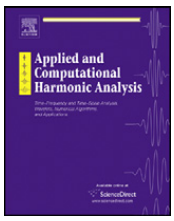
Appl. Comput. Harmon. Anal. 27 (2009) 247–254

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha



Letter to the Editor

Accelerating gradient projection methods for ℓ_1-constrained signal recovery by steplength selection rules

I. Loris [a,*], M. Bertero [b], C. De Mol [c], R. Zanella [d], L. Zanni [d]

[a] Department of Mathematics, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium
[b] Department of Computer and Information Sciences, University of Genova, Via Dodecaneso 35, I-16146 Genova, Italy
[c] Department of Mathematics and ECARES, Université Libre de Bruxelles, Campus Plaine CPI 217, Bd du Triomphe, B-1050 Brussels, Belgium
[d] Department of Pure and Applied Mathematics, University of Modena and Reggio Emilia, Via Campi 213/b, I-41100 Modena, Italy

ARTICLE INFO

Article history:
Available online 27 February 2009
Communicated by Ingrid Daubechies
on 23 February 2009

Keywords:
Sparsity
ℓ_1-penalty
Gradient projection method

ABSTRACT

We propose a new gradient projection algorithm that compares favorably with the fastest algorithms available to date for ℓ_1-constrained sparse recovery from noisy data, both in the compressed sensing and inverse problem frameworks. The method exploits a line-search along the feasible direction and an adaptive steplength selection based on recent strategies for the alternation of the well-known Barzilai–Borwein rules. The convergence of the proposed approach is discussed and a computational study on both well conditioned and ill-conditioned problems is carried out for performance evaluations in comparison with five other algorithms proposed in the literature.

1. Introduction

There has been a vast amount of recent literature dedicated to algorithms for sparse recovery, both in the context of inverse imaging problems and of *compressed sensing*. As an alternative to the usual quadratic penalties used in regularization theory for ill-posed or ill-conditioned inverse problems, the use of ℓ_1-type penalties has been advocated in order to recover regularized solutions having sparse expansions on a given basis or frame, such as e.g. a wavelet system [14]. Denoting by $\boldsymbol{x} \in \mathbb{R}^p$ the vector of coefficients describing the unknown object, by $\boldsymbol{y} \in \mathbb{R}^n$ the vector of (noisy) data and by $\boldsymbol{K}$ the linear operator ($n \times p$ matrix) modeling the link between the two, the inverse problem amounts to finding a regularized solution of the equation $\boldsymbol{K}\boldsymbol{x} = \boldsymbol{y}$. When it is known a priori that $\boldsymbol{x}$ is a sparse vector, one can resort to the following penalized least-squares strategy [9], also referred to as the *lasso* after Tibshirani [29]:

$$\bar{\boldsymbol{x}}(\lambda) = \arg\min_{\boldsymbol{x}} \|\boldsymbol{K}\boldsymbol{x} - \boldsymbol{y}\|^2 + 2\lambda\|\boldsymbol{x}\|_1 \tag{1}$$

where λ is a positive regularization parameter regulating the balance between the penalty and the data misfit terms. The norm $\|\cdot\|$ denotes the usual ℓ_2 norm whereas $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{p} |x_i|$ is the ℓ_1 norm of the vector $\boldsymbol{x}$.

In compressed sensing (also called *compressive sampling*), the aim is to reconstruct a *sparse* signal or object from a small number of linear measurements [6,7,16]. The recovery of such an object can then be achieved by searching for the sparsest solution to the linear system $\boldsymbol{K}\boldsymbol{x} = \boldsymbol{y}$ representing the measurement process, or equivalently by looking for a solution with minimum "ℓ_0-norm." To avoid the combinatorial complexity of the latter problem, one can use as a proxy a convex ℓ_1-norm

* Corresponding author.
E-mail addresses: igloris@vub.ac.be (I. Loris), bertero@disi.unige.it (M. Bertero), demol@ulb.ac.be (C. De Mol), riccardo.zanella@unimore.it (R. Zanella), luca.zanni@unimore.it (L. Zanni).

1063-5203/$ – see front matter
doi:10.1016/j.acha.2009.02.003

minimization strategy. When the data $\boldsymbol{y}$ are affected by measurement errors, the problem is reformulated as a penalized least-squares optimization analogous to (1).

Let us observe that problem (1) is equivalent to the constrained minimization problem:

$$\tilde{\boldsymbol{x}}(\rho) = \arg \min_{\|x\|_1 \leqslant \rho} \|\boldsymbol{K}\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{2}$$

for a certain ρ. One can show that $\bar{\boldsymbol{x}}(\lambda)$ and $\tilde{\boldsymbol{x}}(\rho)$ are piecewise linear functions of λ and ρ. One always has that $\bar{\boldsymbol{x}}(\lambda) = \boldsymbol{0}$ for $\lambda \geqslant \lambda_{\max} \equiv \max_i |(\boldsymbol{K}^T \boldsymbol{y})_i|$. The relationship between λ and ρ is given by $\lambda = \max_i |(\boldsymbol{K}^T(\boldsymbol{y} - \boldsymbol{K}\tilde{\boldsymbol{x}}(\rho)))_i|$ and $\rho = \|\bar{\boldsymbol{x}}(\lambda)\|_1$ [15].

