Project 1 Report

Bowen Yu (boy014@ucsd.edu)

1 High-Level Description

The goal of this project is to improve the efficiency of 1-NN classification by selecting a representative subset of the training data, referred to as prototypes. Nearest neighbor methods are computationally expensive when the training dataset is large because they require comparing the query point with all training samples. By selecting a subset of prototypes that effectively represent the data distribution, we can significantly reduce the computational cost while maintaining high classification accuracy.

My strategy uses K-means clustering, a widely-used unsupervised learning algorithm, to identify representative prototypes. By clustering the data for each class and selecting cluster centroids as prototypes, we ensure that the selected points are both diverse and representative of the underlying data structure. This approach leverages the inherent structure in the data to minimize information loss during prototype selection.

2 Pseudocode

The algorithmn ensures that the selected prototypes are well-distributed across all classes and that the total number of prototypes does not exceed the desired size M. The use of K-means clustering provides a principled way to summarize the data distribution for each class.

3 Experimental Results

I evaluated the proposed prototype selection method on the MNIST dataset, a widely-used benchmark for image classification. Fig 1 summarizes the classification accuracy of a 1-NN classifier for different prototype sizes. The results compare the proposed method (K-means clustering) to using the full training set:

Algorithm 1 Prototype Selection using K-means Clustering

```
Require: Training data $X$, Labels $Y$, Desired prototype size $M$
Ensure: Prototype set $P$ and corresponding labels
 1: Group data by class: $\{X_1, X_2, \ldots, X_C\}$, where $C$ is the number of classes.
 2: Allocate $M_i = M//C$ prototypes per class.
 3: for each class $i \in \{1, 2, \ldots, C\}$ do
 4:     Apply K-means clustering on $X_i$ with $k = M_i$.
 5:     Select cluster centroids as prototypes.
 6: end for
 7: Combine all prototypes: $P = \bigcup_{i=1}^{C} P_i$.
 8: if $|P| > M$ then
 9:     Trim excess prototypes by selecting the closest points to the centroids.
10: end if
11: return $P$ and their corresponding labels.
```

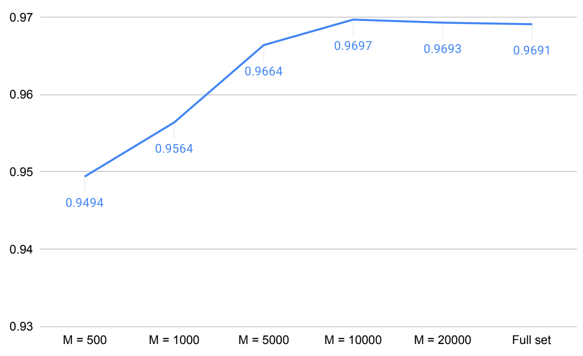

Figure 1: Results of Different Prototypes

3.1 Key Observations

- **Efficiency vs Accuracy:** The results demonstrate that prototype selection reduces the dataset size while maintaining high classification accuracy. For example, using 5000 prototypes (a 91.6% reduction in data size) achieves an accuracy of 0.9664, which is very close to the full dataset's performance of 0.9691.

- **Performance Saturation:** Accuracy saturates when the number of prototypes exceeds 10000, indicating that additional prototypes provide minimal benefits.

- **Insufficient Prototypes:** When the number of prototypes is too small (e.g., 500), accuracy drops significantly to 0.9494, highlighting the need for adequate representation of the data distribution.

4 Critical Evaluation

My method demonstrates several strengths and limitations:

4.1 Strengths

- **Efficiency:** By selecting a small but representative subset of the training data, I achieve a significant reduction in computational cost during classification.

- **Scalability:** The algorithm scales well to large datasets by processing each class independently, which allows for parallelization.

- **Accuracy Preservation:** The method maintains high accuracy, even with substantial reductions in dataset size (e.g., 5000 prototypes).

4.2 Limitations

- **Computational Cost of Clustering:** K-means clustering is computationally expensive, particularly for large datasets. This cost must be considered when applying the method to real-world problems.

- **Fixed Allocation per Class:** The equal allocation of prototypes to each class may not be optimal for imbalanced datasets or classes with varying complexity.

- **Dependence on K-means:** The performance of the method depends on the quality of the clustering, which can be affected by the initialization and the number of clusters.

4.3 Future Directions

To address these limitations, future work could explore:

- Alternative clustering algorithms that are faster or more robust than K-means.

- Dynamic allocation of prototypes based on class distribution or complexity.

- Incorporation of domain-specific knowledge to guide prototype selection.

5 Conclusion

Prototype selection is a powerful technique for reducing the computational cost of nearest neighbor classification. my method, based on K-means clustering, effectively balances efficiency and accuracy. By using a small subset of representative prototypes, I achieve classification performance that is comparable to the full training set while significantly reducing the dataset size. These results demonstrate the potential of prototype selection for practical applications in large-scale machine learning.