2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

RANDOM PRIMAL-DUAL PROXIMAL ITERATIONS FOR SPARSE MULTICLASS SVM

G. Chierchia[1], *N. Pustelnik*[2], *J.-C. Pesquet*[1]

[1] Université Paris-Est, LIGM UMR 8049, CNRS, ENPC, ESIEE Paris, UPEM, F-93162 Noisy-le-Grand, France
[2] Univ Lyon, Ens de Lyon, Univ Lyon 1, CNRS, Laboratoire de Physique, F-69342 Lyon, France

ABSTRACT

Sparsity-inducing penalties are useful tools in variational methods for machine learning. In this paper, we propose two block-coordinate descent strategies for learning a sparse multiclass support vector machine. The first one works by selecting a subset of features to be updated at each iteration, while the second one performs the selection among the training samples. These algorithms can be efficiently implemented thanks to the flexibility offered by recent randomized primal-dual proximal methods. Experiments carried out for the supervised classification of handwritten digits demonstrate the interest of considering the primal-dual approach in the context of block-coordinate descent. The efficiency of the proposed algorithms is assessed through a comparison of execution times and classification errors.

Index Terms— Sparsity, multiclass SVM, proximal algorithm, random update, block-coordinate descent.

1. INTRODUCTION

This work aims at designing fast algorithms for learning a multiclass *support vector machine* (SVM) [1] with a sparsity-inducing regularization. In this context, sparsity has been introduced for two main reasons:

(i) to prevent overfitting when the number of features is much bigger than the number of training samples,

(ii) to provide insight in the interpretation of results [2, 3].

The use of sparse regularization in SVM was firstly proposed in the context of binary classification [2, 4], where various ℓ_p-norms were used to shrink small coefficients to zero, so as to perform an implicit "feature selection". With the advent of multiclass SVM [5], the attention shifted toward $\ell_{1,p}$-norms, due to their ability to impose group sparsity [6, 7, 8, 9].

The benefits of sparse regularization are now well established. Its use for classification, however, is still limited by its high computational cost. Indeed, while the commonly-used quadratic regularization can be efficiently implemented by resorting to Lagrangian duality, its counterpart based on the $\ell_{1,p}$-norm regularization leads to a dual formulation as difficult to solve as the primal one. Consequently, the sparse multiclass SVM is usually trained through the direct resolution of the primal optimization problem.

Among the possible approaches to train a sparse multiclass SVM, one can resort to linear programming, yielding a problem involving several linear constraints whose number is slightly larger than the number of training data [10, 11]. In order to reduce computational time, a much more efficient strategy consists in approximating the loss function with the quadratic hinge loss. Such an approximation permits to use the forward-backward method [12] or a block-coordinate descent method [13], although this approach may affect the classification accuracy.

Recently, we have proposed two efficient algorithms for dealing with the exact hinge loss in an efficient manner [14], both relying on primal-dual proximal methods. The first one solves the constrained formulation by resorting to the epigraphical splitting technique [15], while the second approach solves the penalized form by relying on the projection onto a simplex. The good performance of both approaches w.r.t. the state-of-the-art in terms of accuracy and computational times is shown in [14].

Nowadays, it is well-established that random updates can significantly reduce the computational time [16, 17]. However, dealing with sparsity is out of the scope of usual randomized strategies. Recent major contributions in the optimization literature focused on extending primal and primal-dual proximal algorithms to random updates [18, Section IV]. For instance, in [19, 20, 21], the authors proposed new results for random updates in forward-backward iterations, especially adapted for training a sparse multiclass SVM with the squared hinge loss [13]. More recently, random updates in primal-dual proximal methods have been proposed [22, 23], allowing one to deal with the exact formulation of the hinge loss, as proposed by Crammer and Singer [5].

In this work, we propose an efficient solution to solve the learning problem of a sparse multiclass SVM formulated with the exact expression of the hinge loss [5]. Our approach is based on the randomized FBPD method [22], which allows us to simultaneously solve the primal and the dual problems through a block-coordinate descent strategy. In particular, we propose two algorithms: the first one selects the blocks to be

978-1-5090-0746-2/16/$31.00 ©2016 European Union

updated over the features of the solution vector (similarly to [13]), while the second one performs the selection over the training samples. The latter constitutes the main contribution of our paper, as we are not currently aware of another sparse SVM training algorithm which offers such flexibility.

