Homework 1

CSE 902: Pattern Recognition and Analysis
Instructor: Dr. Arun Ross
Total Points: 120
Due Date: December 4, 2024, 11:00 PM

Please read this carefully:

1. **You are permitted to discuss the following questions with others in the class. However, you *must* write up your *own* solutions to these questions. Any indication to the contrary will be considered an act of academic dishonesty. Copying from *any source* constitutes academic dishonesty.**
2. **A neatly typed report is expected (alternately, you can neatly handwrite the report and then scan it). The report, in PDF format, must be uploaded in D2L by 11:00pm on December 4, 2024. Late submissions will not be graded. In your submission, please include the names of individuals you discussed this homework with and the list of external resources (e.g., AI tools, websites, other books, articles, etc.) that you used to complete the assignment (if any).**
3. **When solving equations or reducing expressions you must explicitly show every step in your computation and/or include the code that was used to perform the computation. Missing steps or code will lead to a deduction of points. If you use any AI agent to solve the problem, you must include the prompts that were used to query the agent.**
4. **Code developed as part of this assignment must be (a) included as an appendix to your report or inline with your solution, and (b) archived in a single zip file and uploaded in D2L. Including the code without the outputs or including the outputs without the code will result in deduction of points.**
5. **The report (PDF) and code (ZIP) must be uploaded as two separate files in D2L.**

---

1. [10 points] Based on the paper “Deep Boltzmann Machines" by Ruslan Salakhutdinov and Geoffrey Hinton (2009), answer the following questions:
   - What distinguishes the architecture of Deep Boltzmann Machines (DBMs) from that of Deep Belief Networks (DBNs), particularly in the way they handle the layers of hidden units and connections?
   - The paper discusses the challenges of training DBMs due to their complex energy landscape. What methods or approximations do the authors propose to overcome these difficulties, and how do these methods ensure efficient learning?
2. [10 points] Based on the paper “Decision Tree SVM: An Extension of Linear SVM for Non-linear Classification" by Feiping Nie, Wei Zhu, Xuelong Li (2020), answer the following questions:
   - How does the Decision Tree SVM (DTSVM) combine the strengths of decision trees and Support Vector Machines (SVMs) to handle non-linear classification tasks, and what are the key differences between DTSVM and traditional SVMs?

- What role does the recursive partitioning mechanism play in the construction of the Decision Tree SVM, and how does it ensure that the non-linear decision boundaries are effectively learned?

3. [10 points] Based on the paper "Dense Associative Memory for Pattern Recognition" by Dmitry Krotov and John J. Hopfield (2016), answer the following questions:

   - How does the Dense Associative Memory (DAM) model proposed in the paper extend the capabilities of traditional Hopfield networks, particularly in terms of capacity and robustness?
   - The paper emphasizes the role of higher-order nonlinear interactions in shaping the energy landscape of the DAM model. How do these interactions enhance pattern recognition performance, and what implications do they have for the stability of stored memories?

4. [30 points] [Textbook] Consider a version of the Balanced Winnow training algorithm (Algorithm 7). Classification of test data is given by line 2. Compare the converge rate of Balanced Winnow with the fixed-increment, single-sample Perceptron (Algorithm 4) on a problem with large number of redundant features, as follows.

   - Generate a training set of 2000 100-dimensional patterns (1000 from each of two categories) in which only the first ten features are informative, in the following way. For patterns in category ω_1, each of the first ten features are chosen randomly and uniformly from the range $+1 \leq x_i \leq 2$, for i = 1, ..., 10. Conversely, for patterns in ω_2, each of the first ten features are chosen randomly and uniformly from the range -2 $\leq x_i \leq$ -1. All other features from both categories are chosen from the range -2 $\leq x_i \leq$ +2.
   - Construct by hand the obvious separating hyperplane.
   - Adjust the learning rates so that your two algorithms have roughly the same convergence rate on the full training set when only the first ten features are considered. That is, assume each of the 2000 training patterns consists of just the first ten features.
   - Now apply your two algorithms to 2000 50-dimensional patterns, in which the first ten features are informative and the remaining 40 are not. Plot the total number of errors versus iteration.
   - Now apply your two algorithms to the full training set of 2000 100-dimensional patterns.
   - What conclusion can you draw?

5. [30 points] To create a two-class problem that SVM struggles to solve effectively, the following factors must be considered:

   (a) **Highly Non-Linear and Complex**: The dataset must have a decision boundary that is highly non-linear, making it difficult to separate using standard kernels.

