Kernel Methods for Machine Learning

Kaggle Challenge

NEGREL Hugo
ENSTA Paris
hugo.negrel@ensta-paris.fr

TOCQUEC Louis
ENS Paris-Saclay
louis.tocquec@ens-paris-saclay.fr

1 ABSTRACT

This work is part of the course "Kernel methods for machine learning" of the MSc. MVA. The aim goal of this work is to design an efficient model for images classification that relies on Kernel methods. Our implementation is available by clicking here : code.

2 DATASET AND PRE-PROCESSING

The training dataset is composed of 5000 images of 32x32 pixels represented as a vector of size 3072 such that the first 1024 values represent pixel intensities on the red channel, and in the same way the next columns represent the green channel and then the blue channel. Each image is associated with a label from 0 to 9 such that the total number of classes is 10.
Before applying any algorithm, we normalize the data. For each image we subtract its smallest value vector and divide everything by the maximum difference in pixel intensity values. A visualization of the image of label 7 is given here :

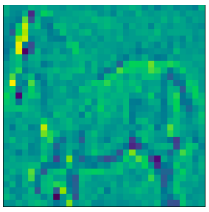

Figure 1 : Image of label 7

3 KERNELS

Kernel methods and their performances are of course influenced by the choice of the kernel. Thus, we need to explore different possibilities in order to design an efficient method. The first kernel we tried was the polynomial kernel defined as :

$$K(x, y) = (x^t y)^d$$

where $d \in \mathbb{N}_+^*$ is the degree which is, in practice, a hyper-parameter to choose (when d=1 it corresponds to the linear kernel).
We also considered the Gaussian kernel which is able to capture non-linearity in the data :

$$K(x, y) = e^{-\frac{||x-y||_2^2}{2\sigma^2}}$$

with $\sigma^2 > 0$ the variance which is, in practice, a hyper-parameter to choose.
At the end, the kernel that achieves the best accuracy was the Gaussian kernel.

4 THE MODEL : SVM

The most famous model for image classification which relies on kernel methods is probably the Support Vector Machines (SVM) [7]. Moreover, it is often the state-of-the-art to perform this task. Thus, we quickly decided to focuses our attention on this model. Mathematically, in binary classification and given training data $(x_i)_{i=1}^N \in \mathbb{R}^{Nxd}$ and their respective labels $(y_i)_{i=1}^N \in \{-1, 1\}^N$, the aim goal of SVM is to solve :

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{N} (1 - y_i f(x_i))_+ \; + \; C||f||_{\mathcal{H}}$$

It is well known, that applying the useful representer theorem and adding the constraints $((1 - y_i[K\alpha]_i)_+ \leq \zeta_i)$ in order to make the objective smooth, the minimization task is equivalent to :

$$\min_{\zeta \in \mathbb{R}^N, \alpha \in \mathbb{R}^N} \sum_{i=1}^{N} \zeta_i \; + \; C\alpha^t K\alpha \;\; \text{s.t} : (1 - y_i[K\alpha]_i)_+ \leq \zeta_i$$

Kernel methods for ML, 2024, Paris

.

We finally compute the equivalent dual problem since strong duality holds for quadratic program :

$$\min_{\alpha \in \mathbb{R}^N} \quad \alpha^t K \alpha - 2\alpha^t y$$
$$\text{s.t} : 0 \leq y_i \alpha_i \leq C \quad \forall i \in [\![1, N]\!]$$

Thus, we end up with an optimization task in finite dimension which is actually quite impressive. Moreover, what make SVM such efficient in practice is the sparsity of the optimal α^*.
In our case, the classification was not binary and so we need to find a way to bypass this issue. One way to do it is to use *One-vs-All* approach. It simply consists to perform for each class a separate binary classification problem, trained such that the samples of the class we are looking for are labeled as 1 and samples of all other classes as -1. Finally, we get the decision function value from each SVM model, and select the class with the highest confidence as the predicted class. Note that we need to use the same numbers of binary SVM as we have classes, i.e 10 classifiers in our case.
Another way is to perform multi-classification task is to use *One-vs-One* approach but it requires 45 binary SVM.

5 FEATURE ENHANCEMENT

Since separating the different raw images can be challenging, we decide to implement methods to reduce the number of features and capture as much as possible the information contains in it.

5.1 Kernel PCA

The first thing that comes in mind is of course the kernel PCA which allows to project data into a lower-dimensional space while preserving the essential structure or relationships within the data.
This algorithm, widely used in data science, was unfortunately not efficient on our task and only improves of 4% the previous pipeline.

5.2 HOG features

After consulting the literature ([3], [6]) on SVM for image classification, we find that another strategy is to use Histogram of Oriented Gradients (HOG) methods [4]. This technique counts occurrences of gradient orientation in the localized portion of an image and is widely used in object detection. It consists in the following steps :

(1) Gradient computation : we compute the gradient of the image using the central differences along each axis g_x and g_y and then we compute the magnitude (1) and the orientation (2) given by :
$$\text{mag}(g_x, g_y) = ||(g_x, g_y)||_2 \quad (1)$$
$$\text{ori}(g_x, g_y) = \text{arctan2}(g_x, g_y) \quad (2)$$
(2) Histogram computation : we divide the image in small part over which we compute a histogram where bins represented orientation angles and bins values represent the magnitude of gradients.

