Primal and dual first order methods for SVM: applications to driver monitoring

Daniela Lupu and Ion Necoara
Automatic Control and Systems Engineering Department
University Politehnica Bucharest
Bucharest, Romania
expd.danny@gmail.com, ion.necoara@acse.pub.ro

***Abstract*—Machine learning tools are become recently very popular for solving real applications from many areas. Most of the learning problems are formulated as optimization problems with simple objective function but large number of constraints of order the number of training data. When considering the dual formulation, usually the objective function is difficult to minimize but the constraints are simple. One relevant application that fits into this pattern is the support vector machine (SVM). A popular approach for solving the primal SVM problem is based on first order methods due to their superior empirical performance. When considering the dual SVM formulation, which has simple constraints, coordinate descent schemes are typically the method of choice in practice due to their cheap iteration. In this paper we present a comparative study of several first order methods for solving primal or dual SVM problems. Numerical evidence on support vector machine classification for automatic detection of driver fatigue supports the effectiveness of such first order methods in real-world problems.**

***Index Terms*—Support vector machine, primal and dual first order methods, driver fatigue monitoring system.**

I. Introduction

In machine learning applications the optimization algorithms involve numerical computation of parameters for a system designed to make decisions based on large amount of data [11], [21]. In particular, one of the most successful formulation for classification of data is the support vector machine (SVM). In the primal formulation of the SVM problem we have a simple objective function, e.g. a quadratic expression with diagonal Hessian, but a large number of linear constraints, equal the number of training data. When considering the dual formulation, usually the objective function is difficult to minimize, but the constraints are simple. The recent success of certain first order optimization methods for SVM problems has motivated increasingly great efforts into developments of new numerical algorithms or into analyzing deeper the existing ones [3], [5], [7]–[9].

Though interior point methods are typically superior in terms of convergence speed and accuracy, the first order methods are able to rapidly provide a suboptimal solution at low computational costs per iteration, which is important in machine learning applications [10]. In this paper we describe several first order methods for solving SVM problems, some of them developed recently by the second author. The optimization algorithms we analyze use first order information combined with random choice of the sets or of the coordinates. More precisely, a popular approach for solving the primal SVM problem is based on first order methods, such as conditional gradient [2], due to their superior empirical performance. However, we show that instead of dealing with the whole set of constraints at each iteration as conditional gradient does, it is more computationally efficient to apply a stochastic gradient variant that uses only one constraint randomly per iteration [8]. When considering the dual SVM formulation, which has simple constraints, we show that coordinate descent schemes are typically the method of choice in practice compared to full projected gradient method due to their cheap iteration [9]. We present a comparative study of these full or partial first order methods for solving primal or dual SVM formulations, accompanied by detailed derivations of their computational complexity.

As application we consider driver fatigue detection using computer vision and support vector machine techniques [17], [18], [20]. Our methodology for fatigue detection is based on Viola-Jones algorithm for face and eyes detection [12] and on machine learning techniques (linear SVM classifier), to classify the eyes of the driver as open or closed. For training the classifier we use the primal and dual first order methods discussed in this paper. Numerical evidence on SVM classification for automatic detection of driver fatigue supports the effectiveness of such first order methods in real problems.

II. Problem Formulation

Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a convex differentiable function. Further, let $\mathbf{W} \subseteq \mathbb{R}^n$ be some non-empty closed convex set. Then, we consider the following convex constrained optimization problem:

$$f^* = \min_{\mathbf{w} \in \mathbf{W}} f(\mathbf{w}) \tag{1}$$

Template (1) covers many applications in machine learning (including SVM), statistics, signal processing, control, by appropriately choosing the objective function and the constraints [1], [5], [6], [17], [21]. We assume that the objective function f is simple, i.e. it is easy to minimize f or we can easily have access to its gradients. On the other hand, in

The research leading to these results has received funding from the Executive Agency for Higher Education, Research and Innovation Funding (UEFISCDI), Romania, under PNIII-P4-PCE-2016-0731, project ScaleFreeNet, no. 39/2017.

