Numerical experiments

11.21 *Log-Chebyshev approximation with bounds.* We consider an approximation problem: find $x \in \mathbf{R}^n$, that satisfies the variable bounds $l \preceq x \preceq u$, and yields $Ax \approx b$, where $b \in \mathbf{R}^m$. You can assume that $l \prec u$, and $b \succ 0$ (for reasons we explain below). We let a_i^T denote the ith row of the matrix A.

We judge the approximation $Ax \approx b$ by the *maximum fractional deviation*, which is

$$\max_{i=1,\ldots,n} \max\{(a_i^T x)/b_i, b_i/(a_i^T x)\} = \max_{i=1,\ldots,n} \frac{\max\{a_i^T x, b_i\}}{\min\{a_i^T x, b_i\}},$$

when $Ax \succ 0$; we define the maximum fractional deviation as ∞ if $Ax \not\succ 0$.

The problem of minimizing the maximum fractional deviation is called the *fractional Chebyshev approximation problem*, or the *logarithmic Chebyshev approximation problem*, since it is equivalent to minimizing the objective

$$\max_{i=1,\ldots,n} |\log a_i^T x - \log b_i|.$$

(See also exercise 6.3, part (c).)

(a) Formulate the fractional Chebyshev approximation problem (with variable bounds) as a convex optimization problem with twice differentiable objective and constraint functions.

(b) Implement a barrier method that solves the fractional Chebyshev approximation problem. You can assume an initial point $x^{(0)}$, satisfying $l \prec x^{(0)} \prec u$, $Ax^{(0)} \succ 0$, is known.

11.22 *Maximum volume rectangle inside a polyhedron.* Consider the problem described in exercise 8.16, *i.e.*, finding the maximum volume rectangle $\mathcal{R} = \{x \mid l \preceq x \preceq u\}$ that lies in a polyhedron described by a set of linear inequalities, $\mathcal{P} = \{x \mid Ax \preceq b\}$. Implement a barrier method for solving this problem. You can assume that $b \succ 0$, which means that for small $l \prec 0$ and $u \succ 0$, the rectangle $\mathcal{R}$ lies inside $\mathcal{P}$.

Test your implementation on several simple examples. Find the maximum volume rectangle that lies in the polyhedron defined by

$$A = \begin{bmatrix} 0 & -1 \\ 2 & -4 \\ 2 & 1 \\ -4 & 4 \\ -4 & 0 \end{bmatrix}, \qquad b = \mathbf{1}.$$

Plot this polyhedron, and the maximum volume rectangle that lies inside it.

11.23 *SDP bounds and heuristics for the two-way partitioning problem.* In this exercise we consider the two-way partitioning problem (5.7), described on page 219, and also in exercise 5.39:

$$\begin{array}{ll} \text{minimize} & x^T W x \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \ldots, n, \end{array} \tag{11.65}$$

with variable $x \in \mathbf{R}^n$. We assume, without loss of generality, that $W \in \mathbf{S}^n$ satisfies $W_{ii} = 0$. We denote the optimal value of the partitioning problem as $p^\star$, and $x^\star$ will denote an optimal partition. (Note that $-x^\star$ is also an optimal partition.)

The Lagrange dual of the two-way partitioning problem (11.65) is given by the SDP

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T \nu \\ \text{subject to} & W + \mathbf{diag}(\nu) \succeq 0, \end{array} \tag{11.66}$$

with variable $\nu \in \mathbf{R}^n$. The dual of this SDP is

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(WX) \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1, \quad i = 1, \ldots, n, \end{array} \tag{11.67}$$

with variable $X \in \mathbf{S}^n$. (This SDP can be interpreted as a relaxation of the two-way partitioning problem (11.65); see exercise 5.39.) The optimal values of these two SDPs are equal, and give a lower bound, which we denote $d^\star$, on the optimal value $p^\star$. Let $\nu^\star$ and $X^\star$ denote optimal points for the two SDPs.

