Handwritten Digit Classification with SVM or KNN

Ileana Bocage
Final Project Report
EEL6825: Pattern Recognition and Intelligent Systems
Electrical and Computer Engineering, Fall 2024
University of Florida

1. INTRODUCTION

A. PROJECT OVERVIEW

Classifying handwritten digits is a popular machine learning problem where the goal was to accurately categorize images of digits (0-9). This project compared the performance of Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) on this task. Both algorithms are widely used but differ in their approach to classification. We used the well-known MNIST dataset, consisting of 70,000 grayscale images (60,000 training images and 10,000 test images), to train and test these models. The primary objective was to determine which algorithm performed better in accurately classifying handwritten digits and to gain insights into their strengths and weaknesses.

B. MOTIVATION

Handwritten digit recognition has practical applications in areas like form processing and digital banking. This project compared two classification methods, SVM and KNN, to optimize their use in real-world image recognition tasks. It aligned with the course objectives by applying key concepts such as data preprocessing, statistical learning theory, and parametric vs. non-parametric techniques. Through this project, we demonstrated handling high-dimensional data and improving model performance with techniques learned in class.

2. BACKGROUND AND RELATED WORK

Handwritten digit recognition has been extensively studied in the field of machine learning, with the MNIST dataset serving as a benchmark for various classification algorithms. Several studies have explored different methods, ranging from traditional machine learning techniques to deep learning models.

Rastogi et al. (2022) conducted a comprehensive study comparing traditional machine learning algorithms like SVM and KNN with deep neural networks (DNNs). Their findings revealed that while DNNs offer superior accuracy, SVM and KNN remain competitive, especially in resource-constrained environments. This highlights the importance of exploring and optimizing these traditional techniques.

Liu et al. (2020) also investigated the performance of various classifiers, including SVM, KNN, and neural networks, on the MNIST dataset. Their research emphasized the significance of preprocessing techniques like feature scaling and dimensionality reduction in improving the performance of non-deep learning models.

Building upon these findings, this project aims to:

1. Compare SVM and KNN: Delve deeper into the differences between performance of SVM and KNN on the MNIST dataset.
2. Optimize Performance: Investigate the impact of preprocessing techniques, such as normalization and PCA, on the accuracy and efficiency of both algorithms.
3. Identify Trade-offs: Analyze the trade-offs between accuracy and computational cost for SVM and KNN, particularly in the context of large-scale datasets.

By addressing these aspects, this project seeks to provide valuable insights into the strengths and limitations of SVM and KNN for handwritten digit recognition and to contribute to the ongoing research in this field.

3. METHODOLOGY

A. APPROACH

This project aimed to classify handwritten digits using Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). The MNIST dataset was used to train and evaluate these models. The first step was to perform data preprocessing, where the images were normalized to a common scale and flattened into one-dimensional feature vectors. The data was split into a training set, validation set, and test set. Principal Component Analysis (PCA) was then applied to the data to reduce the dimensionality of the feature space and to reduce the time required to train the model.

Next, the SVM and KNN algorithms were implemented. For SVM, both linear and Radial Basis Function (RBF) kernels were explored to capture linear and non-linear relationships in the data, and compared on performance. The regularization parameter C was tuned to balance the trade-off between maximizing the margin and minimizing classification errors. For KNN, the Euclidean distance metric was used to measure similarity between data points, and the optimal number of neighbors (k) was determined through cross-validation.

Finally, the trained models were evaluated on a held-out test set using metrics such as accuracy, precision, recall, F1-score, and confusion matrix. The computational efficiency of both models was also assessed.

B. TECHNICAL DETAILS

Data Preprocessing

- *Normalization:* Each pixel value was scaled to the range [0, 1] to ensure consistent feature scaling.
- *Flattening:* Each 28x28 image was reshaped into a 784-dimensional vector.
- *Split:* Data was split into a training set, a validation set, and a test set consisting of 48,000, 12,000, and 10,000 samples respectively.
- *PCA:* PCA was applied to reduce the dimensionality of the feature space to 100 dimensions.

