A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications

Joanna Matuszak, Joanna Wojciechowicz

2024-06-16

Abstract

Machine learning leverages various scientific disciplines to solve complex problems, with classification being one of the key applications. Support Vector Machines (SVMs), introduced by V. Vapnik, are effective for this task due to their use of kernels, which transform data into higher-dimensional spaces where the data can be separated easier. Traditional kernels such as linear, polynomial, and Gaussian may not capture the complex relationships in multimedia data or may not find application to variable length data. To address this, new kernels such as Fisher and Kullback-Leibler (KL) divergence based kernels have been proposed. In this project, we compare the performance of SVMs using these new kernels against traditional ones in the two-class classification task on the CIFAR-10 dataset.

1 Introduction

Machine learning is a highly interdisciplinary field, relying on various scientific disciplines to create models capable of solving complex problems [3]. One key application of machine learning is classification, a supervised learning approach that analyses a given data set to build a model capable of separating data into distinct classes. Among the well-established algorithms for this task is the Support Vector Machine (SVM), introduced by V. Vapnik as a kernel-based model for both classification and regression.

SVMs are particularly powerful because of their ability to utilise kernels. A kernel is a mathematical function that transforms data into a higher-dimensional space, making it easier to find a separating hyperplane when the data is not linearly separable in the original space [2]. This "kernel trick" allows SVMs to handle complex, nonlinear relationships within the data, enhancing their flexibility and accuracy in various applications.

A popular choice of the kernel for SVM is usually linear, polynomial, or Gaussian kernel. Despite many advantages, those three may not be the best choice for dealing with multimedia data. Note that such traditional kernels are based on inner products between individual feature vectors. However, multimedia signals often involve complex relationships and dependencies between multiple feature vectors or components. These kernels might not fully capture the complex relationships within multimedia data.

Another drawback is that traditional SVM kernels are designed to be generic and are not specifically tailored to take advantage of the statistical characteristics or properties of the individual signals or data types to which they are applied.

In order to address those or other possible issues, numerous approaches have been proposed for multimedia classification ([1, 4, 5]). In paper [5] they analyse a Fisher kernel and introduce a kernel based on the Kullback-Leiber divergence and compare the performance of SVMs with those kernels with baseline Generative Mixture Models and baseline Arithmetic Harmonic Sphericity Classifiers. However, we find it interesting to compare proposed Fisher an KL Divergence based kernel directly with well-known linear, polynomial and RBF kernels. Therefore, after discussing those two new approaches, we will perform the image classification on the CIFAR-10 dataset using SVMs with classical kernels and Fisher and KL Divergence based kernels.

2 Fisher Kernel

Let us assume that there exists a generative model that explains our data well. Our goal is to find the distribution

$$p(\mathbf{x}|\theta), \tag{1}$$

where θ is the vector of model's parameters. One of the most popular choices for dealing with speech signals or image classification are the Gaussian Mixture Models [5]. In case of GMMs the parameters of the model θ are the priors, means, and covariance matrices.

Let us assume that each multimedia object X is defined by a sequence of i.i.d. vectors, that is,

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}. \tag{2}$$

In such case, we can express the likelihood of the ensemble being produced by $p(\mathbf{x}|\theta)$ as

$$P(X|\theta) = \prod_{i=1}^{m} p(\mathbf{x}_i|\theta). \tag{3}$$

Having found the generative model explaining the data we can now try to map the data into a new feature space. It can be done with use of the Fisher score [5] given by the gradient of the log-likelihood function with respect to the model parameters, that is

$$\mathbf{U}_\mathbf{X} = \nabla_\theta \log(P(X|\theta)). \tag{4}$$

Summing up, the Fisher score measures how sensitive the log-likelihood is to changes in each parameter. Note that each sequence of vectors X can vary in length, but the Fisher score transforms this sequence into a single fixed-length vector. This is an important advantage since this transformation allows sequences of different lengths to be compared and used in machine learning models that require fixed-length input.

In our case, the parameters θ of the generative model are drawn from the prior probabilities, the mean vector, or the covariance matrix of each Gaussian component in the mixture model. If we choose to use the mean vecors as the model's parameters we can express the Fisher score using a simple formula [5]

$$\nabla_{\mu_\mathbf{k}} \log(P(X|\mu_\mathbf{k})) = \sum_{i=1}^{m} P(k|\mathbf{x}_i)\Sigma_k^{-1}(\mathbf{x}_i - \mu_\mathbf{k}), \tag{5}$$

where $\mu_\mathbf{k}$ is the mean vector for mixture k out of K possible mixtures, $P(k|\mathbf{x}_i)$ is the *a posteriori* probability of mixture k given the observed feature vector $\mathbf{x}_i$.

