# Large Margin Classification with the Progressive Hedging Algorithm

**Boris Defourny**
EECS
University of Liège
Belgium
bdf@montefiore.ulg.ac.be

**Louis Wehenkel**
EECS
University of Liège
Belgium
L.Wehenkel@ulg.ac.be

## Abstract

Several learning algorithms in classification and structured prediction are formulated as large scale optimization problems. We show that a generic iterative reformulation and resolving strategy based on the progressive hedging algorithm from stochastic programming results in a highly parallel algorithm when applied to the large margin classification problem with nonlinear kernels. We also underline promising aspects of the available analysis of progressive hedging strategies.

## 1 Introduction

Decomposition methods are a key ingredient for solving learning problems on large data sets, especially in the field of Support Vector Machines [20, 13, 8]. At the same time, research in optimization often strives to identify algorithms with the best potential for parallel computations [5].

This paper presents a strategy for splitting Support Vector Machines learning problems along the training samples, based on the so-called progressive hedging algorithm (PHA), initially developed for the field of stochastic programming, and parallel by nature [12]. Besides the formal connection with stochastic programming, this paper conveys the progressive hedging strategy to large margin classification in Reproducing Kernel Hilbert Spaces. The resulting algorithm has a linear convergence rate, with inner iteration complexity linear in the size of the data set, except for a single matrix-vector multiplication. With respect to [7], based on a similar dual ascent strategy but without augmented Lagrangians, the present approach allows to derive stopping criteria and handles directly the bias term; more importantly it paves the way towards complexity improvements, as the proximal point theory on which the analysis of the progressive hedging relies quantifies how approximately inner iteration problems could be solved. We believe that guarantees of convergence despite approximations and mild non-convexities will be especially valuable in the context of structured prediction problems, even if here we only investigate what kind of algorithm emerges from ideal conditions.

The paper is organized as follows. Section 2 presents background material. Section 3 presents the generic decomposition strategy and the progressive hedging algorithm. Section 4 develops the calculations for large margin classification, and Section 5 concludes.

## 2 Background

This section recalls that structured prediction is an ill-posed problem, and outlines in such a context the relevance of proximal point methods — on which progressive hedging is based precisely.

**Structured Prediction:** Structured prediction consists in learning a mapping $h : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y}$ has a complex structure, for instance some input dependent $\mathcal{Y}(x)$ for $x \in \mathcal{X}$. With $\mathbb{P}$ the unknown distribution generating the input-output pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ a loss function

penalizing wrong predictions and adapted to the nature of $\mathcal{Y}$, learning means this: Given only a finite sample set $\mathcal{D}_n \in (\mathcal{X} \times \mathcal{Y})^n$ drawn from $\mathbb{P}^n$, find, ideally, the element $h$ of a hypothesis space $\mathcal{H}$ that minimizes the risk

$$\mathcal{R}_{\mathbb{P}}(h) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\{\ell(y, h(x))\} \ . \tag{1}$$

**Empirical Risk Minimization as an Ill-Posed Problem:** Sample sets $\mathcal{D}_n$ allow to approximate the unknown $\mathbb{P}$ by a finite approximation $\hat{\mathbb{P}}$. Replacing $\mathbb{P}$ by $\hat{\mathbb{P}}$ in (1) yields the empirical risk. However, for a fixed size $n$, small perturbations of the sample set may result in a large perturbation of the empirical risk minimizer. Vapnik recognized that learning from finite data sets is thus in practice an *ill-posed problem*. He then chose to built on the regularization method proposed by Tikhonov for addressing ill-posed problems [17], leading to the development of support vector machines (SVM) [19].

**Large Margin Approach to Structured Prediction:** A successful approach to structured prediction [16, 18] consists in letting the structure result from parametric optimization. Given vector-valued joint feature functions $\phi(x, y)$, weights $w \in \mathcal{W}$, and the choice of a hypothesis space $\mathcal{H}$ such that

$$h(x;\ w) = \arg\max_{y\in\mathcal{Y}(x)} w^T\phi(x, y) \ , \tag{2}$$

learning $h \in \mathcal{H}$ reduces to learning $w \in \mathcal{W}$. Note that $y \in \mathcal{Y}(x)$ may actually represent a large system of constraints, so that the task of evaluating $h$ may be far from trivial.