2. Iterative minimization algorithms

Several iterative methods for solving the minimization problems (1) or (2) have been proposed in the literature. For the purpose of comparison with our new acceleration scheme, we will focus on the following algorithms:

1. The Iterative Soft-Thresholding Algorithm ("ISTA") proposed in [8,14,19] goes as follows: $\boldsymbol{x}^{(k+1)} = S_\lambda[\boldsymbol{x}^{(k)} + \boldsymbol{r}^{(k)}]$ where $\boldsymbol{r}^{(k)} = \boldsymbol{K}^T(\boldsymbol{y} - \boldsymbol{K}\boldsymbol{x}^{(k)})$ is the residual in step k and the (nonlinear) soft-thresholding operator acts componentwise as $(S_\lambda[\boldsymbol{x}])_i = x_i - \lambda \operatorname{sgn}(x_i)$ if $|x_i| > \lambda$ and zero otherwise. For any initial vector $\boldsymbol{x}^{(0)}$ and under the condition $\|\boldsymbol{K}\| < 1$, this scheme has been shown to converge to the minimizer $\bar{\boldsymbol{x}}(\lambda)$ defined by (1) [14]. When reinterpreted as a forward–backward proximal scheme, convergence can be seen to hold also for $\|\boldsymbol{K}\| < \sqrt{2}$ [10].
2. The Fast Iterative Soft-Thresholding Algorithm ("FISTA"), proposed in [2], is a variation of ISTA. Defining the operator T by $T(\boldsymbol{x}) = S_\lambda[\boldsymbol{x} + \boldsymbol{K}^T(\boldsymbol{y} - \boldsymbol{K}\boldsymbol{x})]$, the FISTA algorithm is:

$$\boldsymbol{x}^{(k+1)} = T\left(\boldsymbol{x}^{(k)} + \frac{t^{(k)} - 1}{t^{(k+1)}}\left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right)\right), \tag{3}$$

   where $\boldsymbol{x}^{(0)} = \boldsymbol{0}$, $t^{(k+1)} = \frac{1+\sqrt{1+4(t^{(k)})^2}}{2}$ and $t^{(0)} = 1$. It has virtually the same complexity as the ISTA algorithm, but can be shown to have better convergence properties.
3. The GPSR algorithm proposed in [20].
4. The SPARSA algorithm proposed in [30].
5. The Projected Steepest Descent ("PSD") method proposed in [15]: $\boldsymbol{x}^{(k+1)} = P_\Omega[\boldsymbol{x}^{(k)} + \beta^{(k)}\boldsymbol{r}^{(k)}]$, with $\beta^{(k)} = \|\boldsymbol{r}^{(k)}\|^2 / \|\boldsymbol{K}\boldsymbol{r}^{(k)}\|^2$. P_Ω denotes the projection onto the ℓ_1-ball Ω of radius ρ.

Figs. 1–4 provide a visual way to compute the performance of these algorithms in two problem examples. Note that these are the same as in [25], where the reader can find comparisons to yet other methods, including e.g. the ℓ_1-ls method, an interior point algorithm proposed in [24].

3. Gradient projection with adaptive steplength selection

In this section we describe the acceleration scheme we propose for solving the optimization problem (2). This problem is a particular case of the general problem of minimizing a convex and continuously differentiable function $f(\boldsymbol{x})$ over a closed convex set $\Omega \subset \mathbb{R}^p$. Here $\Omega = \{\boldsymbol{x} \in \mathbb{R}^p,\ \|\boldsymbol{x}\|_1 \leqslant \rho\}$. A gradient projection method for solving this problem can be stated as in Algorithm GP.

Some comments about the main steps of Algorithm GP are in order.

First of all, it is worth to stress that any choice of the steplength α_k in a closed interval is permitted. This is very important from a practical point of view since it allows to make the updating rule of α_k problem-related and oriented at optimizing the performance.

If the projection performed in step 2 returns a vector $\boldsymbol{h}^{(k)}$ equal to $\boldsymbol{x}^{(k)}$, then $\boldsymbol{x}^{(k)}$ is a stationary point and the algorithm stops. When $\boldsymbol{h}^{(k)} \neq \boldsymbol{x}^{(k)}$, it is possible to prove that $\boldsymbol{d}^{(k)}$ is a descent direction for f in $\boldsymbol{x}^{(k)}$ and the backtracking loop in step 5 terminates with a finite number of runs; thus the algorithm is well defined [3–5].

The nonmonotone line-search strategy implemented in step 5 ensures that $f(\boldsymbol{x}^{(k+1)})$ is lower than the maximum of the objective function in the last M iterations [23]; of course, if $M = 1$ then the strategy reduces to the standard monotone Armijo rule [3].

Concerning the convergence properties of the algorithm, the following result can be derived from the analysis carried out in [4,5] for more general gradient projection schemes: if the level set $\Omega_0 = \{\boldsymbol{x} \in \Omega:\ f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}^{(0)})\}$ is bounded, then every accumulation point of the sequence $\{\boldsymbol{x}^{(k)}\}$ generated by Algorithm GP is a stationary point of $f(\boldsymbol{x})$ in Ω. We observe that the assumption is trivially satisfied for problem (2) since in this case the feasible region Ω is bounded.

