Marek Pecha; Zachary Langford; David Horák; Richard Tran Mills
Wildfires identification: Semantic segmentation using support vector machine classifier

In: Jan Chleboun and Pavel Kůs and Jan Papež and Miroslav Rozložník and Karel Segeth and Jakub Šístek (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Jablonec nad Nisou, June 19-24, 2022. Institute of Mathematics CAS, Prague, 2023. pp. 173–186.

Persistent URL: http://dml.cz/dmlcz/703198

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

WILDFIRES IDENTIFICATION: SEMANTIC SEGMENTATION USING SUPPORT VECTOR MACHINE CLASSIFIER

Marek Pecha[1,2], Zachary Langford[3], David Horák[1,2], Richard Tran Mills[4]

[1] VŠB–Technical University of Ostrava
17. listopadu 2172/15, Ostrava–Poruba, Czech Republic
[2] Institute of Geonics, Czech Academy of Sciences
Studentská 1768/9, Ostrava–Poruba, Czech Republic
marek.pecha@vsb.cz, david.horak@vsb.cz
[3] Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, United States
langfordzl@ornl.gov
[4] Argonne National Laboratory
9700 S Cass Ave, Lemont, IL, United States
rtmills@anl.gov

Abstract: This paper deals with wildfire identification in the Alaska regions as a semantic segmentation task using support vector machine classifiers. Instead of colour information represented by means of BGR channels, we proceed with a normalized reflectance over 152 days so that such time series is assigned to each pixel. We compare models associated with $\ell 1$-loss and $\ell 2$-loss functions and stopping criteria based on a projected gradient and duality gap in the presented benchmarks.

Keywords: wildfire identification, semantic segmentation, support vector machines, distributed training

MSC: 68T09, 68T45, 68W15

1. Introduction

Global climate change is increasing the frequency and intensity of ecological disturbance; this is particularly true in high latitudes, where projects such as the NASA ABoVE project (`https://above.nasa.gov`) are working to understand the effects of increased climate-driven disturbances. Wildfires are one important source of disturbance, and can significantly affect forest carbon balance. Despite their importance, however, it can be difficult to accurately quantify the effects of wildfire in places such as boreal forests that are far from human habitation and infrastructure. Data from remote sensing platforms and observatory networks can be of great use of this task,

DOI: 10.21136/panm.2022.16

but these data sets can be vast, and analyzing them can require powerful computing resources and tools that are designed to fully utilize them.

Popular methods are based on machine learning approaches including deep learning [8], where U-Net architecture or inception networks are typically used. In this paper, we discuss an alternative approach for wildfire identification in the Alaska regions using semantic segmentation that support vector machine classifiers are exploited. Instead of colour information, we assign changes of normalized reflectance over time to each pixel so that corresponding attributes are represented by time series with 8-day period. Pixels are then categorized using the Monitoring Trends in Burn Severity product. Additionally, we study the influence of stopping criteria on model performance and training time on benchmarks presented in Section 3.2.

2. Support Vector Machines

Support Vector Machines is a set of methods belonging to supervised learning algorithms used for classification, regression, or outliers detection. Since wildfire identification is essentially a binary classification task, i.e. we have to decide if an area is affected by fire or not, we will focus on formulations associated with classification approaches employing SVMs in this paper. Considering underlying structures related to SVMs, we can see them as a single perceptron that finds a learning function (called model in the machine learning community) maximizing a geometric margin between (training) samples and a discriminant hyperplane. This implicit ability guarantees a generalization performance of the model, which can be described by means of a particular case of the Tikhonov regularization in the following form:

$$\underset{f\in\mathscr{H}}{\arg\min}\ \ m^{-1}\sum_{i=1}^{m} V\left(y_i, f\left(\boldsymbol{x}_i\right)\right)^2 + \lambda\|f\|_{\mathscr{H}}^2, \tag{1}$$

where $\mathscr{H}$ is a hypothesis space of functions, $\|\cdot\|_{\mathscr{H}}$ is a norm on the hypothesis space, $f\colon \mathbb{R}^n \to Y$ denotes mapping data (m training samples) to a label space, $V\colon Y \to Y$ is a loss function, and $\lambda \in \mathbb{R}$ is a regularization parameter such that $\lambda = \frac{1}{2C}$. Moreover, this theoretical framework provides us with an explanation related to the regularization perspective of the SVM models so that a trade-off between bias and variance is driven by parameter C.