In line with SVMs, the learning problem can be cast as a margin maximization problem. For instance, we recall the formulation $\text{SVM}_1^{\Delta s}$ proposed by Tsochantaridis et al. [18] with $C > 0$ and slack variables $\xi_i$:

$$\begin{array}{ll} \min_{w,\,\xi} & \frac{1}{2}||w||^2 + (C/n)\sum_{i=1}^n \xi_i,\ s.t.\ \forall i,\ \xi_i \geq 0 \ , \\ & \forall i,\ \forall y \in \mathcal{Y} \setminus \{y_i\} :\ w^T[\phi(x_i, y_i) - \phi(x_i, y)] \geq 1 - \xi_i/\ell(y_i, y) \ . \end{array} \tag{3}$$

**Proximal Point Methods:** Proximal point methods solve optimization programs iteratively from a sequence of regularized programs. According to these methods that provide a unified treatment of several convex programming techniques [6], a solution to a convex program $\min_{x\in X}\{f(x)\}$, with $x$ valued in a Hilbert space normed by $||\cdot||$, can be obtained, assuming that its solution set is nonempty, as the limit point of a sequence $x^\nu$ defined as follows: given a sequence $\{c^\nu\}_{\nu=0}^\infty \subset (0, \bar{c}]$ with $\bar{c} < \infty$ and given an arbitrary initial point $x^0$,

$$x^{\nu+1} \simeq \arg\min_{x\in X}\{f(x) + \frac{c^\nu}{2}||x - x^\nu||^2\} \ . \tag{4}$$

The convergence of $x^\nu$ to a point in the solution set of the original program can be guaranteed provided $x^{\nu+1}$ is close enough to the exact solution of the program in (4). One possible criterion initially studied in [11] is: there exists a sequence $\delta^\nu$ such that

$$||x^{\nu+1} - \arg\min_{x\in X}\{f(x) + \frac{c^\nu}{2}||x - x^\nu||^2\}|| \leq \delta^\nu||x^{\nu+1} - x^\nu|| \ , \qquad \sum_{\nu=0}^\infty \delta^\nu < \infty \ . \tag{5}$$

The stability of the scheme relies on the regularizing effect of the quadratic term in (4). The leeway of using inexact solutions for $x^{\nu+1}$ was initially moderated by the need for controlling the errors through conditions such as (5). However, advances have been made in inexact schemes allowing non-diminishing errors [14]. One will also appreciate, for ill-posed problems, the impact of the regularizing effect itself, viewed as a by-product of the scheme.

## 3 Decomposition Strategy

The decomposition strategy that we propose relies on the idea that a program like (3) may be *split* into $n$ regularized programs $\mathcal{P}_i^\nu$ dedicated to each sample $(x_i, y_i) \in \mathcal{D}_n$. To each variable shared among the samples are added $n$ artificial copies proper to each sample $i$. Programs $\mathcal{P}_i^\nu$ are solved over their own variables, but iterative modifications of the programs will make duplicated variables converge towards a unique value as $\nu \to \infty$.

The modifications are done according to an augmented Lagrangian strategy [2] coupled with an averaging scheme that projects the solution on a particular subspace. In fact, the full solution path followed by the algorithm provides regularized solutions that could be ranked according to a model selection strategy.