Now, we may discuss the choice of the steplengths $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$. Steplength selection rules in gradient methods have received an increasing interest in the last years from both the theoretical and the practical point of view. On the one hand, following the original ideas of Barzilai and Borwein (BB) [1], several steplength updating strategies have been devised to accelerate the slow convergence exhibited in most cases by standard gradient methods, and a lot of effort has been put

Algorithm GP *(Gradient projection method).*

Choose the starting point $\boldsymbol{x}^{(0)} \in \Omega$, set the parameters $\beta, \theta \in (0,1)$, $0 < \alpha_{\min} < \alpha_{\max}$ and fix a positive integer M.
FOR $k = 0, 1, 2, \ldots$ DO THE FOLLOWING STEPS:

Step 1. Choose the parameter $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$;
Step 2. Projection: $\boldsymbol{h}^{(k)} = P_\Omega(\boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)}))$;
If $\boldsymbol{h}^{(k)} = \boldsymbol{x}^{(k)}$ then stop, declaring that $\boldsymbol{x}^{(k)}$ is a stationary point;
Step 3. Descent direction: $\boldsymbol{d}^{(k)} = \boldsymbol{h}^{(k)} - \boldsymbol{x}^{(k)}$;
Step 4. Set $\lambda_k = 1$ and $f_{\max} = \max_{0 \leqslant j \leqslant \min(k, M-1)} f(\boldsymbol{x}^{(k-j)})$;
Step 5. Backtracking loop:
IF $f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}) \leqslant f_{\max} + \beta \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}$ THEN
go to Step 6;
ELSE
set $\lambda_k = \theta \lambda_k$ and go to Step 5;
ENDIF
Step 6 Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}$.

END

Algorithm SS *(Steplength selection for GP).*

IF $k = 0$ THEN
set $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$, $\tau_1 \in (0,1)$ and a nonnegative integer M_α;
ELSE
IF $\boldsymbol{s}^{(k-1)^T} \boldsymbol{z}^{(k-1)} \leqslant 0$ THEN
$\alpha_k = \alpha_{\max}$;
ELSE
$\alpha_k^{(1)} = \max\{\alpha_{\min}, \min\{\frac{\boldsymbol{s}^{(k-1)^T}\boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}, \alpha_{\max}\}\}$;
$\alpha_k^{(2)} = \max\{\alpha_{\min}, \min\{\frac{\boldsymbol{s}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}, \alpha_{\max}\}\}$;
IF $\alpha_k^{(2)}/\alpha_k^{(1)} \leqslant \tau_k$ THEN
$\alpha_k = \min\{\alpha_j^{(2)},\ j = \max\{1, k - M_\alpha\}, \ldots, k\}$;
$\tau_{k+1} = \tau_k * 0.9$;
ELSE
$\alpha_k = \alpha_k^{(1)}$;
$\tau_{k+1} = \tau_k * 1.1$;
ENDIF
ENDIF
ENDIF

into explaining the effects of these strategies [11–13,18,21,22,32]. On the other hand, numerical experiments on randomly generated, library and real-life test problems have confirmed the remarkable convergence rate improvements involved by some BB-like steplength selections [12,13,20,21,28,31,32]. Thus, it seems natural to equip a gradient projection method with a steplength selection that takes into account the recent advances on the BB-like updating rules.

First of all we must recall the two BB rules usually exploited by the main steplength updating strategies. To this end, by denoting with I the $p \times p$ identity matrix, we can regard the matrix $B(\alpha_k) = (\alpha_k I)^{-1}$ as an approximation of the Hessian $\nabla^2 f(\boldsymbol{x}^{(k)})$ and derive two updating rules for α_k by forcing quasi-Newton properties on $B(\alpha_k)$:

$$\alpha_k^{\mathrm{BB1}} = \arg\min_{\alpha_k \in \mathbb{R}} \left\| B(\alpha_k)\boldsymbol{s}^{(k-1)} - \boldsymbol{z}^{(k-1)} \right\| \quad \text{and} \quad \alpha_k^{\mathrm{BB2}} = \arg\min_{\alpha_k \in \mathbb{R}} \left\| \boldsymbol{s}^{(k-1)} - B(\alpha_k)^{-1}\boldsymbol{z}^{(k-1)} \right\|, \tag{4}$$

where $\boldsymbol{s}^{(k-1)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$ and $\boldsymbol{z}^{(k-1)} = \nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^{(k-1)})$. In this way, the steplengths

$$\alpha_k^{\mathrm{BB1}} = \frac{\boldsymbol{s}^{(k-1)^T}\boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}, \qquad \alpha_k^{\mathrm{BB2}} = \frac{\boldsymbol{s}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T}\boldsymbol{z}^{(k-1)}} \tag{5}$$

are obtained.