   (b) **Highly Imbalanced**: Imbalanced datasets can pose a challenge for an SVM.

   (c) **Noisy**: Noise that is added to increase overlap between classes can make the separation task more difficult.

   In this regard, we generate a dataset consisting of two classes ($y \in \{-1, +1\}$) distributed in a double spiral pattern in the 2D plane. The points for each class alternate along the spiral arms, creating a highly non-linear distribution. Gaussian noise is added to the points to further complicate the separation task.

   The resulting spiral dataset can be found here.

   - Plot this data by using different colors and markers for the two classes.

- Train an SVM model with various kernels (linear, RBF, polynomial) on the spiral dataset.
- Evaluate the performance of each kernel in terms of accuracy and decision boundaries. Find out a way to draw the decision boundaries generated by each kernel along with the data points.
- Explain why certain kernels struggle on this dataset.
- Can you create a 2D synthetic dataset where all kernels fail to produce good results?

6. [30 points] In machine learning, accurately assessing a model's performance is critical for ensuring its ability to generalize to unseen data. Estimation and resampling techniques help achieve this by splitting or resampling data in systematic ways to simulate real-world variability. However, the choice of technique can significantly influence the evaluation results.

   Consider a 3D Spiral Dataset consisting of 4 classes. The class labels are indicated at the end of every row. Visualize the dataset by plotting the points using different colors and markers so as to distinguish between the 4 classes.

   The objective of this problem is to evaluate and compare the effectiveness of various resampling and estimation techniques, including **Holdout**, **K-Fold Cross-Validation**, **Stratified K-Fold**, **Leave-One-Out Cross-Validation (LOOCV)**, **Bootstrap**, and **Jackknife**, in assessing the performance of a classifier trained on a complex 3D spiral dataset. The study aims to understand how these methods impact performance metrics, computational cost, and robustness to noise in the data.

   Dataset:

   The **3D Spiral Dataset** should be used for this study:

   - **Structure**: Four intertwined spirals in 3D space, with each spiral representing a distinct class.
   - **Data Points**: $n = 40,000$ points (10,000 per class).
   - **Features**: x_1, x_2, x_3 (representing spatial coordinates).
   - **Classes**: Four ($y \in \{0, 1, 2, 3\}$).
   - **Noise Levels**: Gaussian noise with variance 0.25 is added to introduce complexity.

   Resampling and Estimation Techniques:

   The following resampling and estimation techniques should be applied with specified parameters:

   (a) **Holdout Validation**:
      - Split the dataset into **70% training (28,000 points)** and **30% testing (12,000 points)**.
      - Train the classifier on the training set and evaluate it on the test set.

   (b) **K-Fold Cross-Validation**:
      - Use $k = 5$ and $k = 10$ folds.
      - Divide the dataset into k equal-sized subsets.
      - Train on $k-1$ folds (e.g., 32,000 points for $k = 5$) and test on the remaining fold (e.g., 8,000 points for $k = 5$).

   (c) **Stratified K-Fold Cross-Validation**:
      - Use $k = 5$.
      - Ensure that each fold maintains the class distribution of the original dataset.

   (d) **Leave-One-Out Cross-Validation (LOOCV)**:

- Train on $n-1$ points and test on 1 point, repeated $n = 40,000$ times.

(e) **Bootstrap**:
  - Generate 100 bootstrap samples by randomly sampling **70% of the data with replacement (28,000 points)**.
  - Evaluate using the remaining **out-of-bag (OOB)** samples.

(f) **Jackknife**:
  - Perform $n = 40,000$ iterations, each leaving out one unique sample.
  - Compute overall performance metrics based on the resampled datasets.

Classifier:

- Use **Random Forest Classifier**:
  - **Number of Trees**: 100
  - **Max Depth**: Unlimited
  - **Random State**: 42 for reproducibility

Evaluation Metrics:

The following metrics will be collected for each resampling and estimation technique:

- **Accuracy**: Percentage of correctly classified samples.
- **F1-Score**: Weighted average of precision and recall.
- **Time Complexity**: Total computation time for each technique.

Tasks:

(a) **Model Training and Evaluation**:
  - Train the Random Forest Classifier using each resampling technique.
  - Document the accuracy, F1-score, and computation time.

(b) **Analysis**:
  - Compare metrics across resampling techniques.
  - Highlight trade-offs between computational cost, robustness, and performance.

Expected Outcomes:

- **Performance Table**: A table summarizing the metrics for each resampling technique.
- **Visualizations**: Bar charts comparing accuracy, F1-score, and time complexity across techniques.