many application, including SVM classification, we encounter complicated constraints, e.g. it is computationally prohibitive to project onto the set $\mathbf{W}$ or $\mathbf{W}$ is described by a large number of constraints. The most usual situation is when $\mathbf{W} = \cap_{i=1}^m \mathbf{W}_i$, where each $\mathbf{W}_i$ is simple closed convex set but m is large. By simple we mean that it is easy to construct a barrier function for each $\mathbf{W}_i$ or the projection onto each set $\mathbf{W}_i$ is easy. For example, in SVM the feasible set $\mathbf{W}$ can be written as the intersection of m halfspaces $\mathbf{W} = \{\mathbf{w} : H\mathbf{w} \leq h\}$, that is $\mathbf{W}_i = \{\mathbf{w} : H_i\mathbf{w} \leq h_i\}$, where H_i is the ith row of matrix H. Let us denote by $\mathbf{W}^*$ the optimal set of this problem and for any $\mathbf{w}$ we denote its projection onto $\mathbf{W}^*$ by $\mathbf{w}^*$, denoted $\mathbf{w}^* = \Pi_{\mathbf{W}^*}(\mathbf{w})$. We also assume in the sequel that we have access to the gradient of f, denoted ∇f, or to its conjugate $f^*(\alpha) = \max_{\mathbf{w}\in \text{dom}(f)}[\langle \alpha, \mathbf{w}\rangle - f(\mathbf{w})]$, which is a key assumption in primal or dual first order methods to scale up the numerical algorithms. Based on the previous assumptions we can also easily construct the dual of (1). In particular, if we consider the conjugate of the indicator function $I_{\mathbf{W}}$ of the closed convex set $\mathbf{W}$, so-called support function $\text{supp}_{\mathbf{W}}(\alpha) := \max_{\mathbf{w}\in\mathbf{W}}\langle \mathbf{w}, \alpha\rangle$, then we have:

$$I_{\mathbf{W}}(\mathbf{w}) = \max_{\alpha} \langle \mathbf{w}, \alpha\rangle - \text{supp}_{\mathbf{W}}(\alpha).$$

Replacing this expression in the primal formulation, we obtain the dual problem of (1):

$$\begin{aligned} f^* &= \min_{\mathbf{w}} f(\mathbf{w}) + I_{\mathbf{W}}(\mathbf{w}) \\ &= \min_{\mathbf{w}} [f(\mathbf{w}) + \max_{\alpha} \langle \mathbf{w}, \alpha\rangle - \text{supp}_{\mathbf{W}}(\alpha)] \\ &= \max_{\alpha} [-\text{supp}_{\mathbf{W}}(\alpha) + \min_{\mathbf{w}} (\langle \mathbf{w}, \alpha\rangle + f(\mathbf{w}))] \\ &= \max_{\alpha} -\text{supp}_{\mathbf{W}}(\alpha) - f^*(-\alpha) \\ &= -\min_{\alpha} f^*(-\alpha) + \text{supp}_{\mathbf{W}}(\alpha). \end{aligned} \tag{2}$$

In the sequel, we analyze in details some primal and dual formulations for SVM.

A. Support Vector Machine - Primal

The support vector machine (SVM) is a method that calculates the separating bound that classifies two types of objects or more. We consider the case of binary classification where the goal is to separate into two classes the data thorough a hyperplane, as in Fig. 1. Before formulating the primal problem let us define the geometric margin of (w, b) for a pair from the training set $(x^{(i)}, y^{(i)})_{i=1}^m$ as:

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{\|w\|} \right)^T x^{(i)} - \frac{b}{\|w\|} \right).$$

Through this notion we can measure the quality of the classifier prediction by finding the best bound that maximize the geometric margin. This search is possible by formulating the optimization problem:

$$\begin{aligned} &\min_{w\in\mathbb{R}^n, b\in\mathbb{R}, \zeta\in\mathbb{R}^m} \frac{1}{2}\|w\|^2 + \frac{C}{2}\|\zeta\|^2 \\ &\text{s.t}: \ y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i \ \forall i = 1 : m. \end{aligned} \tag{3}$$

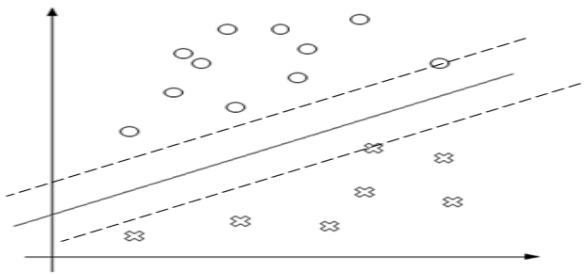

Fig. 1. A binary classification with the linear support vectors machine classifier.

