Data Challenge: Image Classification - Kernel Methods Project Report - 10/03/2016

Achari Berrada Youssef
MVA, ENS Paris Saclay / Ecole Polytechnique
youssef.achari-berrada@polytechnique.edu

Keywords : Histogram of Oriented Gradients, Support Vector Machines, One Versus All Classification

Abstract

In this report, we present the method used as well as the results obtained in the Data Challenge of the course `Machine Learning with Kernel Methods` *in which we have to classify set of images belonging to 10 classes. The goal of the challenge is to code kernel methods without using external libraries like Scikit-learn. However, we are allowed to use general optimization libraries for solving linear and quadratic programs like* cvxopt*. We joined to this report a working code with a file* start.py *in order to reproduce our results and view the implementation used in the project. My pseudoname in the datachallenge is* **yab***. And my final rank in the private leaderboard is 26 with* 0.58400 *of accuracy. I was ranked 16 in the public leaderboard with* 0.61800*, and I made 10 submissions overall.*

1. Introduction

The challenge consists of classifying images representing 10 classes of images, each image is in the size of 32×32 pixels in colors. The challenge provides us with a simple data in .csv files. Xtr.csv for the set of training images of size 5000×3072 and Xte.csv for the set of test images of size 2000×3072. Therefore, each image is represented by an array of lenght 3072. The first 1024 values represent the pixels intensities in the red channel, the next 1024 are for the green channel and the last 1024 represent the blue channel. The challenge provide us also with the labels (between 0 and 9) of the training images in the file Ytr.csv. The 10 classes in the order of their labels are: 0 for `plane`, 1 for `car`, 2 for `bird`, 3 for `cat`, 4 for `deer`, 5 for `dog`, 6 for `frog`, 7 for `horse`, 8 for `boat` and 9 for `lorry`. The goal of the challenge is to predict the labels of the test set of images in Yte.csv file.

2. Approach to the data

First, we perform a **sanity check** on the training set to see if the data is well balanced between the classes of images and we conclude that each class of image is represented by 500 images in the training set. This is important if we want to have good predictions in the test set for our classification task.
Image Visualization: In order to identify the classes and check if the prediction seems to be correct, we create a method in order to visualize the image. To do so, as the data is preprocessed, we create an "inverse preprocessing" function to scale the image to its RGB values in each pixel from 0 to 255. We do the inverse preprocessing for each RGB channel alone and we follow the method:

1. Extract the minimum: $X \leftarrow X - X.min$
2. Divide by the standard deviation: $X \leftarrow \frac{X}{\sqrt{X.var}}$ (This step is important in order to have a good distribution of the intensities in each channel).
3. Normalize between 0 and 1: $X \leftarrow \frac{X}{X.max - X.min}$
4. Rescale from 0 to 255: $X \leftarrow floor(X * 254) + 1$
5. Reshape to the size $32 \times 32 \times 3 \quad (= 3072)$.

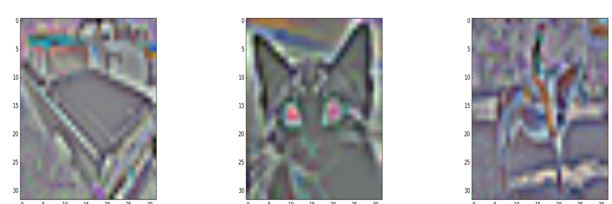

Figure 1: Set of 3 images from the training set for 3 different classes (from left to right: Car, Cat and Horse).

We notice that the images are recognizable. However it is not always the case !

3. Histograms of Oriented Gradients

HOG are Histograms of Oriented Gradients. They were introduced by Dalal and Triggs in the paper [1] for the task of human classification. We wanted to follow this method in order to extract features and perform our classification task based on HOG features of each image. Here are the steps :

- First we divide the image in 16 blocks of 4×4 pixels without overlapping. So we have in total 8×8 blocks.
- For each image, we compute the centered gradient magnitude in each of the RGB channel. So we convolve the matrix with a Sobel Operator, see [3].
- Then, for each channel, and for each block, we add the magnitude gradient of each pixel to the correspondant bins in the Histogram of Oriented gradient of the block.
- As explained in [1] when dealing with RGB images, for each block, we take the HOG of the channel that maximizes the magnitude of the gradient of the block. In [1], they take the maximum w.r.t each pixel, in our method, we take the maximum w.r.t each block.
- Finally, we take output for each image the Histogram of Oriented Gradient in a vector of dimension $8 \times 8 \times 9$ which is equal to 576, dimention of the feature space.

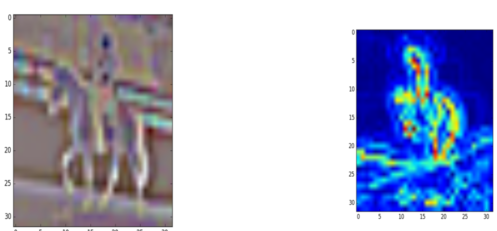

Figure 2: (Left)Image of a Horse from training set and (Right)its Maximum Gradient Magnitude w.r.t each block.

4. Support Vector Machine

We note that we have to do multiclass classification. To do so, we select each class k for $k \in [0, 9]$, and we perform **one vs all** classification by predicting for each image what is the probability of belonging to the class k. We do this task $K = 10$ times, and in final, we output a vector of dimention K telling what is the probability of the image to belong to the class $k \in [0, K]$. Therefore, the prediction will be the **argmax** of this vector for each image.

In our method, we do each one vs all classification task with kernel support vector machine.

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L_{\text{hinge}}(f(x_i), y_i) + \frac{1}{2n\lambda} \|f\|_{\mathcal{H}}^2$$

Where $L_{\text{hinge}}(u) = \begin{cases} 0 & \text{if} \quad u \geq 1 \\ 1 - u & \text{otherwise} \end{cases}$
We choose either the linear kernel or the Gaussian (RBF[1]) Kernel.

$$K(x, x') = \exp(-\frac{\|x - x'\|^2}{2\sigma^2})$$

With the representer theorem, the solution satisfies :

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i K(x_i, x)$$

And due to KKT conditions, the solution is **sparse** on α. Thus, classification of news points only involves kernel evaluations with few support vectors.

Solving linear quadratic program We use the `cvxopt` library to solve the quadratic program of the dual form of the SVM. See [2], page 152.

5. Conclusion

During this challenge, I learned how to compute Histograms of Oriented Gradients. I also learned how to implement support vector method without using external machile learning libraries. In the scientific literature, other kernels can be used in order to do image classification. One that is used frequently is the Histogram Intersection Kernel, see[5].
I Finally thank professors Julien Mairal and Jean-Philippe Vert for the lectures and the quality of the slides they gave us during this course. It was very instructive and interesting as Kernel methods are used in plainty of areas in Machine learning nowadays.

References:

[1] Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection. CVPR 2005.
[2] Julien Mairal, Jean-Philippe Vert. Slides of the course Machine learning with kernel methods, Spring 2017.
[3] Sobel operator, Wikipedia, English.
[4] Histogramme de gradient orienté, Wikipedia, Français.
[5] Subhransu Maji, Alexander C. Berg and Jitendra Malik, Classification using Intersection Kernel Support Vector Machines is Efficient, CVPR2008.

[1] Radial Basis Function