(1641)

Variable Projection Support Vector Machines

Tamás Dózsa [1,2] and Péter Kovács [1]
[1] Eötvös Loránd University, Faculty of Informatics, Department of Numerical Analysis, Budapest, Hungary
[2] Institute for Computer Science and Control, Systems and Control Laboratory, Budapest, Hungary
E-mail: dotuaai@inf.elte.hu

Summary: We introduce an extension of the classical support vector machine classification algorithm with adaptive orthogonal transformations. The proposed transformations are realized through so-called variable projection operators. This approach allows the classifier to learn an informative representation of the data during the training process. Furthermore, choosing the underlying adaptive transformations correctly allows for learning interpretable parameters. Since the gradients of the proposed transformations are known with respect to the learnable parameters, we focus on training the primal form the modified SVM objectives using a stochastic subgradient method. We consider the possibility of using Mercer kernels with the proposed algorithms. We construct a case study using the linear combinations of adaptive Hermite functions where the proposed classification scheme outperforms the classical support vector machine approach. The proposed variable projection support vector machines provide a lightweight alternative to deep learning methods which incorporate automatic feature extraction.

Keywords: SVM, Variable projection, Adaptive orthogonal transformations, Classification, Kernel methods.

1. Introduction

Classical machine learning approaches often rely on predefined feature extraction steps applied to the data before the training of the classification or regression algorithms. Commonly used feature extraction steps range from simple statistical methods like principal component analysis (PCA) [5] to more sophisticated adaptive transformations [1, 6, 7, 8, 19]. Many classification tasks can be successfully solved this way (for example in biological signal processing [6, 7]), however the main limitation of this classical approach is that the resulting representation of the data and the weights of the underlying classifier are optimized separately. Practically this means that ensuring an appropriate representation requires apriori information about the structure of the data.

The introduction of convolutional neural networks [9, 10] addresses this limitation by binding the optimization of data representation to the training of the classifier's weights. Feature extraction in this case is done in an adaptive manner and the resulting data representation is optimal *for the classification task.* This idea gave rise to many popular machine learning algorithms especially in image processing [10].

The application of such approaches to real life problems suffers from the fact that the learned weights of the convolution kernels have no physical meaning [1]. The recent introduction of model based neural network architectures, especially the so-called VP-Net [1] aims to remedy this problem. In a VP-Net, the first few layers (VP-Layers) of a neural network implement adaptive orthogonal transformations. Similarly, to convolutional networks, these layers learn informative features of the data. Training a VP-Net however, often involves the optimization of less parameters than in the case of convolution layers. Another advantage of VP-Net is that if the underlying orthogonal transformations were chosen correctly, then the learned parameters of the VP-layers can be interpreted. VP-Net has already been shown to be an appropriate classifier choice for some problems arising in biological signal processing [1, 11] and autonomous vehicle control [8].

In this work, we investigate the possibility of extending the popular SVM classification algorithm with adaptive orthogonal transformations similarly to the idea of VP-Net. Such an extension can be useful in cases, where computational capacity is limited [8], or the classification task does not require the construction of deep neural networks.

The rest of this paper is organized as follows. In Section 2, we discuss the general form of the adaptive orthogonal transformations which will be used to perform automatic feature extraction. In Section 3 we extend the SVM objective functions using these transformations. In Section 4 we discuss the training of the introduced classifiers using stochastic subgradient descent. Section 5 discusses a numerical experiment. Finally, in Section 6 we draw our conclusions and discuss future steps.

2. Variable Projection Operators

We are going to assume that our data consists of $\boldsymbol{x} \in R^N$ vectors. Our aim is to represent $\boldsymbol{x}$ by $n \ll N$ numbers:

$$\boldsymbol{x} \approx \mathrm{P}_{\Phi(\boldsymbol{\eta})}\boldsymbol{x} = \Phi(\boldsymbol{\eta})(\Phi^{+}(\boldsymbol{\eta})x), \qquad (1)$$

where $\Phi(\boldsymbol{\eta}) \in R^{N\times n}$, $\boldsymbol{\eta} \in R^m$, $\Phi^{+}(\boldsymbol{\eta})$ refers to the Moore-Penrose pseudoinverse of the matrix $\Phi(\boldsymbol{\eta})$ and N, n, m are natural numbers. Thus, we represent the data $\boldsymbol{x}$ with the vector $\Phi^{+}(\boldsymbol{\eta})\boldsymbol{x} \in R^n$ and (1) defines the projection of $\boldsymbol{x}$ onto the column space of $\Phi(\boldsymbol{\eta})$. The

