

Foundation of Artificial Intelligence

Assignement 2

Davide Vitagliano

Academic Year 2023/2024

Sommario

[1. Requirements 1](#_Toc153645194)

[2. Introduction 1](#_Toc153645195)

[3. Designing the solution 1](#_Toc153645196)

[3.1. Support Vector Machine 1](#_Toc153645197)

[3.1.1. Formalization 1](#_Toc153645198)

[3.1.2. Code explanation 2](#_Toc153645199)

[3.1.3. Conclusion 2](#_Toc153645200)

[3.2. Random Forest 2](#_Toc153645201)

[3.2.1. Formalization 2](#_Toc153645202)

[3.2.2. Code explanation 2](#_Toc153645203)

[3.2.3. Conclusion 3](#_Toc153645204)

[3.3. K Nearest Neighbor 3](#_Toc153645205)

[3.3.1. Formalization 3](#_Toc153645206)

[3.3.2. Code Explanation 3](#_Toc153645207)

[3.3.3. Conclusion 3](#_Toc153645208)

[3.4. Naive Bayes 3](#_Toc153645209)

[3.4.1. Formalization 3](#_Toc153645210)

[3.4.2. Code Explanation 3](#_Toc153645211)

[3.4.3. Conclusion 3](#_Toc153645212)

[4. Comparison 3](#_Toc153645213)

# Requirements

Write a handwritten digit classifier for the MNIST database. These are composed of 70000 28x28 pixel gray-scale images of handwritten digits divided into 60000 training set and 10000 test set.

Train the following classifiers on the dataset and use 10 fold cross validation to optimize the hyperparameters:

* SVM using linear, polynomial of degree 2, and RBF kernels;
* Random forests;
* Naive Bayes classifier where each pixel is distributed according to a Beta distribution of parameters:

With

* K-NN;

For this assignment we used GPU acceleration to increase the performance of the training process using NVIDIA CUDA Toolkit.

# Introduction

The first thing to do during the training process is applying feature selection for dimensionality reduction. This is a process of selecting a subset of the original features that are most relevant and informative for the classification. The main benefits of feature selection are:

* Reducing computational cost: fewer features means less time and space required for training and testing models;
* Improving model performance: removing irrelevant or redundant features can reduce noise and overfitting, and increase accuracy and generalization.

In our case we use Principal Components Analysis to reduce the dimensionality of the dataset by maximizing the variance of each dimension.

Hyperparameter tuning is the process of finding the optimal set of hyperparameters for a classifier. Hyperparameters cannot be learned from the training data because they aggressively increase the capacity of a model and can push the loss function to an undesired minimum, causing overfitting to the data, as opposed to correctly mapping the richness of the structure in the data.

We perform 10 fold cross validation in the process of hyperparameter tuning for each classifier. Cross validation is a statistical method used to estimate the skill of the models; it consists of 3 phases:

1. Data splitting: the entire training set is divided into 10 equal parts;
2. Model training and validation: the model is trained on 9 of these folds and validated on the remaining one. This phase is repeated 10 times, each with a different fold used for validation;
3. Performance evaluation: the model’s performance is evaluated on each of the 10 folds and the final performance score is the average of the 10 scores.

We use the accuracy score to evaluate the quality of the different classifiers. The accuracy is the fraction of prediction that a model got right. It can also be calculated in terms of positives and negatives predictions:

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

# Designing the solution

## Support Vector Machine

### Formalization

Support vector machines are a type of supervised learning method that can be used for classification, regression and outlier detection. In the context of the MNIST dataset, SVMs can be used to classify the images into their corresponding digit labels.

Some of the main features of SVMs are:

* They use a kernel function to map the input data into a higher dimensional space, where a linear decision boundary can be found;
* They try to maximize the margin between the decision boundary and the nearest training points of each class, which are called support vectors;
* They can handle both linear and non-linear problems by choosing different kernel functions, such as polynomial, radial basis function, or custom kernels;
* They have a regularization parameter C that controls the trade-off between the complexity of the model and the error on the training data.

There are several hyperparameters that we need to tune:

* **C**: This is the regularization parameter, also known as the cost parameter. This tells the SVM optimization how much we want to avoid misclassifying each training example. A smaller value of C creates a wider margin, which may allow more misclassifications. A larger C creates a narrower margin and thus may reduce the number of misclassifications;
* **Kernel**: This specifies the kernel type to be used in the algorithm. In our case it will be 'linear', 'poly' or 'rbf';
* **Degree**: This is the degree of the polynomial kernel function ('poly') and is ignored by all other kernels. It essentially controls the complexity of the model;
* **Gamma**: This defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. It can be seen as the inverse of the radius of influence of samples selected by the model as support vectors;
* **Coef0**: This is the independent term in the kernel function. It is only significant in 'poly' and 'sigmoid'. If gamma is 0, then coef0 controls the bias effect. Otherwise, the larger gamma is, the higher the bias and the lower the variance;
* **Shrinking**: This is a heuristic method used to speed up the training process. It identifies and removes some of the constraints that are not likely to change the final solution, thereby reducing the size of the problem.
* **Probability**: When this hyperparameter is set to True, the SVM enables the calculation of probability estimates. This is done by fitting an additional logistic regression model on the decision function’s scores. This process involves an internal 5 fold cross validation and can slow down the training time.

### Code explanation

### Conclusion

## Random Forest

### Formalization

### Code explanation

### Conclusion

## K Nearest Neighbor

### Formalization

### Code Explanation

### Conclusion

## Naive Bayes

### Formalization

### Code Explanation

### Conclusion

# Comparison