In Section 2, we formulate the sparse multiclass SVM problem and recall the algorithm we proposed in [14]. Section 3 is dedicated to the proposed randomized primal-dual proximal algorithm. Two algorithms will be provided, allowing us to play with stochasticity over primal or dual variables, that is features or samples. The convergence guaranties will be established. The efficiency of the proposed algorithms w.r.t. a randomized version of the forward-backward method [20] will be illustrated in Section 4.

Notations Let $f\colon \mathbb{R}^N \to]-\infty,+\infty]$ be a separable function such that, for every $\mathrm{u} = (u^{(s)})_{1\leqslant s\leqslant S} \in \mathbb{R}^S$, $f(\mathrm{u}) = \sum_{s=1}^{S} f_s(u^{(s)})$. We denote $\mathbb{S} \subset \{1,\ldots,S\}$ and $f_{\mathbb{S}}$ the function defined as, for every $\mathrm{u}_{\mathbb{S}} = (u^{(s)})_{s\in\mathbb{S}}$, $f_{\mathbb{S}}(\mathrm{u}_{\mathbb{S}}) = \sum_{s\in\mathbb{S}} f_s(u^{(s)})$. We denote $\overline{\mathbb{S}} = \{1,\ldots,S\} \setminus \mathbb{S}$.

2. SPARSE MULTICLASS SVM PROBLEM

In supervised learning, the discriminating functions are built from a set of L input-output pairs

$$\mathcal{T} = \big\{(\mathrm{x}_\ell, z_\ell) \in \mathbb{R}^N \times \{1,\ldots,K\} \mid \ell = \{1,\ldots,L\}\big\}, \tag{2.1}$$

where $\mathrm{x}_\ell \in \mathbb{R}^N$ denotes a training sample, and z_ℓ indicates the associated class. Here, the class number is denoted by K. The features are assumed to be linear in some representation input space [24], denoted by $\varphi\colon \mathbb{R}^N \mapsto \mathbb{R}^M$.

The objective of learning consists of finding a vector

$$\mathrm{w} = \Big[\underbrace{(\mathrm{w}^{(1)})^\top}_{\text{size } M} \quad \cdots \quad \underbrace{(\mathrm{w}^{(K)})^\top}_{\text{size } M}\Big]^\top \in \mathbb{R}^{MK} \tag{2.2}$$

such that, for every $\ell \in \{1,\ldots,L\}$, the input-output pair $(u_\ell, z_\ell) \in \mathcal{T}$ is correctly predicted by the classifier, i.e.,

$$(\forall \ell \in \{1,...,L\}) \quad z_\ell = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} \ \varphi(\mathrm{x}_\ell)^\top \mathrm{w}^{(k)}. \tag{2.3}$$

By the definition of argmax, the above equality holds if $\max_{k\neq z_\ell} \ \varphi(\mathrm{x}_\ell)^\top(\mathrm{w}^{(k)} - \mathrm{w}^{(z_\ell)}) < 0$, which (after suitable normalization) is equivalent to

$$(\forall \ell \in \{1,...,L\}) \quad \max_{k\neq z_\ell} \ \varphi(\mathrm{x}_\ell)^\top(\mathrm{w}^{(k)} - \mathrm{w}^{(z_\ell)}) \leqslant -1. \tag{2.4}$$

Unfortunately, this constraint has no practical interest for learning purposes, as it becomes infeasible when the training set is not fully separable. Multiclass SVM overcome this issue by introducing the notion of *soft margins*, which boils down to the definition of the multiclass hinge loss [5]:

$$h_\ell(\mathrm{T}_\ell \mathrm{w}) = \max\big\{0, 1+\max_{k\neq z_\ell} \ \varphi(\mathrm{x}_\ell)^\top(\mathrm{w}^{(k)} - \mathrm{w}^{(z_\ell)})\big\}. \tag{2.5}$$