Clearly, this problem fits into the primal form (1) based on the following identification:

$$\begin{aligned} &\mathbf{w} = (w^T \ b \ \zeta^T)^T \\ &\mathbf{W} = \cap_{i=1}^M \mathbf{W}_i \ \left(:= \{y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i\} \right). \end{aligned}$$

When the vector of variables $\zeta = 0$, then the hyperplane defined by $w^T x^{(i)} + b$ separates exactly the data into two classes. When $\zeta \neq 0$, then the data are almost separately by an hyperplane. Note that in SVM usually a formulation of the following form is considered [11], [21]:

$$\begin{aligned} &\min_{w\in\mathbb{R}^n, b\in\mathbb{R}, \zeta\in\mathbb{R}^m} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \zeta_i \\ &\text{s.t: } y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i, \ \zeta_i \geq 0, \ \forall i = 1 : m. \end{aligned} \tag{4}$$

In this paper we use the equivalent form (3) instead of the classical formulation (4) since in this case we get less constraints and the objective function is strongly convex in the variables (w, ζ) and thus more adequate for first order methods. In fact in order to get a strongly convex objective function in the full variables $\mathbf{w} = (w, b, \zeta)$ we may also consider adding an l_2 regularization for b as well in (3):

$$\begin{aligned} &\min_{w\in\mathbb{R}^n, b\in\mathbb{R}, \zeta\in\mathbb{R}^m} \frac{1}{2}\|w\|^2 + \frac{C}{2}\|\zeta\|^2 + \frac{C_0}{2}b^2 \\ &\text{s.l: } y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i \ \forall i = 1 : m, \end{aligned} \tag{5}$$

where C_0 is sufficiently small. Clearly, for $C_0 = 0$ we recover the original formulation (3) and for C_0 small enough the optimal solution of (5) is close to an optimal solution of (3). Note that both primal SVM problems (3) and (5) have simple objective functions (e.g. quadratic expression with diagonal Hessian), but a large number of constraints (intersection of half-spaces), thus difficult to project onto this set.

B. Support Vector Machine - Dual

The primal problem (3) can be also reformulated using the dual settings as given in (2). In particular, one has e.g. the following Lagrangian dual problem:

$$\begin{aligned} \max_{\alpha\leq 0} \min_{w\in\mathbb{R}^n, b\in\mathbb{R}, \zeta\in\mathbb{R}^m} &\frac{1}{2}\|w\|^2 + \frac{C}{2}\|\zeta\|^2 + \\ &+ \sum_{i=1}^m \alpha_i [y_i(w^T x_i - b) - 1 + \zeta_i]. \end{aligned}$$

To find the dual form we resolve the unconstrained quadratic minimization subproblem in $\mathbf{w}$ leading to:

$$w = -\sum_{i=1}^{m} \alpha_i y_i x_i, \quad \zeta = -\frac{\alpha}{C}, \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

Making the change of variable $\alpha \to -\alpha$ we get:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i, \quad \zeta = \frac{\alpha}{C}, \quad \sum_{i=1}^{m} \alpha_i y_i = 0,$$

and the dual problem:

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}^m} \frac{1}{2} \| \sum_{i=1}^{m} \alpha_i y_i x_i \|^2 + \frac{1}{2C} \|\alpha\|^2 - \sum_{i=1}^{m} \alpha_i \\ &\text{s.t. :} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \ \alpha \geq 0. \end{aligned} \tag{6}$$

Elaborating on the above expressions in (6) and denoting the kernel matrix K whose entries are given by $K_{ij} = y_i y_j x_i^T x_j$ for all $i, j = 1 : m$, we get:

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}^m} \phi(\alpha) \ \left(:= \frac{1}{2} \alpha^T (K + 1/C I_m) \alpha - \sum_{i=1}^{m} \alpha_i \right) \\ &\text{s.t. :} \quad \alpha \in \Delta := \left\{ \sum_{i=1}^{m} \alpha_i y_i = 0, \ \alpha \geq 0 \right\}. \end{aligned} \tag{7}$$

