ACCELERATING SUPPORT VECTOR MACHINES FOR REMOTE PLATFORMS BY INCREASING SPARSITY

J. L. Garrett[1], *N. K. Singh*[2], *T. A. Johansen*[1], *and I. Necoara*[2]

[1]Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.
[2]Automatic Control and Systems Engineering Department, University Politehnica Bucharest, Romania.

ABSTRACT

The support vector machine (SVM) classification algorithm often achieves quite high accuracy on hyperspectral images, even when trained on small amounts of data. However, SVMs can still be computationally expensive relative to the desired throughput and available resources on remote imaging platforms.

In this paper, the possibility of decreasing the computational costs of SVMs by increasing their sparsity is explored on a few simple hyperspectral scenes. The number of bands is reduced by a factor of up to 20, which roughly corresponds with a $20\times$ reduction in the number of computations, with only few percent decrease in the overall accuracy. Furthermore, it is found that accuracy is fairly insensitive to the method by which different bands are selected to be retained after training.

Index Terms— Classification, Hyperspectral, Machine Learning, Remote Inspection, Remote Sensing

1. INTRODUCTION

Hyperspectral imaging finds more applications as it becomes accessible to remote platforms such as unmanned aerial vehicles and small satellites [1, 2, 3, 4, 5]. While many classification algorithms achieve high accuracy on powerful computers, the operation of classification on edge or embedded processing platforms is more difficult [6]. Many of the most accurate classification techniques utilize the neighbors of a pixel to determine its class [7], but on-board classification is typically limited to pixel-wise classifiers due to computational constraints [6].

Relative to other machine learning classification methods, Support Vector Machines (SVMs) require less memory to store their weights and typically classify a scene with quite high accuracy [6, 8]. However, they also have high computational demands because the inner product between each pixel and each support vector must be computed. Computational demands increase when a non-linear kernel function is incorporated into the inner product.

The research leading to these results has received funding from the NO Grants 2014 – 2021, under Project ELO-Hyp, contract no. 24/2020, and the Research Council of Norway though the AMOS center grant number 223254.

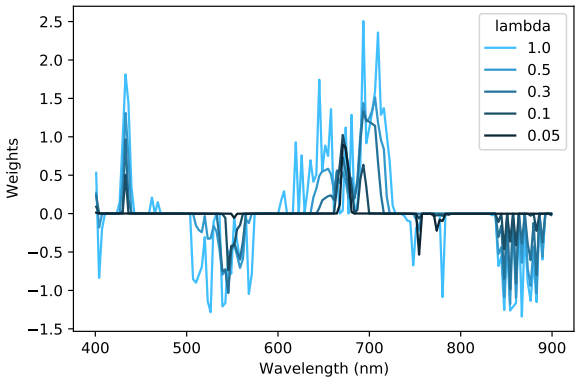

Fig. 1. The sparsity of the support vector which separates the soil class from the other classes in the Samson dataset tuned by varying lambda in the L_1-regularized objective function.

In this paper, SVMs are adapted for on-board processing by using sparsity, via the L_1-norm, to restrict the number of bands (Figs. 1 and 2). Essentially, if a sparse support vector only uses b out of the total p bands, then the algorithm is accelerated by a factor of p/b. Previous applications of sparse SVMs to hyperspectral data have focused its utility as a tool for bands selection and computation reduction [9]. Linear SVM, rather than the more common radial basis function SVM, is used because it does not require the exponential floating point operation, which is relatively slow [6]. As in [9], the objective function is solved in its primal rather than dual form. A one-versus-rest decision function is used to reduce the total number of computations. The SVMs are tested on two simple hyperspectral scenes, Samson and Jasper [10].

