Manhattan (L1) and Euclidean (L2) Distance Metrics in kNN Classification

Introduction to Distance Metrics

In kNN classification, the choice of distance metric can significantly influence the model's performance. The Euclidean distance, or L2 distance, is commonly used for its geometric interpretation in space. However, the Manhattan distance, or L1 distance, can offer advantages in certain contexts, such as when differences in individual dimensions are of interest or when dealing with high-dimensional data.

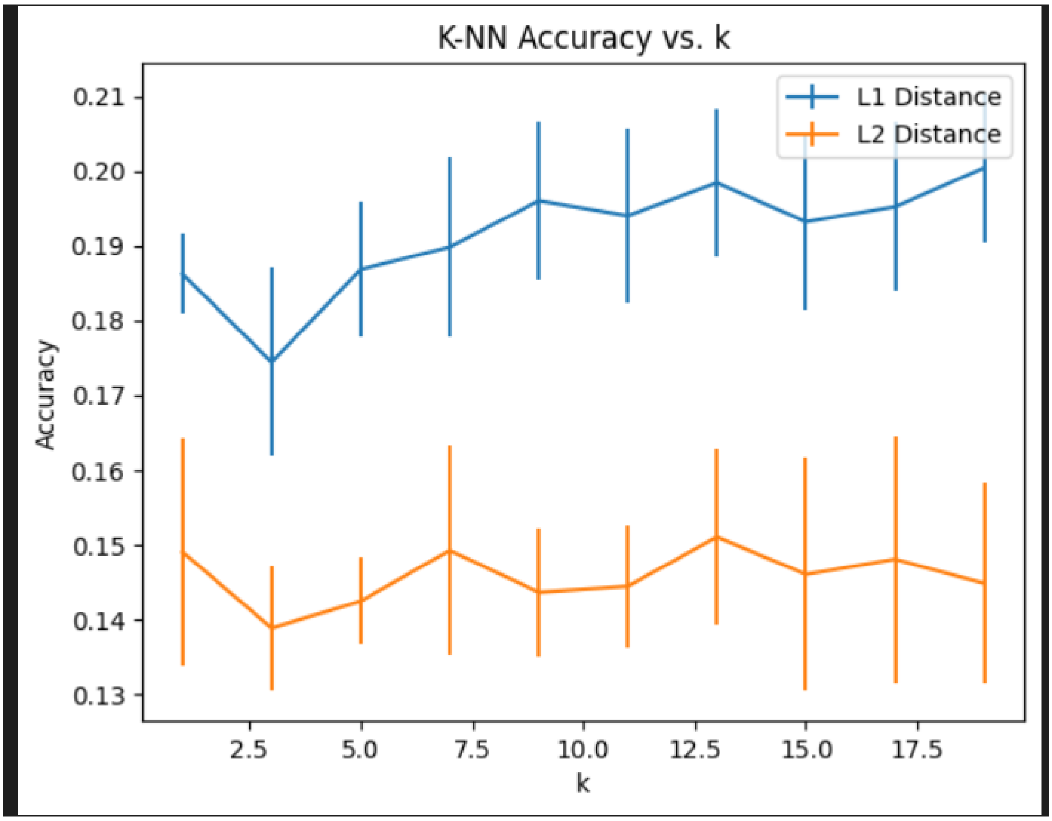
Methodology for Comparing L1 and L2 Distances