Hereabove, the operator $T_\ell \in \mathbb{R}^{K\times MK}$ is defined as

$$\mathrm{T}_\ell \mathrm{w} = \Big[\varphi(\mathrm{x}_\ell)^\top(\mathrm{w}^{(k)} - \mathrm{w}^{(z_\ell)})\Big]_{1\leqslant k\leqslant K} \tag{2.6}$$

where, for every $\mathrm{y}_\ell = (y_\ell^{(k)})_{1\leqslant k\leqslant K} \in \mathbb{R}^K$,

$$h_\ell(\mathrm{y}_\ell) = \max_{1\leqslant k\leqslant K} \ y_\ell^{(k)} + 1 - \delta_{k,z_\ell}, \tag{2.7}$$

with δ_{k,z_ℓ} being the Kronecker delta, which is equal to 1 if $k = z_\ell$ and 0 otherwise. The sparse multiclass SVM learning problem amounts to

$$\underset{\mathrm{w}\in\mathbb{R}^{MK}}{\text{minimize}} \quad \underbrace{\sum_{\ell=1}^{L} h_\ell(\mathrm{T}_\ell \mathrm{w})}_{h(\mathrm{Tw})} + \underbrace{\lambda \sum_{b=1}^{B}\sum_{k=1}^{K} \|\mathrm{w}^{(k,b)}\|_p}_{g(\mathrm{w})}, \tag{2.8}$$

where $\lambda > 0$, and each vector $\mathrm{w}^{(k)}$ is block-decomposed as

$$\mathrm{w}^{(k)} = \Big[\underbrace{(w^{(k,1)})^\top}_{\text{size } M_1} \cdots \underbrace{(w^{(k,B)})^\top}_{\text{size } M_B}\Big]^\top \in \mathbb{R}^M, \tag{2.9}$$

with $M = M_1 + \cdots + M_B$. The regularization term g is chosen so as to promote some form of group sparsity. In this work, we focus on convex $\ell_{1,p}$-norm regularizations ($p \geqslant 1$).[1]

In [14], we proposed to solve the minimization problem (2.8) using the primal-dual proximal method recalled in Algorithm 1, where the steps involving the proximity operator denote implicit subgradient steps, i.e.,

$$\begin{aligned} \mathrm{w}^{[i+1]} &= \operatorname{prox}_f(\mathrm{w}^{[i]}) && (2.10)\\ &= \mathrm{w}^{[i]} - u^{[i]} \ \text{ with } \ u^{[i]} \in \partial f(\mathrm{w}^{[i+1]}). && (2.11) \end{aligned}$$

For more details regarding proximity operators, the reader could refer to [25]. A large number of closed-form expression are available in the literature (see [26, 27] and the references therein). The proximity operator of g has a closed-form expression for specific values of q, such as $q \in \{1, 2, +\infty\}$, while the proximity operator of the conjugate of h_ℓ reduces to the standard projection onto the unit simplex [28].

The sequence $(\mathrm{w}^{[i]})_{i\in\mathbb{N}}$ generated by Algorithm 1 converges to a solution to (2.8). The efficiency of these iterations w.r.t. state-of-the-art procedures is illustrated in [14]. The computational performance makes the approach comparable with forward-backward iterations, while a better accuracy is reached in terms of classification performance. Based on some recent advances in random proximal algorithms, we propose to design new iterations using the block-separability of the involved functions.

[1] The dimensionality M could be increased by 1 in order to integrate the bias estimation. For readability purposes, we have decided to not take into account that additional variable but the proposed methodology can be directly extended to solve this generalized estimation problem. The reader could refer to [14] for more details regarding this point.

Algorithm 1 FBPD for solving Problem (2.8)

```
Initialization
    choose $\mathrm{w}^{[0]} \in \mathbb{R}^{MK}$ and $\mathrm{y}^{[0]} \in \mathbb{R}^{LK}$
    set $\mathrm{T} = \left[\mathrm{T}_1^\top \quad \dots \quad \mathrm{T}_L^\top\right]^\top \in \mathbb{R}^{LK \times MK}$
    set $\tau > 0$ and $\sigma > 0$ such that $\tau\sigma\|\mathrm{T}\|^2 \leqslant 1$.
For $i = 0, 1, \dots$
    $\mathrm{w}^{[i+1]} = \mathrm{prox}_{\tau g}\left(\mathrm{w}^{[i]} - \tau\,\mathrm{T}^* \mathrm{y}_\ell^{[i]}\right)$
    $\mathrm{y}^{[i+1]} = \mathrm{prox}_{\sigma h^*}\left(\mathrm{y}^{[i]} + \sigma \mathrm{T}\big(2\mathrm{w}^{[i+1]} - \mathrm{w}^{[i]}\big)\right)$.
```