2. TRAINING A SPARSE SVM

In this section, the sparse SVM problem formulation is recast into a least-squares problem and the algorithm used to solve it is introduced. Then, the training dynamics and the effects of metaparameters are described (Fig. 3). Only execution of the SVM must be on-board. The training of the support vectors

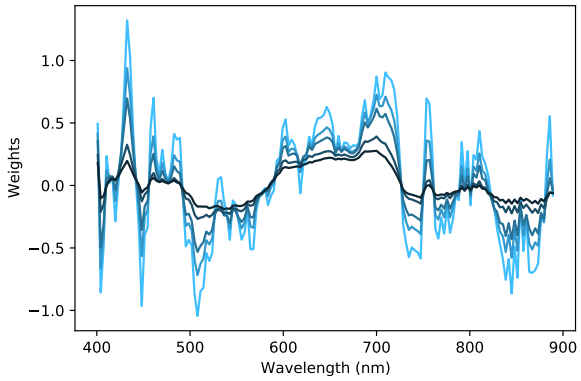

Fig. 2. The support vectors which separate the soil class from the other classes in the Samson dataset show minimal sparsity for varying lambda in the L_2-regularized objective function.

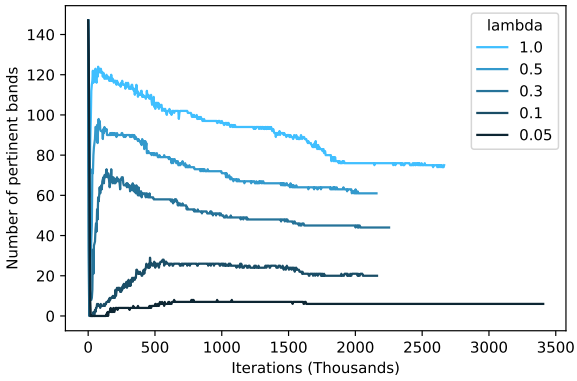

Fig. 3. The evolution of the number of bands with a weight > 0.1 in the support vector separating the soil from the other classes during training, for several different values of λ.

can be done on the ground, without significant computational limits.

2.1. Problem formulation

Consider a dataset, $\{(\mathbf{z}_i, y_i)\}_{i=1}^N$ of N spectra, $\mathbf{z}_i$, each with p bands and a class label y_i. Then, the sparse linear SVM classification objective function is:

$$\min_{w,d,u} \ \lambda \sum_{i=1}^{N} u_i + \|w\|_1 \qquad (1)$$
$$\text{subject to}: \ y_i(w^T z_i + d) \geq 1 - u_i, \quad u_i \geq 0 \quad \forall i,$$

where $w \in \mathbb{R}^p$ is the support vector, $d \in \mathbb{R}$ is the offset, u_i is a slack variable, and λ specifies the importance of classification accuracy relative to sparsity. The variables w and d parameterize hyperplane $w^T z + d = 0$ that separates the classes. Problem (1) can be recast as a linear programming problem [11] and then as a least-squares problem [12]:

$$\text{find } x \in \mathcal{Y}: \ Ax = b, \ Cx \leq d, \qquad (2)$$

where, by considering Y as a vector containing all y_i's and Z be the matrix containing z_i vectors, one can compute the following:

$$x \in \mathcal{Y} = [0, \infty)^{2(p+N+1)}, \qquad (3)$$
$$A = [\mathbf{1}_{2p}^T, \ \lambda \cdot \mathbf{1}_N^T, \ 0, \ 0, \ -\mathbf{1}_N^T], \qquad (4)$$
$$C = \begin{bmatrix} \mathbf{C} & \mathbf{0}_N \\ \mathbf{0}_{2p+N+2} & -\mathbf{C}^T \end{bmatrix}, \qquad (5)$$

where, $\mathbf{C} = [-Y \circ Z^T, \ Y \circ Z^T, \ \mathbf{I}_N, \ -Y, \ Y]$. Here $\mathbf{1}$ denotes a vector of all ones, $\mathbf{0}, \mathbf{I}$ denote the zero and identity matrix, respectively, and $\circ$ is the Hadamard product (elementwise multiplication). In this context, A is a vector of size $2(p + N + 1)$ and C is a matrix of size $2(p + N + 1) \times 2(p + N + 1)$. Moreover, this problem (2) can be reformulated equivalently as an optimization problem with functional constraints:

$$\min_{x \in \mathcal{Y}} \ f(x) \qquad (6)$$
$$\text{subject to} \ \ C_\xi^T x - d_\xi \leq 0 \quad \forall \xi \in \Omega_2,$$

where C_ξ^T are (block) rows partition of matrix C, and $f(x) = \frac{1}{2}(Ax - b)^2$.