SVM

- *Kernel Function*: Linear and RBF kernels were both implemented.
- *Regularization:* The regularization parameter C was tuned to control the trade-off between maximizing the margin and minimizing training errors. It was determined that a C value of 1.0 gave results with high accuracy.

KNN

- *Distance Metric:* Euclidean distance was used to measure the distance between data points.
- *Neighbor Selection:* The k-nearest neighbors of a query point were determined based on their Euclidean distance. A k value of 3 was used.
- *Majority Voting:* The majority class among the k-nearest neighbors was assigned as the predicted class.
- *Cross-validation:* Cross-validation was used to give a better accuracy score and to determine an optimal k value.

C. EXPERIMENTAL DESIGN

The MNIST dataset was divided into training and testing sets. The training set was used to train both the SVM and KNN models. For SVM, various hyperparameters, such as the regularization parameter C and the kernel type (linear or RBF), were tuned to optimize performance. For KNN, the optimal number of neighbors (k) was determined through cross-validation.

Once trained, both models were evaluated on the validation set and then on the held-out test set. Performance metrics, including accuracy, precision, recall, and F1-score, were calculated to assess the models' predictive capabilities.

An analysis was conducted to compare the strengths and weaknesses of the models. The impact of different hyperparameters, preprocessing techniques, and model architectures on the overall performance was investigated. By comparing the accuracy, precision, recall, F1-score, and computational efficiency of the two models, insights were gained into their suitability for different applications.

4. RESULTS AND ANALYSIS

A. RESULTS

SVM ACCURACY

Linear Kernel:
SVM Accuracy on validation set: 0.69
SVM Accuracy on test set: 0.71

Radial Bais Function (RBF) Kernel:
SVM Accuracy on validation set: 0.98
SVM Accuracy on test set: 0.98

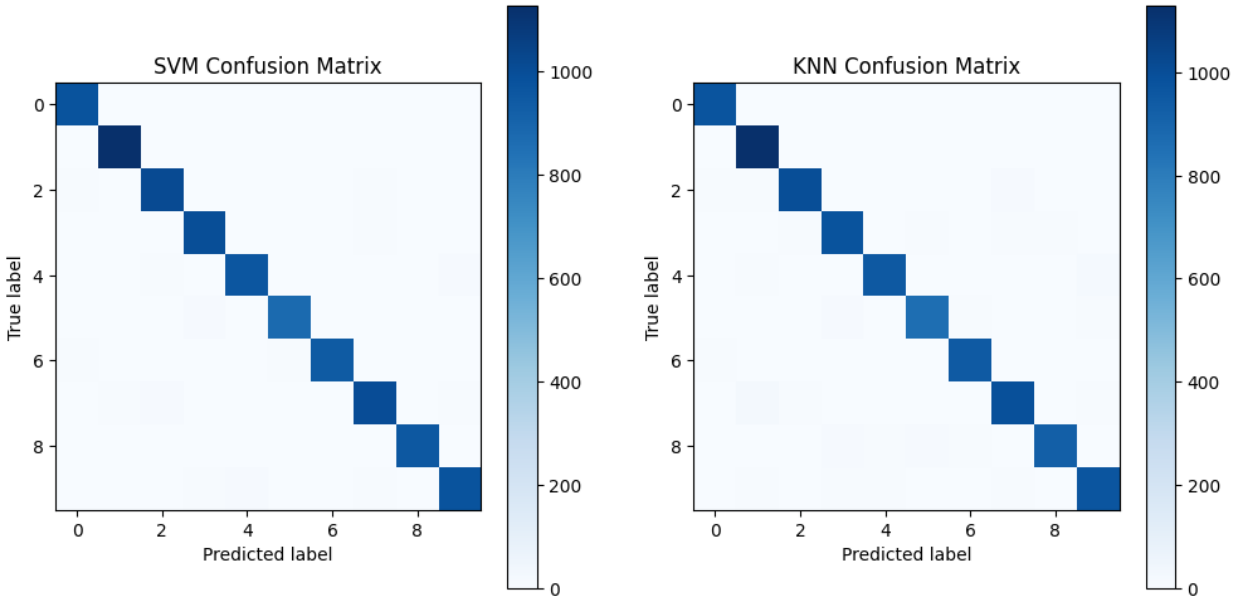
SVM Classification Report:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	980
1	0.99	0.99	0.99	1135
2	0.98	0.98	0.98	1032
3	0.98	0.99	0.98	1010
4	0.98	0.98	0.98	982
5	0.98	0.98	0.98	892
6	0.99	0.98	0.99	958
7	0.98	0.98	0.98	1028
8	0.98	0.98	0.98	974
9	0.98	0.97	0.97	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