After the dataset is loaded and the labels have been printed, the images are read as grayscale using cv2.IMREAD_GRAYSCALE and the cv2.imread() function and is then appended to the TRAIN_DATA. The code is performing classification on a dataset using two different distance metrics: L1 distance (also known as Manhattan distance) and L2 distance (also known as Euclidean distance). The dataset is divided into five folds for cross-validation, and the mean and standard deviation of the classification accuracies are calculated over the five folds for each value of k from 1 to 20. The code uses the numpy and matplotlib.pyplot libraries. The variable k_max is set to 20, which is the maximum value of k to be used in k-NN classification. The functions l1_dist and l2_dist calculate the L1 and L2 distances between two points in the dataset, respectively. The variables mean_accuracies_l1, std_accuracies_l1, mean_accuracies_l2, and std_accuracies_l2 will store the mean and standard deviation of the classification accuracies for L1 and L2 distance metrics, respectively, for each value of k. The code then enters a loop over the odd values of k from 1 to 20. For each value of k, the code performs 5-fold cross-validation. The inner loop iterates over the five folds, using one fold for validation and the other four folds for training. For each validation point, the L1 and L2 distances between that point and all training points are calculated. The distances are sorted, and the k training points with the smallest distances are selected as the k-nearest neighbors. The class labels of these neighbors are counted, and the class label that occurs most frequently is assigned as the predicted class label for the validation point. The classification accuracy is calculated as the number of correctly predicted validation points divided by the total number of validation points. The accuracy is stored for each fold, and the mean and standard deviation of the accuracies are calculated over the five folds. Finally, the mean and standard deviation of the accuracies for each value of k are plotted using the matplotlib.pyplot library. The plot will show how the classification accuracy varies with the number of nearest neighbors (k) and the choice of distance metric.

After that we can see in the plot that the L1 accuracy is better than the L2 accuracy for this experiment.

Results

Accuracy Results: A table and accompanying plots were generated to display the average accuracy obtained with kNN classifiers using L1 and L2 distances across various values of K. These results illustrate how each distance metric performs in classifying the dataset.

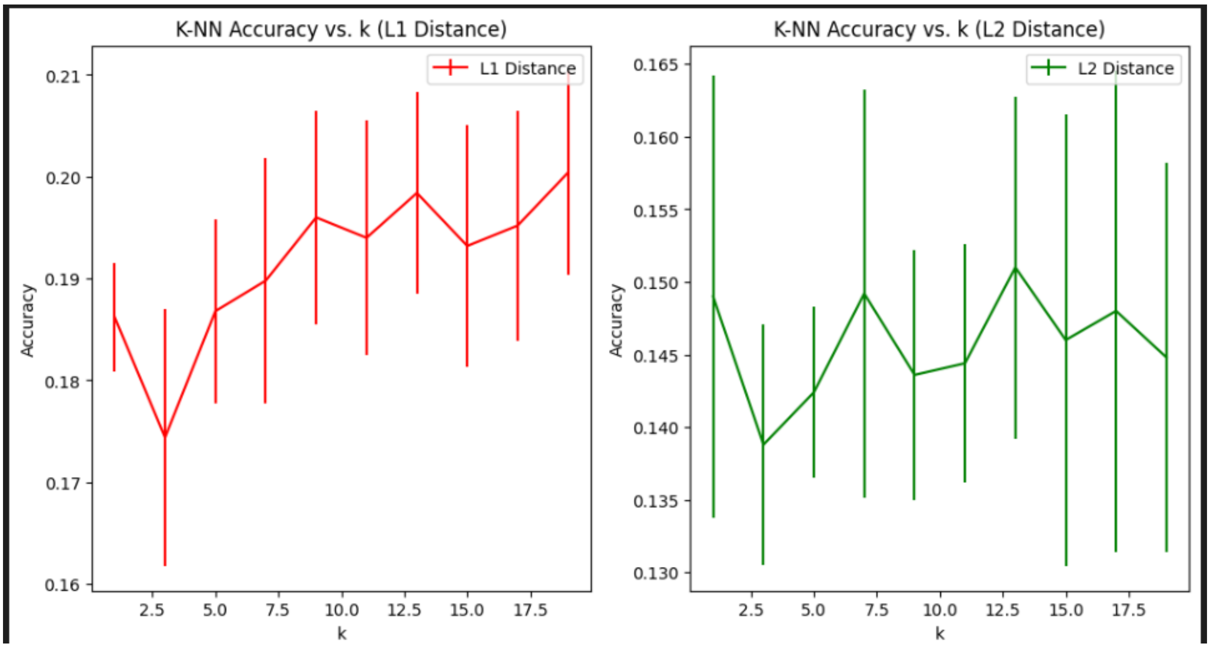


Optimal K Values: The analysis identified the optimal K values for both L1 and L2 distance metrics, highlighting differences in how these metrics aggregate neighbor votes to classify new instances.