(a) Implement a barrier method that solves the SDP (11.66) and its dual (11.67), given the weight matrix W. Explain how you obtain nearly optimal ν and X, give formulas for any Hessians and gradients that your method requires, and explain how you compute the Newton step. Test your implementation on some small problem instances, comparing the bound you find with the optimal value (which can be found by checking the objective value of all 2^n partitions). Try your implementation on a randomly chosen problem instance large enough that you cannot find the optimal partition by exhaustive search (*e.g.*, $n = 100$).

(b) *A heuristic for partitioning.* In exercise 5.39, you found that if $X^\star$ has rank one, then it must have the form $X^\star = x^\star (x^\star)^T$, where $x^\star$ is optimal for the two-way partitioning problem. This suggests the following simple heuristic for finding a good partition (if not the best): solve the SDPs above, to find $X^\star$ (and the bound $d^\star$). Let v denote an eigenvector of $X^\star$ associated with its largest eigenvalue, and let $\hat{x} = \mathbf{sign}(v)$. The vector $\hat{x}$ is our guess for a good partition.

Try this heuristic on some small problem instances, and the large problem instance you used in part (a). Compare the objective value of your heuristic partition, $\hat{x}^T W \hat{x}$, with the lower bound $d^\star$.

(c) *A randomized method.* Another heuristic technique for finding a good partition, given the solution $X^\star$ of the SDP (11.67), is based on *randomization*. The method is simple: we generate independent samples $x^{(1)}, \ldots, x^{(K)}$ from a normal distribution on $\mathbf{R}^n$, with zero mean and covariance $X^\star$. For each sample we consider the heuristic approximate solution $\hat{x}^{(k)} = \mathbf{sign}(x^{(k)})$. We then take the best among these, *i.e.*, the one with lowest cost. Try out this procedure on some small problem instances, and the large problem instance you considered in part (a).

(d) *A greedy heuristic refinement.* Suppose you are given a partition x, *i.e.*, $x_i \in \{-1, 1\}$, $i = 1, \ldots, n$. How does the objective value change if we move element i from one set to the other, *i.e.*, change x_i to $-x_i$? Now consider the following simple greedy algorithm: given a starting partition x, move the element that gives the largest reduction in the objective. Repeat this procedure until no reduction in objective can be obtained by moving an element from one set to the other.

Try this heuristic on some problem instances, including the large one, starting from various initial partitions, including $x = \mathbf{1}$, the heuristic approximate solution found in part (b), and the randomly generated approximate solutions found in part (c). How much does this greedy refinement improve your approximate solutions from parts (b) and (c)?

11.24 *Barrier and primal-dual interior-point methods for quadratic programming.* Implement a barrier method, and a primal-dual method, for solving the QP (without equality constraints, for simplicity)

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x \\ \text{subject to} & Ax \preceq b, \end{array}$$

with $A \in \mathbf{R}^{m \times n}$. You can assume a strictly feasible initial point is given. Test your codes on several examples. For the barrier method, plot the duality gap versus Newton steps. For the primal-dual interior-point method, plot the surrogate duality gap and the norm of the dual residual versus iteration number.

l1_ls: A Matlab Solver for Large-Scale ℓ_1-Regularized Least Squares Problems

Kwangmoo Koh deneb1@stanford.edu
Seungjean Kim sjkim@stanford.edu
Stephen Boyd boyd@stanford.edu

May 15, 2008

l1_ls solves ℓ_1-regularized least squares problems (LSPs) using the truncated Newton interior-point method described in [KKL+07].

1 The problems

l1_ls solves an optimization problem of the form

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \|x\|_1, \tag{1}$$

where the variable is $x \in \mathbf{R}^n$ and the problem data are $A \in \mathbf{R}^{m \times n}$ and $y \in \mathbf{R}^m$. Here, $\lambda \geq 0$ is the regularization parameter. We refer to the problem (1) as an *ℓ_1-regularized least squares problem*.

l1_ls can also solve ℓ_1-regularized LSPs with nonnegativity constraints:

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|^2 + \lambda \sum_{i=1}^n x_i \\ \text{subject to} & x_i \geq 0, \quad i = 1, \ldots, n. \end{array} \tag{2}$$

2 Calling sequences

The package l1_ls has two solver files, l1_ls.m and l1_ls_nonneg.m: l1_ls.m solves problem (1) and l1_ls_nonneg.m solves problem (2). Both solvers supports two calling styles. The simple calling style works when A is given as a matrix. The more complex calling style handles the case when A and its transpose are given as operators.