2.2. Stochastic subgradient projection algorithm for constrained least-squares

Recently, a stochastic subgradient projection algorithm for constrained least-squares (SSP-LS) problem (6) was proposed by two of the authors [12]. The SSP-LS algorithm considers a independent random variable ξ_k sampled from probability distribution $\mathbf{P}_1$. At each iteration, it performs a first proximal subgradient step to minimize the objective function and then a second subsequent stochastic subgradient step to minimize the feasibility violation of the sampled constraint.

Algorithm 1 SSP-LS

Require: $x_0 \in \mathcal{Y}$, $\alpha > 0$, $\beta \in (0, 2)$
  for $\delta >$ tol **do**
    Sample $\xi_k \sim \mathbf{P}_1$
    $v_k \leftarrow x_k - \alpha A^T(Ax_k - b)$
    $z_k \leftarrow (1 - \beta)v_k + \beta \Pi_{\mathcal{C}_{\xi_k}}(v_k)$
    $x_{k+1} \leftarrow \Pi_{\mathcal{Y}}(z_k)$
    compute δ
  end for

Here, $\Pi_{\mathcal{Y}}(\cdot)$ denotes the projection onto $\mathcal{Y}$ and $\delta = \max(|Ax - b|, \|\max(0, Cx - d)\|)$. The probability distribution can be chosen e.g., dependent on the (block) rows of matrix C:

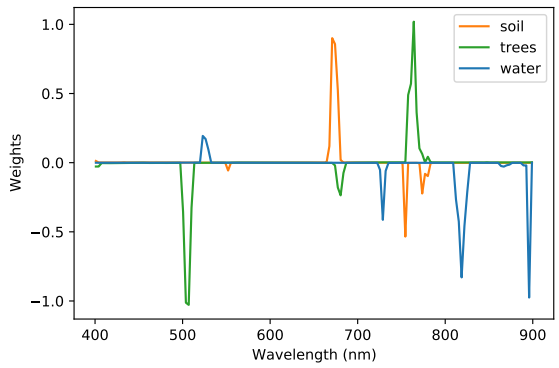

Fig. 4. The support vectors separating the Samson classes, trained with $\lambda = 0.05$.

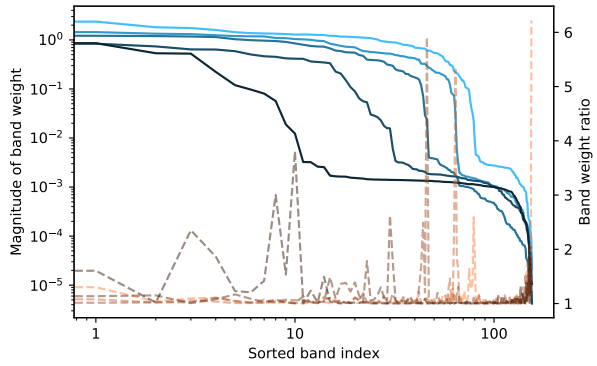

Fig. 5. The ordered magnitude of the band weights for the soil-vs-rest support vectors trained with varied λ (solid, left). The ratio of two subsequent weights, peaks indicate the number of relevant bands (dashed, right).

$$\mathbf{P}_1(\xi = \xi_k) = \frac{\|C_{\xi_k}\|_F^2}{\|C\|_F^2},$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. It has been proved in [12] that this stochastic algorithm converges linearly.

2.3. Training Dynamics

The SSP-LS algorithm successfully converged for the Samson and Jasper hyperspectral datasets, trained with 20% of the labelled data. The number of bands contributing to the support vector (magnitude greater than 0.1) showed non-monotonic behavior during training (Fig. 3). At initialization, all the bands contributed, but a minimum appeared after several hundred iterations. After a few thousand iterations, another local maximum in the number of contributing bands appeared, followed by a slow decrease. Thus, it is critical to give SSP-LS sufficient time to converge.