At this point, inspired by the steplength alternations successfully implemented in recent gradient methods [21,32], we propose a steplength updating rule for GP which adaptively alternates the values provided by (5). The details of the GP steplength selection are given in Algorithm SS. This rule decides the alternation between two different selection strategies by means of the variable threshold τ_k instead of a constant parameter as done in [21] and [32]. This trick makes the choice of τ_0 less important for the GP performance and, in our experience, seems able to avoid the drawbacks due to the use of

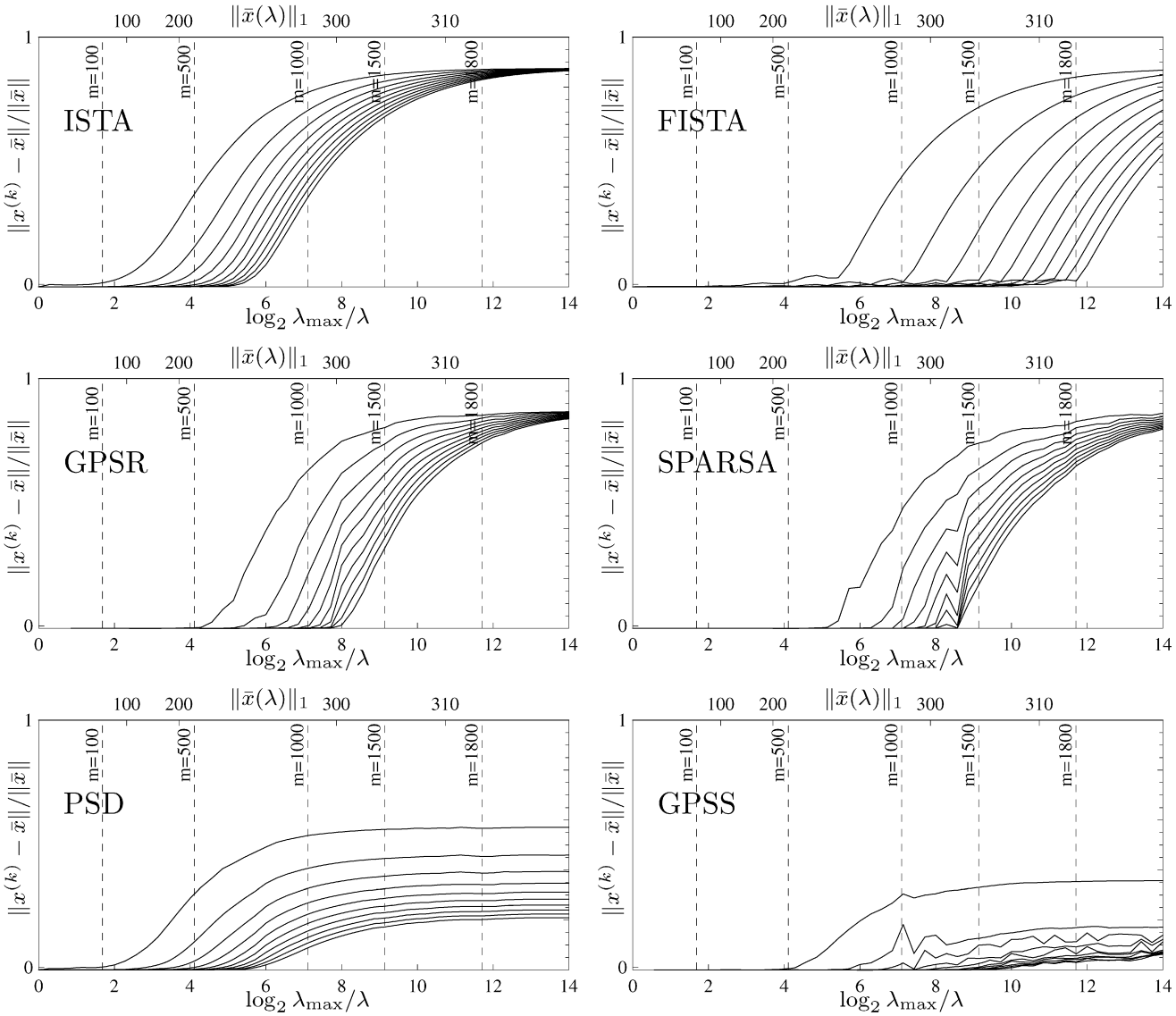


Fig. 1. Approximation isochrones in the case of the Gaussian random matrix for $t = 6, 12, \ldots, 60$ seconds.

the same steplength rule in too many consecutive iterations. In the following we denote by GPSS Algorithm GP equipped with the steplength selection SS.

We end this section by describing the setting for the GPSS parameters used in the computational study of this work:

- *line-search parameters*: $M = 1$ (monotone line-search), $\theta = 0.5$, $\beta = 10^{-4}$;
- *steplength parameters*: $\alpha_{\min} = 10^{-10}$, $\alpha_{\max} = 10^{10}$, $\alpha_0 = \max\{\alpha_{\min}, \min\{\|P_\Omega(\boldsymbol{x}^{(0)} - \nabla f(\boldsymbol{x}^{(0)}))\|_\infty^{-1}, \alpha_{\max}\}\}$, $\tau_1 = 0.5$, $M_\alpha = 2$.

In our experience the above setting often provides satisfactory performance; however, it cannot be considered optimal for every application and a careful parameter tuning is always advisable.

4. Numerical experiments

To assess the performances of our GPSS algorithm and estimate the gain in speed it can provide with respect to Algorithms 1 to 5, we perform some numerical tests. To this purpose we adopt the methodology proposed in [25] and based on the notion of *approximation isochrones*. It improves on the comparisons made for a single value of λ or ρ, i.e. for a single level of sparsity of the recovered object.