In the following part of this paper, we introduce certain C-SVM formulations that are associated with classification tasks concerning non-linearly separable (training) samples and their relaxed-bias versions, where a bias term is considered as a scaled parameter and included in an optimization problem by means of augmenting the normal vector of a hyperplane $\boldsymbol{w}$ and samples with an additional dimension.

2.1. Soft maximum-margin classifier

Let us start with a standard SVM formulation introduced by Vapnik et al. in [3]. It was initially developed as a supervised binary classifier, i.e. an algorithm that determines a function (model), which maps a training sample to a label (related

to 2 categories in this case) such that it adapts itself to unseen data drawn from the same distribution as the training ones. This essential model ability is called generalization.

To describe the training phase of the SVM classifier more in detail, let us firstly denote the training data set as follows:

$$T := \{(\boldsymbol{x}_1, y_1)\,,\ (\boldsymbol{x}_2, y_2)\,,\ \ldots,\ (\boldsymbol{x}_m, y_m)\}, \tag{2}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ ($n \in \mathbb{N}$) is an i-th sample and $y_i \in \{-1, 1\}$ is its label, m is a number of training samples. Further, let us consider that the samples are linearly separable, i.e. it exists a separating hyperplane between the clusters of samples belonging to these two categories. A model of a linear SVM is then represented in the form of a maximum-margin hyperplane H so that:

$$H = \langle \boldsymbol{w}, \boldsymbol{x} \rangle - \hat{b}, \tag{3}$$

where $\boldsymbol{w}$ is a normal vector of the hyperplane H, and $\hat{b} = \frac{b}{\|\boldsymbol{w}\|}$ is a scalar called a bias term that determines an offset in a direction of $\boldsymbol{w}$, or $-\boldsymbol{w}$ in a case when $\hat{b}$ is negative. Let us denote a bias $\hat{b}$ as b for a more convenient notation in equations in the following text. Remark that the maximum margins are defined by means of locations associated with support vectors, and the width between these margins is equal to $\frac{2}{\|\boldsymbol{w}\|}$.

Maximizing the distance d corresponds to regularization of the weights $\boldsymbol{w}$, which is basically the prevention of overfitting a model to the training data set T. Regarding the constraints arising from geometric margins, we can write an optimization problem for finding a normal vector $\boldsymbol{w}$ and a bias b as follows:

$$\underset{\boldsymbol{w},\ b}{\arg\min}\ \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle\ \text{ s.t. }\ \begin{cases} y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b\right) \geq 1, \\ i \in \{1, 2, \ldots, m\}, \end{cases} \tag{4}$$

the constraint $y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b\right) \geq 1$ can be interpreted so that all training categorical samples must lie on or above corresponding margins equal to -1 and 1, respectively. Note, a solution of the optimization problem (4) exists only when the training samples T are linearly separable. To sort out the separability issue, we can exploit the soft-margin SVM [3]. An idea beyond the approach is based on adding an auxiliary (regularization) term to (4), particularly, $C\sum_{i=1}^{m}\xi_i$ [1], and, also, an additional relaxation of the constraints related to the margins such that:

$$\underset{\boldsymbol{w},\ b,\ \xi_i}{\arg\min}\ \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C\sum_{i=1}^{m}\xi_i\ \text{ s.t. }\ \begin{cases} y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b\right) \geq 1 - \xi_i, \\ \xi_i \geq 0,\ i \in \{1, 2, \ldots, m\}, \end{cases} \tag{5}$$

[1] The term $C\sum_{i=1}^{m}\xi_i$ regularises misclassification errors and restricts the complexity of the classifier in sense of overfitting a classification model.

where $\xi_i := \max\{0, 1 - [\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b]\}$. Essentially, the function quantifies the error between the predicted and correct sample classification $\boldsymbol{x}_i$. If sample $\boldsymbol{x}_i$ is correctly classified, a value of the hinge loss function equals 0. In order of sample misclassification, a value of hinge loss function is proportional to the distance between the respective margin and a misclassified sample.

The parameter C is a user-defined penalty, which determines the influence associated with the misclassification of samples on the objective function. Generally, a higher value of C increases the importance of minimizing the hinge loss functions ξ_i and also maximizing $\|\boldsymbol{w}\|$. This leads to minimizing the width of the margin and may cause overfitting of a classifier to a training data set consequently. It means a model has a high variance. A smaller value of the penalty C results in a wider margin that may cause a large number of misclassifications, i.e. a high bias of a model[2]. The goal is to find a reasonable value of C such that a resulting model balances a bias-variance tradeoff. Typically, the value is determined using hyperparameter optimization techniques, e.g. grid-search combined with cross-validation.