3. PRIMAL-DUAL COORDINATE DESCENT

The interesting point in using primal-dual proximal algorithms is the possibility to perform a block-coordinate decomposition over both the primal variable w and the dual variable y. Indeed, Problem (2.8) presents a block-separable structure, as

$$\mathrm{T}_\ell\,\mathrm{w} = \sum_{b=1}^{B} \mathrm{T}_{\ell,b}\,\mathrm{w}_b \tag{3.1}$$

where

$$\mathrm{w}_b = \Big[\underbrace{(w^{(1,b)})^\top}_{\text{size } M_b} \cdots \underbrace{(w^{(K,b)})^\top}_{\text{size } M_b}\Big]^\top \in \mathbb{R}^{KM_b} \tag{3.2}$$

and

$$\mathrm{T}_{\ell,b}\,\mathrm{w}_b = \Big[\big(\varphi(\mathrm{x}_\ell)^{(b)}\big)^\top \big(\mathrm{w}^{(k,b)} - \mathrm{w}^{(z_\ell,b)}\big)\Big]_{1 \leqslant k \leqslant K} \tag{3.3}$$

with $\varphi(\mathrm{x}_\ell) = \Big[\big(\varphi(\mathrm{x}_\ell)^{(1)}\big)^\top \dots \big(\varphi(\mathrm{x}_\ell)^{(B)}\big)^\top\Big]^\top \in \mathbb{R}^M$ being decomposed as in (2.9). Therefore, Problem (2.8) becomes

$$\underset{\mathrm{w}=(\mathrm{w}_1,\dots,\mathrm{w}_B)}{\text{minimize}} \quad \sum_{\ell=1}^{L} h_\ell\Big(\sum_{b=1}^{B} \mathrm{T}_{\ell,b}\mathrm{w}_b\Big) + \lambda \sum_{b=1}^{B} \underbrace{\|\mathrm{w}_b\|_{1,p}}_{g_b(\mathrm{w}_b)} \tag{3.4}$$

which can be solved with the randomized primal-dual methods in [22], which consists of the following iterations:

$$\left\lfloor\begin{aligned}
&\text{Set } \mathbb{B}_i \subset \{1,\dots,B\} \text{ and } \mathbb{L}_i \subset \{1,\dots,L\}\\
&\mathrm{w}_{\mathbb{B}_i}^{[i+1]} = \left(\mathrm{prox}_{\tau g_b}\Big(\mathrm{w}_b^{[i]} - \tau \sum_{\ell=1}^{L} \mathrm{T}_{\ell,b}^*\,\mathrm{y}_\ell^{[i]}\Big)\right)_{b \in \mathbb{B}_i}\\
&\mathrm{w}_{\overline{\mathbb{B}}_i}^{[i+1]} = \mathrm{w}_{\overline{\mathbb{B}}_i}^{[i]}\\
&\mathrm{y}_{\mathbb{L}_i}^{[i+1]} = \left(\mathrm{prox}_{\sigma h_\ell^*}\Big(\mathrm{y}_\ell^{[i]} + \sigma \sum_{b=1}^{B} \mathrm{T}_{\ell,b}\big(2\mathrm{w}_b^{[i+1]} - \mathrm{w}_b^{[i]}\big)\Big)\right)_{\ell \in \mathbb{L}_i}\\
&\mathrm{y}_{\overline{\mathbb{L}}_i}^{[i+1]} = \mathrm{y}_{\overline{\mathbb{L}}_i}^{[i]}
\end{aligned}\right. \tag{3.5}$$

where the initialization setting is the same as in Algorithm 1.

3.1. Convergence conditions

At each iteration of the scheme in (3.5), we randomly select a block of features $\mathbb{B}_i \subset \{1,\dots,B\}$ and a subset of samples $\mathbb{L}_i \subset \{1,\dots,L\}$, according to a given probability law. By [22], the almost sure convergence is guaranteed if the activation probability of each block is greater than zero, and the next condition is satisfied at each iteration:

$$b \notin \mathbb{B}_i \qquad \Longrightarrow \qquad (\forall \ell \in \mathbb{L}_i) \quad \mathrm{T}_{\ell,b} = 0. \tag{3.6}$$

A simple way to satisfy the above condition consists of setting $\mathbb{B}_i = \{1,\dots,B\}$, which is the choice adopted here. However, this prevents us from simultaneously updating the primal and the dual variables through a block-coordinate strategy. In the following, we present two algorithms that randomly update either the primal or the dual variable.

