YBIGTA ML Assignment

Jungmook, Kang

Contents

1 Ensemble Method

1.1 Description of the Dataset

For the ensemble task, you will be looking at the breast cancer dataset provided by the UC Irvine Machine Learning Repository.

- There are 569 instances.
- 30 features with continuous values:
  - radius1
  - texture1
  - perimeter1
  - area1
  - smoothness1
  - compactness1
  - concavity1
  - concave_points1
- These features are computed from a digitized image of a fine needle aspirate of a breast mass.
- Essentially, these features capture the characteristics of the breast mass image.

1.2 Task Objective

Our task with the ensemble method: Binary Classification.
The model's main goal is to determine whether or not a patient has breast cancer based on the features provided by the dataset.

1.3 Model Training and Evaluation

- Finish the #TODO's in Section 1.3 to train the decision tree model and random forest model as well as obtain predictions from these trained models. You are expected to tune the hyperparameters and fill in the blanks to allow the for loop to run. Report the graphs as well as the train and test accuracy for both models in the report.

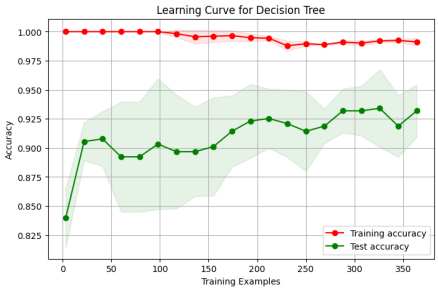

Figure 1: Learning Curve for Decision Tree

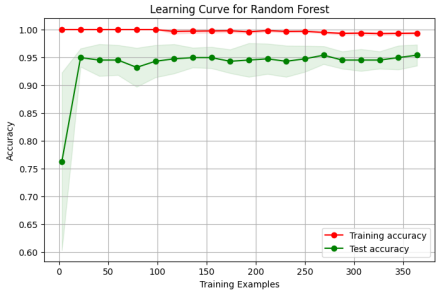

Figure 2: Learning Curve for Random Forest

Figure 3: Comparison of Learning Curves for Models

- What is the better model and please provide evidence and supporting arguments that back your decision?

  Training Accuracy Comparison: The Decision Tree model has a training accuracy close to 1.0, suggesting it memorizes the training data, while the Random Forest model also achieves near 1.0 accuracy, which is expected due to its ensemble nature. Both models effectively learn the training data.

  Test Accuracy Comparison: The Decision Tree stabilizes around 0.92 - 0.93 in test accuracy, whereas the Random Forest achieves a higher test accuracy of around 0.95. This suggests that Random Forest generalizes better to unseen data.

  Overfitting Analysis: The Decision Tree exhibits a large accuracy gap between training and test data (7-8%), indicating overfitting, whereas the Random Forest model maintains a smaller accuracy gap (4-5%), leading to better generalization.

  Model Stability: The Decision Tree model's test accuracy fluctuates more, indicating less stability, while the Random Forest model maintains a smoother test accuracy curve, suggesting greater robustness.

  Based on these observations, the **Random Forest** model is the preferred choice due to its higher test accuracy, reduced overfitting, better generalization, and greater stability. Thus, **Random Forest is the preferred model.**

2 Principal Component Analysis

2.1 Description of the Dataset

For the PCA task, we will be using the wine dataset provided by the UC Irvine Machine Learning Repository.

- There are 178 instances where there are three main cultivars of wine (three different clusters).
- 13 features:
  - Alcohol
  - Malic Acid
  - Ash
  - Magnesium
  - ... and many more

2.2 Task Objective

Our main goal with using PCA is to use dimension reduction on the dataset. However, we would also like to see whether or not our three different cultivars of wine can be clearly separated in this reduced-dimensional space.

2.3 Implementing PCA

- Finish the #TODO's in section 2.1) to implement a custom version of the PCA algorithm. Report the plots.

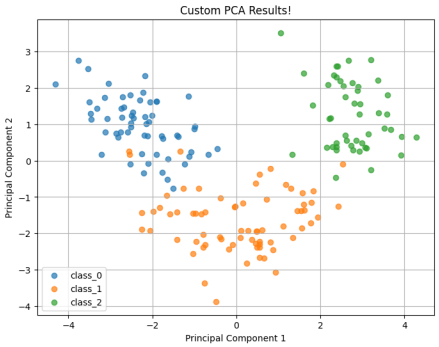

Figure 4: Custom PCA

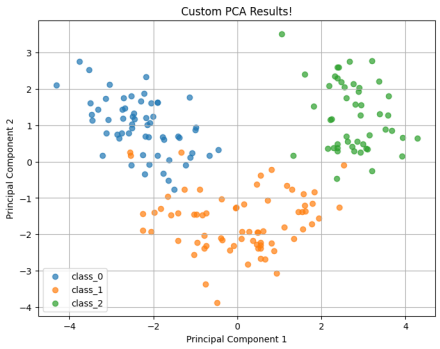

Figure 5: Learning Curve for Random Forest

Figure 6: Sklearn PCA

- What are the benefits and the disadvantages of PCA? Could you provide another dimensionality reduction method that can be used apart from PCA?

Benefits of PCA

- **Dimensionality Reduction:** PCA reduces the number of dimensions while retaining most of the data's variance, simplifying visualization and computation.
- **Feature Extraction:** PCA creates new, uncorrelated features (principal components) that often better represent the data.
- **Noise Reduction:** By focusing on the most significant components, PCA reduces the effect of noise in the data.