To reduce the number of unknowns and employ our approach based on a deterministic approach that uses the MPRPG [4] as an underlying solver, it stands for Modified Proportioning with Reduced Gradient Projection, we can modify the primal formulation (5) so that it turns into an optimization problem with the following structure:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha} \in \Omega} \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{A} \boldsymbol{\alpha} - \boldsymbol{b}^T \boldsymbol{\alpha}, \tag{6}$$

where Ω is a convex closed set defined by means of box constraints $\Omega := \{\boldsymbol{\alpha} \in \mathbb{R}^m \mid \boldsymbol{u} \leq \boldsymbol{\alpha} \leq \boldsymbol{l}\}$. Practically, we can obtain a formulation analogous to the structure (6) by dualizing primal formulation (5) such that:

$$\arg\min_{\boldsymbol{\alpha}} \ \frac{1}{2} \boldsymbol{\alpha}^T \underbrace{\boldsymbol{Y}^T \boldsymbol{K} \boldsymbol{Y}}_{\boldsymbol{G}} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{1} \ \text{ s.t. } \begin{cases} \boldsymbol{o} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \\ \boldsymbol{y}^T \boldsymbol{\alpha} = 0, \end{cases} \tag{7}$$

where $\boldsymbol{Y} = diag(\boldsymbol{y})$, $\boldsymbol{y} = [y_1,\ y_2,\ \ldots,\ y_m]^T$, and $\mathbf{1} = [1,\ 1,\ \ldots,\ 1] \in \mathbb{R}^m$. $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ is the SPS (Symmetric Positive Semi-definite) matrix of inner products called the Gram matrix such that $\boldsymbol{K} = \boldsymbol{X}^T \boldsymbol{X}$, where $\boldsymbol{X} = [\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots, \boldsymbol{x}_m]$. $\boldsymbol{G}$ denotes the Hessian matrix, which is SPS either. Exploiting a derivative of the Lagrangian with respect to $\boldsymbol{\xi}$, we can determine the vector of Lagrange multipliers $\boldsymbol{\beta}$ so that $\boldsymbol{\beta} = C\mathbf{1} - \boldsymbol{\alpha}$, thus $\boldsymbol{\beta}$ does not occur in (7). This formulation is called dual ℓ1-loss.

For obtaining a solution of the original (primal) problem, we introduce dual to primal reconstruction formulas as follows:

$$\boldsymbol{w} = \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha}, \tag{8}$$

[2]In this case, the term bias corresponds to a systematic error arising from wrong assumptions that may lead to missing relevant relations between features and labels caused by means of a low capability of a model.

associated with the normal vector of the separating hyperplane, and the bias is reconstructed by means of:

$$b = \frac{1}{\operatorname{card}(J)} \left(\boldsymbol{X}_{*J}^T \boldsymbol{w} - \boldsymbol{y}_J \right) \mathbf{1}_J^T, \tag{9}$$

where $J = \{i \mid 0 < \alpha_i < C,\ i = 1, 2, \ldots, k\}$ is the support vector index set, $\operatorname{card}(J)$ presents its cardinality, $\boldsymbol{X}_{*J}$ denotes the submatrix of the matrix $\boldsymbol{X}$ with the column indices belonging to J; $\boldsymbol{y}_J$ and $\mathbf{1}_J$ are subvectors of the vectors $\boldsymbol{y}$ and $\mathbf{1}$, respectively. Using the reconstructed normal vector $\boldsymbol{w}$ and bias b, we set the decision rule:

$$\operatorname{sgn}\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\right) = \begin{cases} +1 \ldots\ \boldsymbol{x}_i \in \text{Class A}, \\ -1 \ldots\ \boldsymbol{x}_i \in \text{Class B}. \end{cases} \tag{10}$$

2.2. Hessian matrix regularization

The Hessian matrix $\boldsymbol{G}$ corresponding to the dual formulation (7) is SPS, which implies the underlying optimization problem has a non-unique solution. In this subsection, we modify the primal formulation (5) in such a way that the Hessian in dual formulation becomes SPD (Symmetric Positive Definite) [10, 9]. It implies that the resulting optimization problem is strictly convex, and its solution is unique. An idea beyond the adjustment is based on substitution $\ell 1$-norm of loss function by the $\ell 2$-norm, i.e. the squared loss function, in the objective function so that (7) results into the following form:

$$\underset{\boldsymbol{w},\ b,\ \xi_i}{\arg\min} \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + \frac{C}{2} \sum_{i=1}^{m} \xi_i^2 \ \text{ s.t. } \begin{cases} y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \right) \geq 1 - \xi_i, \\ i \in \{1, 2, \ldots, m\}. \end{cases} \tag{11}$$