This algorithm captures how the value of pixels change in the image and allows to detect direction in which the shape evolves. This new strategy has significantly increased the performance of our model, achieving 28% of accuracy on the Kaggle page by training the model only on 500 images.

5.3 Color Histogram

Even with this evolution in the accuracy of the model, we were still close to the benchmark performances. Since HOG focuses more on the classification, the natural idea is to implement a strategy that capture now the evolution of colors in an image. That's why we implement the color Histogram algorithm. Combining these two methods is a widely democratized practice that has proven itself on numerous occasions [5]. It allows our model to increase to 34% on the Kaggle competition.

5.4 Histogram of intensities

We also implement a function that computes row-wise histograms for a gray scale image, which captures information about the intensity distribution along each row. This approach is not particularly widespread but proved to be effective in our problem by increasing the accuracy up to 38% on the Kaggle page when being associated with HOG.

5.5 Kernel Fisher Discriminant Analysis

Kernel Fisher Discriminant Analysis (KFDA) is a supervised machine learning technique similar to PCA for classification. Initially coined for linear separation between gaussian clusters, it was naturally extended

to non-linear separation with kernel methods, see [2], whether one considers vanilla classification or multi-class. Let c the number of class considered. We implemented multi-class KFDA as follows. If we note $S_W^\phi = \sum_{i=1}^{c} \sum_{n=1}^{l_i} (\phi(x_n^i) - m_i^\phi)(\phi(x_n^i) - m_i^\phi)^\top$ and $S_B^\phi = \sum_{i=1}^{c} l_i (m_i^\phi - m^\phi)(m_i^\phi - m^\phi)^\top$, where l_i is the number of elements in class i and $m_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \phi(x_j^i)$. We now aim at maximising : $\dfrac{W^\top S_B^\phi W}{W^\top S_W^\phi W}$ meaning one wants which turns out to be equivalent to maximizing thanks to the kernel trick : $\dfrac{A^\top M A}{A^\top N A}$, where $M = \sum_{j=1}^{c} l_j (M_j - M_\star)(M_j - M_\star)^\top$ and $N = \sum_{j=1}^{c} K_j (\mathbf{I} - \frac{1}{l_j}\mathbf{1}) K_j^\top$. To be more precise, one has $(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$ and $(M_\star)_j = \frac{1}{l} \sum_{k=1}^{l} k(x_j, x_k)$. Solving this maximisation problem is equivalent to computing the eigenvectors related to the $(c-1)$ largest eigenvalues of $N^{-1}M$. The main idea behind maximizing this objective is to identify the direction(s) along which the difference between means as large as possible while making the variance within class as small as possible [8]. The projection of a test dataset X_{test} is then done naturally by computing $A^{*\top} k(X_{\text{tr}, X_{\text{test}}})$. Eventually, this method was not employed for our final classification, as we preferred to concentrate on its competitor. Similarly to a PCA, one could then plot the projected data, and proceed to a SVM.

6 RESULTS

Our final model achieves 0.385 of accuracy on the Kaggle page and combine HOG features with a cell size of 8x8 pixels and 9 orientation bins with histogram intensities features with 12 bins per row. The kernel was the Gaussian one with $\sigma = 1$. We train our model on a small number of samples (1500) due to time-consuming training process. Note that these methods were trained using transformed samples in grayscale image format. Implementation : https://github.com/L-Tocquec/Kernel-Methods-for-ML-Kaggle-challenge/blob/main/SVM.py

RÉFÉRENCES

[1] Michael Arbel. 2024. Copy of Data Challenge - Kernel methods (2023-2... https://kaggle.com/competitions/data-challenge-kernel-methods-2023-2024-extension

[2] Mark Crowley Benyamin Ghojogh, Fakhri Karray. 2019. Fisher and Kernel Fisher Discriminant Analysis : Tutorial. (2019).

[3] Harihara Santosh Dadi and GK Mohan Pillutla. 2016. Improved face recognition rate using HOG features and SVM classifier. *IOSR Journal of Electronics and Communication Engineering* 11, 04 (2016), 34–44.

[4] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Vol. 1. Ieee, 886–893.

[5] Lujun Jin, Jian Cheng, and Hu Huang. 2010. Human tracking in the complicated background by particle filter using color-histogram and hog. In *2010 International Symposium on Intelligent Signal Processing and Communication Systems*. IEEE, 1–4.

[6] Bambang Sugiarto, Esa Prakasa, Riyo Wardoyo, Ratih Damayanti, Listya Mustika Dewi, Hilman F Pardede, Yan Rianto, et al. 2017. Wood identification based on histogram of oriented gradient (HOG) feature and support vector machine (SVM) classifier. In *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 337–341.

[7] V Vapnik et al. [n. d.]. The nature of statistical learning theory 1995 New York Springer 10.1007. *Google Scholar Google Scholar Digital Library Digital Library* ([n. d.]).

[8] Yuqian Duan Xijun Liang, Shenyu Du. 2023. Kernel-based Algorithms for Image Classification : A Review. *Research Square* (2023).