We notice that the dual form (7) has simpler constraints than the primal problem (3). In fact, we can project onto the *simplex* $\Delta = \{\alpha : \ y^T \alpha = 0, \ \alpha \geq 0\}$ in $\mathcal{O}(m \log m)$ flops [4]. Moreover, once a dual solution is obtained we can easily recover a primal solution. Indeed, since the complementarity condition $\alpha_i [y_i (w^T x_i - b) - 1 + \zeta_i] = 0$ holds, then for any $\alpha_i > 0$ we can recover b from $y_i (w^T x_i - b) - 1 + \zeta_i = 0$. Thus, a primal solution can be recovered as:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i, \quad \zeta = \frac{\alpha}{C}, \quad b = w^T x_i - (1 - \zeta_i)/y_i.$$

III. First Order Methods for SVM

In the following we describe several primal and dual first order methods for solving SVM problems that use first order information combined with random choice of sets or of coordinates. We also provide detailed derivations of their computational complexity on this specific application. From our best knowledge this is the first comparative study of the methods considered below for SVM.

A. Primal First Order Methods

Since primal SVM problem (3) has complicated linear constraints (intersection of many half-spaces) and since there are efficient solvers for linear programs, one possible candidate for solving (3) is the conditional gradient algorithm, see [2] for a detailed description of this method:

Conditional Gradient

Given $\mathbf{w}_0 \in \mathbf{W}$, for $k \geq 0$ do:
1. Compute the gradient $\nabla f(\mathbf{w}_k)$
2. Solve the linear program:
   $s_k = \arg\min_{s \in \mathbf{W}} \langle \nabla f(\mathbf{w}_k), s \rangle$
3. Compute the new iterate:
   $\mathbf{w}_{k+1} = (1 - \gamma_k)\mathbf{w}_k + \gamma_k s_k$.

where the stepsize γ_k can be chosen using line search, constant, or variable such as $\gamma_k = 2/(k+1)$. Its convergence behavior is given in next theorem:

Theorem 3.1: [2] Let f be a convex function with Lipschitz continuous gradient of constant L_f and $\mathbf{W}$ be a convex set with finite diameter $\text{diam}_W = \max_{\mathbf{w}_1, \mathbf{w}_2 \in \mathbf{W}} \|\mathbf{w}_1 - \mathbf{w}_2\| < \infty$. Then, the iterates of conditional gradient with $\gamma_k = 2/(k+1)$ satisfy:

$$f(\mathbf{w}_k) - f^* \leq \frac{2 L_f \text{diam}_W^2}{k+2}.$$

In the particular case of SVM problem (3), since the objective function is simple, we can compute immediately the gradient:

$$\nabla f(\mathbf{w}_k) = (w_k^T \ 0 \ \ C\zeta_k^T)^T$$

and the linear program we need to solve is:

$$\begin{aligned} s_k = &\arg \min_{s=(w,b,\zeta)} (w_k^T \ 0 \ C\zeta_k^T) s \\ &\text{s.t. :} \ y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i \ \forall i = 1 : m. \end{aligned}$$

However, since the feasible set $\mathbf{W} = \{\mathbf{w} : \ y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i \ \forall i = 1 : m\}$ is usually unbounded in order to ensure convergence we need to impose some additional box constraints $\mathbf{w}_l \leq \mathbf{w} \leq \mathbf{w}_u$, for appropriately chosen $(\mathbf{w}_l, \mathbf{w}_u)$. Moreover, conditional gradient needs to compute the solution of a linear program at each iteration whose overall complexity is of order $\mathcal{O}((m+n)^3)$ [10], and requires knowledge of the entire feasible set $\mathbf{W}$. Therefore, it is not adequate in applications where the data arrives in streams. Next, we present a gradient descent algorithm with random projections that uses at each iteration only one set $\mathbf{W}_i = \{\mathbf{w} : y^{(i)}(w^T x^{(i)} - b) \geq 1 - \zeta_i\}$ from the intersection $\mathbf{W} = \cap_{i=1}^m \mathbf{W}_i$, see [8] for a detailed description:

Gradient with Random Projections

Given any $\mathbf{w}_0$, for $k \geq 0$ do:
1. Compute gradient step:
   $\mathcal{G}(\mathbf{w}_k) = \mathbf{w}_k - \gamma_k \nabla f(\mathbf{w}_k)$
2. Choose randomly an index $i \in [1 : m]$
3. Compute the new iterate:
   $\mathbf{w}_{k+1} = \Pi_{\mathbf{W}_i}(\mathcal{G}(\mathbf{w}_k))$,

where the stepsize γ_k can be chosen constant or variable such as $\gamma_k = \gamma_0/k$. Recall that $\Pi_{\mathbf{W}}$ denotes the projection operator onto the set $\mathbf{W}$. The convergence behavior of this scheme is given next:

Theorem 3.2: [8] Let f be strongly convex function with constant σ_f and with gradient Lipschit with constant L_f.