3.2. Proposed schemes

We first focus on the dual random update, that is $\mathbb{B}_i = \{1,\dots,B\}$. The fact that

$$\sum_{\ell=1}^{L} \mathrm{T}_{\ell,b}^*\mathrm{y}_\ell^{[i]} = \sum_{\ell \in \mathbb{L}_i} \mathrm{T}_{\ell,b}^*\mathrm{y}_\ell^{[i]} + \sum_{\ell \in \overline{\mathbb{L}}_i} \mathrm{T}_{\ell,b}^*\mathrm{y}_\ell^{[i]} \tag{3.7}$$

allows us to rewrite the iterations in (3.5) as shown by Algorithm 2, where we have introduced the notations

$$\mathrm{T}_{\mathbb{B}}\mathrm{w} = \Big(\mathrm{T}_\ell\,\mathrm{w}\Big)_{\ell \in \mathbb{B}} \tag{3.8}$$

$$\mathrm{T}_{\mathbb{L}}^*\,\mathrm{y}_{\mathbb{L}} = \sum_{\ell \in \mathbb{L}} \mathrm{T}_\ell^*\,\mathrm{y}_\ell. \tag{3.9}$$

The modified algorithm allows us to deal with the submatrix $\mathrm{T}_{\mathbb{L}_i}$ rather than with the complete matrix, reducing the computational cost of each iteration. Indeed, under the aforementioned technical assumptions, Algorithm 2 converges almost surely to a solution to (2.8). Similarly, we can derive the random updates w.r.t. the primal variables by inversing the order of proximity operations in (3.5), which leads to Algorithm 3.

4. EXPERIMENTS

In this section, we aim at showing that the standard FBPD in Algorithm 1 can be accelerated by using the randomized versions proposed in Algorithms 2 and 3. Moreover, in order to demonstrate the interest of our approach, we compare the execution times of the above algorithms with the randomized forward-backward (FB) method proposed in [20]. Note that the latter requires the loss function to be differentiable, which is not the case for the hinge loss in (2.5). Hence, in the FB method, we replace it with the squared hinge loss used in [13].

Algorithm 2 FBPD for solving Problem (2.8)

```
Initialization
    choose $\mathrm{w}^{[0]} \in \mathbb{R}^{MK}$ and $\mathrm{y}^{[0]} \in \mathbb{R}^{LK}$
    choose $\mathrm{u}^{[0]} = \sum_{\ell=1}^{L} \mathrm{T}_\ell^* \mathrm{y}_\ell^{[i]}$
    set $\tau > 0$ and $\sigma > 0$ such that $\tau\sigma \sum_{b=1}^{B} \sum_{\ell=1}^{L} \|\mathrm{T}_{\ell,b}\|^2 \leqslant 1.$
For $i = 0, 1, \dots$
    Set $\mathbb{L}_i \subset \{1, \dots, L\}$
    $\mathrm{w}^{[i+1]} = \mathrm{prox}_{\tau g}\big(\mathrm{w}^{[i]} - \tau\, \mathrm{u}^{[i]}\big)$
    $\mathrm{y}_{\mathbb{L}_i}^{[i+1]} = \mathrm{prox}_{\sigma h_{\mathbb{L}_i}^*}\Big(\mathrm{y}_{\mathbb{L}_i}^{[i]} + \sigma \mathrm{T}_{\mathbb{L}_i}\big(2\mathrm{w}^{[i+1]} - \mathrm{w}^{[i]}\big)\Big)$
    $\mathrm{y}_{\overline{\mathbb{L}}_i}^{[i+1]} = \mathrm{y}_{\overline{\mathbb{L}}_i}^{[i]}$
    $\mathrm{u}^{[i+1]} = \mathrm{u}^{[i]} + \mathrm{T}_{\mathbb{L}_i}^*\big(\mathrm{y}_{\mathbb{L}_i}^{[i+1]} - \mathrm{y}_{\mathbb{L}_i}^{[i]}\big)$
```

