# Nonsmooth optimization techniques for semi–supervised classification[*]

A. Astorino[§]  A. Fuduli[**]

**Abstract**

We apply nonsmooth optimization techniques to classification problems, with particular reference to the TSVM (Transductive Support Vector Machine) approach, where the considered decision function is nonconvex and nondifferentiable and then difficult to minimize.

We present some numerical results obtained by running the proposed method on some standard test problems drawn from the binary classification literature.

**Keywords**: semi–supervised learning, nonsmooth optimization, bundle methods.

## 1 Introduction

In the last years a remarkable research work has been performed in different related disciplines aiming at conceiving general models for machine learning. Some research issues in this area are pattern classification, regression estimation and novelty detection. In particular, many mathematical techniques have been proposed to improve the generalization of the classification models and to reduce, at the same time, the computational cost.

The objective of pattern classification (see [9, 23, 25, 26]) is in fact to categorize data into different classes and several algorithms have been devised for automatically distinguishing among different samples, on the basis

[*] This work has been developed within the project *Matematica e Diagnosi Medica* by "Centro E. Fermi". The research has been partially supported by the Italian "Ministero dell'Istruzione, dell'Università e della Ricerca Scientifica", under FIRB project *Large Scale Nonlinear Optimization* (RBNE01WBBB).

[§] Istituto di Calcolo e Reti ad Alte Prestazioni C.N.R., c/o Dipartimento di Elettronica Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS), Italia. E-mail: astorino@icar.cnr.it

[**] Dipartimento di Matematica, Università della Calabria, 87036 Rende (CS), Italia. E-mail: antonio.fuduli@unical.it

of their patterns. The number of applications that require a categorization is very large. Some examples are text and web classification, object recognition in machine vision, gene expression profile analysis, DNA and protein analysis and many others.

More formally, given a set of labelled and/or unlabelled objects characterized by some features, the task of pattern classification is to express, for each object, one particular feature, the class label, as a function of the remaining ones. We refer to this function as to the "prediction function". On the basis of the available information coming from the dataset, the aim of a pattern classification algorithm is to construct a good prediction function, in terms of number of misclassified objects.

In this paper, we consider a mathematical programming approach for the semi–supervised classification. As the name suggests, the semi–supervised classification is a compromise between supervised and unsupervised classification, whose objective is to take advantage from both of them.

In the supervised case, most of the learning models apply the inductive inference concept, where the prediction function, derived only from the labelled input data, is used to predict the label of any future object. A well known supervised classification technique is the Support Vector Machine (SVM) approach [9, 23], where a classifier, the so–called support vector machine, is constructed by generating a hyperplane far away from the labelled objects. More specifically, in the binary classification, where the aim is to distinguish between two different classes, the output of a SVM model is a hyperplane staying in the middle between two parallel hyperplanes, each of them supporting, respectively, one class. The distance between these two parallel hyperplanes is called the margin: it is a measure of the generalization capability of the SVM. We remark that a considerable drawback of the supervised learning algorithms is that they require a large number of labelled training data in order to construct accurate classifiers.

On the other hand, in the unsupervised classification, since all the objects are unlabelled, the training is not supervised and the aim is to cluster the data, on the basis of their similarity.

In the semi–supervised learning only partial information is available about the data labels. In particular, referring to the training set as to the set of the labelled objects and to the testing set as to the set of the unlabelled objects, the basic idea is to construct the classifier on the basis of the information coming from both of them. In this sense the semi–supervised learning models apply the transductive inference concept [26], where the prediction function is derived from the information concerning all the available data, i.e. both the training and the testing sets. This function is not aimed at predicting the class label for newly incoming samples, but only at making a decision about the currently available objects. This is the case of the Transductive Support Vector Machine (TSVM) approach whose variants are implemented in [4, 7, 13, 15], where some knowledge on the testing set is taken into account during the training procedure.

There are many real–world problems where labelling often requires fairly expensive human labor, whereas unlabelled data are abundant being easier to obtain. Some examples are medical diagnosis, web categorization, text processing and bioinformatics: in these cases the semi–supervised learning is very useful and has been recently object of a remarkable research work.

When the labelled points are relatively few, a frequent drawback in the classification is the possibility to overfit the training data with a consequent loss of generality. For this reason, the key idea of the semi–supervised classification is to use, in the training phase, also the unlabelled data to improve the generalization.