Analysing the formulation above, we can simply observe the term that quantifies misclassification error $\sum_{i=1}^{m} \xi_i^2 \geq 0$. Therefore, we do not consider $\xi_i \geq 0$ as a constraint. The formulation (11) is called the primal $\ell 2$-loss SVM. As in the case of the $\ell 1$-loss SVM, we derive a dual formulation. Using the Lagrange duality and evaluating the Karush–Kuhn–Tucker conditions, the primal formulation (11) transforms into the dual one so that for any $C > 0$:

$$\underset{\boldsymbol{\alpha}}{\arg\min}\ \frac{1}{2} \boldsymbol{\alpha}^T \left(\boldsymbol{G} + C^{-1} \boldsymbol{I} \right) \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{1} \ \text{ s.t. } \begin{cases} \mathbf{0} \leq \boldsymbol{\alpha}, \\ \boldsymbol{y}^T \boldsymbol{\alpha} = \mathbf{0}. \end{cases} \tag{12}$$

While the Hessian $\boldsymbol{G}$ is regularized by a matrix $C^{-1}\boldsymbol{I}$, it avoids linear dependency of columns also arising from possible multicollinearity of the training samples. Then, the matrix becomes full-rank SPD. The optimization problem and the quality of its solution are practically data-driven, i.e. highly dependent on the data nature. Therefore, we can say precisely that the associated optimization problem could be more computationally stable, and a convergence rate of an underlying solver could

be faster than in the case of the $\ell 1$-loss SVM. On the other hand, $\ell 1$-loss SVM could produce a sparse and more robust model in the sense of performance score. Then, we adapt the support vector index set J such that:

$$J = \{i \mid 0 < \alpha_i,\ i = 1, 2,\ \ldots, k\} \tag{13}$$

for the reconstruction formulas (8), (9) related to normal vector $\boldsymbol{w}$ of hyperplane H and bias b, respectively.

2.3. Relaxed-bias approaches

The standard soft-margin SVM solves the problem of finding a classification model in the form of the maximal-margin hyperplane (3). In the case of the relaxed-bias classification [7], we do not consider the bias b in a classification model. However, we include it into the problem by means of augmenting the vector $\boldsymbol{w}$ and each sample $\boldsymbol{x}_i$ with an additional dimension so that $\widehat{\boldsymbol{w}} \leftarrow \begin{bmatrix} \boldsymbol{w} \\ B \end{bmatrix}$, $\widehat{\boldsymbol{x}}_i \leftarrow \begin{bmatrix} \boldsymbol{x}_i \\ \gamma \end{bmatrix}$, where $\gamma \in \mathbb{R}^+$ is a user-defined variable, which is typically set to 1. Let $p \in \{1, 2\}$ then the problem of finding a hyperplane $\widehat{H} = \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \rangle$ can be formulated as a constrained optimization problem in the following primal formulation:

$$\underset{\widehat{\boldsymbol{w}},\ \xi_i}{\arg\min}\ \frac{1}{2}\langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \rangle\ +\ \frac{C}{p}\sum_{i=1}^{n} \widehat{\xi}_i^{\,p}\ \text{ s.t. }\ \begin{cases} y_i \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}}_i \rangle \geq 1 - \widehat{\xi}_i, \\ \widehat{\xi}_i \geq 0 \ \text{ if } p = 1,\ i \in \{1, 2, \ldots, n\}, \end{cases} \tag{14}$$

where $\widehat{\xi}_i = \max\{0,\ 1 - y_i \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}}_i \rangle\}$ is the hinge loss function related to augmented samples $\widehat{\boldsymbol{x}}_i$. Generally, we can say the minimizer associated with formulation (14) corresponding to a rotation of the separating hyperplane $\widehat{H} \in \mathbb{R}^n$ in a one-dimension higher feature-space $\mathbb{R}^{n+1}$ such that the maximizing of geometric margins are satisfied.