3 Kullback-Leibler Divergence Based Kernel

The next approach will not be based on the assumption that there exists a generative model that represents all data. Instead, we will start with fitting a statistical model

$$p(\mathbf{x}|\theta_i) \tag{6}$$

for each object. By an object, we mean a sequence of vectors $X_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$. There are many possibilities of choosing a type of model for this task, however, the ones that seem to be promising are again Gaussian Mixture Models [5]. In this case, the parameters θ_i for the object i are priors, mean vectors, and covariance matrices. In particular, we will focus on the single full covariance Gaussian model, for which the parametes for the object are only mean vector and covariance matrix.

Having estimated the distribution $p(\mathbf{x}|\theta_i)$, we move from the original space to the space of the PDFs. Therefore, we can replace the kernel computation in the original space with kernel computations in the space of PDFs. We have

$$K(X_i, X_j) \Longrightarrow K(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)). \tag{7}$$

In order to compute the kernel, we first need to choose a distance measure that we will apply in the new feature space. Note that we will be operating on PDFs, so the distance measure should be

a measure that compares two distributions. We choose the symmetric Kullback-Leiber divergence [5] given by

$$D(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)) = \int_{-\infty}^{\infty} p(\mathbf{x}|\theta_i) \log\left(\frac{p(\mathbf{x}|\theta_i)}{p(\mathbf{x}|\theta_j)}\right) d\mathbf{x} + \int_{-\infty}^{\infty} p(\mathbf{x}|\theta_j) \log\left(\frac{p(\mathbf{x}|\theta_j)}{p(\mathbf{x}|\theta_i)}\right) d\mathbf{x}. \tag{8}$$

Based on the Kullback-Leiber divergence we define the kernel as [5]

$$K(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)) = \exp\left(-\gamma D(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)) + \beta\right), \tag{9}$$

where γ is a scaling parameter and β is a shifting parameter.

Now, we will investigate further the case for single full covariance Gaussian model, for which the parameters (vector of means and covariance matrix) can be computed explicitly. In this case, the KL divergence can be computed directly [5] from the formula

$$D(p(x|\theta_i), p(x|\theta_j)) = \mathrm{tr}(\Sigma_i\Sigma_j^{-1}) + \mathrm{tr}(\Sigma_j\Sigma_i^{-1}) - 2S + \mathrm{tr}\left((\Sigma_i^{-1} + \Sigma_j^{-1})(\mu_i - \mu_j)(\mu_i - \mu_j)^T\right), \tag{10}$$

where S is the dimensionality of the original data $\mathbf{x}$. Note that in this case (from here we assume $\beta = 0$), the kernel defined in (9) is already a normalized kernel. We have

$$\begin{aligned} K(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_i)) &= \exp\left(-\gamma D(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_i))\right) = \\ &= \exp\left(-\gamma\left(\mathrm{tr}(\Sigma_i\Sigma_i^{-1}) + \mathrm{tr}(\Sigma_i\Sigma_i^{-1}) - 2S + \mathrm{tr}\left((\Sigma_i^{-1} + \Sigma_i^{-1})(\mu_i - \mu_i)(\mu_i - \mu_i)^T\right)\right)\right) = \\ &= \exp\left(-\gamma\left(S + S - 2S\right)\right) = 1. \end{aligned} \tag{11}$$

In the case of Gaussian Mixture Models, there is no analytical solution for θ_i, the parameters can be found using e.g. the Expectation Maximisation algorithm. In addition, the computation of the divergence (8) is not direct. It can be approximated or evaluated using Monte Carlo methods.

4 CIFAR-10 Image Classification

Next, we will write our own implementation of the discussed kernels, and then we will test them in the image classification task. We choose the CIFAR-10 dataset [6]. It consists of 50,000 32x32 colour training images and 10,000 test images, labelled in 10 categories.

We choose two classes (cars and planes), 250 images from each one, divide the dataset into train and test, and perform the classification. It is worth noticing that this choice of the data set makes the classification task challenging, since, as we can see in Fig. 1, CIFAR-10 consists of hard to distinguish objects with low definition. First, we investigate the performance of the SVM with three standard kernels. Since each image has the same size, we can use the classical kernels, such as the linear kernel, the polynomial kernel, and the RBF kernel. We will compare their performance with two kernels discussed before, that is, the Fisher kernel and the Kullback-Leibler divergence based kernel. For Fisher kernel we chose the GMM with 2 components and for KL divergence based kernel we used the single full covariance Gaussian model.