columns of $\Phi(\boldsymbol{\eta})$ consist of discrete samplings of a (usually complete and orthonormal) function system in the Lebesgue space $L_2(R)$. The function system that defines $\Phi(\boldsymbol{\eta})$ depends on the parameter vector $\boldsymbol{\eta}$ in a nonlinear fashion and the operator $\mathrm{P}_{\Phi(\boldsymbol{\eta})}$ is referred to as a variable projection operator [2, 19]. Clearly, the feature extraction scheme (1) depends on the parameter $\boldsymbol{\eta}$. A common procedure to obtain a good representation is to solve

$$\min_{\eta \in R^m} r_2(\boldsymbol{\eta}; \boldsymbol{x}) = \min_{\eta \in R^m} ||\boldsymbol{x} - P_{\Phi(\boldsymbol{\eta})}\boldsymbol{x}||_2^2. \quad (2)$$

Provided that the partial derivatives of the functions generating the columns of $\Phi(\boldsymbol{\eta})$ are known with respect to $\boldsymbol{\eta}$, (2) can be solved using a gradient based method [2, 19].

We note that solving the optimization task (2) allows for several adaptive representations for the vector $\boldsymbol{x}$. Suppose the columns of $\Phi(\boldsymbol{\eta})$ consist of smooth functions. Then, once the parameter vector $\boldsymbol{\eta}$ that minimizes (2) has been determined, one could replace $\boldsymbol{x}$ with its smooth approximation $\mathrm{P}_{\Phi(\boldsymbol{\eta})}\boldsymbol{x}$ or the residual transformation $\boldsymbol{x} - \mathrm{P}_{\Phi(\boldsymbol{\eta})}\boldsymbol{x}$. In some applications [6, 8] such data representations are preferable, however in this work, for simplicity, we are only going to discuss the extension of SVM classifiers with the transformation $\boldsymbol{x} \to \Phi^+(\boldsymbol{\eta})(\boldsymbol{x})$.

3. VP-SVM Objectives

Given a training set with q number of examples $\{(\boldsymbol{x}_i, y_i)\}_{1 \le i \le q}$, $\boldsymbol{x}_i \in R^N, y_i \in \{-1,1\}$, the purpose of an SVM classifier is to identify a hyperplane which separates the examples $\boldsymbol{x}_i$. In the simplest (linear) case, the hyperplane is given by its normal vector $\boldsymbol{w} \in R^N$, whose components are found through an optimization process known as training. Most commonly, the training of SVM classifiers is posed as a linear programming problem and convex optimization tools are used to find the optimal hyperplane [12-14]. In real life applications, SVM classifiers often rely on Mercer kernels [3, 4, 12, 13]. These allow us to transform the data examples $\boldsymbol{x}_i$ to a high dimensional reproducing kernel Hilbert space (RKHS), where a separating hyperplane might more easily be identified. In this (nonlinear) case, dual formulations of the above-mentioned linear programming problem are solved using convex optimization. When training such nonlinear SVM classifiers, the dual problems are preferable, as they allow to express the above-mentioned transformations as inner products of the examples $\boldsymbol{x_i}$.

For the proposed VP-SVM classifiers, we consider gradient based algorithms for training. This is because, the adaptive transformations (2) by which we enhance the SVM objectives are optimized using gradient based methods. Fortunately, extensive literature exists on how to train SVM objectives with such methods [3, 4], focusing mostly on the primal form of the optimization problem. For this reason, we take as starting points the SVM objectives discussed in [3]. It is well-known [3], that the primal form of the linear SVM classifier objective can be reduced to minimizing the following expression with respect to $\boldsymbol{w}$ and b:

$$C \sum_{i=1}^{q} \max(0, 1 - y_i (\boldsymbol{w}^T \boldsymbol{x_i} + b)) + ||\boldsymbol{w}||^2, \quad (3)$$

where $\boldsymbol{w} \in R^N, C, b \in R$.