Analysis

Performance Differences: The comparison revealed nuanced differences in model accuracy when employing L1 versus L2 distances. For instance, L1 distance might show resilience against outliers or perform better in high-dimensional spaces due to its linear nature.

Choosing Between L1 and L2: The choice between L1 and L2 distances should consider the dataset's characteristics and the specific classification task. L1 distance can be particularly effective in scenarios where dimensions are not equally relevant or when the data contains many outliers.



Discussion on CIFAR-10 Dataset

Applying kNN with both L1 and L2 distances to a complex and high-dimensional dataset like CIFAR-10 further emphasizes the limitations of distance-based classifiers in handling intricate patterns and the vast variety of features. In comparison, NNs and CNNs are better equipped to extract hierarchical features and achieve superior classification performance on such datasets.

Conclusion

The exploration of Manhattan (L1) and Euclidean (L2) distance metrics in the context of kNN classification provides valuable insights into how distance calculations impact model performance. While both metrics have their merits, the optimal choice depends on the specific dataset and task requirements. The comparative analysis also reinforces the notion that more sophisticated models like NNs and CNNs offer substantial benefits for complex image classification tasks like those presented by the CIFAR-10 dataset.

Figures and Tables

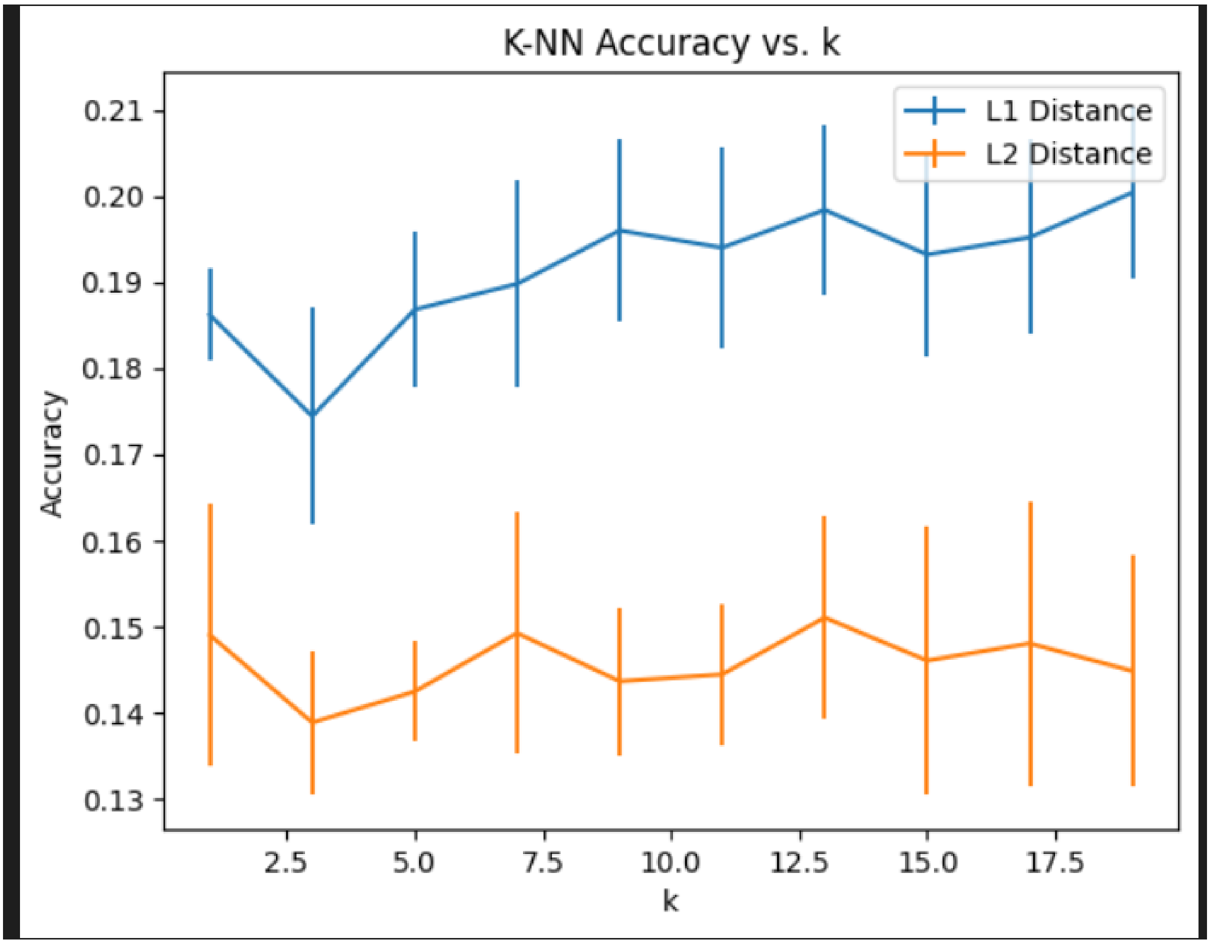


Figure X: Comparison of Average Accuracy using L1 vs. L2 Distance

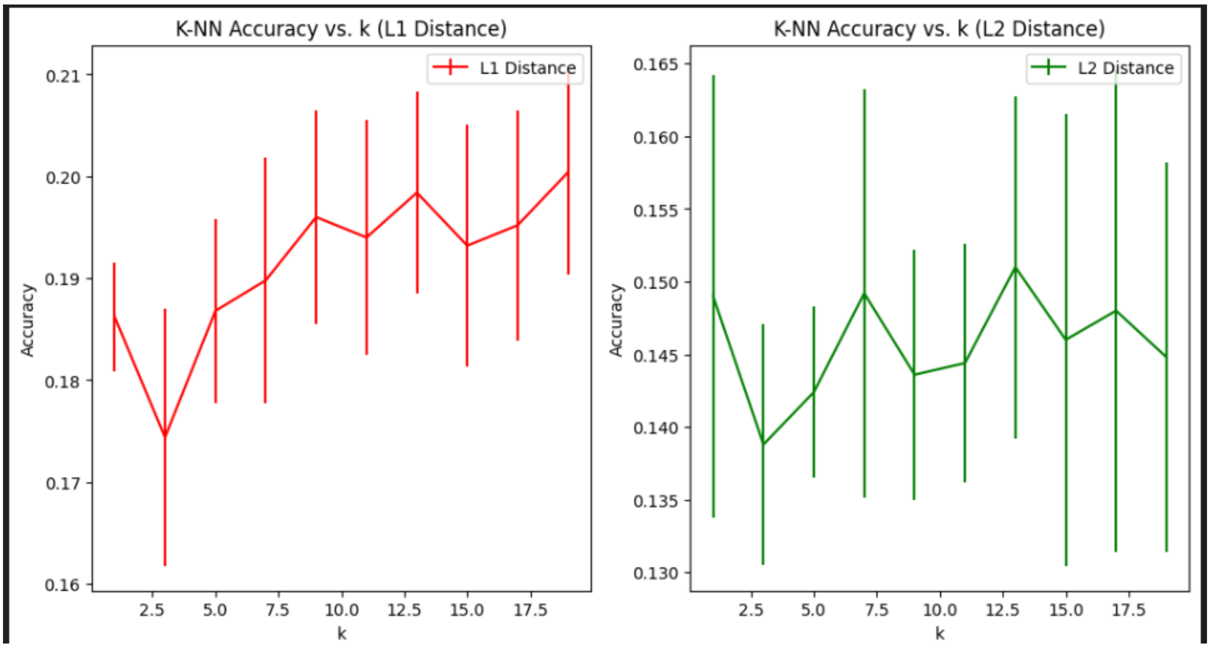


Table X: Summary of Optimal K Values and Corresponding Accuracies for L1 and L2 Distances