In literature, different algorithms have been proposed for semi–supervised learning such as large margin classifiers [4, 7, 13, 15], graph based approaches [2, 7, 16, 28, 29], algorithms based on a preliminary clustering step [5, 13] and their combinations. In this paper we propose to use the TSVM approach to solve semi–supervised classification problems. In particular, we adopt the model described in [7] focusing on the peculiar features of the optimization problem to be solved, which can be formulated as a nonsmooth nonconvex and unconstrained minimization problem. To tackle this problem we use a descent method derived from the one described in [11].

The paper is organized as follows. In section 2 we analyze the TSVM objective function and study its characteristics. In section 3 we describe a bundle approach for minimizing this objective function. Finally, we present the numerical results obtained by running the proposed method on some standard test problems drawn from the binary classification literature. Some conclusions are reported in the last section.

Throughout the paper we adopt the following notations. We denote by $\|\cdot\|$ the Euclidean norm in $\mathbb{R}^n$ and by $a^Tb$ the inner product of the vectors $a$ and $b$. Moreover the *subdifferential* of a convex function $f$ at any point $x$ is denoted by $\partial f(x)$. We recall that the sudifferential of a convex function $f$ at the point $x$ is the set of the *subgradients* of $f$ at $x$, i.e. the set of vectors $g \in \mathbb{R}^n$ satisfying the *subgradient inequality*

$$f(y) \geq f(x) + g^T(y - x) \quad \forall\ y \in \mathbb{R}^n.$$

## 2 Transductive Support Vector Machine

Most of the semi–supervised learning algorithms utilize the cluster assumption, which can be stated in two equivalent ways:

- two points that are connected by a path going through high density regions should have the same label;
- the decision boundary should lie in a low density region.

This is achieved in the TSVM approach.

Different mathematical programming TSVM models exist in literature. We mention here the contributions by Joachims [15], Bennett & Demiriz [4], Fung & Mangasarian [13] and Chapelle & Zien [7]. In all such works the basic idea is to obtain the best support vector machine on the basis of the labelled data, having as few as possible unlabelled points in the margin zone. This is obtained by penalizing all the hyperplanes characterized by having unlabelled points in the margin area.

In particular, given a training set of $p$ samples

$$\{(x_1, y_1), (x_2, y_2), ..., (x_p, y_p)\}$$

with $x_i \in \mathbb{R}^n$, $i = 1, \dots p$, and $y_i \in \{-1, 1\}$ being a binary label, for $i = 1, ..., p$, and a testing set of $q$ unlabelled samples in $\mathbb{R}^n$

$$\{x_{p+1}, x_{p+2}, ..., x_{p+q}\},$$

the problem of finding a hyperplane far away from both the labelled and unlabelled points can be formulated as follows:

$$\left\{ \begin{array}{ll} \displaystyle\min_{w,b,\xi} & \displaystyle\frac{1}{2}\|w\|^2 + C_1 \sum_{i=1}^{p} \xi_i + C_2 \sum_{i=p+1}^{p+q} r(w^T x_i + b) \\ & y_i[w^T x_i + b] \geq 1 - \xi_i \qquad i = 1, ..., p, \end{array} \right. \tag{1}$$

where $r$ is the margin penalty function involving the unlabelled data and $C_1$ and $C_2$ are positive weight parameters. In [4, 7, 13, 15] function $r$ is chosen as

$$r(t) = \left\{ \begin{array}{ll} 1 - |t| & \text{for } -1 < t < 1 \\ 0 & \text{otherwise,} \end{array} \right. \tag{2}$$

and the proposed algorithms are characterized by different approaches to solve problem (1).

The algorithm proposed in [15] adopts first the supervised SVM approach for labelling the objects of the testing set. Then the current solution is improved by switching the labels of some testing samples, selected on the basis of appropriate heuristic techniques.

In [4] the authors formulate the semi–supervised support vector machine problem as a mixed integer program. Since they introduce a binary variable for each unlabelled point, the problem can be difficult to solve for a large number of unlabelled data. To avoid this difficulty, in [13] a concave minimization problem is tackled and a stationary point is found by solving successive linear programs.