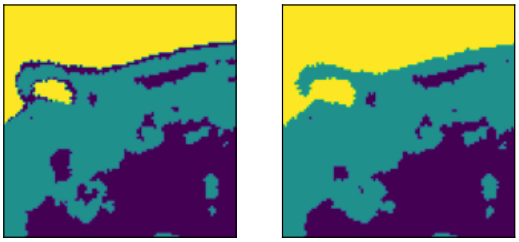

Fig. 6. The true Samson labels (left) compared to those found with the $\lambda = 0.05$ SVM (right): water (yellow), soil (purple), trees (teal).

Only the parameter λ, which controls the sparsity, affected the training dynamics much. If set below 0.1, the training required notably more time. The step size, α affected the number of required iterations little, while β near 1 slowed the algorithm, but did not alter the overall sequence. For the experiments shown here, $\alpha = 0.5$ and $\beta = 1.96$. The same general behavior was seen for training the support vector for each class, although the final number of pertinent bands and required iterations varied.

3. SPARSITY AND ACCURACY

The SSP-LS algorithm trained sparse support vectors for all the classes in both the Samson and Jasper images using a one versus rest decision function (Fig. 4). The mean was removed and the standard deviation was normalized before processing. The ordered magnitude of the weights of the support vector contains multiple flat regions, as also observed by [9], which verifies that the support vectors truly are sparse (Fig. 5).

norm	λ	Soil	Trees	Water	Overall
l1	1.00	0.948	0.995	0.994	0.979
	0.50	0.929	0.994	0.997	0.973
	0.30	0.917	0.993	0.999	0.970
	0.10	0.866	0.996	1.000	0.954
	0.05	0.838	0.995	1.000	0.944
l2	1.00	0.961	0.995	0.989	0.982
	0.50	0.956	0.995	0.991	0.981
	0.30	0.949	0.995	0.993	0.979
	0.10	0.929	0.993	0.995	0.972
	0.05	0.914	0.993	0.998	0.968

Table 1. Accuracy of SVMs applied to Samson

The accuracy of these support vectors is then compared in Table 1 to linear support vectors trained with L_2 regularization using the *liblinear* library [13]. Sparsity only imparts a small penalty on the accuracy of the classification, and although it grows with decreasing λ, it still produces a clear image (Fig. 6).

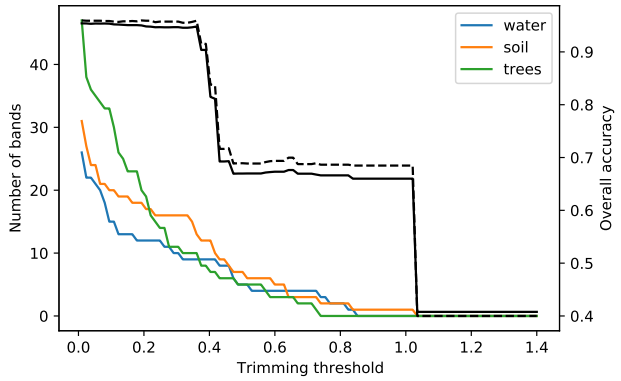

Fig. 7. The number of non-zero bands in the support vectors (left axis) and the resulting accuracy on the training (dashed) and test (solid) data, trained with $\lambda = 0.1$.

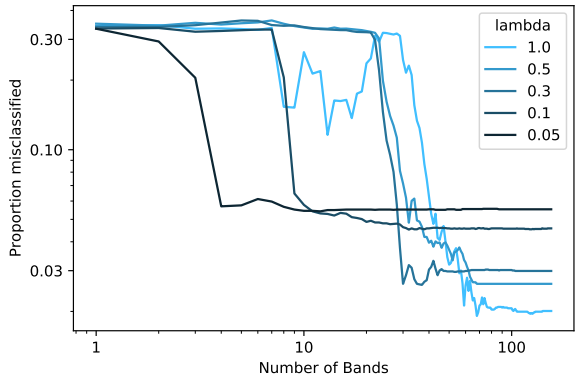

Fig. 8. The overall accuracy as a function of the number of bands retained in each support vector, for each λ setting.