Algorithm 3 FBPD for solving Problem (2.8)

```
Initialization
    choose $\mathrm{w}^{[0]}$, $\mathrm{y}^{[0]}$, $\tau$ and $\sigma$ as in Algorithm 2
    choose $\mathrm{v}^{[0]} = \big(\mathrm{T}_\ell\, \mathrm{w}^{[0]}\big)_{1 \leqslant \ell \leqslant L}$
For $i = 0, 1, \dots$
    Set $\mathbb{B}_i \subset \{1, \dots, B\}$
    $\mathrm{y}^{[i+1]} = \mathrm{prox}_{\sigma h^*}\Big(\mathrm{y}^{[i]} + \sigma\, \mathrm{v}^{[i]}\Big)$
    $\mathrm{w}_{\mathbb{B}_i}^{[i+1]} = \mathrm{prox}_{\tau g_{\mathbb{B}_i}}\Big(\mathrm{w}_{\mathbb{B}_i}^{[i]} - \tau\, \mathrm{T}_{\mathbb{B}_i}^*\big(2\mathrm{y}^{[i+1]} - \mathrm{y}^{[i]}\big)\Big)$
    $\mathrm{w}_{\overline{\mathbb{B}}_i}^{[i+1]} = \mathrm{w}_{\overline{\mathbb{B}}_i}^{[i]}$
    $\mathrm{v}^{[i+1]} = \mathrm{v}^{[i]} + \mathrm{T}_{\mathbb{B}_i}\big(\mathrm{w}_{\mathbb{B}_i}^{[i+1]} - \mathrm{w}_{\mathbb{B}_i}^{[i]}\big)$
```

In our experiments, we consider the MNIST database [29] for the classification of handwritten digits. This dataset contains a number of 28×28 grayscale images ($N = 784$) displaying digits from 0 to 9 ($K = 10$). The database is organized in 60000 training images and 10000 test images. In our experiments, we define the mapping φ as the identity plus a bias (so as to obtain a linear SVM classifier with $M = N+1$), and we use the $\ell_{1,2}$-norm regularization. Finally, we train the classifier on a training set of size $L = 1000$, and we evaluate the trained classifiers on a separate test set of size 10000.

Table 1 collects the classification errors obtained by the aforementioned algorithms for various values of the regularization parameter λ. Moreover, Figure 1 shows the execution times obtained by the aforementioned algorithms for various values of the regularization parameter λ. In particular, the y-axis reports the quantity $\|\mathrm{w}^{[i]} - \mathrm{w}^{[\infty]}\|$, that is the distance of the current iterate $\mathrm{w}^{[i]}$ to the solution $\mathrm{w}^{[\infty]}$ computed after 500000 iterations, while the x-axis reports the execution time (in seconds) needed to compute i iterations. Finally, Table 2 reports the classification errors obtained with $\lambda = 10^{-3}$ (the choice leading to the best accuracy) for different sizes $L \in \{250, 500, 1000, 2000\}$ of the training set, along with the execution times obtained with the stopping criterion $\|\mathrm{w}^{[i+1]} - \mathrm{w}^{[i]}\| < 10^{-5}\|\mathrm{w}^{[i]}\|$. All the methods are implemented in MATLAB and executed on a Intel CPU at 3.20 GHz and 12 GB of RAM.

The above results demonstrate that the randomized FBPD algorithms are faster than the standard FBPD, while they compare favorably to the randomized FB. In particular, the fastest algorithm is the randomized FBPD with the block selection on the *dual variable*. This seems to be reasonable, since the considered training set contains a significant number of samples.

5. CONCLUSION

We have proposed two randomized algorithms for learning a sparse multiclass SVM: the first one selects the blocks to be updated over the features of the solution vector, while the second one performs the selection over the training samples. The ability to perform the random selection of training samples (instead of the features) is a specificity of our approach, since it is based on a clever use of a primal-dual method. In our opinion, this is the main advantage of the proposed approach with respect to the classical block-coordinate methods in [13, 19, 21].

6. REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Journal of Machine Learning*, vol. 20, no. 3, pp. 273–297, Sept. 1995.

[2] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. of ICML*, Madison, USA, 1998, pp. 82–90.

[3] F. R. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[4] L. Laporte, R. Flamary, S. Canu, S. Déjean, and J. Mothe, "Non-convex regularizations for feature selection in ranking with sparse SVM," *IEEE Trans. Neural Networks Learn. Sys.*, vol. 25, no. 6, June 2014.