### 3.1 Separable Reformulation

Consider an initial convex program

$$\min_{w,\,\xi_i} \quad f(w) + \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(\xi_i), \quad \text{s.t.} \quad \forall\, i \,:\, \xi_i \in \Xi_i(w), \quad w \in W \ . \tag{6}$$

Note that (3) is of the form (6) by identifying

$$f(w) = ||w||^2/2, \quad f_i(\xi_i) = C\,\xi_i, \quad W = \mathbb{R}^m,$$
$$\Xi_i(w) = \{\xi_i \geq 0 :\ \forall y \in \mathcal{Y} \setminus \{y_i\},\ w^T[\phi(x_i, y_i) - \phi(x_i, y)] \geq 1 - \xi_i/\ell(y_i, y)\} \ .$$

Consider also

$$\min_{w_i,\,\xi_i} \quad \tfrac{1}{n}\textstyle\sum_{i=1}^n [\, f(w_i) + f_i(\xi_i)\,], \quad \text{s.t.} \quad \forall\, i \,:\, \xi_i \in \Xi_i(w_i), \quad w_i \in W, \tag{7a}$$

$$w_i = \tfrac{1}{n}\textstyle\sum_{j=1}^n w_j \ . \tag{7b}$$

Programs (6) and (7a-7b) are equivalent since (7b) enforces $w_1 = w_2 = \cdots = w_n$. However, the second formulation is such that the objective and constraints (7a) are separable in $(w_i, \xi_i)$, while (7b) are coupling constraints. For brevity, define $g_i(w_i, \xi_i) = f(w_i) + f_i(\xi_i)$.

Now, the reformulation (7a-7b) has the structure of a so-called *two-stage stochastic program with recourse* [3], with *first-stage decisions* $w_i = w$ constant with $i$, and *second-stage decisions* $\xi_i$ adapted to $i$. This observation allows us to adapt techniques initially developed in the context of stochastic programming. In particular, we will exploit the decomposition approach of the *progressive hedging algorithm* (PHA) of Rockafellar and Wets [12] for solving stochastic programs, closely related to the method of partial inverses of Spingarn [15] and the alternating linearization approach of Kiwiel et al. [9].

### 3.2 Progressive Hedging

The algorithm introduces variables $\mu_i^\nu$, $1 \leq i \leq n$. They can be interpreted as messages transmitted to samples $i$ at each iteration $\nu$ so as to converge to a unique solution $w$.

**Definition 1 (Progressive Hedging Algorithm)** *Let $c > 0$, $\epsilon > 0$; $w^\nu, \mu_i^\nu \in \mathbb{R}^m\ \forall\, i$.*

1. *Initialization step: Set $w^0 = 0$, $\mu_i^0 = 0$. Set $\nu = 0$.*
2. *Solving step: For each $i$, solve approximately*

   $$\mathcal{P}_i^\nu : \min_{w_i,\,\xi_i} \quad g_i(w_i, \xi_i) + w_i^T\mu_i^\nu + \frac{c}{2}||w_i - w^\nu||^2 \quad \textit{s.t.} \quad \xi_i \in \Xi_i(w_i), \quad w_i \in W \ . \tag{8}$$

   *Let $(w_i^\dagger, \xi_i^\dagger)$ denote a near-optimal solution to $\mathcal{P}_i^\nu$.*
3. *Averaging step: Define $\bar{w} = \frac{1}{n}\sum_{i=1}^n w_i^\dagger$.*
4. *Update step: Set $w^{\nu+1} = \bar{w}$. Set $\mu_i^{\nu+1} = \mu_i^\nu + c\,(w_i^\dagger - \bar{w})$.*
5. *Termination step: Stop if $||w^{\nu+1} - w^\nu||^2 + \frac{1}{n}\sum_{i=1}^n c^{-2}||\mu_i^{\nu+1} - \mu_i^\nu||^2 < \epsilon$. Otherwise, set $\nu$ to $\nu + 1$ and return to step 2.*

Note that the update step for the messages $\mu_i^\nu$ is akin to a gradient ascent step. Thus, it should not come as a surprise that the convergence is rather slow, but nearly independent of the problem dimension. We will argue, following [1], that for ill-posed problems, early stopping of gradient ascent iterations is in fact appropriate, insofar as high-accuracy solutions do not generalize well, and following [4], that for large scale learning, convergence rates do not fully determine performances.

Three propositions adapted from [12] describe the ideal behavior of the algorithm.