The simple calling sequence of l1_ls is

```
>> [x,status] = l1_ls(A,y,lambda,rel_tol,quiet);
```

The more complex calling sequence of l1_ls is

```
>> [x,status] = l1_ls(A,At,m,n,y,lambda,rel_tol,quiet);
```

Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems

Mário A. T. Figueiredo, Robert D. Nowak, Stephen J. Wright

***Abstract*—Many problems in signal processing and statistical inference involve finding sparse solutions to under-determined, or ill-conditioned, linear systems of equations. A standard approach consists in minimizing an objective function which includes a quadratic (squared ℓ_2) error term combined with a sparseness-inducing (ℓ_1) regularization term.*Basis pursuit*, the *least absolute shrinkage and selection operator* (LASSO), wavelet-based deconvolution, and *compressed sensing* are a few well-known examples of this approach. This paper proposes gradient projection (GP) algorithms for the bound-constrained quadratic programming (BCQP) formulation of these problems. We test variants of this approach that select the line search parameters in different ways, including techniques based on the Barzilai-Borwein method. Computational experiments show that these GP approaches perform well in a wide range of applications, often being significantly faster (in terms of computation time) than competing methods. Although the performance of GP methods tends to degrade as the regularization term is de-emphasized, we show how they can be embedded in a continuation scheme to recover their efficient practical performance.**

I. Introduction

A. Background

There has been considerable interest in solving the convex unconstrained optimization problem

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^k$, $\mathbf{A}$ is an $k \times n$ matrix, τ is a nonnegative parameter, $\|\mathbf{v}\|_2$ denotes the Euclidean norm of $\mathbf{v}$, and $\|\mathbf{v}\|_1 = \sum_i |v_i|$ is the ℓ_1 norm of $\mathbf{v}$. Problems of the form (1) have become familiar over the past three decades, particularly in statistical and signal processing contexts. From a Bayesian perspective, (1) can be seen as a maximum *a posteriori* criterion for estimating $\mathbf{x}$ from observations

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{2}$$

where $\mathbf{n}$ is white Gaussian noise of variance σ^2, and the prior on $\mathbf{x}$ is Laplacian (that is, $\log p(\mathbf{x}) = -\lambda\|\mathbf{x}\|_1 + \mathbf{K}$) [1], [25], [54]. Problem (1) can also be viewed as a regularization technique to overcome the ill-conditioned, or even singular, nature of matrix $\mathbf{A}$, when trying to infer $\mathbf{x}$ from noiseless observations $\mathbf{y} = \mathbf{A}\mathbf{x}$ or from noisy observations as in (2).

The presence of the ℓ_1 term encourages small components of $\mathbf{x}$ to become exactly zero, thus promoting sparse solutions [11], [54]. Because of this feature, (1) has been used for more than three decades in several signal processing problems where sparseness is sought; some early references are [12], [37], [50], [53]. In the 1990's, seminal work on the use of ℓ_1 sparseness-inducing penalties/log-priors appeared in the literature: the now famous *basis pursuit denoising* (BPDN, [11, Section 5]) criterion and the *least absolute shrinkage and selection operator* (LASSO, [54]). For brief historical accounts on the use of the ℓ_1 penalty in statistics and signal processing, see [41], [55].

Problem (1) is closely related to the following convex constrained optimization problems:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 \leq \varepsilon \tag{3}$$

and

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq t, \tag{4}$$

where ε and t are nonnegative real parameters. Problem (3) is a *quadratically constrained linear program* (QCLP) whereas (4) is a *quadratic program* (QP). Convex analysis can be used to show that a solution of (3) (for any ε such that this problem is feasible) is either $\mathbf{x} = 0$, or else is a minimizer of (1), for some $\tau > 0$. Similarly, a solution of (4) for any $t \geq 0$ is also a minimizer of (1) for some $\tau \geq 0$. These claims can be proved using [49, Theorem 27.4].