Disadvantages of PCA

- **Loss of Interpretability:** Principal components are linear combinations of original features, making them harder to interpret.
- **Linear Assumption:** PCA assumes linear relationships in the data and may not perform well for datasets with complex non-linear relationships.
- **Sensitive to Scaling:** PCA is affected by the scaling of features. If features are not standardized, results may be biased toward features with larger variance.

Alternative Dimensionality Reduction Method: t-SNE

- **t-SNE:** A non-linear dimensionality reduction technique primarily used for data visualization.
- Maps high-dimensional data into 2D or 3D spaces while preserving the local structure of the data.
- Focuses on preserving the relative distances between neighboring points.
- Excellent for visualizing clusters in high-dimensional data.
- Handles non-linear relationships better than PCA.
- Computationally expensive for large datasets.
- Poor interpretability for features or embeddings.

3 Support Vector Machine

3.1 Description of the Dataset

For the SVM task, we will be using the iris dataset provided by sklearn.

- 150 instances
- 4 features:
  - Sepal Length
  - Sepal Width
  - Petal Length
  - Petal Width
- These features describe the physical characteristics of Iris flowers.

Our task with the SVM is to classify flowers into different species of Iris. Since we are implementing a binary SVM, we will reduce the number of classes by one (from three to two). Additionally, we will only use two features to simplify visualization and better understand the decision boundary created by our SVM model.

3.2 Description of the Sub-gradient Form of the SVM

The SVM objective is to find a decision boundary (hyperplane) that maximizes the margin between the two classes while ensuring correct classification. Hence, we are looking to minimize the loss function below:

$$L(w, b) = \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i + b))$$

- The first term (LHS) adjusts the margin between the two classes. It effectively tries to find a decision boundary that maximizes the distance between the two classes.
- λ controls the tradeoff between the margin maximization term and the hinge loss term. It determines whether the model focuses more on maximizing the margin or classifying correctly.
- The second term (RHS) represents the hinge loss, which penalizes the SVM if it classifies a point/sample incorrectly.

3.3 Hard Margin vs Soft Margin SVM

Hard Margin SVM

The objective of a Hard Margin SVM is to find a hyperplane that perfectly separates the two classes with the largest possible margin. It assumes that the data is linearly separable.

$$\begin{aligned} \text{Minimize:} \quad & \frac{1}{2}\|w\|^2 \\ \text{Subject to:} \quad & y_i(w \cdot x_i + b) \geq 1, \quad \forall i \end{aligned}$$

Soft Margin SVM

The Soft Margin SVM introduces a regularization parameter and slack variables ξ_i to allow for some misclassified points, making it suitable for non-linearly separable or noisy data.

$$\begin{aligned} \text{Minimize:} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{Subject to:} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \end{aligned}$$

Key Differences

- **Hard Margin:** Assumes perfectly linearly separable data, no tolerance for misclassifications.
- **Soft Margin:** Allows for some misclassified points, controlled by the parameter C.

Advantages and Disadvantages

- **Hard Margin SVM:**
  - **Advantages:**
    - Works well when data is clean and linearly separable.
    - Provides a unique solution.
  - **Disadvantages:**
    - Fails when the data is not linearly separable.
    - Sensitive to noise and outliers.
- **Soft Margin SVM:**
  - **Advantages:**
    - Handles non-linearly separable data and noisy datasets.

    - Provides flexibility by allowing some misclassifications.
- **Disadvantages:**
    - Requires careful tuning of the regularization parameter C.
    - May lead to overfitting if C is not properly set.

3.4 SVM implement

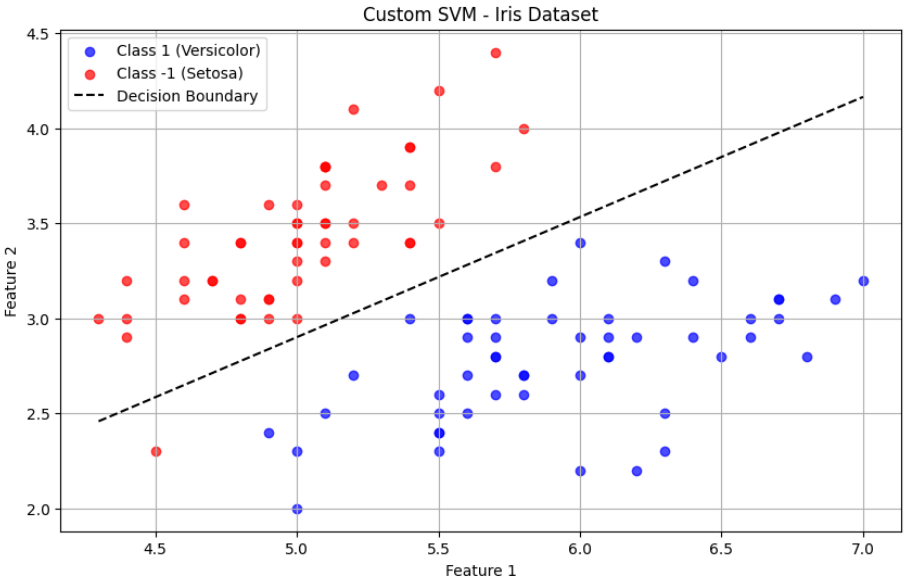

Figure 7: Custom SVM