Choosing (2) as the margin penalty function, problem (1) can be rewritten in the following unconstrained form:

$$\min_{w\in\mathbb{R}^n,\ b\in\mathbb{R}} h(w,b), \tag{3}$$

where

$$h(w,b)=\frac{1}{2}\|w\|^2+C_1\sum_{i=1}^{p}\max\{0,1-y_i w^T x_i+b\}+C_2\sum_{i=p+1}^{p+q}\max\{0,1-|w^T x_i+b|\}.$$

In passing from formulation (1) to formulation (3) the resulting problem comes out to be of the unconstrained type. This seems advantageous especially when the input dataset is very large. On the other hand, the main drawback is that function $h$ is not differentiable and, moreover, due to the third term involving the unlabelled points, it is even nonconvex.

A method for solving problem (3) is reported in [7], where the authors perform a standard gradient descent method on a smooth approximation of the objective function. Our approach is, indeed, to adopt some recently proposed methods [11] of the bundle type, which are capable to cope with both nonconvexity and nonsmoothness.

## 3 Bundle methods for nonsmooth minimization

Bundle methods have been initially conceived by Lemaréchal [18] and Wolfe [27] for minimizing convex nondifferentiable functions. They are based on the construction of a polyhedral model approximating from below the objective function. This approximation becomes more and more accurate as the algorithm proceeds. In particular, suppose that we want to solve the following unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{4}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex, not necessarily differentiable, function and assume that, at any point $x$, the function value $f(x)$ and a subgradient $g \in \partial f(x)$ are computable. We recall that a necessary and sufficient condition for a point $x^*$ to be a minimum of $f$ is that $0 \in \partial f(x^*)$.

In bundle methods, at a generic iteration we maintain a bundle $B$ of the type:

$$B \overset{\triangle}{=} \{(x_i, f(x_i), g_i) \mid i \in I\},$$

i.e. a set of triplets indexed by the index set $I$, such that $g_i \in \partial f(x_i)$, for all $i \in I$. In correspondence to $B$, it is possible to construct the following polyhedral approximation of $f$:

$$\hat{f}(x) \overset{\triangle}{=} \max_{i \in I} \{f(x_i) + g_i^T(x - x_i)\}, \tag{5}$$

called the "cutting plane function" and obtained as the pointwise maximum of $|I|$ affine functions, the linearizations of $f$ rooted at the points $x_i$, for $i \in I$.

The new iterate $x^+$ is obtained by pursuing two conflictive objectives [1]: on one hand we want to minimize the polyhedral function $\hat{f}$; on the other hand we want to minimize the distance between $x^+$ and a particular point, say $y$, named the "stability center" and generally coinciding with the best point in terms of function value among the $x_i$'s, for $i \in I$. In other words, the new point $x^+$ is computed as the solution of the following optimization problem:

$$\min_{x \in \mathbb{R}^n} \gamma \hat{f}(x) + \frac{1}{2}\|x - y\|^2, \tag{6}$$

where the parameter $\gamma > 0$ is aimed at tuning the trade off between the two objectives.

Problem (6) can be easily rewritten as a quadratic program of the type:

$$\begin{cases} \min\limits_{v,d} \quad \gamma v + \dfrac{1}{2}\|d\|^2 \\ \\ \qquad\quad v \geq g_i^T d - \alpha_i \quad i \in I, \end{cases} \tag{7}$$

where $d \overset{\triangle}{=} x - y$ is the "displacement" from $y$, and

$$\alpha_i \overset{\triangle}{=} f(y) - [f(x_i) + g_i^T(y - x_i)] \tag{8}$$

is the $i$-th "linearization error" at $y$, which is nonnegative because $f$ is convex.

Program (7) is the subproblem that characterizes each iteration of a bundle method. More specifically, indicating by $(v^+, d^+)$ its optimal solution, the new iterate $x^+$ is computed as $y + d^+$; moreover, the stability center $y$ (which plays the role of the current iterate) is updated to $x^+$ whenever an appropriate decrease condition on $f$ is satisfied at $x^+$.

It is easy to show that $v^+ \leq 0$ and that, whenever $v^+ = 0$, the point $y$ is a minimum of $f$.

Now we shortly focus on the case where $f$ is nonsmooth and nonconvex. First of all, we observe that, with respect the convex case, the classical concept of the subdifferential of $f$ at a point $x$ is replaced by the Goldstein $\epsilon$-subdifferential (see [19]), denoted by $\partial_\epsilon^G f(x)$: it is the convex hull of the Clarke subdifferentials [8] of $f$ computed at the points within a sphere centered at $x$ and of radius $\epsilon > 0$. As a consequence, a necessary condition for a point $x^*$ to be a minimizer of $f$ is

$$0 \in \partial_\epsilon^G f(x^*), \tag{9}$$

for $\epsilon > 0$.

