Data Challenge - Kernel Methods for Machine Learning

Carlos Cuevas Villarmín Javier Alejandro Lopetegui González
Université Paris-Saclay
M1-AI

November 28, 2024

Abstract

Multi-class classification is a classic machine learning problem, and in recent years, significant advancements have been noted in the literature. In this challenge, we experiment with several kernel techniques to classify images. This brief report seeks to provide an overview of our methods, with a particular emphasis on feature extraction and classification.

1 Introduction

The data we work with consist on: 5000 classified images for training and 2000 images as test set. Each image is represented by a vector of size 3072 where the first 1024 values represent pixel intensities on the red channel, then the next 1024 represent the green channel, and the last 1024 entries, the blue channel and as the images are preprocessed the values are between -1 and 1. There is a leaderboard accessible to the public, and the score is determined using around half of the test results. The remaining 50% will determine the ultimate outcomes.The goal of the data challenge is to learn how to implement machine learning algorithms, gain understanding about them and adapt them to structural data.

The structure of the report is the following: Section 2 talks about the preprocessing part of the work, how we have increase the data set, how we detect local features of the images and what is the technique we have implemented to transform them to global features. On the other hand, Section 3 introduces the models and the kernels considered for the classification task. Finally, in 4 we will explain our best approaches and briefly talk about our workflow.

2 Preprocessing

In this section we will introduce the approaches used for data augmentation, feature extraction and classification.

2.1 Data augmentation techniques

In order to increase the data for training the models and with the aim of making the pedictor invariant to some linear transformations we increase the training set by adding to the set rotated and flipped versions of the images.

2.2 Feature extraction

Having generated a greater set to train the models we have selected different techniques for feature extraction. The chosen classical algorithms in computer vision to detect and describe local features are:

- **Histograms of Oriented Gradientes (HOG).** It calculates gradients in an image and divides it into cells. Within each cell, gradient orientations are binned into histograms. These histograms are then normalized to account for changes in illumination and contrast. Finally, the normalized histograms are concatenated to form a feature vector representing the distribution of gradient orientations in the image,

- **Scale-Invariant Feature Transform (SIFT).** It operates by identifying keypoints through scale-space extrema detection, finding significant points regardless of scale changes. Keypoint localization refines their positions based on image feature characteristics like contrast and curvature. Each keypoint is assigned a dominant orientation to achieve rotation invariance. SIFT generates distinctive feature descriptors by considering local gradient orientations and magnitudes around keypoints. These descriptors are designed to be invariant to changes in scale, rotation, and illumination. Finally, keypoints from different images are matched based on the similarity of their descriptors,

- **Descriptive And Illumination-invariant Spatial sYstem (DAISY).** The descriptor computes histograms of gradient orientations within circular regions around each pixel in a local image patch, offering multi-scale adaptation, rotation invariance, and illumination robustness. These histograms are organized in a grid pattern and concatenated into a feature vector.

2.3 Global feature extraction

We use the technique known as **bag of words**. It consists on grouping local feature descriptors using the K-means technique. Every cluster is regarded as a bag, and every word represents a local feature description. Then, for each image we compute its bow representation.

3 Classifiers and kernels

Once the data has been preprocessed, we classify them using a classifier. We have consider kernel SVM one versus one classifier, kernel SVM one versus all classifier and kernel ridge classifier. Kernel ridge classifier has been defined by adapting the well-known kernel ridge regression classifier to output 1 in case the predicted label is the target one and -1 otherwise.

On the other hand, the implemented kernels are:

- **Linear Kernel.** $K(x,y) = x^T y$.
- **Gaussian Kernel.** $K(x,y) = exp(\frac{-||x-y||_2^2}{2\sigma^2})$.
- **Polynomial Kernel.** $K(x,y) = (1 + x^T y)^p$.
- **Laplacian Gaussian Kernel.** $K(x,y) = exp(\frac{-\sum_i |x_i - y_i|}{\sigma^2})$.

4 Experiments and Results

We have coded in python and the implementation is publicly available in GitHub.

Data visualization. We have transformed data in order to previsualize the images and try to identify to which label corresponds each number.

Experiments. After several trials, there are three combinations that have reported good values. First, we validate the quality of the trials with a portion of the training data. Once the reported values on this subset of the training were higher than our last submission to the challenge we redo the experiment considering the full training set for training.

- HOG + SIFT + Kernel Ridge Classifier. After finetuning some hyperparamenters the accuracy achieved in the public leaderboard is 0.644.
- HOG + Kernel Ridge Classifier. We have achieved an accuracy of 0.629 in the public leaderboard of the challenge.
- HOG + Kernel SVM One-vs-One. In this experiment the score in the public leaderboard was 0.564.

In all the scenarios data was fully augmented. It means that all the images were also flipped and rotated and not just a few of them, i.e., the ratio of augmentation was set to 1.

Hyperparameters fine-tuning.

For the approach we used in the best submission we modified the parameters for sift features extractor in order to be able to detect more key points in images. We changed the value of the threshold to discard low contrast point on images (c_dog) and parameters related to the blur level considered for the images (sigma_min and sigma_in). By doing that we reached a more significant number of key points per image what directly impact on the quality of the representation. Performing a more detailed hyperparameters fine tuning should enhance the results obtained.

5 Conclusions and further work

In this challenge, we have considered some feature extraction techniques mainly SIFT, DAISY and HOG. Having dealt with the problem of the nature of the data and preprocessed the samples we have tried several models for classification with kernel methods such as kernel SVM and Kernel Ridge Classifier. We realize that after data augmentation and local feature extraction the computational cost could be very high. Some experiments could not be carried out correctly due to lack of time and they have been left out of this challenge. However, we can conclude that Kernel Ridge Classifier with data preprocessing is a good combination to obtain an interesting model and probably exploring more complex feature extraction techniques and fine-tuning more precisely the hyperparameters the trained model would achieve a high accuracy.

References

[1] Tomasi, C. (2012). Histograms of oriented gradients. Computer Vision Sampler, 1-6.

[2] Markel, J. (1972). The SIFT algorithm for fundamental frequency estimation. IEEE Transactions on Audio and Electroacoustics, 20(5), 367-377.

[3] Tola, E., Lepetit, V., & Fua, P. (2009). Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE transactions on pattern analysis and machine intelligence, 32(5), 815-830.

[4] Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019, June). An overview of bag of words; importance, implementation, applications, and challenges. In 2019 international engineering conference (IEC) (pp. 200-204). IEEE.