For values of λ in a given interval $\lambda_{\min} \leqslant \lambda \leqslant \lambda_{\max}$, one computes the minimizer $\bar{\boldsymbol{x}}(\lambda)$ of (1). When the number of nonzero components in $\bar{\boldsymbol{x}}(\lambda)$ is not too large, this can be done by means of the direct (noniterative) *homotopy* method [27] or LARS algorithm [17]. Then, for a fixed and given computation time, one runs one of the algorithms for each value of λ (or ρ). The relative error $e^{(k)} = \|\boldsymbol{x}^{(k)}(\lambda) - \bar{\boldsymbol{x}}(\lambda)\|/\|\bar{\boldsymbol{x}}(\lambda)\|$ reached at the end of the computation is plotted as a function of

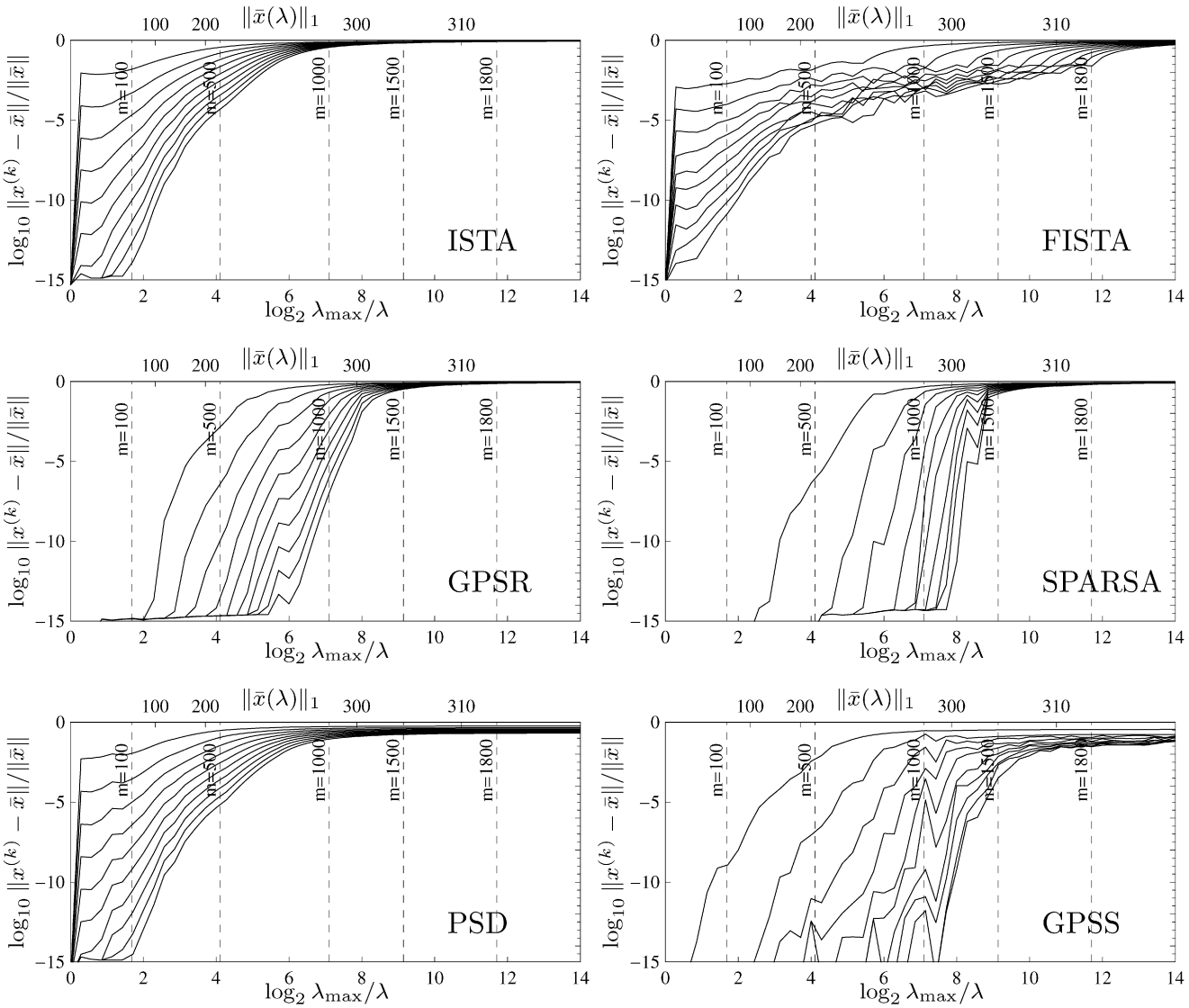


Fig. 2. The same as Fig. 1 but in a semi-log plot.

λ and hence this plot is just the approximation isochrone showing the degree of accuracy reached in the given amount of computing time for each value of λ. A set of such plots allow to quickly grasp the performances of a given algorithm in various parameter regimes and to easily compare it with other methods; it reveals in one glance under which circumstances the algorithms do well or fail. The paper [25] also demonstrates the fact that the relative performances of the algorithms may strongly depend on the specific application one considers, and in particular on the properties of the linear operator $\boldsymbol{K}$ modeling the problem.