In literature, differently from the convex case, there are few algorithms tackling problem (4), when $f$ is nonconvex. Some references are [10, 17, 19, 24]. In particular we focus on the recent approach reported in [11], which is an extension to nonconvex functions of a classical bundle method. In the following, we just remark the main differences with respect to the convex case.

Let $y$ the current stability center at a certain iteration. When the objective function is not convex, the nonnegativity of the linearization error $\alpha_i$, for $i \in I$, is no longer guaranteed and the polyhedral function (5) is not necessarily a lower approximation of $f$. To overcome these difficulties, the initial idea proposed in [11] is to partition the index set $I$ of the bundle in two sets $I_+$ and $I_-$ defined as follows:

$$I_+ \stackrel{\triangle}{=} \{i \mid \alpha_i \geq 0\} \quad \text{and} \quad I_- \stackrel{\triangle}{=} \{i \mid \alpha_i < 0\}\ . \tag{10}$$

Then, in correspondence to $I_+$ and $I_-$, two different polyhedral functions are constructed:

$$\hat{f}_+(x) \stackrel{\triangle}{=} \max_{i \in I_+} \{f(x_i) + g_i^T(x - x_i)\} \tag{11}$$

and

$$\hat{f}_-(x) \stackrel{\triangle}{=} \min_{i \in I_-} \{f(x_i) + g_i^T(x - x_i)\}, \tag{12}$$

which are convex and concave, respectively. These functions can be easily expressed in terms of the linearization errors and of the displacement $d$ from $y$, as follows:

$$\Delta_+(d) \overset{\triangle}{=} \max_{i \in I_+} \{g_i^T d - \alpha_i\} + f(y) \tag{13}$$

and

$$\Delta_-(d) \overset{\triangle}{=} \min_{i \in I_-} \{g_i^T d - \alpha_i\} + f(y). \tag{14}$$

Observing that $\Delta_+(0) < \Delta_-(0)$, the new displacement $d^+$ is computed such that $\Delta_+(d^+) \leq \Delta_-(d^+)$. Looking at problem (7), this corresponds to solve the following quadratic program:

$$\begin{cases} \min\limits_{v,d} \quad \gamma v + \dfrac{1}{2}\|d\|^2 \\ \qquad v \geq g_i^T d - \alpha_i \quad i \in I_+ \\ \qquad v \leq g_i^T d - \alpha_i \quad i \in I_- \ . \end{cases} \tag{15}$$

As in the convex case, indicating by $(v^+, d^+)$ the optimal solution of program (15), the new iterate $x^+$ is computed as $y + d^+$; moreover the stability center $y$ is updated to $x^+$ whenever an appropriate decrease condition on $f$ is fulfilled at $x^+$.

Technical details of the method can be found in [11], where the convergence of the algorithm to a stationary point satisfying condition (9) is discussed, under the hypotheses that $f$ is weakly semismooth [20] and the set

$$\mathcal{F}_0 \overset{\triangle}{=} \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$$

is compact.

## 4 Numerical experiments

We have adapted the bundle method described above for solving the TSVM problem (3). The implementation has been written in double precision Fortran 77, under a Windows XP system.

We have run our algorithm, called TSVM$^{Bundle}$, for several values of the parameters $C_1$ and $C_2$. We have always set $C_2 \leq C_1$, as it appears reasonable to charge a greater weight to the labelled points than to the unlabelled ones.

We have chosen some test problems drawn from the binary classification literature, described in table 1. The first six datasets are taken from the UCI Machine Learning Repository [21], Galaxy is the dataset used in galaxy discrimination with neural networks [22], while the last two test problems are described in [7].

