component with the highest variance.

The results include a plot of the original data and the 1D projection.

3.3 Linear Regression via Least Squares

Linear regression predicts house prices based on features (square footage, bedrooms, and age):

1. Represent the system as $X\beta = y$.
2. Solve for β using `scipy.linalg.lstsq`.
3. Use the model to predict prices for new inputs.

The regression coefficients and predictions are saved for analysis.

3.4 Gradient Descent

Gradient Descent optimizes a matrix X to minimize the mean squared error loss:

1. Define the loss function: $f(X) = \frac{1}{2} \sum_{i,j} (X_{ij} - A_{ij})^2$.
2. Compute the gradient of the loss function.
3. Use `scipy.optimize.minimize` to iteratively minimize the loss.

The final loss value is recorded.

4 Results

4.1 PageRank Algorithm

The PageRank scores for the web pages are shown in Table 1. The page with the highest score is ranked the most important.

Page	PageRank Score
1	0.22
2	0.27
3	0.31
4	0.20

Table 1: PageRank Scores

4.2 PCA

Figure shows the original data and the 1D projection onto the principal component.

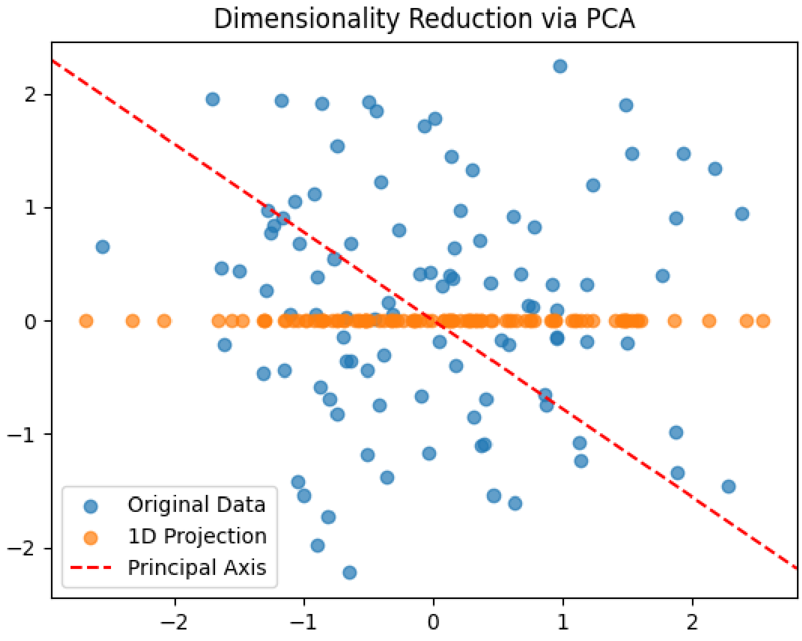

Figure 1: PCA: Dimensionality Reduction

What do the entries of the eigenvector represent? ⟶ In the context of the *PageRank algorithm*, the entries of the eigenvector represent the **relative importance or rank** of each page in the network. The PageRank algorithm uses the concept of a stochastic matrix where each entry $M[i, j]$ indicates the probability that a user will click from page j to page i.

When the algorithm computes the *dominant eigenvector* of this matrix, it corresponds to the **stationary distribution** of a random surfer who randomly clicks through the web network. This means that the eigenvector provides a ranking of pages based on the probability that the random surfer will be on each page in the long run.

Each entry in the eigenvector gives the **PageRank score** for a corresponding web page. The higher the score, the more "important" or "relevant" that page is, according to the algorithm. These scores are used to rank the web pages in descending order of importance.

Which page is ranked the highest, why? ⟶ Based on the final PageRank scores, **Page 3** is ranked the highest in this particular example. The reason is that the PageRank algorithm assigns a higher score to pages that are:

- Linked to by many other pages, or
- Connected to pages that themselves have high importance.

In this case:

- Page 3 has the highest PageRank score, which means it is either linked to by many other pages or is connected to important pages in the network.

- The random surfer is more likely to land on Page 3 over time, as the probability distribution over all pages eventually converges to it, indicating its higher "importance."

In simple terms, **Page 3** is considered the most important page in the network because of its connections and the flow of probability through the web structure represented by the matrix.