norm	λ	Trees	Water	Dirt	Road	Overall
l1	1.00	0.948	0.994	0.968	0.907	0.974
	0.50	0.977	0.997	0.967	0.899	0.976
	0.30	0.977	0.998	0.968	0.884	0.975
	0.10	0.975	0.999	0.960	0.862	0.971
	0.05	0.978	1.000	0.942	0.827	0.965
l2	1.00	0.975	0.993	0.964	0.914	0.974
	0.50	0.975	0.993	0.963	0.911	0.974
	0.30	0.975	0.994	0.962	0.907	0.973
	0.10	0.975	0.994	0.963	0.901	0.973
	0.05	0.975	0.996	0.962	0.886	0.972

Table 2. Accuracy of SVMs applied to Jasper

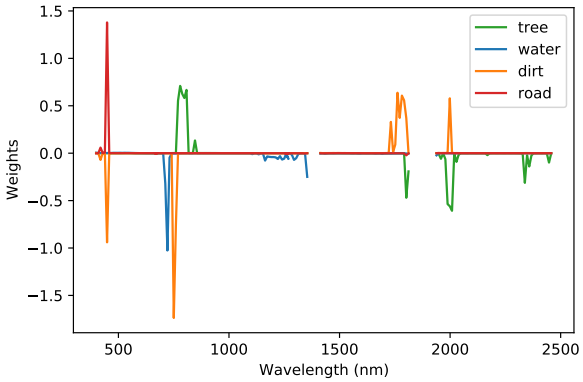

Fig. 9. The support vectors found for the classes in Jasper with $\lambda = 0.05$.

To utilize the improved speed associated with the sparse vectors, it is necessary to pick how many bands to set to 0, so that they can be ignored at execution. In [9], the number of ordered bands is chosen by (i) ordering all the bands by the magnitude of their weight in the support vector, then (ii) selecting only those bands which occur in the ordering before the steepest decrease in magnitude. However, unlike that previous study, there were multiple disjoint local maxima and the slope only reached values less than 10, rather than 10^5.

Two different strategies were tested for selecting bands: picking a fixed threshold or fixed number of bands. Thresholding, in which all bands below a set threshold were set to zero, resulted in a different number of bands for each support vector (Fig. 7). The accuracy remained quite stable over a wide range of thresholds, even as the number of retained bands changed. Steep drops of over 20% reduction in accuracy occurred at a few points. Surprisingly, the accuracy increased as the number of bands decreased at a few threshold values.

Selecting the M bands with the largest weights for each support vector allowed for simpler comparison between support vectors with different sparsity (Fig. 8). As the sparsity increases (decreasing λ), the vectors achieve their maximum accuracy with fewer bands, as few as 4. However, as the total number of bands is increased, the less sparse support vectors perform better.

When the experiments were repeated on the Jasper hyperspectral image, sparse support vectors were again found for every class (Fig. 9). The overall accuracy of the sparse support vectors actually exceeded that of the L_2-regularized vectors for a few values of λ, unlike on the Samson scene (Table 2). The general form of the accuracy as a function of the number of non-zero bands was similar to the Samson scene, although a bit less structured (Fig. 10).

4. CONCLUSIONS

Sparse support vectors achieve performance similar to standard linear support vectors. The degree of sparsity can be controlled and for simple scenes such as Samson and Jasper it is possible to achieve around 95% overall accuracy using

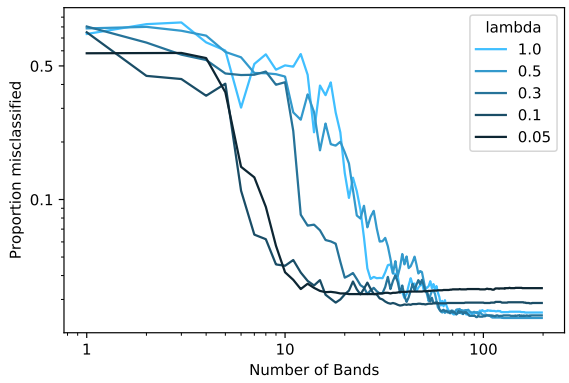

Fig. 10. The overall accuracy as a function of the number of bands retained in each support vector, for each λ setting.

just a few percent of the total bands for each support vector, corresponding to more than $20\times$ increase in the execution speed. These bands are different for each of the support vectors, which indicates that the detailed spectral information is still being used.