We first focus on the test problems 1–7, for which we have performed a ten–fold cross–validation. In table 2 we compare our results with those reported in [4, 5, 12]. In particular, the classification error is averaged over the results of the ten–fold cross–validation. The reported results of our algorithm are the best ones for the different tested values of the weighting parameters $C_1$ and $C_2$. The algorithms taken into account are the following:

- $\text{SVM}^{RLP}$: the inductive linear 1–norm SVM [3, 6];
- $\text{S}^3\text{VM}$: the transductive version [4, 5] of the linear 1–norm SVM;
- $\text{SVM}^{Light}$ [14];
- $\text{TSVM}^{Light}$: the transductive version [15, 16] of $\text{SVM}^{Light}$.

| # | Dataset | Dimension | Points |
|---|---|---|---|
| 1 | Cancer | 9 | 699 |
| 2 | Diagnostic | 30 | 569 |
| 3 | Heart | 13 | 297 |
| 4 | Pima | 9 | 769 |
| 5 | Ionosphere | 34 | 351 |
| 6 | Sonar | 60 | 208 |
| 7 | Galaxy | 14 | 4192 |
| 8 | g50c | 50 | 550 |
| 9 | g10n | 10 | 550 |

Table 1: Datasets

For Pima and Galaxy datasets, since no data on ten–fold cross–validation are available in [4, 5], we report (see the starred test problems in table 2) also the numerical results obtained by randomly selecting, for each trial, a testing set constituted by 50 unlabelled points. This restriction is imposed in [4, 5], due to the failure of the used integer programming solver in tackling classification problems characterized by large testing sets. We adopt the same restriction only for comparison purposes, as our algorithm is capable to deal with larger sets.

| Dataset | SVM$^{RLP}$ | S$^3$VM | SVM$^{Light}$ | TSVM$^{Light}$ | SVM$^{Bundle}$ | TSVM$^{Bundle}$ |
|---|---|---|---|---|---|---|
| Cancer | 3.60 | 3.40 | - | - | 2.57 | 2.29 |
| Diagnostic | 3.50 | 3.30 | - | - | 1.93 | 1.93 |
| Heart | 17.30 | 16.00 | 13.50 | 16.30 | 13.67 | 12.00 |
| Pima | - | - | 23.60 | - | 22.60 | 22.47 |
| Pima$^*$ | 22.00 | 22.20 | - | - | 20.80 | 20.80 |
| Ionosphere | 10.90 | 10.60 | 12.00 | 15.72 | 11.71 | 11.43 |
| Sonar | 28.10 | 21.90 | - | 25.24 | 21.43 | 20.48 |
| Galaxy | - | - | - | 5.90 | 4.44 | 4.44 |
| Galaxy$^*$ | 6.40 | 5.40 | - | - | 5.40 | 5.40 |

Table 2: Testing error average %

For evaluating the performance of the semi–supervised approach versus the supervised case, we report in the SVM$^{Bundle}$ column the results of the inductive version of our method, obtained for $C_2$ equal to zero. We note that the transductive case never performs worse than the inductive one.

In table 3 we compare our results, relative to the g50c and g10n datasets, with those reported in [7] and obtained by running the TSVM$^{Light}$ and $\nabla$TSVM algorithms; the latter is a standard gradient descent method [7] on a smooth approximation of the objective function $h$ of problem (3). As for the cross–validation, we have used in this case the same ten different splits provided by the authors in [7].

| Dataset | TSVM$^{Light}$ | $\nabla$TSVM | TSVM$^{Bundle}$ |
|---|---|---|---|
| g50c | 6.87 | 5.80 | 4.02 |
| g10n | 14.36 | 9.82 | 8.60 |

Table 3: Testing error average %

In table 2 and 3, for each dataset, the best result has been underlined.

## 5 Conclusions

In this paper we have applied a bundle type method for semi–supervised classification problems.

Our algorithm has performed best on all datasets except for the Ionosphere test problem, where however the performance appears comparable.

In our opinion the main advantage of the proposed method is the possibility to train the classifier on the basis of a large number of labelled and unlabelled points. This could be useful especially in many real–world classification problems, where labelling is expensive, whereas unlabelled data are abundant being easier to obtain.

The results of our experimentation suggest that sophisticated nonsmooth optimization techniques can be fruitfully adopted for classification problems.

## References

[1] A. Astorino, A. Fuduli, and M. Gaudioso, *Analysis of regularization techniques in convex nondifferentiable optimization*, in Operations Research Proceedings 1996, Berlin, Heidelberg, 1997, pp. 20–25.

[2] M. Belkin and P. Niyogi, *Semi–supervised learning on Riemannian manifolds*, Machine Learning, 56 (2004), pp. 209–239.