Moreover, assume that the stepsize is chosen as $\gamma_k = \frac{\gamma_0}{k}$. Then, there exists a constant $M(L_f, \sigma_f, \mathbf{w}^*) > 0$ such that the iterates of gradient descent algorithm with random projections satisfy the following convergence rate in expectation:

$$\mathbf{E}\left[\|\mathbf{w}_k - \mathbf{w}^*\|^2\right] \leq \frac{M(L_f, \sigma_f, \mathbf{w}^*)}{k}.$$

We used the notation $\mathbf{w}^*$ for an optimal solution of the primal SVM problem. Note that the previous theorem requires strongly convex objective function, therefore for SVM we usually consider the l_2 regularization formulation (5), for some sufficiently small C_0. Moreover, note that the computation of the gradient step $\mathcal{G}(\mathbf{w}_k)$ is numerically cheap and the projection $\Pi_{\mathbf{W}_i}$ onto the half-space $\mathbf{W}_i$ can be computed in closed form in $\mathcal{O}(m+n)$ operations, which is much cheaper than the overall complexity $\mathcal{O}((m+n)^3)$ for solving the linear programm corresponding to one iteration of the conditional gradient scheme.

B. Dual First Order Methods

In the dual formulation of SVM given in (7), the feasible set is simple, that is we can project onto the *simplex* $\Delta = \{\alpha : y^T\alpha = 0,\ \alpha \geq 0\}$ in $\mathcal{O}(m \log m)$ flops [4]. Thus, one possible candidate for solving this problem (7) is the full dual projected gradient method, see [10] for details:

Full Dual Gradient

Given $\alpha_0 \in \Delta$, for $k \geq 0$ do:
1. Compute gradient step:
   $\mathcal{G}_\phi(\alpha_k) = \alpha_k - \gamma_k \nabla\phi(\alpha_k)$
2. Compute the new iterate:
   $\alpha_{k+1} = \Pi_\Delta(\mathcal{G}_\phi(\alpha_k))$,

where the stepsize γ_k can be chosen constant or using line search. The convergence behavior of this scheme is given in the next theorem:

Theorem 3.3: [10] Let ϕ be strongly convex function with constant σ_ϕ and with gradient Lipschit with constant L_ϕ. Moreover, assume that the stepsize is chosen constant $\gamma_k = 1/L_\phi$. Then, the iterates of the full dual projected gradient algorithm satisfy:

$$\|\alpha_k - \alpha^*\|^2 \leq \left(1 - \frac{2\sigma_\phi}{\sigma_\phi + L_\phi}\right)\|\alpha_0 - \alpha^*\|^2.$$

Here α^* denotes an optimal solution of (7). Note that for the dual SVM problem (7) the Hessian $K_C = K + 1/CI_m$ is positive definite, since $K \succeq 0$, and thus $\sigma_\phi = \lambda_{\min}(K_C)$ and $L_\phi = \lambda_{\max}(K_C)$. Moreover, since the projections step can be computed efficiently in $\mathcal{O}(m \log m)$ operations, the main computational bottleneck of the full dual gradient scheme is the computation of the full gradient $\nabla\phi(\alpha) = K_C\alpha - e$, usually done in $\mathcal{O}(m^2)$ operations. Here e denotes the vector with all entries 1. However, we show below that applying a random coordinate descent scheme to dual SVM problem (7) we can update the iterates in at most $\mathcal{O}(m)$ operations (since K is a dense matrix). Indeed, the random coordinate descent algorithm has the following steps, see [9] for more details:

Random Coordinate Descent

Given $\alpha_0 \in \Delta$, for $k \geq 0$ do:
1. Choose uniformly at random $(i, j) \in [1 : m]$
2. Solve the subproblem:
   $(s_i^*, s_j^*) = \arg \min_{(s_i, s_j) \geq 0, y_i s_i + y_j s_j = 0} \Phi_{ij}(s_i, s_j)$
2. Compute the new iterate:
   $\alpha_{k+1} = \alpha_k + s_i^* e_i + s_j^* e_j$,

where $\Phi_{ij}(s_i, s_j) = \nabla_i\phi(\alpha_k)s_i + L_{ij}/2s_i^2 + \nabla_j\phi(\alpha_k)s_j + L_{ij}/2s_j^2$, $\nabla_i\phi$ denotes the ith component of $\nabla\phi$, i.e. $\nabla_i\phi = e_i^T\nabla\phi$, and L_{ij} is the Lipschitz constant of the partial gradient $(\nabla_i\phi, \nabla_j\phi)$. The convergence of this scheme is given next:

Theorem 3.4: [10] Let ϕ be strongly convex function with constant σ_ϕ and with gradient Lipschitz with constant L_ϕ. Then, there exists a constant $\tau = \tau(\sigma_\phi, L_\phi, m) \in (0,\ 1)$ such that the iterates of the random coordinate descent algorithm satisfy in expectation:

$$\mathbf{E}\left[\phi(\alpha_k)\right] - f^* \leq \tau^k(\phi(\alpha_0) - f^*).$$

When the random coordinate descent algorithm is applied on the dual SVM problem (7), we have that

$$\nabla_i\phi(\alpha) = K_C(i,:)\alpha - 1,$$

usually computed in $\mathcal{O}(m)$ flops. Moreover, for this application $L_{ij} = K_C(i,i) + K_C(j,j)$. Finally, the subproblem can be computed in closed form in $\mathcal{O}(1)$ operations, thus the complexity per iteration of this scheme is $\mathcal{O}(m)$, that is m times cheaper than for the full dual gradient. In the next section we present numerical evidence for assessing the practical convergence behavior of the four algorithms described above on the driver fatigue monitoring system.

IV. Driver Fatigue Monitoring System Using Support Vector Machines

We consider as main application the driver fatigue monitoring system. Driver fatigue is one of the leading causes of traffic accidents. In this section we briefly present the fatigue monitoring system which exploits the driver's facial expression to detect and alert fatigued drivers [17], [19], [20]. The presented approach adopts the Viola-Jones classifier [12] to detect the driver's facial features. The correlation coefficient template matching method is then applied to derive the state of each feature on a frame by frame basis. An SVM classifier is finally integrated within the system to classify the facial appearance as either fatigued or otherwise. Using this simple and cheap implementation, the overall system achieved an accuracy of prediction of about 87%.

In Fig.2 we depict the flowchart of the proposed system whose process steps are similar to [20]. The first step of the strategy implies a face detection. Once the searching area is restrained, the eyes detection phase begins. If any of this steps fails then a new frame is acquired and the detection process is repeated. Otherwise, if the detection is successful then we apply the SVM classifier to obtain the state of the eye. Our contribution is achieved at this step, that is we implement and compare the

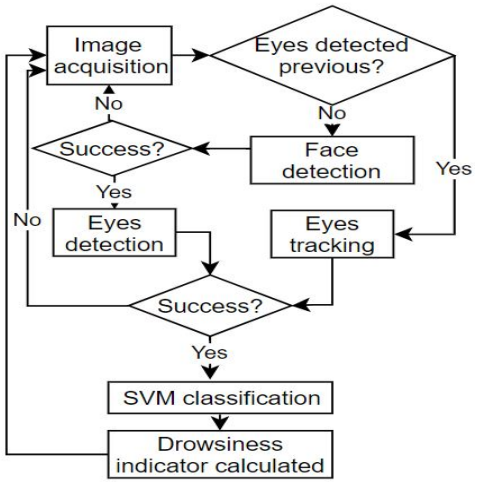

Fig. 2. Flowchart of the proposed system.

four optimization algorithms described in the previous section on the corresponding primal and dual SVM problems using real data from [16] to obtain the eye classifiers. After the first successful detection, the eyes tracking is triggered. If the track fails then we go back to the detection phase. The environment in which the application was developed is Matlab R2016.