3. Wildfire identification as semantic segmentation task

Semantic segmentation is a computer vision task for which most recent methods are based on deep learning approaches, where neural networks of U-Net type architectures are typically used. Actually, semantic segmentation is associated with image classification at a pixel level. It means that every pixel is assigned to a category such that an image segmentation mask is created. In common semantic segmentation, labelled colour images with the BGR (blue-green-red) channel order are used as inputs. The pre-trained encoder of the U-Net extracts features and patterns from spatial images, and the decoder projects these lower resolution feature onto the pixel space in higher resolution to get a dense classification.

We show up an alternative approach that exploits a spectral reflectance corrected for the atmospheric condition instead of colour information. An essential idea of these corrections follows up simulating the propagation of electromagnetic

waves in a geogas system to obtain surface reflectance without emission, e.g. remove a contribution of atmospheric aerosol scattering.

To estimate a spectral surface reflectance corresponding to 500 m spatial resolution at a pixel, we use the MODIS (Moderate Resolution Imaging Spectroradiometer) instrument that data was extracted by the Google Earth Engine running at cloud https://earthengine.google.com. Essentially, the reflectance is a ratio of reflected energy to incident radiation $\frac{\phi_r}{\phi}$ as a function of the wavelength. The MODIS product called MOD09A1 (https://modis.gsfc.nasa.gov/data/dataprod/mod09.php) provides 7 bands associated with this electromagnetic spectrum ranging from 459 nm to 2155 nm as an 8-day composite. To describe a region affected by fire, we study changes in normalized reflectance over time periods so that features corresponding to each pixel are represented by time series related to the 7 bands mentioned above with an 8-day period. The pixels are then categorized using boundaries collected from Monitoring Trends in Burn Severity (https://www.mtbs.gov/). Such samples are being classified using SVM implemented in our in-house software PermonSVM.

3.1. PermonSVM: Classification tool based on PETSc framework

The PermonSVM package [11] is a part of the PERMON toolbox . This toolbox is designed for usage in a distributed environment containing hundreds or thousands of computational cores. Technically, it is an extension of the core package called PermonQP [6], from which it inherits environment basic structures, initialization routines, a build system, and utilizes computational routines implemented in the core package PermonQP. Programmatically, a core functionality associated with the PERMON toolbox is written on top of the PETSc framework [1]. It follows the same design and coding style that makes it easy to use for anyone familiar with PETSc.

PermonSVM currently supports parallel reading of the SVMLight, HDF5, and PETSc binary file formats, solutions of more than 4 problem formulations of the related classification problems, k-fold and stratified k-fold cross-validation. The underlying QP problem related to SVM with implicitly represented the Hessian matrix, in which the Gram matrix $\boldsymbol{X}^T\boldsymbol{X}$ is not assembled, and is computed by means of solvers provided by the PermonQP package or PETSc framework. All PERMON modules are developed as open-source software under the BSD–2–Clause license.

3.2. Benchmarks

We present results associated with state-of-the-art investigating wildfire detection so that data was collected and processed in a way already mentioned above. In our experiments, we study wildfires in the Alaska regions in 2004. The wildfires across Alaska are the dominant disturbance, and creating frameworks for quantification is important to long-term scientific projects such as the U.S. Department of Energy project Next Generation of Arctic Ecosystem Experiments. The 2004 Alaska wildfire season was the worst on record in the U.S. state of Alaska in terms of area burned (27,000 km^2). Looking at Table 1, our toy data set used to present

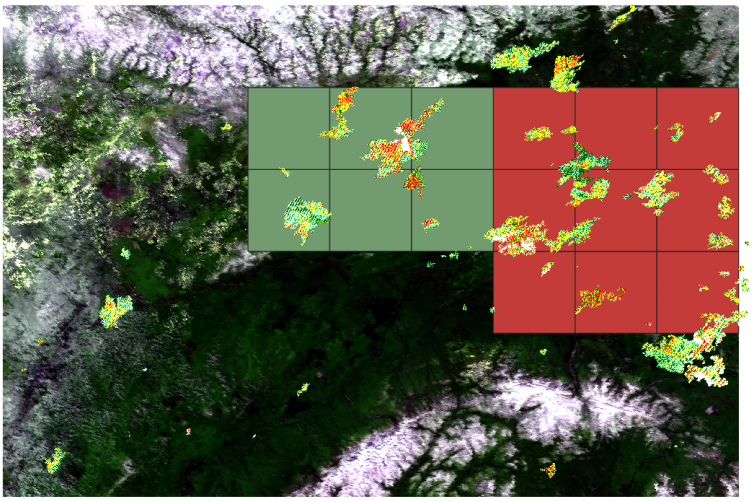

Figure 1: Areas in Alaska affected by wildfires that we model in our experiments. Red squares represent the training data set and green ones are related to test data set. Data are accumulated over 152 days from May to Semptember

state-of-the-art results contains $500,000$ samples split into training and test data sets consisting of $360,000$ and $240,000$ samples, respectively. These samples are associated with changing reflectance over 152 days from May to September.

