Image Classification – Kernel Methods for Machine Learning Course

Juliette Achddou [*]
achddou@enst.fr

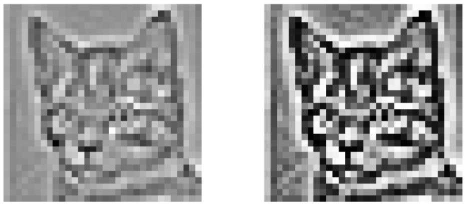
Salma El Alaoui Talibi[*]
salma.el-alaoui-talibi@polytechnique.edu

Camille Jandot [*]
jandot@enst.fr

March 12, 2017

1 Introduction

In this challenge, we tackled the task of classifying 32×32 pixels images, belonging to 10 different classes using kernel methods. We first tried to visualize the images. To do so, we performed histogram equalization, as the contrast was quite poor in the originals images (Figure 1).



(a) Before equalization (b) After equalization

Figure 1: Effect of histogram equalization

The training set was composed of $5,000$ images ($2,000$ images in the test set), where each of the 10 classes was equally represented. Each image is described by 3 channels of 32×32 pixels, which is $3,072$ features: with as few as $5,000$ training examples, using a SVM classifier on the raw data was thus rather unsuccessful.

Hence, we chose to use the histogram of oriented gradients (HOG) image descriptors [DT]. Then, we reduced the dimension using kernel PCA, before applying a kernel SVM classifier.

2 Building Features with HOG

The histogram of oriented gradients is an image descriptor that aggregates the gradients' orientations, weighted by the magnitude of the gradients in local areas of the image. We tested HOG with different sets of configurations (signed vs. unsigned – the orientation is signed, ie. can take values up to 360° – vs. unsigned (180°), on an RGB image (equalized or not) vs. a black and white image, with cells of size 4 pixels vs. cells of size 8). What seems to yield the best results with a Kernel PCA and a SVM classifier (both with Gaussian kernels) was the (signed, equalized RGB, cells of size 8) configuration. When trying different kernels for the PCA and the SVM classifier, we kept the previous configuration and did not validate on the other possible configurations.

3 Dimension Reduction (Kernel PCA)

Histograms of gradients in the latter configuration provide $4 \times 4 \times 18 \times 3 = 864$ features per image (4×4 cells, 18 orientation bins, 3 channels). We decided to reduce the dimension using Kernel PCA. We tried linear ($K(x,y) = \langle x, y \rangle$), Gaussian ($K(x,y) = \exp(-\gamma \|x-y\|^2)$), and Laplacian ($K(x,y) = \exp(-\sigma^{-1}\|x-y\|)$) kernels. We selected the two latter kernels because of the fact that they are translation invariant.

We first computed the first $n_{\text{components}}$ eigenvectors and eigenvalues (u_i, λ_i) of the centered Gram matrix K on the training set (ordered by decreasing eigenvalues). The projections are given by $K \cdot \alpha / \sqrt{\lambda}$, where '$/$' denotes the term by term division.

4 Kernel SVM

The classical SVM formulation intends to solve a binary classification problem and its dual formulation is:

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i y_i - \sum_{i=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$$

subject to $0 \leq y_i \alpha_i \leq \frac{1}{2\lambda n}\ \forall i \in 1, \ldots, n$.
We implemented two variations of SVM for multiclass classification: Crammer-Singer ([CS]) and the famous One-vs-One (OvO). As in the dimension reduction step, we chose to try linear, Gaussian and Laplacian kernels.

4.1 Crammer-Singer

Multiclass SVMs classify a vector $x \in \mathbb{R}^d$ into one of k classes using the rule:

$$\hat{y} = \text{argmax}_{m \in [k]} w_m^{\intercal} x \tag{1}$$

Where $w_m \in \mathbb{R}^d$ can be thought of as a prototype representing the m^{th} class and the inner product $w_m^{\intercal} x$ as the

[*] Master Data Sciences, Team Sacaju

score of the m^{th} class with respect to x. Thus, equation 1 chooses the class with the highest score.