**Proposition 1** *Assume that the programs $\mathcal{P}_i^\nu$ are solved exactly, and denote by $(w_i^*, \xi_i^*)$ the unique minimizer of $\mathcal{P}_i^\nu$. Recall the definition of $w^{\nu+1}$ and $\mu_i^{\nu+1}$ from the update step. Then the sequence $(w^\nu, \{\mu_i^\nu\}_{i=1}^n)$ for $\nu = 1, 2, \ldots$ converges to an optimal point, in the sense that the accumulation point $w^\infty$ is optimal for (7a-7b), while $\{\mu_i^\infty\}_{i=1}^n$ forms an optimal solution to the Lagrangian dual of (7a-7b). In particular, $w^\infty$ is optimal for* (6)*.*

**Proposition 2** *Assume that the programs $\mathcal{P}_i^\nu$ are solved exactly. Define the distance of the iterate $(w^\nu, \{\mu_i^\nu\}_{i=1}^n)$ to its limit point as*

$$\delta_c^\nu = [\tfrac{1}{n}\textstyle\sum_{i=1}^n \big(\ ||w^\nu - w^\infty||^2 + c^{-2}||\mu_i^\nu - \mu_i^\infty||^2\ \big)]^{1/2}\ .$$

*Then, as $\nu \to \infty$, the sequence $\delta_c^\nu$ strictly decreases to 0.*

**Proposition 3** *Assume that the programs $\mathcal{P}_i^\nu$ are solved exactly. If the objective functions are linear-quadratic and the feasible sets $\Xi_i(w)$ are polyhedral, and if the original program (7a-7b) and its dual have unique optimal solutions, then the convergence rate is linear, in the sense that there is some $\theta_c \in [0,1)$ such that $\delta_c^{\nu+1} \le \theta_c\, \delta_c^\nu$.*

## 4 Application to Large Margin Binary Classification

We set $\mathcal{X} = \mathbb{R}^m$, $\mathcal{Y} = \{+1, -1\}$, and focus on the large margin classification problem

$$\max_{w,\,b,\,\xi_i} \quad \tfrac{1}{2}||w||^2 + \tfrac{C}{n}\textstyle\sum_{i=1}^n \xi_i \quad s.t.\ \forall i:\ y_i(w^T x_i + b) \ge 1 - \xi_i, \quad \xi_i \ge 0\ .$$

Since $w$ and $b$ must be constant with $i$, the averaging and termination steps in the PHA now involve $w$, $b$ and the variables (copies and multipliers) associated with them.

To the variables $w$ and $b$ we associate the copies $w_i$, $b_i$ and the multipliers $\mu_i \in \mathbb{R}^m$ and $\beta_i \in \mathbb{R}$ respectively. The programs $\mathcal{P}_i^\nu$ in (8) become

$$\begin{array}{ll} \min_{w_i,\,b_i,\,\xi_i} & \tfrac{1}{2}||w_i||^2 + C\xi_i + w_i^T\mu_i^\nu + b_i\beta_i^\nu + \tfrac{c}{2}||w_i - w^\nu||^2 + \tfrac{c}{2}||b_i - b^\nu||^2 \\ \text{s.t.} & y_i(w_i^T x_i + b_i) \ge 1 - \xi_i, \quad \xi_i \ge 0\ . \end{array} \tag{9}$$

By considering the dual, one obtains an analytical solution to (9). Denoting the projection operator on the interval $[0, C]$ by $\Pi_{[0,C]}\{x\} = \max\{0, \min\{C, x\}\}$, one has

$$\alpha_i^* = \Pi_{[0,C]}\left\{\frac{y_i\,[\,x_i^T(\mu_i^\nu - c\,w^\nu)/(1+c) + (\beta_i^\nu - c\,b^\nu)/c\,] + 1}{||x_i||^2/(1+c) + 1/c}\right\} \tag{10}$$

$$w_i^*(\alpha_i^*) = (\alpha_i^* y_i x_i - \mu_i^\nu + c\,w^\nu)/(1+c) \qquad b_i^*(\alpha_i^*) = (\alpha_i^* y_i - \beta_i^\nu + c\,b^\nu)/c\ . \tag{11}$$