KNN RESULTS
KNN Accuracy on validation set: 0.97
KNN Accuracy on test set: 0.97
Cross-validation score: 0.97

KNN Classification Report:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	980
1	0.96	1.00	0.98	1135
2	0.98	0.97	0.98	1032
3	0.97	0.97	0.97	1010
...				
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000



B. ANALYSIS

The linear kernel in SVM performed poorly due to its inability to model complex, non-linear relationships in the dataset, underscoring the limitations of linear decision boundaries for handwritten digit classification, where class overlap is frequent. In contrast, the RBF kernel excelled by effectively capturing these non-linear patterns, leading to high precision, recall, and F1-scores across all classes. The confusion matrix highlights minimal misclassifications, confirming the RBF kernel's strength in separating overlapping classes and offering superior performance.

Both SVM and KNN exhibited strong overall accuracy, with SVM achieving 98% and KNN 97%. SVM's RBF kernel was particularly adept at handling non-linear decision boundaries, resulting in fewer misclassifications. KNN, though slightly less accurate, remained competitive with a closely matched test accuracy. However, KNN's reliance on nearest neighbor searches increases computational costs during prediction, especially with large datasets, whereas SVM's higher training complexity leads to faster inference times. Additionally, KNN's sensitivity to the choice of k and susceptibility to noisy data make its performance more variable, whereas SVM benefits from effective kernel selection. Overall, SVM with the RBF kernel proved more robust and accurate for this task, while KNN offers a simpler, reliable alternative for contexts favoring ease of implementation.

5. DISCUSSION AND CONCLUSION

A. CONCLUSIONS

This project demonstrated the effectiveness of SVM and KNN, combined with PCA for dimensionality reduction, in handwritten digit classification. SVM with the RBF kernel achieved a 98% test accuracy, showcasing its ability to handle complex, overlapping patterns, making it ideal for intricate datasets. KNN performed well with 97% accuracy, affirming its reliability despite its simplicity. PCA's role in reducing computational complexity while maintaining essential features enhanced efficiency without compromising performance. These findings emphasize the importance of selecting appropriate classifiers and dimensionality reduction methods in pattern recognition, particularly for high-dimensional datasets like handwritten digits. This has practical implications in real-world applications such as automated postal services, check processing, and document digitization, where accurate digit recognition is essential.

B. FUTURE WORK

Future research could explore several avenues to enhance this project. First, implementing hyperparameter optimization techniques, such as grid search or Bayesian optimization, could further

refine SVM and KNN performance by identifying optimal parameters. Additionally, integrating ensemble methods like bagging or boosting could enhance classification accuracy by combining the strengths of multiple models. Another potential improvement involves experimenting with advanced feature extraction techniques, such as convolutional neural networks (CNNs), to capture more intricate patterns in the data. Furthermore, expanding the dataset to include more diverse and complex handwritten digits or testing the models on other handwritten datasets could assess the generalizability of the approach. Finally, real-time deployment of the system in practical applications, such as mobile digit recognition or automated form processing, could provide valuable insights into its operational robustness and scalability.

6. REFERENCES

- R. Rastogi, C. Verma, D. Sharma and P. K. Goyal, "A Comparative Statistical Analysis Between ML Algorithms & DNN Techniques Using MNIST Dataset," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 317-321, doi: 10.1109/ICAC3N56670.2022.10074302.

- W. Liu, J. Wei and Q. Meng, "Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition," 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 2020, pp. 587-590, doi: 10.1109/ AEECA49918.2020.9213482.

- J. Allibhai, “Building a K-nearest-neighbors (K-nn) model with Scikit-Learn,” Medium, https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a (accessed Dec. 2, 2024).