The Crammer-Singer formulation for multiclass SVM estimates $w_1 \dots w_k$ by solving the following optimization problem:

$$\text{minimize}_{w_1 \dots w_k} \frac{1}{2} \sum_{m=1}^{k} \|w_m\|^2 \qquad (2)$$
$$+C \sum_{i=1}^{n} \left[1 + \max_{m \neq y_i} w_m^\intercal x_i - w_{y_i}^\intercal x_i \right]$$

We can see that equation 2 means that for each training sample, we suffer no loss if the score of the correct class is larger than the score of the closest class by at least 1.

In order to solve this problem, the dual of equation 2 is minimized using a dual decomposition method, which consists in updating at every iteration a small subset of dual variables, keeping all others fixed. We follow the approach presented in [BFU], in which the restricted problem is reduced to the Euclidean projection onto the positive simplex.

4.2 One-vs-One SVM

Multiclass SVM can be tackled training $\frac{N \cdot (N-1)}{2}$ SVM classifiers, where N denotes the number of classes. Each classifiers opposes 2 classes and decides which one is the image is. Then, we aggregate the votes: the final class is the one that was the returned the most by the $N \cdot (N-1)$ classifiers. Our classifiers were built solving the dual SVM formulation with a QP-solver.

4.3 Comparison regarding Computation Time

We compared the execution time of the aforementioned SVM implementation in Table1.

In our pipeline, OvO seemed to yield slightly better performances than Crammer-Singer, so we quickly put aside Crammer-Singer, given that even though OvO is slower than Crammer Singer, it remains tractable.

Table 1: Execution Time

	TRAINING TIME	PREDICTION TIME
CRAMMER-SINGER	18 S	0.2 S
ONE-VS-ONE	67 S	4 S

5 Results

We chose the parameters for HOG with a Kernel PCA (Gaussian) followed by a SVM with Gaussian kernel. We did not recompute all the features in every possible configuration (size of cells, signed/unsigned, ...) to select the hyperparameters of the PCA and SVM.

We selected the hyperparameters using 5-fold cross validation. Our best cross-validation score was obtained using both PCA and SVM with a Gaussian kernel, with the following parameters. Table 2 provides the cross-validation scores obtained for different pairs of kernels (PCA and SVM).

- Kernel PCA: $\gamma = 1$, $n_{\text{components}} = 500$,
- SVM (OvO): $\gamma = 3$, $C = 100$.

Table 2: Cross Validation Scores

(PCA / SVM)	LINEAR	GAUSSIAN	LAPLACIAN
LINEAR	50.0 %	55.4%	54.8 %
GAUSSIAN	52.8 %	**57.2%**	56.0%
LAPLACIAN	53.2 %	56.7%	56.4%

We found that the performance of the Crammer-Singer formulation, although comparable, was slightly lower that that of the One vs One version. In order to compare, our best configuration (reached 56.1% accuracy) with Crammer-Singer was:

- Kernel PCA (Gaussian kernel): $\gamma = 0.5$, $n_{\text{components}} = 500$,
- SVM (Gaussian kernel): $\gamma = 3$, $C = 0.016$.

6 Conclusion

Our best score was obtained using HOG features with a Gaussian kernel PCA and a SVM with Gaussian kernel. This pipeline reached **59.7%** accuracy on the public leaderboard and **59.4%** accuracy on the private leaderboard. We could have also tried other image descriptors such as SIFT or kernel descriptors, and tested some different kernels (chi2, histogram intersection kernel,...).

References

[BFU] Mathieu Blondel, Akinori Fujino, and Naonori Ueda. Large-scale multiclass support vector machine training via euclidean projection onto the simplex. In *Pattern Recognition (ICPR), 2014 22th International Conference on*.

[CS] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*.

[DT] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005*.