We computed the following semantic segmentation results on KAROLINA, which is a combination of HPE Apollo 2000 and Apollo 6500 systems used for HPC workloads such as AI and other data-intensive applications, for example.

mod09ak_2004	#Wildfires	#Background	#Attributes
Training	$46,851$ (13.01%)	$313,149$ (86.99%)	133
Test	$28,351$ (11.81%)	$211,649$ (88.19%)	133

Table 1: The mod09ak_2004 data set description related to training and test ones. Proportions of classes in the data sets are pointed out as percents.

A critical part of any data-related pipeline is associated with stopping criteria. Choosing the right strategy to terminate an underlying optimization solver influences the quality of a resulting model. In our experiments, we explored an optimization solver called MPRGP employed in our classification problems that models were computed employing relaxed-bias formulations for $\ell 1$ and $\ell 2$ loss types presented in Section 2.3. An expansion of an active set was performed using a projected conjugate (CG) gradient, and $\Gamma = 100$ was set to determine proportionality. Misclassification errors were penalized with $C = 1$, i.e. a default value. A standard stopping criterion used in MPRGP involves a norm of projected gradient $\|g^p\|$ compared with a relative norm of a dual right-hand side $\boldsymbol{b}_{\text{dual}}$ as follows:

$$\|\boldsymbol{g}^p\| \leq \epsilon \|\boldsymbol{b}_{\text{dual}}\|. \tag{15}$$

However, this terminating condition does not take into account model quality. A reasonable approach could be based on monitoring a loss function and including it in a stopping criterion. In the case of SVM, we consider a specific type of a loss function called a hinge loss function $\xi := \max\{0, 1 - [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - b]\}$ defined in Section 2.1 and this term is incorporated in a primal functional. Moreover, we can prove there is no gap between primal and dual functional at its optimal solution for the case of the $\ell 2$-loss SVM formulation. It holds a strong duality. Regarding these properties, we can use stopping criteria based on a duality gap for the $\ell 2$-loss SVM as follows:

$$|p\left(\boldsymbol{w}, b, \boldsymbol{\xi}\right) - d(\boldsymbol{\alpha})\,| \leq \epsilon |p\left(\boldsymbol{w}, b, \boldsymbol{\xi}\right)|, \tag{16}$$

where

$$p\left(\boldsymbol{w}, b, \boldsymbol{\xi}\right) = \frac{1}{2}\left\langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \right\rangle \; + \; \frac{C}{2}\sum_{i=1}^{n} \widehat{\xi}_i^2, \tag{17}$$

and

$$d\left(\boldsymbol{\alpha}\right) = \frac{1}{2}\boldsymbol{\alpha}^T\left(\boldsymbol{G} + C^{-1}\boldsymbol{I}\right)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{1} \tag{18}$$

are a primal and a dual functional related to relaxed-bias $\ell 2$-loss SVM formulations, respectively; ϵ represents a relative tolerance. The attained results are summarized in Table 2 and Table 3.

Dataset	Loss	Stop. criteria	Hessian mult.	Loss val.	Train. time [s]
mod09ak_2004	$\ell 1$	(15)	2962	2.28e4	22.67
	$\ell 2$	(15)	1025	3.03e4	6.96
		(16)	1029	3.00e4	15.60

Table 2: Attained results using 64 MPI processes (KAROLINA). Solver: MPRGP so that an expansion step is performed using the projected CG step, $\Gamma = 100$ in proportion criterion, a relative tolerance ϵ was set to 0.1; penalty $C = 1$.

The overall performance of attained models does not significantly differ as measured by the $F1$ score, which is a harmonic mean of precision and sensitivity, as presented in Table 3. However, we can see that some models perform slightly better than others when we compare them using other metrics. Analyzing the influence of the proposed stopping criteria based on the duality gap on model scores, we can see that the $\ell 2$-loss model, when the training process was terminated using the condition (16), behaves slightly better both precision and sensitivity scores on a test data set than the $\ell 2$-loss model trained employing the MPRGP solver stopped by means of the terminate condition (15).

