Image classification with kernel methods
Team name: HarderBetterFasterKernel

Bianca Marin Moreno
bianca.marin-moreno@polytechnique.edu

Leonardo Martins Bianco
leonardo.martins_bianco@ens-paris-saclay.fr

Objectives. In this report, we detail our developments concerning the data challenge for the course on kernel methods at MVA. The objective was to classify images from a subset of the CIFAR-10 dataset using exclusively kernel methods and our own code, i.e., without the aid of libraries such as `libsvm` or `sklearn`. Our code is available at `https://github.com/leobianco/kernel_methods`.

1 Methods used

Approach. Our approach was to start from classical and simple models, and then progressively increase their accuracy by testing different kernels on them, as well as by processing the data.

1.1 First method: kernel ridge regression.

Our first implementation is a kernel ridge regression. In order to deal with multiple classes we transform the label vectors in one-hot encodings and use the approach introduced in [7]. Without any data processing, for $\lambda = 1$ and using a Gaussian Kernel with $\sigma^2 = 1.8$ (parameters we obtained doing a grid search), this method achieves a 0.24 score in both public and private leaderboards.

1.2 Second method: SVM on extracted features.

The following implementations are based on standard approaches when using kernel methods for computer vision [4]. It consists in using kernel classifiers (most typically support vector machines) on feature representations of the data. More precisely, one first extracts features from the images (characteristics such as edges, textures, colors, etc), then one represents the data as a function of the presence of these features, and finally one classifies the vectors resulting from this representation.

1.2.1 Feature extraction

Analysis of data. In order to choose a method for feature extraction, we take into account the characteristics observed in the data. In our dataset, all images have gone through a color preprocessing, meaning that all had approximately the same tone. We also noticed that given the resolution of 32×32 of the images, its subject is in most cases located in the center, i.e., there are no biases of relative positioning. The resolution being relatively low, we noticed that there is no sense in taking texture into account for classification. We thus concluded that the most important features for our data were the edges present in the images.

Histogram of oriented gradients (HOG). Since we are taking only edges into account as features, the HOG descriptor seems appropriate for the task of feature extraction. For a given image, it consists in applying a convolution with specific filters resulting in the spatial derivatives of the image, which highlights its edges, then storing weighted histograms of the orientations of these gradients.

Other feature extractors. We point out that there is another common line of feature extraction called the *bag of visual words*. It consists in first extracting recurring patches of the images by using a SIFT descriptor, clustering these to find a representer of those patches (these are the features), then considering histograms of these features. We did not pursue this approach for two reasons. First, we believe that the relative positioning on the images is not important, thus we see less interest in the SIFT descriptor. Then, given the low resolution of the images, the patches extracted and the words derived from them would probably not be as insightful as those extracted in higher resolution images (e.g., "bicycle rim", or "nose") since less structure fits in low resolutions.

1.2.2 Classification

Multiclass SVMs. There are ten class labels in the given dataset. Classically, SVMs are binary classifiers, but there are some adaptations to perform multiclass classification with them. Namely, one can directly switch to a multiclass loss function [1], or implement several binary SVMs between different classes and use a voting scheme to decide the predicted label. The latter approach splits in two possible implementations: either one fits one SVM separating each class from all the others (resulting in $N_{\text{classes}} = 10$ models), an approach called "one vs. all", or one fits one SVM distinguishing each pair of classes (resulting in $N_{\text{classes}}(N_{\text{classes}} - 1)/2 = 45$ models), an approach called "one vs. one". We decided to stick to the "one vs. all" method for its simplicity and comparable performance [2].

Model selection and training. Model selection in this case consisted in choosing a kernel and its corresponding parameters, as well as the hyperparameters for the multiple SVMs (regularization parameters), and parameters associated to HOG's feature extraction procedure. We brute-forced our choices: for each kernel selected, all other parameters were chosen by a standard grid search with five-fold cross-validation on some test range of parameters. The training part consists in determining the parameters of the SVM classifiers, i.e., in solving constrained quadratic programming problems. Only for this optimization part we were allowed to use the `cvxopt` library. For some models we also perform data augmentation on the data before trainning, that consists on horizontal reflections on the images.

Kernels. We first implemented the standard kernels studied in class such as the Linear kernel, the Gaussian kernel, and the Polynomial kernel. As we compare histograms obtained by HOG, we implemented the Chi squared kernel, a kernel built for this type of task, and that in the literature seems to be largely used when dealing with computer vision with kernel methods.

1.2.3 Results

Our best submission on the public leaderbord uses the Multiclass SVM method with $C = 10$, HoG extraction, the Chi squared kernel, and data augmentation, giving a score of 0.581. The data augmentation part increased the score from 0.554 to 0.581. However, the best score in the private leaderboard (0.588) is obtained using the same SVM method with $C = 10$, but with a Gaussian Kernel for $\sigma^2 = 1$, and no data augmentation. We did not try the Gaussian kernel with data augmentation, thus we can not conclude wether it would perform better or worse.

2 Other methods of interest

In our researches we found more recent kernel methods that according to the literature would perform better in an image classification task then the standard SVM "one vs. all". Although we did not code them as time was short and we gave more importance to standard methods, we describe briefly some of these methods and we acknowledge that they might have given better scores in the challenge.

Pyramid match kernel. The pyramid match kernel was introduced in [3] with the main idea of bulding a kernel that is able to match unordered features sets into histograms, and then computing a weighted histogram intersection in the space. Work done in [5] shows that using pyramid match kernels could increase the performance of image classification tasks.

Kernel convolutional networks. Kernel convolutional networks were introduced in [6] with the goal of making a link between convolutinal neural networks and kernel methods, teaching CNNs to be invariant.

References

[1] Koby Crammer and Yoram Singer. "On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines". In: *J. Mach. Learn. Res.* 2 (Mar. 2002), pp. 265–292. ISSN: 1532-4435.

[2] Kai-Bo Duan and S. Sathiya Keerthi. "Which Is the Best Multiclass SVM Method? An Empirical Study". In: *Multiple Classifier Systems*. Ed. by Nikunj C. Oza et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 278–285. ISBN: 978-3-540-31578-0.

[3] K. Grauman and T. Darrell. "The pyramid match kernel: discriminative classification with sets of image features". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. 2005, 1458–1465 Vol. 2. DOI: `10.1109/ICCV.2005.239`.

[4] Christoph H. Lampert. *Kernel Methods in Computer Vision*. Hanover, MA, USA: Now Publishers Inc., 2009. ISBN: 1601982682.

[5] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, pp. 2169–2178. DOI: `10.1109/CVPR.2006.68`.

[6] Julien Mairal et al. "Convolutional Kernel Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: `https://proceedings.neurips.cc/paper/2014/file/81ca0262c82e712e50c580c032d99b60-Paper.pdf`.

[7] Katuwal Rakesh and P.N. Suganthan. "An Ensemble of Kernel Ridge Regression for Multiclass Classification". In: *Procedia Computer Science* 108 (2017). International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland, pp. 375–383. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2017.05.109`. URL: `https://www.sciencedirect.com/science/article/pii/S1877050917306622`.