Optimizing Nearest Neighbor Classification on MNIST: A Prototype Selection and Clustering Approach

Zhicheng Wang[a]

[a]University of California San Diego, 9500 Gilman Dr, La Jolla, 92093, CA, United States

Abstract

In this project we investigated strategies for optimizing nearest neighbor classification by selecting a representative subset of prototypes from the MNIST dataset. This approach aimed to increase classification speed and accuracy.

1. Implementation Details

1.1. Pseudocode

Algorithm 1 Prototype Selection from Training Set

Input: Training set T, Number of prototypes M
Output: Prototype set P of size M
$P \leftarrow \emptyset$ ▷ Initialize P as an empty set
for each class c in T **do**
  $T_c \leftarrow$ Extract all samples belonging to class c
  Perform clustering on T_c using *ClusteringMethod*
  Determine number of prototypes N_c to select from T_c based on *SamplingMethod*
  for each cluster in T_c **do**
    Select prototypes based on *DistanceMetric*
  end for
  Add selected prototypes to P
end for
return P

1.2. Samplings

In our analysis, we employed two sampling methods to optimize our approach to data representation and analysis: stratified sampling and a variant of adaptive sampling method.

The approach of the adaptive method by calculating a weight for each label, which inversely correlates with its accuracy—labels that are harder to classify (i.e., have lower accuracy) are assigned greater importance in the sampling process. Where our model performs poorly are likely to benefit from a denser sampling of prototypes, thereby providing more information and potentially improving overall model performance.

1.3. Clustering Methods

We also employed two distinct clustering methods: K-means and Gaussian Mixture Models.

K-means Clustering: K-means is known for its simplicity and efficiency, making it an ideal choice for initial explorations of data clustering. Given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, K-means clustering aims to partition the n observations into $k(\leq n)$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares. Mathematically, the objective is to find:

$$\min_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

where μ_i is the mean of points in S_i.

Gaussian Mixture Models: To address the limitations of K-means and capture more complex cluster shapes, we also applied GMM to our dataset. Unlike K-means, GMM accommodates clusters with different sizes and covariance structures, offering a more flexible approach to modeling the data distribution.

The model is defined as follows:

$$p(x|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{1}$$

where:

- x is a data point.
- Θ represents the parameters of the model, including the means μ_k, covariances Σ_k, and mixture coefficients π_k for each component k.
- K is the number of components in the mixture.
- π_k are the mixing coefficients that satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.
- $\mathcal{N}(x|\mu_k, \Sigma_k)$ is the probability density function of the Gaussian distribution for component k, defined as:

$$\mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right) \tag{2}$$

where:

- d is the dimensionality of the data point x.
- $|\Sigma_k|$ is the determinant of the covariance matrix Σ_k.

The goal of GMM clustering is to maximize the likelihood of the data given the model, which can be formulated as:

$$\log p(X|\Theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right) \tag{3}$$

where $X = \{x_1, x_2, \ldots, x_N\}$ is the set of all data points.

The parameters of the model (Θ) are estimated using the Expectation-Maximization (EM) algorithm, which iterates between the following two steps:

Expectation step (E-step): Calculate the posterior probabilities that associate each data point with a given distribution, known as responsibilities.

$$\gamma(z_{ik}) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{4}$$

Maximization step (M-step): Update the parameters μ_k, Σ_k, and π_k based on the responsibilities.

$$\mu_k^{new} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(z_{ik}) x_i \tag{5}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(z_{ik})(x_i - \mu_k^{new})(x_i - \mu_k^{new})^T \tag{6}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{7}$$

where $N_k = \sum_{i=1}^{N} \gamma(z_{ik})$ is the effective number of points assigned to cluster k.

2. Data Descriptive Analysis

After employing t-SNE (t-distributed Stochastic Neighbor Embedding) to visualize the distribution of the MNIST dataset, a observation was made regarding the spatial arrangement of the data points corresponding to the different labels. The visualization suggests that the clusters formed by these labels do not adopt a uniform circular shape which indicate of varying densities and shapes among the clusters. Given this complexity, the insights gained from the t-SNE visualization motivate us to explore alternative clustering strategies, such as Gaussian Mixture Models (GMM), which can more flexibly accommodate the diverse shapes and densities observed.

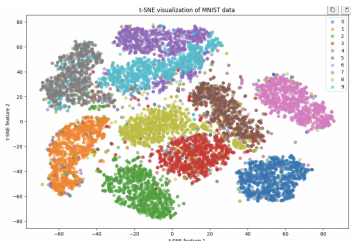

Figure 1: T-SNE Visualization

3. Models and Evaluations

3.1. *Baseline Model*

For the baseline model, we use a 1-NN classifier using random prototype selection, leveraging bootstrapping to assess accuracy variability and compute a confidence interval for a more reliable performance metric.