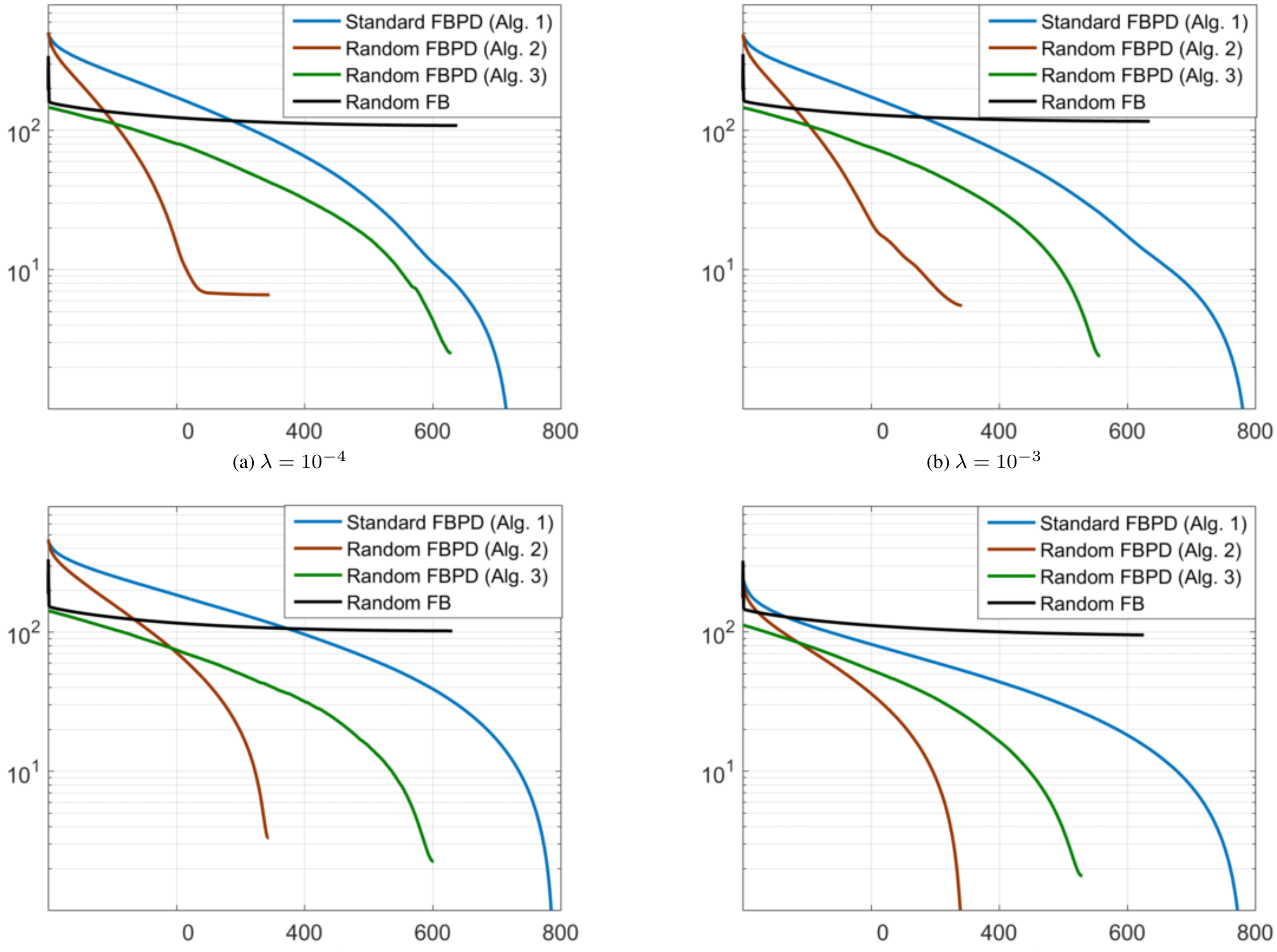
[5] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–392, 2001.

Table 1. Classification errors measured over a test set of 10000 samples.

λ	Standard FBPD (Alg. 1)	Random FBPD (Alg. 2)	Random FBPD (Alg. 3)	Random FB [20]
10^{-4}	13.11 %	13.40 %	14.43 %	13.72 %
10^{-3}	13.08 %	13.29 %	14.48 %	13.70 %
10^{-2}	13.17 %	13.25 %	14.65 %	13.86 %
10^{-1}	15.15 %	14.85 %	16.83 %	15.35 %

Table 2. Classification errors and execution times obtained with a different number of training samples.