Dataset	Loss	Stopping criteria	precision [%]	sensitivity [%]	F1
mod09ak_2004	$\ell 1$	(15)	84.12	94.58	0.89
	$\ell 2$	(15)	83.58	92.81	0.89
		(16)	85.33	93.13	0.89

Table 3: Influence of stopping criteria on the model performance scores on the test data set.

As we mentioned in Section 2.2, the $\ell 1$-loss model could be a more robust in the sense of its performance than the one based on the $\ell 2$-loss function. It could be ambitious to make such a conclusion merely by looking at the performance scores since we can see that a precision score is higher for $\ell 2$-loss and, on the other side, sensitivity is higher for $\ell 1$ loss. Nevertheless, it differs in the value of loss functions, which represent overall misclassification errors, pointed out in Table 2. From this table, we can easily see that the $\ell 1$-loss-based model generalizes a training data set better than the $\ell 2$-loss-based one. Assuming the training times of each model, we can see that evaluating the time of stopping criteria (16) is time-consuming and almost 2 times slower than for (15), and training the $\ell 1$-loss model is nearly 3.3 times slower than for $\ell 2$-loss trained to employ the MPRGP solver terminated using the condition (15). From the observations above, it seems the $\ell 2$-loss model that its training was stopped exploiting the stopping criteria (16) could be a good trade-off among $\ell 1$-loss and $\ell 2$-loss models, when the stopping condition (15) was used in during the models training.

Dataset	Loss	$\sum$ Hes. mult.	Loss val.	Train. time [s]
mod09ak_2004	$\ell 1$	34622	1.80e5	73.82
	$\ell 2$	51967	2.24e5	112.28

Table 4: Solutions related to the complete SVM formulations using SMALXE + MPRGP. A default stopping condition is used. Results are attained using 64 MPI processes on KAROLINA. Setting of an inner solver: $\Gamma = 100$, a relative tolerance $\epsilon_{\text{inner}} = 0.1$; $\epsilon_{\text{outer}} = 1\text{e}-2$ and divtol $=$ 1e10 for an outer loop (SMALXE). Misclassification penalty $C = 1$.

The attained models presented above can be viewed as solutions related to a special case of the Tikhonov regularization (1) such that a bias term b is relaxed. This approach simplifies the SVM formulations (7), (12), i.e. the complete SVM formulations with bounds and equality constraints. It leads to problems that are numerically cheaply to solve than the original ones. We demonstrate computational demands on training models employing the complete SVM formulations in the following numerical experiments. We employed the Semimonotonic augmented Lagrangian (SMALXE) algorithm [5] that is "pass-through" solver taking care of equality constraints (a default stopping condition for SMALXE is used in the following numerical experiments). By this approach, we splitted (7), or (12) for $\ell 2$-loss case, into two

sub-problems such that an equality constraint and bounds are handled separately, one after another. An outer loop is performed using the augmented Lagrangians and bound constrained optimization problem is computed by means of an inner solver – MPRGP in our case. The results are summarized in Table 4 and Table 5.

Looking at elapsed times presented in Table 4, we can see that training a model is 3.26 times slower for the complete $\ell 1$-loss formulation (7) than in case of a relaxed formulation of this problem (14) (for $p = 1$). Moreover, we can observe that a value of a loss function (a quantification of misclassification error) is 7.89 times higher than its relaxed version. It is similar to training a model employing the complete $\ell 2$-loss formulation when a default stopping condition is exploited. A training time is 16.1 times slower, and a value associated with a loss function is 7.39 times higher.

Dataset	Loss	precision [%]	sensitivity [%]	F1
mod09ak_2004	$\ell 1$	82.80	96.18	0.89
	$\ell 2$	82.98	95.63	0.89

Table 5: The best perfomance scores of models trained employing the complete SVM formulations (on the test data set).

The performance scores of models on the test data set are summarized in Table 5. They do not significantly differ from the scores attained employing the relaxed versions of the SVM formulation in Table 3; however, a true positive rate (sensitivity) is slightly higher. It means that models identify fire occurrences (true positives) better than the ones with relaxed bias at the cost of decreasing precision, i.e. a false positive rate.

Dataset	Solver	precision	sensitivity	F1
mod09ak_2004	XGBoost	97.05	89.00	0.93

Table 6: Results attained using the XGBoost solver.