It is licit to convey augmented Lagrangian methods to Hilbert spaces thanks to their connection with proximal point methods [10]. For classification in a Reproducing Kernel Hilbert Space (RKHS) induced by a positive definite kernel $k$, we proceed as follows. We take as induction hypothesis the representations

$$w^\nu(\cdot) = \textstyle\sum_{j=1}^n \omega_j^\nu k(x_j, \cdot) \qquad w_i^*(\cdot) = \sum_{j=1}^n \omega_{ij}^* k(x_j, \cdot) \qquad \mu_i^\nu(\cdot) = \sum_{j=1}^n \pi_{ij}^\nu k(x_j, \cdot)\ ,$$

which holds true at iteration $\nu = 0$ with $\omega_j^0 = 0$, $\pi_{ij}^0 = 0$. From (10), (11) we establish the expressions of $\alpha_i^*, \omega_{ij}^*$ and obtain, by the PHA steps, update formulae for $\omega_j^{\nu+1}$ and $\pi_{ij}^{\nu+1}$:

$$\alpha_i^* = \Pi_{[0,C]}\left\{\frac{\sum_{j=1}^n y_i a_{ij} K(x_i, x_j) + y_i\,(\beta_i^\nu/c - b^\nu) + 1}{k(x_i,x_i)/(1+c) + 1/c}\right\} \quad \text{with } a_{ij} = \frac{\pi_{ij}^\nu - c\,\omega_j^\nu}{1+c}\ ,$$

$$\omega_{ij}^* = \frac{\alpha_i^* y_i \delta_{ij} + c\,\omega_j^\nu - \pi_{ij}^\nu}{1+c} \quad \text{with } \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \ne j \end{cases}, \qquad \begin{array}{l} \omega_j^{\nu+1} = \frac{1}{n}\sum_{i=1}^n \omega_{ij}^*\ , \\ \pi_{ij}^{\nu+1} = \pi_{ij}^\nu + c\,(\omega_{ij}^* - \omega_j^{\nu+1})\ . \end{array}$$

The induction hypothesis is thus verified. The expression of $b^{\nu+1}$ will come from (11) with the new $\alpha_i^*$ and from the PHA steps.

Now, besides the Gram matrix $K$ defined by $K_{ij} = k(x_i, x_j)$, storage requirements can be made linear in $n$. Let $\mathbf{1} \in \mathbb{R}^n$ denote a column vector of ones, $\mathrm{diag}\{z\}$ the diagonal matrix with diagonal elements $z_i$, and check by induction that $\pi_{ij}^\nu$, $\omega_{ij}^*$ can always be written in matrix form as

$$[\pi_{ij}^\nu] = \mathbf{1}\,{\pi_A}^T + \mathrm{diag}\{\pi_B\}\ , \qquad\qquad [\omega_{ij}^*] = \mathbf{1}\,{\omega_A^*}^T + \mathrm{diag}\{\omega_B^*\}\ ,$$

for some $\pi_A$, $\pi_B$, $\omega_A^*$, $\omega_B^* \in \mathbb{R}^n$. Notice that $\omega_A^*$ can be evaluated prior to $\alpha^*$. Having only vectors in $\mathbb{R}^n$ also simplifies the expression of the norms in the termination step (calculations are lengthy and omitted here). Hence the final form of the algorithm, with $\Pi_{[0,C]}\{\cdot\}$ now understood componentwise, and $\odot$ denoting the componentwise product.

**Definition 2 (Kernel Progressive Hedging for Large Margin Classification)**

---

**Inputs:** $C > 0$, *Gram matrix* $K \in \mathbb{R}^{n \times n}$, *target labels* $y \in \mathbb{R}^n$ *with* $y_i \in \{+1, -1\}$.
**Algorithm Parameters:** *Proximal parameter* $c > 0$, *termination tolerance* $\epsilon > 0$.
**Outputs:** *Weights* $\omega^\nu \in \mathbb{R}^n$, *bias* $b^\nu \in \mathbb{R}$ *for the classifier* $\hat{y}(x) = sign\{\sum_{i=1}^n \omega_i k(x_i, x) + b\}$.