A. The Detection Phase

In the detection step we use two types of detectors: one for the face and a second for the eyes. In our implementation we applied the Viola-Jones algorithm with the HAAR-like features that are fast to calculate due to the integral images concept, see [12] for more details. However, the simple HAAR classifiers are usually weak, so the Viola-Jones approach develops a stronger classifier by organizing the previous ones in a cascade structure. This method is precise, but the main disadvantage is that the process is long. Matlab is equipped with an image processing toolbox that includes face and eyes detectors as described above. They are applied on a gray image, so first a change of color spectrum is needed as the frame acquired is a RGB image. The face detector is initialized with the command:

```
vision.CascadeObjectDetector
    ('FrontalFaceCART')
```

and the detector is called with

```
bbox = step(detector,Image)
```

and returns a bound box with the following information (x,y,width,height). For the eye detectors the command for the initialization has the region of interest option activated, i.e for the left eye:

```
vision.CascadeObjectDetector
 ('LeftEye','UseROI',true)
```

Practically, the bound box procured by the face detector will be the frontier of the new area of search instead of the entire image. The eyes detectors are called with:

```
bbox = step(detector,Image,ROI)
```

Even if the detection is fast, repeating this action for each frame will slow the processing phase, thus the need of a tracker is imperative.

B. The Tracking Phase

The solution selected for the feature tracker is based on the Kanade-Lucas-Tomasi work [13], [14], and it is named the KLT tracker. For this method to work we first need to establish the type of features that will be used in the process. The extraction of the characteristics was based on the fact that the pupil of the eye is a circle. Thus, we choose for this task the Circle Hough Transformation (CHT) [15]. The CHT is applied in the eye zone that it already detected and the points of this specific geometric figure are recorded taking into account the whole frame. Then, again Matlab has a command that resolves the CHT:

```
imfindcircles(Image,rVal/rRange)
```

To estimate the radius of the pupil we use the information from the eye detector and the human geometric face aspects. After the extraction of the features, the points of interest are used in the KLT tracker. The tracking method uses the detected features for monitoring. Each frame is analyzed based on the previous one. The aim of this approach is to align a template (the previous frame), denoted with $T(p)$, to an input image $I(p)$, where p is the vector that has the image coordinates (x, y). For a better alignment between the template and the input image, we employ a minimization of the differences, which are obtained with the help of the l_2 norm:

$$\sum_p \|I(W(p;d)) - T(p)\|^2,$$

where $W(\cdot,\cdot)$ is the operator for translation. The algorithm is designed to eliminate the points for which the distance from a frame to another are bigger than a certain threshold and to go back to detection if there are less than e.g. three points to follow. This approach is faster and has a low computational cost, but it looses points when occlusion take place.

C. The Classification Phase

After identifying the eyes, next step is to classify them as closed or open and based on this information to calculate the fatigue indicator. In order to train and test the classifiers we used a database of real images of size $24x24$ pixels from [16]. We select a total of 2.400 images from which half were for the closed eye state and the other half for the open eye state. For the training process we use 800 images (one-third of the total number of images from the database), half for the closed eye and half for the open eye. After extracting the features from the training data (144 or 900) using the procedures described above, the linear SVM classifiers were obtained with the primal and dual first order optimization methods described in the previous sections. The SVM classifiers were then tested on the remaining images. The classifiers are binary, thus have only the two states: open (coded as 1), and closed (coded as

Features	Gradient Random Proj.			Conditional Gradient			Random Coordinate			Dual Gradient		
	obj	iter	time	obj	iter	time	obj	iter	time	obj	iter	time
144	20.17	127	54	20.02	392	304	19.93	23	5	19.94	93	27
900	11.05	471	173	10.98	697	789	10.87	61	11	10.89	518	58

TABLE I
COMPARISON OF THE FOUR ALGORITHMS ON PRIMAL OR DUAL SVM.

Fig. 3. Database samples used in learning.

-1). For each eye (left and right) we create a separate detector and classifier, but we have not observed substantial differences in the classification accuracy of the left/right eye. The fatigue indicator takes into account the eye status as follows: if the eye status is closed for 6 frames consecutively, then an alarm is triggered as it is considered that the driver is asleep; if an alternation is detected (blink), then we consider that the person is tired and a break is suggested.