The LASSO approach to regression has the form (4), while the basis pursuit criterion [11, (3.1)] has the form (3) with $\varepsilon = 0$, *i.e.*, a linear program (LP)

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{5}$$

Problem (1) also arises in wavelet-based image/signal reconstruction and restoration (namely deconvolution); in those problems, matrix $\mathbf{A}$ has the form $\mathbf{A} = \mathbf{R}\mathbf{W}$, where $\mathbf{R}$ is (a matrix representation of) the observation operator (for example, convolution with a blur kernel or a tomographic projection), $\mathbf{W}$ contains a wavelet basis or a redundant dictionary (that is, multiplying by $\mathbf{W}$ corresponds to performing an inverse wavelet transform), and $\mathbf{x}$ is the vector of representation coefficients of the unknown image/signal [24], [25], [26].

M. Figueiredo is with the *Instituto de Telecomunicações* and Department of Electrical and Computer Engineering, *Instituto Superior Técnico*, 1049-001 Lisboa, **Portugal**. R. Nowak is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706, **USA**. S. Wright is with Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, **USA**.

This work was partially supported by NSF, grants CCF-0430504 and CNS-0540147, and by *Fundação para a Ciência e Tecnologia*, POSC/FEDER, grant POSC/EEA-CPS/61271/2004.

as an EM algorithm, in the context of image deconvolution problems [45], [25]. IST can also be derived in a *majorization-minimization* (MM) framework[2] [16], [26] (see also [23], for a related algorithm derived from a different perspective). Convergence of IST algorithms was shown in [13], [16]. IST algorithms are based on bounding the matrix $\mathbf{A}^T\mathbf{A}$ (the Hessian of $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$) by a diagonal $\mathbf{D}$ (*i.e.*, $\mathbf{D}-\mathbf{A}^T\mathbf{A}$ is positive semi-definite), thus attacking (1) by solving a sequence of simpler denoising problems. While this bound may be reasonably tight in the case of deconvolution (where $\mathbf{R}$ is usually a square matrix), it may be loose in the CS case, where matrix $\mathbf{R}$ usually has many fewer rows than columns. For this reason, IST may not be as effective for solving (1) in CS applications, as it is in deconvolution problems.

Finally, we mention matching pursuit (MP) and orthogonal MP (OMP) [5], [17], [20], [56], which are greedy schemes to find a sparse representation of a signal on a dictionary of functions. (Matrix $\mathbf{A}$ is seen as an n-element dictionary of k-dimensional signals). MP works by iteratively choosing the dictionary element that has the highest inner product with the current residual, thus most reduces the representation error. OMP includes an extra orthogonalization step, and is known to perform better than standard MP. Low computational cost is one of the main arguments in favor of greedy schemes like OMP, but such methods are not designed to solve any of the optimization problems above. However, if $\mathbf{y}=\mathbf{A}\mathbf{x}$, with $\mathbf{x}$ sparse and the columns of $\mathbf{A}$ sufficiently incoherent, then OMP finds the sparsest representation [56]. It has also been shown that, under similar incoherence and sparsity conditions, OMP is robust to small levels of noise [20].

C. Proposed Approach

The approach described in this paper also requires only matrix-vector products involving $\mathbf{A}$ and $\mathbf{A}^T$, rather than explicit access to $\mathbf{A}$. It is essentially a gradient projection (GP) algorithm applied to a quadratic programming formulation of (1), in which the search path from each iterate is obtained by projecting the negative-gradient direction onto the feasible set. (See [3], for example, for background on gradient projection algorithms.) We refer to our approach as GPSR (*gradient projection for sparse reconstruction*). Various enhancements to this basic approach, together with careful choice of stopping criteria and a final debiasing phase (which finds the least squares fit over the support set of the solution to (1)), are also important in making the method practical and efficient.

Unlike the MM approach, GPSR does not involve bounds on the matrix $\mathbf{A}^T\mathbf{A}$. In contrasts with the IP approaches discussed above, GPSR involves only one level of iteration. (The approaches in [11] and [36] have two iteration levels—an outer IP loop and an inner CG, PCG, or LSQR loop. The ℓ_1-*magic* algorithm for (3) has three nested loops—an outer log-barrier loop, an intermediate Newton iteration, and an inner CG loop.)