We test the different algorithms on two different operators arising typically either from a compressed sensing or from an inverse problem. In both cases the matrix $\boldsymbol{K}$ is of size 1848×8192. In the first case, the elements of $\boldsymbol{K}$ are taken from a Gaussian distribution with zero mean and variance such that $\|\boldsymbol{K}\| = 1$. This matrix is rather well conditioned and can serve as a paradigm of compressed sensing applications. It is applied to a sparse vector and perturbed by additive Gaussian noise (about 2%) to yield the data $\boldsymbol{y}$. The second matrix models a severely ill-conditioned linear inverse problem that finds its origin in a problem of seismic tomography described in detail in [26].

For both operators, the minimizer $\bar{\boldsymbol{x}}(\lambda)$ is computed for 50 different values of λ (or equivalently, 50 different values of ρ). Then, for each iterative algorithm, we make plots having the relative error $e^{(k)}$ on the vertical axis and $\log_2 \lambda_{\max}/\lambda$ on the bottom horizontal axis (on the top horizontal axis the value of $\rho = \|\bar{\boldsymbol{x}}\|_1$ is also reported). The number of nonzero components m in $\bar{\boldsymbol{x}}(\lambda)$ is indicated by vertical dashed lines. In each plot we report the isochrone lines that correspond to a given amount of computer time. In this way one can see how close, for the different values of λ, the iterates approach the minimizer after a given time. Let us remark that although the reported computing times are of course specific to a given computer and implementation, the overall behavior of the isochrones should be fairly general. For example, the fact that they get very close to each other in some places can be interpreted as a bottleneck feature of the algorithm.

In Fig. 1, we report the results for the ISTA, FISTA, GPSR, SPARSA, PSD and our new algorithm GPSS for the case of the Gaussian random matrix. The proposed GPSS algorithm compares favorably with the other five, especially for small values