As mentioned before, several strategies [3, 4] exist for solving the optimization problem (3), however for simplicity, in this work we focus on obtaining a solution using a subgradient based method (see Section 4). An adaptive feature extraction step can be easily added to (3) by

$$C \sum_{i=1}^{q} \max(0, 1 - y_i (\boldsymbol{w}^T (\Phi^+(\boldsymbol{\eta})\boldsymbol{x}_i) + b) + + ||\boldsymbol{w}||^2 + R(\boldsymbol{\eta}), \quad (4)$$

where this time $\boldsymbol{w} \in R^n$ and $R(\boldsymbol{\eta})$ is a regulatory term to ensure a good representation of the data and to avoid the problem of vanishing (sub)gradients with respect to $\boldsymbol{\eta}$:

$$R(\boldsymbol{\eta}) := \frac{\alpha}{q} \sum_{i=1}^{q} \frac{||\boldsymbol{x}_i - P_{\Phi(\boldsymbol{\eta})}\boldsymbol{x_i}||_2^2}{||\boldsymbol{x}_i||_2^2} \quad (5)$$

In (5), $\alpha \in R$ is a penalty parameter for the regulatory term. In Section 4 we provide the subgradients of (4) with respect to $\boldsymbol{\eta}$. These can be calculated provided that the partial derivatives of $\Phi(\boldsymbol{\eta})$ are known [1, 2].

One of the main advantages of support vector machines is their ability to be used with Mercer kernels [3]. Even though in this case, usually the dual form of the SVM objective is considered, popular algorithms exist which optimize the primal objective as well [3, 4]. In this work, we consider the objective presented in [3]:

$$\min_{f \in \mathcal{H}} \lambda ||f||_{\mathcal{H}}^2 + \sum_{i=1}^{q} \max\big(0, 1 - y_i f(\boldsymbol{x}_i)\big), \quad (6)$$

where $\lambda = 1/C, \mathcal{H}$ is a reproducing kernel Hilbert space and by the representer theorem [20] we can look for the solution f in the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{q} \beta_i k(\boldsymbol{x}, \boldsymbol{x}_i) \ (\boldsymbol{\beta} \in \mathrm{R}^q), \quad (7)$$

provided that the inner product $k(\cdot,\cdot)$ of $\mathcal{H}$ is known. We omitted the bias parameter b from (6) to simplify the expression, but it could easily be incorporated as well. Because of (7), supplementing the objective (6) with orthogonal transformations naively would yield a problem that has a complexity of $\mathcal{O}(q^2)$ for every update of the parameters. This is because evaluating the term $||f||_{\mathcal{H}}^2$ using (7) requires us to calculate the inner products $k(\boldsymbol{x_i}, \boldsymbol{x}_j)$ for every possible index i, j. To overcome this, we propose using instead the modified objective

$$\min_{\boldsymbol{\beta} \in R^q, \boldsymbol{\eta} \in R^m} \sum_{i=1}^{q} \max\big(0, 1 - y_i \sum_{j=1}^{q} \beta_j k(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)\big) + R(\boldsymbol{\eta}), \quad (9)$$

where $R(\boldsymbol{\eta})$ is the regulatory term from (5) and $\tilde{x}_i$ denotes the transformed vector $\Phi^+(\boldsymbol{\eta})\boldsymbol{x}_i$. This notation will be used throughout the rest of the paper. Similarly, to the linear case, this regulatory term is responsible for ensuring that the projections $\mathrm{P}_{\Phi(\boldsymbol{\eta})}\boldsymbol{x}_i$ approximate the examples $\boldsymbol{x_i}$ well, while keeping the problem of vanishing gradients at bay. In our experiments (Section 5) we show that problems can be constructed when the simplified data representation through the proposed method allows for easier separation of the classes.

4. Training the VP-SVM Classifiers

We note that the proposed VP-SVM objectives (4) and (9) are not differentiable everywhere. Because of this, we can only utilize subgradient based methods to perform the minimization task. The subgradients of a function $f: R^N \to R$ are defined at the point $\boldsymbol{x}$ as the set of vectors $\boldsymbol{g} \in R^N$ for which

$$f(\boldsymbol{z}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^T(\boldsymbol{z} - \boldsymbol{x}), \tag{10}$$

for all $\boldsymbol{z}$ in the domain of f. If the function f is convex and differentiable, then the subgradient coincideces with the gradient. The above-mentioned objective functions are subdifferentiable with respect to the learnable parameters. In addition, $\boldsymbol{x}$ is a minimizer point of a convex function f if and only if $0 \in \partial f(\boldsymbol{x})$, or in other words 0 is a subgradient of f at $\boldsymbol{x}$. This property allows for the construction of optimization algorithms using the notion of subgradients.