However, most SVM libraries are not suited for executing with a different subset of bands for each comparison, and, even if they were, switching between subsets would entail some additional computational cost. This difficulty could be avoided if an algorithm were developed such that the same subset of bands were used for each support vector.

Many SVMs applied to hyperspectral data utilize kernels, which were not explored here. While kernels enable SVMs to classify more complex data sets, they also vastly increase the cost of execution. Further research could investigate whether the methods described above could be used to accelerate SVMs using radial basis function or polynomial kernels.

The code used to train and execute these SVMs can be found on the ELO-Hyp github page: github.com/ELO-Hyp/.

5. REFERENCES

[1] Y Zhong, X Wang, Y Xu, S Wang, T Jia, X Hu, J Zhao, L Wei, and L Zhang, "Mini-UAV-borne hyperspectral remote sensing: From observation and processing to applications," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 4, pp. 46–62, 2018.

[2] R H Becker, M Sayers, D Dehm, R Shuchman, K Quintero, K Bosse, and R Sawtell, "Unmanned aerial system based spectroradiometer for monitoring harmful algal blooms: A new paradigm in water quality monitoring," *Journal of Great Lakes Research*, vol. 45, no. 3, pp. 444–453, 2019.

[3] M Abdellatif, H Peel, A G Cohn, and R Fuentes, "Hyperspectral imaging for autonomous inspection of road pavement defects," in *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*. International Association for Automation and Robotics in Construction, 2019, pp. 384–392.

[4] R Booysen, R Jackisch, S Lorenz, R Zimmermann, M Kirsch, P A Nex, and R Gloaguen, "Detection of rees with lightweight UAV-based hyperspectral imaging," *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.

[5] B Raeissi, M A Bashir, J L Garrett, M Orlandic, T A Johansen, and T Skramstad, "Detection of different chemical binders in coatings using hyperspectral imaging," *Journal of Coatings Technology and Research*, vol. 19, no. 2, pp. 559–574, 2022.

[6] A Alcolea, M E Paoletti, J M Haut, J Resano, and A Plaza, "Inference in supervised spectral classifiers for on-board hyperspectral imaging: An overview," *Remote Sensing*, vol. 12, no. 3, 2020.

[7] G Camps-Valls, D Tuia, L Bruzzone, and J A Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE signal processing magazine*, vol. 31, no. 1, pp. 45–54, 2013.

[8] F Melgani and L Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[9] S Chepushtanova, C Gittins, and M Kirby, "Band selection in hyperspectral imagery using sparse support vector machines," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX*. SPIE, 2014, vol. 9088, pp. 432–446.

[10] F Zhu, Y Wang, B Fan, S Xiang, G Meng, and C Pan, "Spectral unmixing via data-guided sparsity," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5412–5427, 2014.

[11] C Bhattacharyya, L Grate, A Rizki, D Radisky, F Molina, M Jordan, M Bissell, and I Mian, "Simultaneous classification and relevant feature identification in high-dimensional spaces: Application to molecular profiling data," *Signal Processing*, vol. 83, no. 4, pp. 729–743, Apr. 2003.

[12] I Necoara and N K Singh, "Stochastic subgradient for composite convex optimization with functional constraints," *arXiv*, vol. 2204.08204, 2022.

[13] R E Fan, K W Chang, C J Hsieh, X R Wang, and C J Lin, "Liblinear: A library for large linear classification," *the Journal of machine Learning research*, vol. 9, pp. 1871–1874, 2008.