GPSR is able to solve a sequence of problems (1) efficiently for a sequence of values of τ. Once a solution has been obtained for a particular τ, it can be used as a "warm-start" for a nearby value. Solutions can therefore be computed for a range of τ values for a small multiple of the cost of solving for a single τ value from a "cold start." This feature of GPSR is somewhat related to that of LARS and other homotopy schemes, which compute solutions for a range of parameter values in succession. In particular, "warm-starting" allows using GPSR within a continuation scheme (as suggested in [31]). IP methods such as those in [11], [36], and ℓ_1-*magic* have been less successful in making effective use of warm-start information, though this issue has been investigated in various contexts (see, *e.g.*, [30], [35], [61]). To benefit from a warm start, IP methods require the initial point to be not only close to the solution but also sufficiently interior to the feasible set and close to a "central path," which is difficult to satisfy in practice.

[2]Also known as bound optimization algorithms (BOA). For a general introduction to MM/BOA, see [33].

II. Proposed Formulation

A. Formulation as a Quadratic Program

The first key step of our GPSR approach is to express (1) as a quadratic program; as in [28], this is done by splitting the variable $\mathbf{x}$ into its positive and negative parts. Formally, we introduce vectors $\mathbf{u}$ and $\mathbf{v}$ and make the substitution

$$\mathbf{x}=\mathbf{u}-\mathbf{v},\quad \mathbf{u}\geq 0,\ \mathbf{v}\geq 0. \tag{6}$$

These relationships are satisfied by $u_i=(x_i)_+$ and $v_i=(-x_i)_+$ for all $i=1,2,\ldots,n$, where $(\cdot)_+$ denotes the *positive-part operator* defined as $(x)_+=\max\{0,x\}$. We thus have $\|\mathbf{x}\|_1=\mathbf{1}_n^T\mathbf{u}+\mathbf{1}_n^T\mathbf{v}$, where $\mathbf{1}_n=[1,1,\ldots,1]^T$ is the vector consisting of n ones, so (1) can be rewritten as the following bound-constrained quadratic program (BCQP):

$$\begin{array}{ll}\min\limits_{\mathbf{u},\mathbf{v}} & \frac{1}{2}\|\mathbf{y}-\mathbf{A}(\mathbf{u}-\mathbf{v})\|_2^2+\tau\,\mathbf{1}_n^T\mathbf{u}+\tau\,\mathbf{1}_n^T\mathbf{v},\\ \text{s.t.} & \mathbf{u}\geq 0\\ & \mathbf{v}\geq 0.\end{array} \tag{7}$$

Note that the ℓ_2-norm term is unaffected if we set $\mathbf{u}\leftarrow\mathbf{u}+\mathbf{s}$ and $\mathbf{v}\leftarrow\mathbf{v}+\mathbf{s}$, where $\mathbf{s}\geq 0$ is a shift vector. However such a shift increases the other terms by $2\,\tau\,\mathbf{1}_n^T\mathbf{s}\geq 0$. It follows that, at the solution of the problem (7), $u_i=0$ or $v_i=0$, for $i=1,2,\ldots,n$, so that in fact $u_i=(x_i)_+$ and $v_i=(-x_i)_+$ for all $i=1,2,\ldots,n$, as desired.

Problem (7) can be written in more standard BCQP form,

$$\begin{array}{ll}\min\limits_{\mathbf{z}} & \mathbf{c}^T\mathbf{z}+\frac{1}{2}\,\mathbf{z}^T\mathbf{B}\mathbf{z}\equiv F(\mathbf{z}),\\ \text{s.t.} & \mathbf{z}\geq 0,\end{array} \tag{8}$$

where

$$\mathbf{z}=\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix},\quad \mathbf{b}=\mathbf{A}^T\mathbf{y},\quad \mathbf{c}=\tau\,\mathbf{1}_{2n}+\begin{bmatrix}-\mathbf{b}\\ \mathbf{b}\end{bmatrix}$$

and

$$\mathbf{B}=\begin{bmatrix}\mathbf{A}^T\mathbf{A} & -\mathbf{A}^T\mathbf{A}\\ -\mathbf{A}^T\mathbf{A} & \mathbf{A}^T\mathbf{A}\end{bmatrix}. \tag{9}$$