---

*Set* $\nu$ *to 0. Initialize vectors* $\alpha, \omega_A^*, \omega_B^*, \omega^\nu, \pi_A^\nu, \pi_B^\nu, b^*, \beta^\nu \in \mathbb{R}^n$ *to 0. Initialize* $b^\nu \in \mathbb{R}$ *to 0.*
*Define constants* $g \in \mathbb{R}^n$: $g_i = K_{ii}$, $d \in \mathbb{R}^n$: $d_i = (K_{ii}/(1+c) + 1/c)^{-1}$ .
*Repeat for* $\nu = 0, 1, \ldots$

$$\omega_A^* = \frac{c\,\omega^\nu - \pi_A^\nu}{1+c} \quad , \quad \alpha = \Pi_{[0,C]}\left\{ d \odot \left[ y \odot \left( \frac{g \odot \pi_B^\nu}{1+c} - K\,\omega_A^* \right) + \beta^\nu / c - b^\nu\,\mathbf{1} \right] + \mathbf{1} \right\} ,$$

$$\omega_B^* = \frac{y \odot \alpha - \pi_B^\nu}{1+c} \quad ,$$

$$\omega^{\nu+1} = \omega_A^* + \frac{1}{n}\,\omega_B^* \ , \quad \pi_A^{\nu+1} = \pi_A^\nu - \frac{c}{n}\,\omega_B^* \ , \quad \pi_B^{\nu+1} = \pi_B^\nu + c\,\omega_B^* \ ,$$

$$b^* = \frac{y \odot \alpha - \beta^\nu}{c} + b^\nu\,\mathbf{1} \ , \quad b^{\nu+1} = \frac{\mathbf{1}^T b^*}{n} \ , \quad \beta^{\nu+1} = \beta^\nu + c\,(b^* - b^{\nu+1}\,\mathbf{1})$$

*until the stopping criterion is met, either the exact but quadratic-in-*$n$ *criterion:*

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 < \epsilon \ ,$$

*where* $\delta_1 = (\omega^{\nu+1} - \omega^\nu)^T K (\omega^{\nu+1} - \omega^\nu)$ , $\quad \delta_2 = \frac{1}{n} g^T (\omega_B^* \odot \omega_B^*) - \frac{1}{n^2} {\omega_B^*}^T K \omega_B^*$ ,

$$\delta_3 = (b^{\nu+1} - b^\nu)^2 \ , \quad \delta_4 = \frac{1}{n}(b^* - b^{\nu+1}\mathbf{1})^T (b^* - b^{\nu+1}\mathbf{1}) \ ,$$

*or the following approximate criterion:* $\quad \frac{1}{n} g^T (\omega_B^* \odot \omega_B^*) < \epsilon$ .

---

**Proposition 4** *The criterion* $\frac{1}{n} g^T (\omega_B^* \odot \omega_B^*) < \epsilon$ *is an admissible stopping criterion. It reduces to* $||\omega_B^*||^2 < n\epsilon$ *if the kernel is normalized (that is, if* $g_i = K(x_i, x_i) = 1$*).*

*Proof:* The algorithm enforces progressively $b_i = b$ and $w_i(\cdot) = w(\cdot)$ in the RKHS by enforcing progressively $b_i^* = b^\nu$ for all $i$ and $\omega_{ij}^* = \omega_j^\nu$ for all $i, j$. In matrix form, at convergence we have in particular $[\omega_{ij}^*] = \mathbf{1}{\omega_A^*}^T + \text{diag}\{\omega_B^*\} = \mathbf{1}\omega^{\nu T}$, implying $\omega_B^* = 0$. □

The decreasing property of the exact stopping criterion is lost. If $\epsilon$ is large, one may need to run a minimal number of iterations before checking the approximate stopping condition.