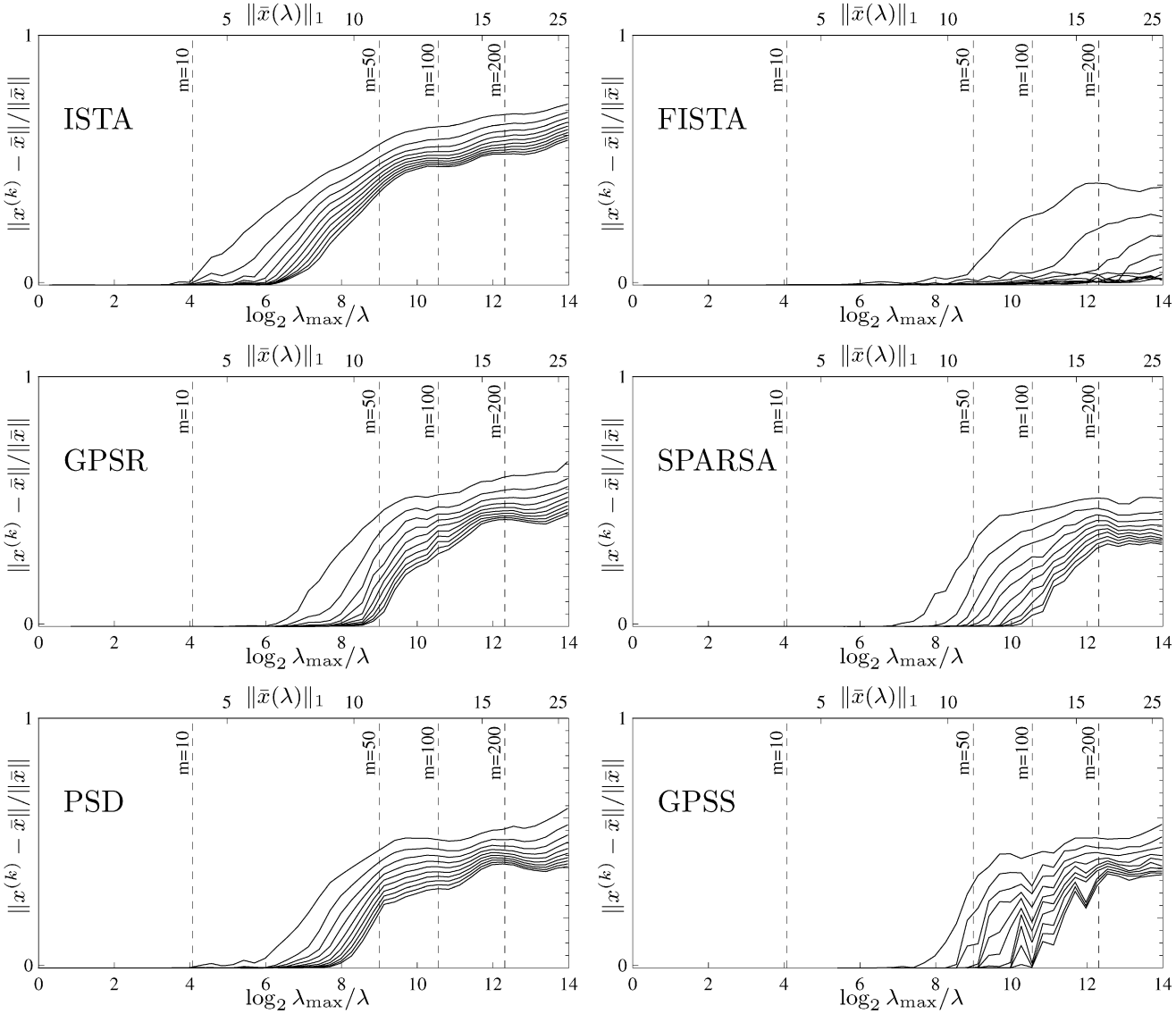


Fig. 3. Approximation isochrones for the seismic inverse problem for $t = 1, \ldots, 10$ minutes.

of λ. Experiments made by varying the parameter M showing no significant difference, we report here only the results obtained with $M = 1$ (monotonic line search). However, the behavior for large penalties is not clearly visible in Fig. 1. It is better demonstrated when using a logarithmic scale for the relative error on the vertical axes as reported in Fig. 2.

In Figs. 3 and 4, we report the results for the case of the ill-conditioned matrix arising from the seismic inverse problem. Clearly, for this operator, ISTA, GPSR and PSD have a lot of difficulty in approaching the minimizer for small values of λ (lines not approaching $e = 0$). The FISTA algorithm appears to work best for small penalty parameters whereas GPSS and SPARSA compete for the second place in such instance. From Fig. 4, we see that the GPSS and SPARSA algorithms are performing best for large values of λ.

The reported encouraging numerical results call of course for further experiments, but we believe that they are sufficiently representative to allow honest extrapolation to reliable conclusions holding more generally. As seen, the proposed GPSS algorithm performs well for the compressed sensing problem: for small values of λ, it clearly outperforms the other algorithms (see Fig. 1) whereas it is still competitive for larger values of λ. In the ill-conditioned inversion problem, GPSS an SPARSA appear to perform better than all other tested algorithms for large values of λ, whereas they are challenged by the FISTA method for smaller values.

Acknowledgments

I.L. and C.D.M. are supported by grant GOA-062 of the VUB. I.L. is supported by grant G.0564.09N of the FWO-Vlaanderen. M.B., R.Z. and L.Z. are partly supported by MUR grant 2006018748.