In [4] an efficient subgradient based algorithm is presented to minimize the objective functions (3) and (6). The adaptation of this algorithm to train VP-SVM will be part of our future work, however in this study we used the stochastic subgradient descent (SSGD) [14] method for training. In each step of the training process, we randomly select a single example $\boldsymbol{x_i}$, calculate the subgradient of the objective function with respect to the learnable parameters, then update the parameters. This in practice means that the objective functions being minimized change from the form presented in section 3. For example, instead of calculating the subgradients of objective (3), in single step of SSGD we would subdifferentiate the modified loss

$$q \cdot C \cdot \max(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x_i})) + ||\boldsymbol{w}||^2, \tag{11}$$

where q denotes the total number of examples and C is a hyper-parameter that controls the tradeoff between the margin and the hinge loss. Again, the bias parameter was omitted for simplicity. We can modify the proposed linear VP-SVM objective (4) similarly by

$$J(\boldsymbol{w}, \boldsymbol{\eta})_{\boldsymbol{i}} := q \cdot C \cdot \max(0, 1 - y_i(\boldsymbol{w}^T\tilde{x}_i) + \\ + ||\boldsymbol{w}||^2 + R(\boldsymbol{\eta}, \boldsymbol{x}_i), \tag{12}$$

where for better scalability we also change the regularization term (5) to

$$R(\boldsymbol{\eta}, \boldsymbol{x}_i) := \alpha \cdot \frac{||\boldsymbol{x_i} - P_{\Phi(\boldsymbol{\eta})}\boldsymbol{x_i}||_2^2}{||\boldsymbol{x_i}||_2^2} \tag{13}$$

The subgradients of (12) exist with respect to $\boldsymbol{\eta}$ and $\boldsymbol{w}$, and can be expressed as

$$\frac{\partial J(\boldsymbol{w},\boldsymbol{\eta})_i}{\partial \boldsymbol{w}} = \begin{cases} \boldsymbol{w} - C \cdot q \cdot y_i \tilde{x}_i, if\ s > 0 \\ \boldsymbol{w}, else \end{cases}, \tag{14}$$

where $s := 1 - y_i(\boldsymbol{w}^T\tilde{x}_i)$ and

$$\frac{\partial J(\boldsymbol{w},\boldsymbol{\eta})_i}{\partial \boldsymbol{\eta}} = \begin{cases} -C \cdot q \cdot y_i \cdot \frac{\partial \tilde{x}_i}{\partial \boldsymbol{\eta}} + \frac{\partial R(\boldsymbol{\eta},\boldsymbol{x}_i)}{\partial \boldsymbol{\eta}}, if\ s > 0 \\ \frac{\partial R(\boldsymbol{\eta},\boldsymbol{x}_i)}{\partial \boldsymbol{\eta}}, else \end{cases}, \tag{15}$$

In (15), the gradients $\frac{\partial \Phi^+(\boldsymbol{\eta})x_i}{\partial \boldsymbol{\eta}} = \frac{\partial \tilde{x}_i}{\partial \eta}$ and $\frac{\partial R(\boldsymbol{\eta},\boldsymbol{x}_i)}{\partial \boldsymbol{\eta}}$ can be calculated provided that the partial derivatives of $\Phi(\boldsymbol{\eta})$ are known with respect to $\boldsymbol{\eta}$ [1, 2]. For the exact formulas we refer to [1]. Once (14) and (15) have been calculated, the stochastic subgradient descent algorithm updates the parameters with

$$\boldsymbol{\eta} \to \boldsymbol{\eta} - \gamma_i \cdot \frac{\partial J(\boldsymbol{w},\boldsymbol{\eta})_i}{\partial \boldsymbol{\eta}} \text{ and} \\ \boldsymbol{w} \to \boldsymbol{w} - \gamma_i \cdot \frac{\partial J(\boldsymbol{w},\boldsymbol{\eta})_i}{\partial \boldsymbol{w}} \tag{16}$$

If the learning rate γ_i is sufficiently small, then the above-described stochastic subgradient descent algorithm is guaranteed to converge [14].