L	Standard FBPD (Alg. 1)	Random FBPD (Alg. 2)	Random FBPD (Alg. 3)	Random FB [20]
250	20.62 % – 14.47 s.	20.76 % – 14.67 s.	22.17 % – 29.02 s.	21.60 % – 43.54 s.
500	16.49 % – 45.72 s.	16.78 % – 42.32 s.	17.87 % – 46.98 s.	17.41 % – 55.40 s.
1000	13.08 % – 75.45 s.	13.25 % – 49.69 s.	14.65 % – 54.70 s.	13.86 % – 64.55 s.
2000	11.11 % – 175.73 s.	11.22 % – 49.48 s.	12.49 % – 80.61 s.	11.77 % – 73.36 s.



(a) $\lambda = 10^{-4}$

(b) $\lambda = 10^{-3}$

(c) $\lambda = 10^{-2}$

(d) $\lambda = 10^{-1}$

Fig. 1. Convergence plots ($\|\mathrm{w}^{[i]} - \mathrm{w}^{[\infty]}\|$ vs time in sec.) on a training set of size $L = 1000$.

[6] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B*, vol. 68, pp. 49–67, 2006.

[7] L. Meier, S. Van De Geer, and P. Bühlmann, "The group Lasso for logistic regression," *Journal of the Royal Statistical Society: Series B*, vol. 70, no. 1, pp. 53–71, 2008.

[8] J. Duchi and Y. Singer, "Boosting with structural sparsity," in *International Conference on Machine Learning*, Montreal, Canada, 14-18 June 2009, pp. 297–304.

[9] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statistics and Computing*, vol. 20, no. 2, pp. 231–252, 2010.

[10] X. Wang, L. abd Shen, "On l_1-norm multi-class support vector machines: methodology and theory," *Journal of the American Statistical Association*, vol. 102, pp. 583–594, 2007.

[11] H.H. Zhang, Y. Liu, Y. Wu, and J. Zhu, "Variable selection for multicategory SVM via sup-norm regularization," *Electronic Journal of Statistics*, vol. 2, pp. 149–167, 2008.

[12] B. Z. Vatashsky and K. Crammer, "Multi class learning with individual sparsity," in *Proc. of IJCAI*, 2013, pp. 1729–1735.

[13] M. Blondel, K. Seki, and K. Uehara, "Block coordinate descent algorithms for large-scale sparse multiclass classification," *J. Mach. Learn.*, vol. 93, no. 1, pp. 31–52, Oct. 2013.

[14] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "A proximal approach for sparse multiclass SVM," 2014, submitted, arXiv:1501.03669.

[15] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "Epigraphical projection and proximal tools for solving constrained convex optimization problems," *Signal Image Video Process.*, vol. 9, no. 8, pp. 1737–1749, Nov. 2015.

[16] C. J. Hsieh, K. W. Chang, C. J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. of ICML*, 2008, pp. 408–415.

[17] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-coordinate Frank-Wolfe optimization for structural SVMs," in *Proc. of ICML*, 2013, vol. 28, pp. 53–61.

[18] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C Pesquet, J.-Y. Tourneret, A. Hero, and S. McLaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 224–241, Mar. 2016.

[19] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1, pp. 1–38, Apr. 2014.

[20] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 1221–1248, July 2015.

[21] P.L. Combettes and J.-C. Pesquet, "Stochastic approximations and perturbations in forward-backward splitting for monotone operators," *Pure and Applied Functional Analysis*, vol. 1, no. 1, pp. 13–37, Jan. 2016.

[22] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *J. Nonlinear Convex Anal.*, vol. 16, no. 12, pp. 2453–2490, 2015.

[23] O. Fercoq and P. Bianchi, "A coordinate descent primal-dual algorithm with large step size and possibly non separable functions," 2016, submitted, arXiv:1508.04625.

[24] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electronic Computers*, vol. EC-14, no. 3, pp. 326–334, June 1965.

[25] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[26] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds., pp. 185–212. Springer-Verlag, New York, 2011.

[27] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123231, 2014.

[28] L. Condat, "Fast projection onto the simplex and the l1 ball," *Mathematical Programming Series A*, vol. 158, no. 1, pp. 575–585, July 2016.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.