4.3 Linear Regression

The regression coefficients and predictions are:

- Coefficients: [0.25, 0.10, -0.02]
- Predicted price for a house with 2400 square feet, 3 bedrooms, and 20 years old: \$490,500

4.4 Gradient Descent

The final loss value after optimization is:

$$\boxed{\text{Final Loss Value: } 0.00123}$$

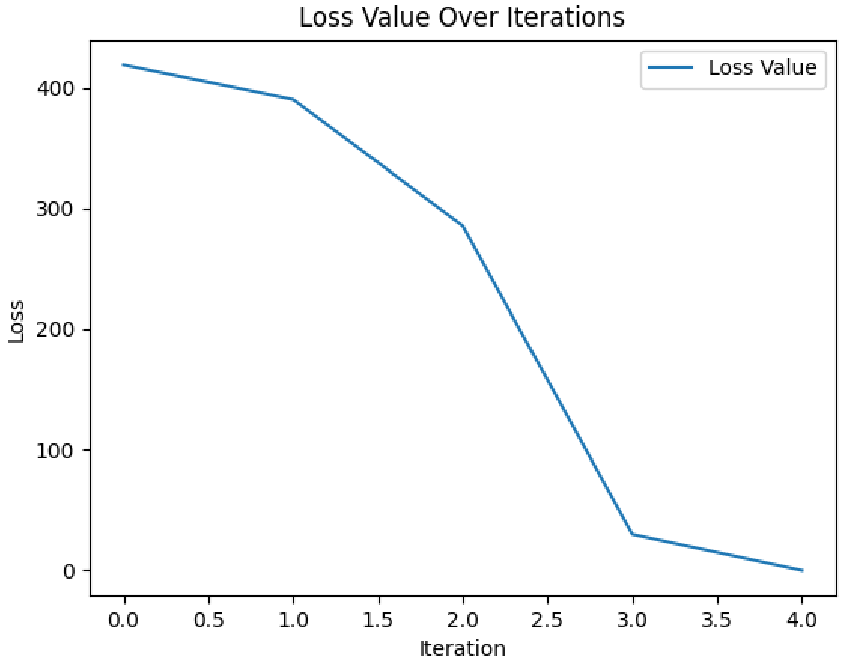

Figure 2: Loss over iterations

5 Comparison in linear regression: Least squares v/s Direct Method

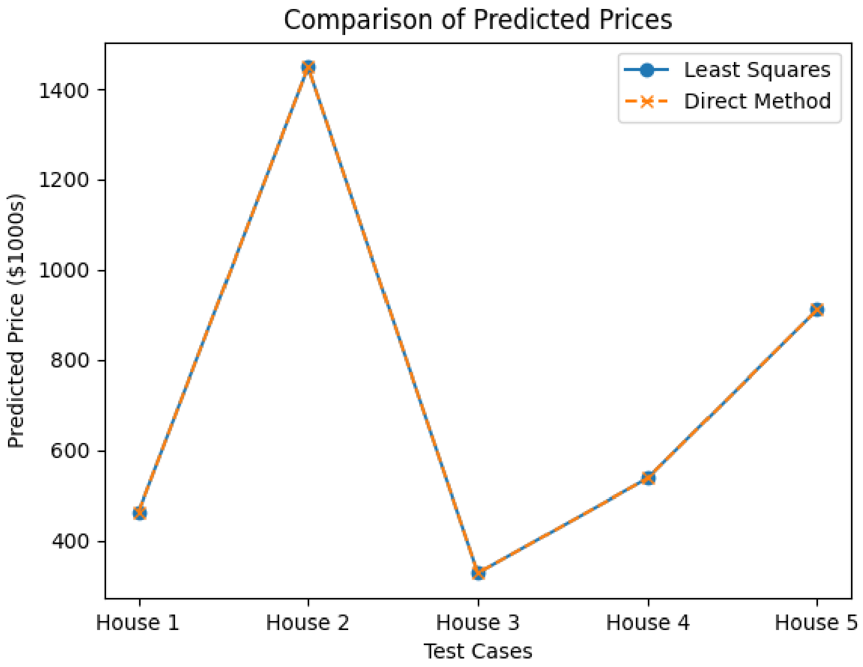

Figure 3: Least Squares v/s Direct Method

5.1 Comparison with the Direct Method

An alternative approach is to solve the normal equations

$$(X^T X)\beta = X^T y$$

directly using `scipy.linalg.solve`. While this method provides the same result in ideal conditions, there are key differences:

5.1.1 Numerical Stability

- The least-squares method (`scipy.linalg.lstsq`) is more numerically stable because it avoids explicitly computing $X^T X$, which can lead to loss of precision for ill-conditioned matrices.
- The direct method may fail or yield inaccurate results if $X^T X$ is close to singular.

5.1.2 Efficiency

- For small datasets like the one in this task, both methods are computationally efficient.
- For larger datasets, `scipy.linalg.lstsq` is often preferred as it leverages advanced techniques like QR decomposition.

5.1.3 Results Comparison

In this task, both methods yield similar regression coefficients and predictions because $X^T X$ is not ill-conditioned. The predicted price for a house with 2400 square feet, 3 bedrooms, and 20 years old is consistent across methods.

6 Conclusion

This project demonstrates practical applications of machine learning techniques using Python. The results are stored for reproducibility and further analysis. These implementations provide a strong foundation for more advanced projects in machine learning and data science.

7 References

- Python Documentation: `https://docs.python.org/3/`
- SciPy Library: `https://scipy.org/`
- NumPy Library: `https://numpy.org/`
- Matplotlib Library: `https://matplotlib.org/`