We compared the performance scores of attained classification models employing PermonSVM with a model trained by means of the XGBoost (eXtreme Gradient Boosting) solver [2]. It is based on a boosted tree method. The results are presented in Table 6. The overall scores measured by means of F1 score are higher for a model trained by XGBoost. However, PermonSVM produces models with higher sensitivity over precision, while the XGBoost model has a higher precision over sensitivity. This means PermonSVM models perform better at predicting positive events (wildfires) over determining pixel areas that are non-affected by fire. Predicting more false negatives (FN) over false positives (FP) would be more acceptable for natural hazard applications than the other way around.

4. Conclusions

We studied state-of-the-art semantic segmentation for wildfire identification in the Alaska regions so that a classification part was based on the SVM methods implemented in the toolbox PERMON for distributed computing, specifically in an extension called PermonSVM. Instead of BGR channels associated with pixel colour information, we assigned time series monitoring changes in reflectance over 152 days. In the presented numerical experiments, we focused on the influence of two stopping criteria based on a norm of projected gradient and a duality gap on model performance for the relaxed $\ell 2$-loss SVM. Attained results were compared and discussed with the $\ell 1$-loss SVM (relaxed). As an underlying solver for model training, we employed the MPRGP solver – a deterministic active-set method.

From the qualities of models in the sense of performance scores and training times, it seems than a terminating training process using stopping criteria based on duality gap for $\ell 2$-loss is a good trade-off between the $\ell 1$-loss and the $\ell 2$-loss models that a training process stopped exploiting a terminating condition incorporating a projected gradient. Such attained model performs better than $\ell 2$-loss case. However, the training process was almost 2 times slower. Compared to the $\ell 1$-loss model, it performs worse in the sense of a hinge-loss function value even so the training process is 1.45 times faster.

We compared the attained models employing relaxed formulations related to the SVM problems with models trained using the complete SVM formulations for both $\ell 1$-loss and $\ell 2$-loss functions as well. From the numerical experiments, we concluded that it is suitable to use relaxed versions of the SVM formulation for training models related to our classification problem because it takes a longer time to train models using complete SVM formulations than in the cases of their relaxed versions and attained models are slightly worse.

We studied qualities related to SVM models trained by means of PermonSVM with boosted tree methods implemented in XGBoost software. We observed that PermonSVM produces models with a higher sensitivity over precision (better at predicting positive events (wildfires) over determining pixel areas that are non-affected by fire). In contrast, the XGBoost model has a higher precision over sensitivity. We think predicting more false negatives (FN) over false positives (FP) would be more acceptable for natural hazard applications than the other way around.

5. Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140) and by Grant of SGS No. SP2022/42, VŠB-Technical University of Ostrava, Czech Republic. The research has been also supported by European Union's Horizon 2020 research and innovation programme under grant agreement number 847593 and by The Czech Radioactive Waste Repository Authority (SÚRAO) under grant agreement number SO2020-017. Further, we acknowledge that the results of this research have been achieved using

the DECI resource Archer2 based in Great Britain at EPCC with support from the PRACE aisbl.

R. T. Mills was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, and by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.

References

[1] Balay, S. et al.: PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory, 2022.

[2] Chen, T. and Guestrin, C.: In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[3] Cortes, C. and Vapnik, V.: Support-vector networks. Machine Learning (1995).

[4] Dostal, Z. and Schoberl, J.: Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. Computational Optimization and Applications **30** (2005).

[5] Dostál, Z.: *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*, vol. 23. SOIA, Springer, New York, US, 2009.

[6] Hapla, V. et al.: PermonQP, 2022. URL http://permon.vsb.cz/qp/.

[7] Hsieh, C. J., Chang, K. W., Lin, C. J., Keerthi, S. S., and Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: *Proceedings of the 25th international conference on Machine learning - ICML'08*. ACM Press, 2008.

[8] LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning. Nature **521** (2015), 436.

[9] Lee, C. P. and Lin, C. J.: A study on l2-loss (squared hinge-loss) multiclass SVM. Neural Computation **25** (2013), 1302–1323.

[10] Pecha, M. and Horák, D.: Analyzing l1-loss and l2-loss support vector machines implemented in PERMON toolbox. In: *Lecture Notes in Electrical Engineering*, pp. 13–23. Springer International Publishing, 2020.

[11] PERMON: PermonSVM, 2022. URL http://github.com/permon/permonsvm.