In the case of each kernel, we create an SVM model and tune its hyperparameters using 5-fold cross-validation. The hyperparameters are:

- `C` (regularization parameter) in case of linear kernel,
- `C` (regularization parameter), `degree`, `gamma` (scaling factor for the dot product of the input features before the polynomial term is applied), `coef0` (an independent term added to the product) in case of polynomial kernel,
- `C` (regularization parameter), `gamma` (scaling factor $\gamma = \frac{1}{\sigma^2}$) in case of RBF kernel,
- `C` (regularization parameter) in case of Fisher kernel,
- `C` (regularization parameter), `gamma` (scaling factor) in case of KL Divergence based kernel.

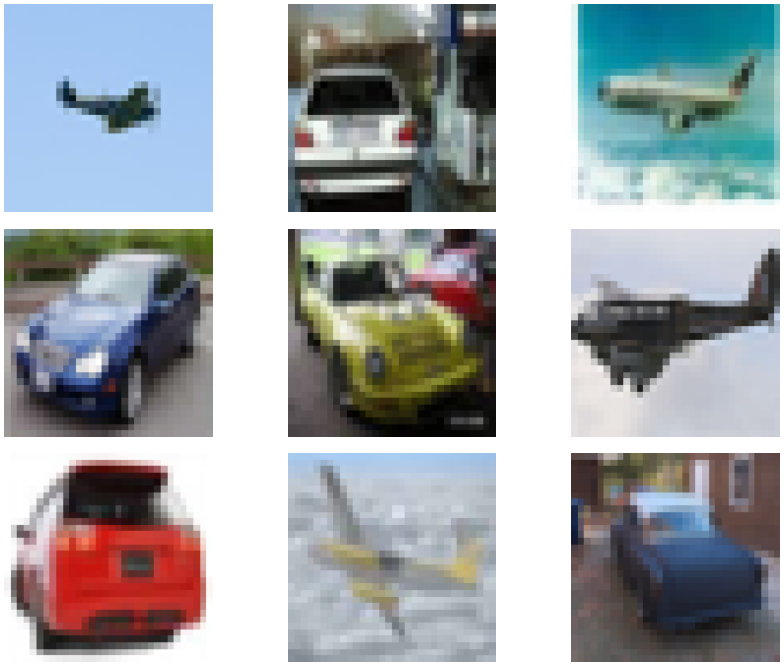

Figure 1: Examples of images from two chosen classes from CIFAR-10 dataset.

In Tables 1, 2, 3, 4 and 5 we can see the performance of the models after hyperparameter tuning. The best performance we got for:

- `C` = 0.05 for the SVM with linear kernel,
- `C` = 0.5, `degree` = 3, `gamma` = *"scale"*, `coef0` = 1 for the SVM with polynomial kernel,
- `C` = 100, `gamma` = 0.001 for SVM with RBF kernel,
- `C` = 5 for SVM with Fisher kernel,
- `C` = 10, `gamma` = 0.05 for SVM with KL Divergence based kernel.

The highest accuracy was achieved by the SVM with RBF kernel ($accuracy = 0.85$). This was closely followed by the SVM with the polynomial kernel ($accuracy = 0.84$), and the SVM with the KL Divergence based kernel ($accuracy = 0.83$). The SVM with the linear kernel obtained an accuracy of 0.78, while the SVM with the Fisher kernel had the lowest accuracy at 0.67.

The KL Divergence based kernel performed very well, with results nearly matching those of the RBF and polynomial kernels. However, the Fisher kernel did not yield satisfactory results, as its accuracy was the lowest among all the models studied.

Examining the precision for each class, which is the ratio of correctly predicted positive observations to the total predicted positives, we observe that for all models except the one with the Fisher kernel, the precision was higher for class 1 (cars). The SVM with the RBF kernel achieved the highest macro average precision and weighted average precision, both at 0.85. In contrast, the SVM with the Fisher kernel had the lowest scores, with a macro average precision of 0.68 and a weighted average precision of 0.68.

Regarding recall for each class, which is the ratio of correctly predicted positive observations to all actual positives in the class, the values were similar across all models except for the SVM with the Fisher kernel. For class 0 (planes), the recall was 0.50, while for class 1 (cars), it was 0.81. This indicates that the model correctly predicted the majority of observations in class 1 (cars) but struggled with class 0 (planes).