Similarly, to the linear case, the proposed nonlinear version of the VP-SVM objective (9) can be trained using the SSGD algorithm. In this case, the objective to be optimized can be expressed as

$$J(\boldsymbol{\beta}, \boldsymbol{\eta})_i := qC \max\left(0, 1 - y_i \sum_{j=1}^{q} \beta_j k(\tilde{x}_i, \tilde{x}_j)\right) + \\ + R(\boldsymbol{\eta}, \boldsymbol{x}_i), \tag{17}$$

where $R(\boldsymbol{\eta}, \boldsymbol{x}_i)$ is defined by (13). The subgradients of (17) with respect to the learnable parameters are given as

$$\frac{\partial J(\boldsymbol{\beta},\boldsymbol{\eta})_i}{\partial \boldsymbol{\beta_j}} = \begin{cases} -Cq\, y_i\, k(\tilde{x}_i, \tilde{x}_j), if\ u > 0 \\ 0, else \end{cases}, \tag{18}$$

and

$$\frac{\partial J(\boldsymbol{\beta},\boldsymbol{\eta})_i}{\partial \boldsymbol{\eta}} = \begin{cases} \frac{\partial R(\boldsymbol{\eta},\boldsymbol{x}_i)}{\partial \boldsymbol{\eta}} - Cq\, y_i \sum_{j=1}^{q} \beta_j \frac{\partial k(\tilde{x}_i,\tilde{x}_j)}{\partial \boldsymbol{\eta}}, if\ u > 0 \\ \frac{\partial R(\boldsymbol{\eta},\boldsymbol{x}_i)}{\partial \boldsymbol{\eta}}, else \end{cases} \tag{19}$$

In (18) and (19) the subgradients are determined by the magnitude of

$$u := 1 - y_i \sum_{j=1}^{q} \beta_j k(\tilde{x}_i, \tilde{x}_j)$$

Furthermore, in (19) the partial derivatives with respect to $\boldsymbol{\eta}$ clearly depend on the choice of the kernel $k(\cdot,\cdot)$. If, for example one chooses the popular radial basis kernel function defined as

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

then $k(\tilde{x}_i, \tilde{x}_j) := k(\Phi^+(\boldsymbol{\eta})\boldsymbol{x}_i, \Phi^+(\boldsymbol{\eta})\boldsymbol{x}_j)$ becomes

$$e^{\frac{-\left\|\Phi^+(\boldsymbol{\eta})x_i - \Phi^+(\boldsymbol{\eta})x_j\right\|_2^2}{\sigma^2}} = e^{\frac{-\left\|\Phi^+(\boldsymbol{\eta})\,(x_i - x_j)\right\|_2^2}{\sigma^2}}$$

and $\frac{\partial k(\Phi^+(\boldsymbol{\eta})x_i, \Phi^+(\boldsymbol{\eta})x_j)}{\partial \boldsymbol{\eta}}$ can be given as

$$-e^{\frac{-\left\|\Phi^+(\boldsymbol{\eta})(x_i - x_j)\right\|_2^2}{\sigma^2}} \cdot 2 \cdot \Phi^+(\boldsymbol{\eta})(\boldsymbol{x}_i - \boldsymbol{x}_j) \cdot \frac{1}{\sigma^2} \cdot \frac{\partial \Phi^+(\boldsymbol{\eta})(x_i - x_j)}{\partial \boldsymbol{\eta}}$$

For a fixed kernel, in each step of the SSGD algorithm the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ can be updated similarly to the linear case (16) once the subgradients (18) and (19) have been calculated.

5. Experiments

We demonstrate the utility of the proposed VP-SVM classifier (9) through an example problem. In our case study, the training examples $\boldsymbol{x}_i \in R^N$ $(i = 1, \dots, q)$ consisted of the linear combinations of the first two so-called adaptive Hermite functions [17]. These functions are closely related to the classical Hermite orthogonal polynomials [15, 16]. Namely, let

$$h_k(x), \quad (x \in R)$$

denote the k-th Hermite polynomial. These form a complete orthogonal system in the weighted Lebesgue space $L_2^w(R)$, where the weight function is given as $w(x) = e^{-x^2}$. We can then introduce the so called Hermite functions by

$$\varphi_k(x) := \\ = h_k(x) \cdot e^{-x^2/2} \cdot \sqrt{\pi^{1/2} 2^k k!} \quad (k \in N, x \in R) \tag{20}$$

The functions defined in (20) form a complete system in $L_2(R)$ and are orthonormal with respect to the usual inner product:

$$\langle \varphi_k, \varphi_j \rangle = \int_{-\infty}^{\infty} \varphi_k(x)\varphi_j(x)dx = \delta_{kj}$$

These functions and their subsequent generalizations form the basis of many signal processing applications [1, 6, 8, 11, 17, 18]. To properly describe the training set for our case study, we need to mention the affine argument transformations of (20)

$$\varphi_k^{\lambda,\tau}(x) := \\ = \sqrt{\lambda}\varphi_k(\lambda(x - \tau)) \; (x, \tau \in R, \lambda > 0, k \in N), \tag{21}$$

known as adaptive Hermite functions [17]. Adaptive Hermite functions have been especially useful for modeling quasi-periodic signals with quasi-compact support.