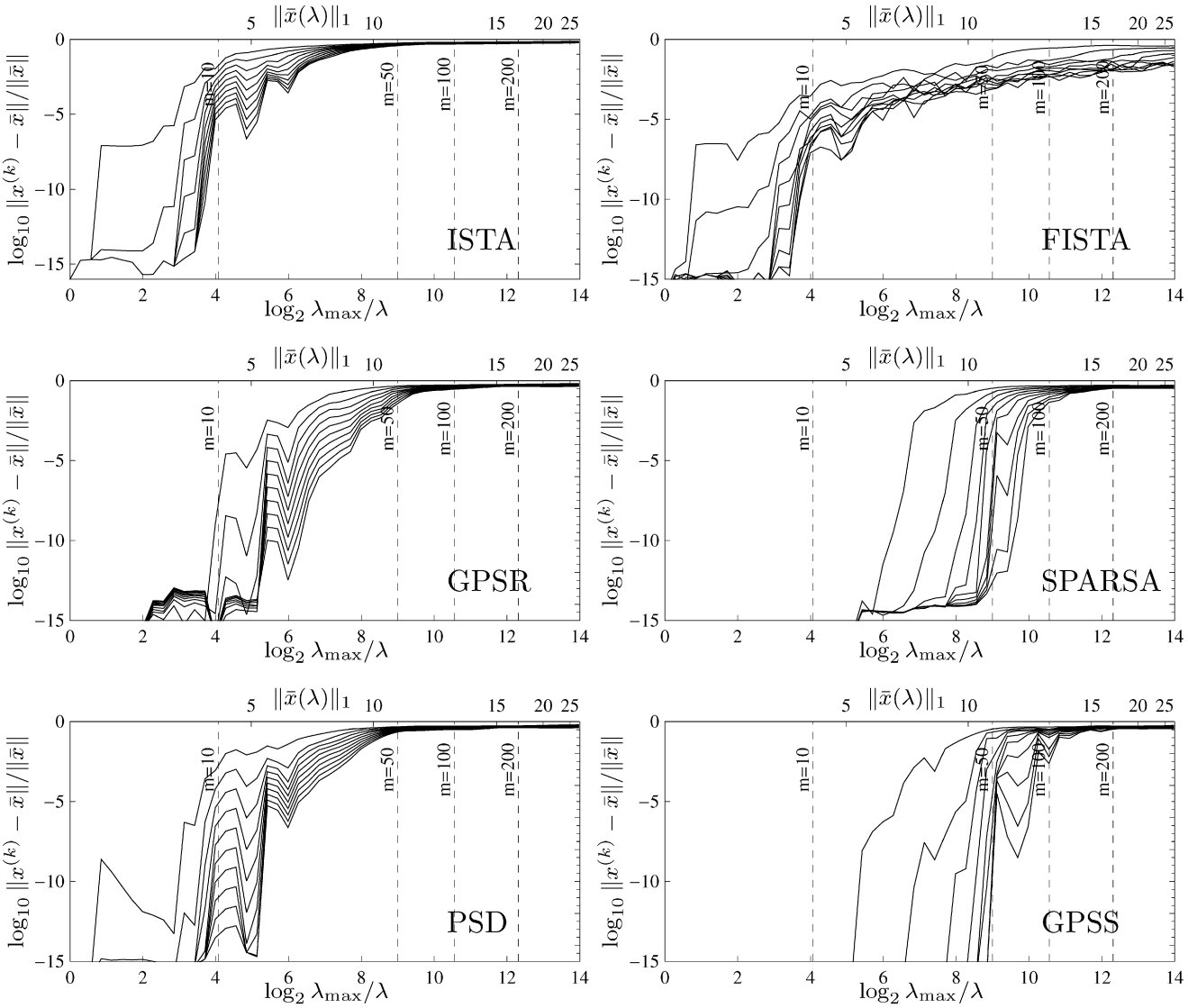


Fig. 4. The same as in Fig. 3 in a semi-log plot.

References

[1] J. Barzilai, J.M. Borwein, Two point step size gradient methods, IMA J. Numer. Anal. 8 (1988) 141–148.
[2] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (2009) 183–202.
[3] D.P. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, 1999.
[4] E.G. Birgin, J.M. Martínez, M. Raydan, Inexact spectral projected gradient methods on convex sets, IMA J. Numer. Anal. 23 (2003) 539–559.
[5] S. Bonettini, R. Zanella, L. Zanni, A scaled gradient projection method for constrained image deblurring, Inverse Problems 25 (2009) 015002.
[6] E. Candès, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inform. Theory 52 (2006) 489–509.
[7] E. Candès, T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, IEEE Trans. Inform. Theory 52 (2006) 5406–5425.
[8] A. Chambolle, An algorithm for total variation minimization and applications, J. Math. Imaging Vision 20 (2004) 89–97.
[9] S.S. Chen, D. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.
[10] P.L. Combettes, V.R. Wajs, Signal recovery by proximal forward–backward splitting, Multiscale Model. Simul. 4 (2005) 1168–1200.
[11] Y.H. Dai, R. Fletcher, On the asymptotic behaviour of some new gradient methods, Math. Program. 103 (2005) 541–559.
[12] Y.H. Dai, R. Fletcher, New algorithms for singly linearly constrained quadratic programming problems subject to lower and upper bounds, Math. Program. 106 (2006) 403–421.
[13] Y.H. Dai, W.W. Hager, K. Schittkowski, H. Zhang, The cyclic Barzilai–Borwein method for unconstrained optimization, IMA J. Numer. Anal. 26 (2006) 604–627.
[14] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math. 57 (2004) 1413–1457.
[15] I. Daubechies, M. Fornasier, I. Loris, Accelerated projected gradient method for linear inverse problems with sparsity constraints, J. Fourier Anal. Appl. 14 (2008) 764–792.
[16] D.L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (2006) 1289–1306.
[17] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2004) 407–499.
[18] R. Fletcher, On the Barzilai–Borwein method, Technical Report NA/207, Department of Mathematics, University of Dundee, Dundee, UK, 2001.

[19] M.A.T. Figueiredo, R.D. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Trans. Image Process. 12 (2003) 906–916.
[20] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, IEEE J. Sel. Topics Signal Process. 1 (2007) 586–597.
[21] G. Frassoldati, G. Zanghirati, L. Zanni, New adaptive stepsize selections in gradient methods, J. Ind. Manag. Optim. 4 (2008) 299–312.
[22] A. Friedlander, J.M. Martínez, B. Molina, M. Raydan, Gradient method with retards and generalizations, SIAM J. Numer. Anal. 36 (1999) 275–289.
[23] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line-search technique for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.
[24] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, A method for large-scale ℓ_1-regularized least squares, IEEE Trans. Sel. Topics Signal Process. 1 (2007) 606–617.
[25] I. Loris, On the performance of algorithms for the minimization of ℓ_1-penalized functionals, Inverse Problems 25 (2009) 035008.
[26] I. Loris, G. Nolet, I. Daubechies, F.A. Dahlen, Tomographic inversion using ℓ_1-norm regularization of wavelet coefficients, Geophys. J. Internat. 170 (2007) 359–370.
[27] M.R. Osborne, B. Presnell, B.A. Turlach, A new approach to variable selection in least squares problems, IMA J. Numer. Anal. 20 (2000) 389–403.
[28] T. Serafini, G. Zanghirati, L. Zanni, Gradient projection methods for quadratic programs and applications in training support vector machines, Optim. Methods Softw. 20 (2005) 343–378.
[29] R. Tibshirani, Regression selection and shrinkage via the lasso, J. R. Stat. Soc. Ser. B 58 (1996) 267–288.
[30] S. Wright, R. Nowak, M. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process. (2009), in press.
[31] L. Zanni, An improved gradient projection-based decomposition technique for support vector machines, Comput. Manag. Sci. 3 (2006) 131–145.
[32] B. Zhou, L. Gao, Y.H. Dai, Gradient methods with adaptive step-sizes, Comput. Optim. Appl. 35 (2006) 69–86.