**Choice of the proximal parameter** $c$ **:** Choosing $c$ is subject to a trade-off, because the effect of $c$ on the advances towards the primal and the dual solutions are antagonist (as noted in [12], Prop. 5.3). Concretely, observe from Def. 2 that $c$ intervenes in the updates of $\omega_B^*$ and $\pi_B^*$ in the denominator and numerator respectively. Using the leeway provided by the proximal point theory, we propose the following dynamical update rule for $c$ : Choose lower and upper bounds $0 < \underline{c} \leq \overline{c} \leq +\infty$ , and then prior to each iteration $\nu$, define, with componentwise square root and absolute value,

$$c^\nu = \max\left\{\underline{c},\ \min\left\{\overline{c},\ \sqrt{|\omega_B^*|\,|\pi_B^*|}\right\}\right\} \ .$$

Then replace $c$ in Def. 2 by $c^\nu \in \mathbb{R}^n$ with componentwise divisions.

**Complexity:** Evaluating $\delta_1$, the second term of $\delta_2$, and $K\omega_A^*$ is quadratic in $n$. The first stopping criterion is the one backed by the theory. However, the evolution of the distance to the solution is well captured by the first term of $\delta_2$, at least after a few iterations. Hence our suggestion of the second stopping criterion, justified a posteriori by Prop. 4. With this second criterion, at this stage the only operation which is not linear in $n$ is the evaluation of the matrix-vector multiplication $p = K\omega_A^*$.

**Distributed kernel evaluations:** Although the exact evaluation of $p = K\omega_A^*$ is at worst quadratic in $n$ for dense Gram matrices, it is not too hard to come up with a scheme for distributing the calculations among different machines, by exploiting block decompositions of the Gram matrix, and storing the blocks locally.

## 5 Conclusion

We have shown on the large margin classification problem the benefits of a decomposition approach from stochastic programming. The analysis of the progressive hedging strategy is based on the theory of proximal point methods; we leave as future work improvements in the complexity of the resulting algorithms by combining approximations in the evaluation of inner iterations and advances in the field of proximal point theory itself.


### Acknowledgments

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.


## References


[1] A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with application to tomography. *SIAM Journal on Optimization*, pages 79–108, 2001.

[2] D. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic Press, 1982.

[3] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.

[4] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 20 (NIPS-2007)*, pages 161–168, 2008.

[5] J. Eckstein. *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*. PhD thesis, Massachusetts Institute of Technology, 1989.

[6] J. Eckstein. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.

[7] T. Hazan, A. Man, and A. Shashua. A parallel decomposition solver for SVM: Distributed dual ascend using Fenchel duality. In *IEEE Conference of Computer Vision and Pattern Recognition*, 2008.

[8] T. Joachims, T. Finley, and C.N.J. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77:27–59, 2009.

[9] K.C. Kiwiel, C.H. Rosa, and A. Ruszczyński. Proximal decomposition via alternating linearization. *SIAM Journal on Optimization*, 9:153–172, 1999.

[10] R.T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976.

[11] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 15:877–898, 1976.

[12] R.T. Rockafellar and R.J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematical Programming*, 16:119–147, 1991.

[13] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML-2007)*, pages 807–814, 2007.

[14] M. Solodov and B. Svaiter. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Analysis*, 7:323–345, 1999.

[15] J.E. Spingarn. Application of the method of partial inverses to convex programming: Decomposition. *Mathematical Programming*, 32:199–233, 1985.

[16] B. Taskar, S. Lacoste-Julien, and M.I. Jordan. Structured prediction, dual extragradient and Bregman projections. *Journal of Machine Learning Research*, 7:1627–1653, 2006.

[17] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill posed problems*. W.H. Winston and Sons (distributed by Wiley), 1977.

[18] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[19] V.N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[20] L. Zanni, T. Serafini, and G. Zanghirati. Parallel software for training large scale Support Vector Machines on multiprocessor systems. *Journal of Machine Learning Research*, 7:1467–1492, 2006.