We constructed each example for our dataset by fixing the affine parameters $\boldsymbol{\eta} := (\lambda, \tau)$ and taking the linear combinations of the first two adaptive Hermite functions:

$$\boldsymbol{x}_i := c_{0,i}\varphi_0(\boldsymbol{\eta}) + c_{1,i}\varphi_1(\boldsymbol{\eta}), \\ (c_{0,i}, c_{1,\mathrm{i}} \in R, i = 1, \dots, q) \tag{22}$$

where $\varphi_{\boldsymbol{k}}(\boldsymbol{\eta})$ is an equidistant sampling of (21). The parameter $\boldsymbol{\eta} := (\lambda, \tau)$ remained constant for every example $\boldsymbol{x}_i$, and two classes were introduced through linear parameters $c_{0,i}$ and $c_{1,\mathrm{i}}$. This was done in a way, that the points given as $(c_{0,i}, c_{1,i})$ were randomly chosen from two concentric circles as shown in Fig. 1.

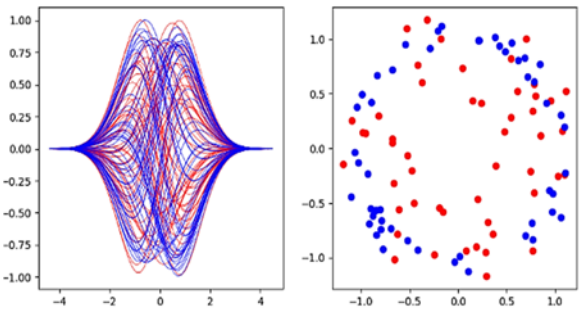

Fig. 1. LEFT: $\boldsymbol{x_i}$ training examples generated by adaptive Hermite functions. RIGHT: Corresponding $(\boldsymbol{c_{0,i}}, \boldsymbol{c_{1,i}})$ parameters from (22).

In the VP-SVM, the columns of the matrix $\Phi(\boldsymbol{\eta})$ also contained the adaptive Hermite functions (21). The idea in this case was to demonstrate, that if the correct $\boldsymbol{\eta} := (\lambda, \tau)$ are learned during the training process, then the examples shown on left side of Fig. 1 become more easily separable (the right side of Fig. 1). We minimized the objective function (17) using the methods described in Section 4. We compared the accuracy of VP-SVM on the set with a classical nonlinear SVM classifier also trained using SSGD. Both classifiers used an RBF kernel with the same σ parameter. Accuracy scores on the training and on the test, sets are shown in Fig. 2 during each epoch of the training.

By Fig. 2, the above-described experiment provides a proof of concept for the utility of the proposed VP-SVM classification algorithm. Since many anomalies detection and classification tasks have been studied, where features are extracted by orthogonal transformations [6, 8, 19], we are hopeful that the proposed algorithms can also be used to solve real world problems. VP-SVM provides a lightweight model-based machine learning approach for classification problems. This can be especially useful when model driven machine learning methods have been shown to provide good results, however the available computational capacity does not allow for using deep neural networks [8].

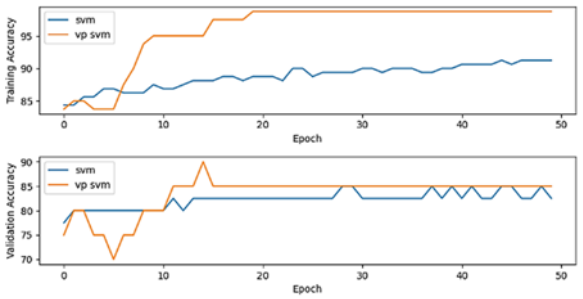

Fig. 2. TOP: training accuracies of VP-SVM (orange) and SVM (blue) using RBF kernels. BOTTOM: accuracy scores on unseen (validation) data at different epochs.