[3] K. P. Bennett and E. Bredensteiner, *Geometry in learning*, in Geometry at Work, C. Gorini, ed., Mathematical Association of America, Washington, D.C., 2000, pp. 132–145.

[4] K. P. Bennett and A. Demiriz, *Semi–supervised support vector machines*, in Advances in Neural Information Processing Systems, M. S. Kearns, S. A. Solla, and D. A. Cohn, eds., vol. 12, MIT Press, Cambridge, MA, 1998, pp. 368–374.

[5] ——, *Optimization approaches to semi–supervised learning*, in Complementarity: Applications, Algorithms and Extensions, M. Ferris, O. Mangasarian, and J. S. Pang, eds., Kluwer Academic Publishers, Boston, 2001, pp. 121–141.

[6] K. P. Bennett and O. L. Mangasarian, *Robust linear programming discrimination of two linearly inseparable sets*, Optimization Methods and Software, 1 (1992), pp. 23–34.

[7] O. Chapelle and A. Zien, *Semi–supervised classification by low density separation*, in Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005, pp. 57–64.

[8] F. H. Clarke, *Optimization and nonsmooth analysis*, John Wiley and Sons, 1983.

[9] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[10] A. Fuduli, M. Gaudioso, and G. Giallombardo, *A DC piecewise affine model and bundling technique in nonconvex nonsmooth minimization*, Optimization Methods and Software, 18 (2004), pp. 89–102.

[11] ——, *Minimizing nonconvex nonsmooth functions via cutting planes and proximity control*, SIAM Journal on Optimization, 14 (2004), pp. 743–756.

[12] G. Fung and O. L. Mangasarian, *Proximal support vector machine classifiers*, in Proceedings KDD-2001, San Francisco, August 26-29 2001, pp. 77–86.

[13] ——, *Semi-supervised support vector machines for unlabeled data classification*, Optimization Methods and Software, 15 (2001), pp. 29–44.

[14] T. Joachims, *Making large-scale SVM learning practical*, in Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, 1999.

[15] ——, *Transductive inference for text classification using support vector machines*, in International Conference on Machine Learning, 1999, pp. 200–209.

[16] ——, *Transductive learning via spectral graph partitioning*, in Proceedings of the International Conference on Machine Learning, 2003, pp. 290–297.

[17] K. C. Kiwiel, *Methods of descent for nondifferentiable optimization*, vol. 1133 of Lecture notes in mathematics, Springer-Verlag, 1985.

[18] C. Lemaréchal, *An algorithm for minimizing convex functions*, in Proceedings IFIP '74 Congress, J. Rosenfeld, ed., North-Holland, Amsterdam.

[19] M. Mäkelä and P. Neittaanmäki, *Nonsmooth optimization*, World Scientific, 1992.

[20] R. Mifflin, *An algorithm for constrained optimization with semismooth functions*, Mathematics of Operations Research, 2 (1977), pp. 191–207.

[21] P. M. Murphy and D. W. Aha, *UCI repository of machine learning databases*, in www.ics.uci.edu/∼mlearn/MLRepository.html, 1992.

[22] S. Odewahn, E. Stockwell, R. Pennington, R. Humphreys, and W. Zumach, *Automated star/galaxy discrimination with neural networks*, Astronomical Journal, 103 (1992), pp. 318–331.

[23] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in kernel methods. Support vector learning*, MIT Press, Cambridge, MA, 1999.

[24] H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis*, SIAM Journal on Optimization, 1 (1992), pp. 121–152.

[25] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, 2004.

[26] V. Vapnik, *The nature of the statistical learning theory*, Springer Verlag, New York, 1995.

[27] P. Wolfe, *A method of conjugate subgradients for minimizing nondifferentiable functions*, in Nondifferentiable optimization, M. Balinski and P. Wolfe, eds., vol. 3 of Mathematical Programming Study, North-Holland, Amsterdam, 1975, pp. 145–173.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, *Learning with local and global consistency*, in Advances in Neural Information Processing System, MIT Press, Cambridge, MA, 2003.

[29] D. Zhou and B. Schölkopf, *Learning from labeled and unlabeled data using random walks*, in Proceedings of the 26th DAGM Symposium, C. E. Rasmussen, H. H. Büthoff, M. A. Giese, and B. Schölkopf, eds., Springer, Berlin, Germany, 2004, pp. 237–244.