Overall, considering macro average and weighted average statistics, the SVM with the RBF kernel outperformed the others and the SVM with Fisher kernel gave the worst results.

	Precision	Recall	F1-Score	Support
0	0.75	0.78	0.77	46
1	0.81	0.78	0.79	54
Accuracy			0.78	100
Macro Avg	0.78	0.78	0.78	100
Weighted Avg	0.78	0.78	0.78	100

Table 1: Classification Report for the SVM model with linear kernel and $C = 0.05$.

	Precision	Recall	F1-Score	Support
0	0.81	0.85	0.83	46
1	0.87	0.83	0.85	54
Accuracy			0.84	100
Macro Avg	0.84	0.84	0.84	100
Weighted Avg	0.84	0.84	0.84	100

Table 2: Classification Report for the SVM model with polynomial kernel $C = 0.5$, $degree = 3$, $gamma =$ "$scale$" and $coef0 = 1$.

	Precision	Recall	F1-Score	Support
0	0.84	0.83	0.84	46
1	0.85	0.87	0.86	54
Accuracy			0.85	100
Macro Avg	0.85	0.85	0.85	100
Weighted Avg	0.85	0.85	0.85	100

Table 3: Classification Report for the SVM model with RBF kernel, $C = 100$ and $gamma = 0.001$.

	Precision	Recall	F1-Score	Support
0	0.70	0.50	0.58	46
1	0.66	0.81	0.73	54
Accuracy			0.67	100
Macro Avg	0.68	0.66	0.65	100
Weighted Avg	0.68	0.67	0.66	100

Table 4: Classification Report for the SVM model with Fisher kernel and $C = 5$.

	Precision	Recall	F1-Score	Support
0	0.78	0.87	0.82	46
1	0.88	0.80	0.83	54
Accuracy			0.83	100
Macro Avg	0.83	0.83	0.83	100
Weighted Avg	0.83	0.83	0.83	100

Table 5: Classification Report for the SVM model with KL divergence based kernel, $C = 10$ and $gamma = 0.05$.

5 Critical assesment

Looking at our results that we presented in Chapter 4 we can see that for chosen dataset with the images of the same size, we cannot see the benefit of using presented kernels instead of well-known and already implemented in Python RBF kernel. Worth mentioning is the fact that in the case of Fisher and KL divergence based kernels the computation time was significantly larger than the one for

classical kernels. However, we note that proposed solution (KL divergence based kernel) gave similar results to RBF and polynomial one, therefore it could be beneficial to use it in situations where the standard kernels cannot be used.

6 Future work

Future work on this topic can extend the experiments to encompass the entire CIFAR-10 data set, including images from 10 different classes. Additionally, more detailed simulations can be conducted by tuning the hyperparameters on a denser grid.

Given that the Fisher kernel and the Kullback-Leibler divergence based kernel are designed for multimedia data, not just images, their performance can also be evaluated on an audio dataset. Moreover, both kernels can be utilised for image data sets that contain images of various sizes, making it beneficial to conduct experiments on such data as well.

These directions will provide a more comprehensive understanding of the kernels' capabilities across different types of multimedia data.

7 Summary

In this work, we investigated kernels designed for multimedia data, specifically the Fisher kernel and the KL divergence-based kernel. We provided a theoretical introduction to these kernels, implemented them in Python, and evaluated their performance on a subset of the CIFAR-10 dataset, focussing on two classes (cars and planes). We compared these multimedia-specific kernels with classical, well-known kernels such as linear, polynomial, and RBF kernels.

Our experiments demonstrated that the KL divergence-based kernel performed nearly as well as the RBF kernel, yielding satisfactory results by effectively classifying both classes. In contrast, the Fisher kernel was the least effective among the kernels we studied. Therefore, we can conclude that the KL divergence-based kernel is a promising alternative in scenarios where standard kernels are inadequate or when a multimedia-specific kernel is required.

References

[1] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 3, pages III–513, 2003.

[2] Colin Campbell. Kernel methods: a survey of current techniques. *Neurocomputing*, 48(1-4):63–84, 2002.

[3] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[4] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Image categorization using fisher kernels of non-iid image models. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2184–2191, 2012.

[5] Pedro J. Moreno, Purdy P. Ho, and Nuno Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, page 1385–1392, Cambridge, MA, USA, 2003. MIT Press.

[6] TensorFlow Datasets. Cifar-10 dataset. https://www.tensorflow.org/datasets/catalog/cifar10?hl=pl. Accessed: 2024-06-16.