6. Conclusion

By utilizing variable projections, we proposed novel extensions of the usual SVM objectives. We discussed a training scheme using stochastic subgradient method for the proposed classifiers. We demonstrated the efficiency of our method through a simulated test scenario. The proposed VP-SVM extends the classical SVM classifiers with an automatic feature extraction scheme defined by adaptive orthogonal transformations. The learned parameters of the feature extraction can be interpreted [1].

The latter property is of utmost importance when applying the proposed classification scheme to biomedical signal processing tasks (such as illness recognition), which will be part of our future work. In addition, VP-SVM provides a lightweight alternative to similar deep learning-based feature learning and classification schemes such as [1, 10]. Using VP-SVM can be helpful in situations where computational capacity is limited [8].

Acknowledgements

This project was supported by the NVKDP Cooperative Doctoral Program by the Hungarian Ministry of National Development and the National Research, Development and Innovation Fund. This project was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1]. P. Kovács, et al., VPNET: Variable projection networks, *International Journal of Neural Systems*, Vol. 32, Issue 1, 2022, 21500544.

[2]. G. H. Golub, V. Pereyra, Separable nonlinear least squares: The variable projection method and its applications, *Inverse Problems*, Vol. 19, Issue 2, 2003, R1.

[3]. O. Chapelle, Training a support vector machine in the primal, *Neural Computation*, Vol. 19, Issue 5, 2007.

[4]. S. Shalev-Shwartz, et al., Pegasos: Primal estimated subgradient solver for SVM, *Mathematical Programming*, Vol. 127, Issue 1, 2011, pp. 3-30.

[5]. H. Abdi, L. J. Williams, Principal component analysis, *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2, Issue 4, 2010, pp. 433-459.

[6]. T. Dózsa, G. Bognár, P. Kovács, Ensemble learning for heartbeat classification using adaptive orthogonal transformations, in *Proceedings of the International Conference on Computer Aided Systems Theory (EUROCAST'19)*, 2019, pp. 355-363.

[7]. P. Addison, Wavelet transforms and the ECG: A review, *Physiological Measurement*, Vol. 26, Issue 5, 2005, R155-99.

[8]. T. Dózsa, J. Radó, J. Volk, Á. Kisari, A. Soumelidis, P. Kovács, Road abnormality detection using piezoresistive force sensors and adaptive signal models, *IEEE Transactions on Instrumentation and Measurement*, Vol. 71, 2022, pp. 1-11.

[9]. J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, Vol. 61, 2015, pp. 85-117,

[10]. A. Krizhevsky., I. Sutskever, G. Hinton, ImageNet classification with deep convolutional networks, in *Proceedings of the Conference Neural Information Processing Systems (NIPS'12)*, 2012, pp. 1-9.

[11]. T. Dózsa, C. Böck, G. Bognár, J. Meier, P. Kovács, Color classification of visually evoked potentials by means of Hermite functions, in *Proceedings of the 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 251-255.

[12]. S. Boyd, L. Vandenberghe, Convex Optimization, *Cambridge University Press*, 2004.

[13]. S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, *J. of Mach. Learning Res.*, Vol. 2, 2001, pp. 242-264.

[14]. J. Shawe-Taylor, S. Shiliang, A review of optimization methodologies in support vector machines, *Neurocomputing*, Vol. 74, Issue 17, 2011, pp. 3609-3618.

[15]. G. Szegő, Orthogonal Polynomials, 3rd Ed., *AMS Colloquium Publications*, 1967.

[16]. W. Gautschi, Orthogonal Polynomials, Computation and approximation, in Numerical Mathematics and Scientific Computation, *Oxford University Press*, 2004.

[17]. T. Dózsa, P. Kovács, ECG signal compression using adaptive Hermite functions, in *Proceedings of the International Conference on ICT Innovations*, 2015, pp. 245-254.

[18]. P. Kovács, C. Böck, T. Dózsa, J. Meier, M. Huemer, Waveform modeling by adaptive weighted Hermite functions., in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)*, 2019, pp. 1080-1084.

[19]. G. H. Golub, V. Pereyra, Separable nonlinear least squares: The variable projection method and its applications, *Inverse Problems*, Vol. 19, 2003, R1.

[20]. G. S. Kimeldorf, G. Wahba, A correspondence between Bayesian estimation on stochastic processes and smoothing by splines, *The Annals of Mathematical Statistics*, Vol. 41, Issue 2, 1970, pp. 495-502.