Using the simple and cheap first order optimization algorithms previously presented for solving either the primal or dual formulation of the linear SVM problem derived from the 800 images, the overall system achieves an accuracy of about 87% for both eyes when we consider 900 features and of about 83% for 144 features. Note that while the differences between the four algorithms is significant regarding the objective function, CPU time and number of iterations (see Table 1), their classification accuracy is not that different, in general between 86% and 88% for 900 features. Moreover, the reader should note that in the table we report the full number of iterations for all the methods, that is for gradient based on random projection we divided the total number of iteration by $n + m$ and for random coordinate descent scheme by $m/2$. From the table we also observe a very good behavior for the random coordinate descent scheme, both in terms of CPU time and objective function over the other three first order methods. Therefore, numerical evidence supports the effectiveness of these primal and dual first order methods based on random choice of the sets or of the coordinates in real-world problems.

V. CONCLUSIONS

We have presented a comparative study of four gradient type methods for solving primal or dual SVM problems for classification. Our preliminary numerical simulations on SVM classification for automatic detection of driver fatigue have showed superior numerical performance of those methods based on random choice of the sets or of the coordinates compared to full projections or gradients, respectively. In our future work we plan to investigate the effectiveness of such first order methods in other real-world problems as well.

REFERENCES

[1] D. Blatt and A.O. Hero, *Energy based sensor network source localization via projection onto convex sets*, IEEE Transactions on Signal Processing, 54(9): 3614–3619, 2006.

[2] M. Jaggi, *Revisiting Frank-Wolfe: projection-free sparse convex optimization*, International Conference on Machine Learning, 2013.

[3] A. Juditsky and A. Nemirovski, *First order methods for nonsmooth convex large-scale optimization (I): general purpose methods*, Optimization for Machine Learning, 121-148, 2011.

[4] K. Kiwiel, *On linear-time algorithms for the continuous quadratic Knapsack problem*, J. Optimization Theory Applications, 2007.

[5] E. Moulines and F.R. Bach, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning*, In Advances in Neural Information Processing Systems, 2011.

[6] I. Necoara, V. Nedelcu and I. Dumitrache, *Parallel and distributed optimization methods for estimation and control in networks*, Journal of Process Control, 21(5): 756–766, 2011.

[7] A. Patrascu and I. Necoara, *Nonasymptotic convergence of stochastic proximal point algorithms for constrained convex optimization*, Journal of Machine Learning Research, 2018.

[8] I. Necoara, *Random algorithms for convex minimization over intersection of simple sets*, Technical Report, UPB, 2017.

[9] I. Necoara and A. Patrascu, *A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints*, Computational Optim. and Applications, 57(2): 307–337, 2014.

[10] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer, Boston, 2004.

[11] S. Theodoridis, *Machine Learaning: A Bayesian and Optimization Perspective*, Academic Press, 2015.

[12] V. Paul and M. Jones, *Robust real-time face detection*, International Journal of Computer Vision, 57(2): 137–154, 2004.

[13] B. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, Proceedings of Artificial Intelligence Conference, 674–679, 1981.

[14] C. Tomasi and T. Kanade, *Detection and tracking of point features*, International Journal of Computer Vision, 1991.

[15] R. Duda and P. Hart, *Use of the Hough transformation to detect lines and curves in pictures*, Communications of the ACM, 15(1): 11–15, 1972.

[16] F. Song, X. Tan, X. Liu and S. Chen, *Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients*, Pattern Recognition, 47(9): 2825–2838, 2014.

[17] A. Colic, O. Marques and B. Furht, *Driver Drowsiness Detection: Systems and Solutions*, Springer, 2014.

[18] H. Singh, J. Bhatia and J. Kaur, *Eye tracking based driver fatigue monitoring and warning system*, India International Conference on Power Electronics, 2011.

[19] W. Horng, C. Chen, Y. Chang and C. Fan, *Driver fatigue detection based on eye tracking and dynamic template matching*, IEEE Conference on Networking, Sensing and Control, 7–12, 2004.

[20] V. Dahiphale, R. Sathyanarayana and M. Mukhedkar, *Computer Vision System for Driver Fatigue Detection*, International Journal of Advanced Research in Electronics and Communication Engineering, 4(9), 2015.

[21] V. Vapnik, *Statistical learning theory*, John Wiley, 1998.