Table 1: Baseline Model Performance with 1-NN and Random Selection

Time (s)	Accuracy (%)	M	Confidence Interval
0.9	87.77	1000	95.0%: 87.63% - 89.48%
1.7	93.84	5000	95.0%: 93.37% - 93.88%
2.5	95.00	10000	95.0%: 94.65% - 95.20%

3.2. *K-Means Clustering*

3.2.1. *stratified sampling*

Table 2: K-NN Model Performance with closest point prototype selection

Model	Time	Accuracy (%)	M	Metric
1-NN	1m 38s	92.48	1000	L2
1-NN	5m 38s	94.63	5000	L2
1-NN	12m 31s	95.57	10000	L2
1-NN	59s	92.58	1000	L1
1-NN	4m 13s	94.74	5000	L1
1-NN	11m 21s	95.61	10000	L1
3-NN	59s	91.99	1000	L1
3-NN	13m 34s	94.88	5000	L1
3-NN	17m 32s	95.73	10000	L1

3.2.2. *adaptive sampling*

Table 3: K-NN Model Performance with closest point prototype selection

Model	Time	Accuracy (%)	M	Metric
1-NN	1m 38s	92.73	1000	L1
1-NN	6m 43s	94.53	5000	L1
1-NN	13m 8s	95.60	10000	L1
3-NN	1m 8s	92.84	1000	L1
3-NN	8m 59s	94.68	5000	L1
3-NN	14 min 14 s	95.78	10000	L1

Note: Through our experimental analysis, it was observed that the 3-NN classifier generally outperforms classifiers with other values of K in terms of accuracy. Furthermore, the L1 distance metric consistently outperformed the L2 distance metric across various scenarios. This suggests that for the specific dataset and problem context we are working with, a lower number of nearest neighbors and the utilization of the L1 metric for distance calculation are more effective in achieving higher classification accuracy.

3.3. *Gaussian Mixture Model clustering*

3.3.1. *stratified sampling*

Table 4: K-NN Model Performance with closest point prototype selection

Model	Time	Accuracy (%)	M	Metric
1-NN	23m	92.55	1000	L1
1-NN	63m	94.99	5000	L1
1-NN	220m	96.27	10000	L1
3-NN	23m	92.84	1000	L1
3-NN	84m	94.93	5000	L1
3-NN	253m	95.98	10000	L1

3.3.2. *adaptive sampling*

Table 5: K-NN Model Performance with closest point prototype selection

Model	Time	Accuracy (%)	M	Metric
1-NN	20m	92.05	1000	L1
1-NN	60m	94.92	5000	L1
1-NN	213m	95.60	10000	L1
3-NN	20m	92.78	1000	L1
3-NN	60m	95	5000	L1
3-NN	232m	96.32	10000	L1

4. Results

The baseline model's performance was conducted using bootstrapping to assess accuracy variability and compute a confidence interval for a more reliable performance metric. Table 1 demonstrates that the accuracy improves with an increase in the size of the subset(M), reaching up to 95.00% accuracy with 10,000 samples.

In our K-Means clustering experiments, we explored the effects of stratified and adaptive sampling techniques on model performance, using both L1 and L2 distance metrics. The results, as shown in Table 2, indicate that using the L1 distance metric generally yields higher accuracy. Notably, the 1-NN classifier with L1 distance consistently outperforms its L2 counterpart.

GMM clustering was evaluated under stratified and adaptive sampling conditions to understand its behavior in comparison to K-Means clustering. The highest accuracy achieved was 96.32% with 10,000 samples using the 3-NN classifier, indicating the model's effectiveness in handling complex data distributions which is 1.32% better than the baseline model.

5. Discussion

5.1. *Limitations*

There were several limitations encountered due to constraints in time and computational resources, which inevitably led to the premature conclusion of our experimental analysis. Despite these constraints, our study has laid a foundational understanding of the performance of K-Means clustering, Gaussian Mixture Models, and nearest neighbors classifiers across various configurations.

5.2. *Future Work*

For future studies, it is worth considering algorithms that produce probabilities of points belonging to each label. Probabilistic models, such as Bayesian classifiers or logistic regression, could offer a wider perspective on classification tasks. This approach could lead to better decisions when selecting subsets of the training set, thereby enhancing the model's accuracy and reliability.

Another direction involves integrating clustering methods with probabilistic approaches. Such a hybrid method could take advantage of the strength of clustering algorithms in identifying natural groupings within the data while utilizing probabilistic models to assign labels to these clusters based on the likelihood of membership. The only problem I suspect in this combinations is when it comes to balancing the two.