# Optimization Methods for $\ell_1$-minimization

Sangwoon Yun

Computational Sciences
Korea Institute for Advanced Study

December 11, 2009
2009 NIMS Thematic Winter School

# Outline

- Greedy Algorithm
- Optimization Methods for Convex Relaxation
- Extensions

# Greedy Algorithm

1. starting from $x^0 = 0$
2. iteratively constructs a $k$-term approximant $x^k$ by maintaining a set of active columns and, at each stage, expanding that set by one additional column
3. column chosen at each stage maximally reduced the $\|Ax - b\|_2$ error in approximating $b$ from the currently active columns
4. $\|Ax - b\|_2$ error is evaluated; if it now falls below a specified threshold, the algorithm terminates.

- Matching Pursuit
- Orthogonal Mathing Pursuit
- Stagewise OMP
- Regularized OMP
- C0mpressive Sampling MP

when the solution is sparse and the columns of $A$ sufficiently incoherent

# Optimization Methods for Convex Relaxation

Homotopy Method (Osborne et al. '00, Donoho & Tsaig '06) for $\ell_1$ QC, BPDN, LASSO:

- find the full path of solutions for all nonnegative values of the scalar parameters in the various formulations
- only the submatrix of $A$ corresponding to nonzero components of the current vector $x$ need be known explicitly, so that if $x$ has few nonzeros, these methods may be competitive even for problems of very large scale.

$\ell_1$-magic (Candés et al. '05) for $\ell_1$ QC:

- $\ell_1$ QC is solved by recasting it as a second-order cone program (SOCP), then applying a primal log-barrier approach.
- for each value of the log-barrier parameter, smooth unconstrained subproblem is solved using Newton's method with line search, where the Newton equations may be solved using CG.

$\ell_1$-regularized least squares (Kim et al. '07) ($\ell_1$-ls) for BPDN:

- interior-point method (with log-barrier)
- using preconditioned conjugate gradient (PCG) for solving the linear equations at each iteration

Gradient Projections for Sparse Reconstruction (Figueiredo et al. '07) (GPSR) for BPDN:

- gradient projection method for solving bound constrained quadratic programming reformulation of BPDN:

$$\begin{array}{ll} \min\limits_{x^+,x^-} & \frac{1}{2}\|A(x^+ - x^-) - b\|_2^2 + \mu(e^T x^+ + e^T x^-) \\ \text{s.t} & x^+, x^- \geq 0 \end{array}$$

- in order to accelerate convergence, a technique based on Barzilai-Borwein (BB) steps is used.

Iterative Shrinkage/Thresholding (Figueiredo and Nowak '03, Daubechies et al. '04) (IST) for BPDN:

- consists of a soft-thresholding/shrinkage step and a gradient step

$$d^k = \arg\min_d (A^T(Ax^k - b))^T d + d^T H^k d + \mu \|x^k + d\|_1,$$

where $H^k = \alpha I$ for some positive constant $\alpha$.

Fixed Point Continuation (Hale et al. '07) (FPC) for BPDN:

- similar to IST
- the parameter $\mu$ in BPDN determines the amount of shrinkage and, therefore, the speed of convergence
- in practice, $\mu$ is decreased in a continuation scheme.

Sparse Reconstruction by Separable Approximation (Wright et al. ’09) (SpaRSA) for BPDN:

- similar to IST, also like FPC, continuation is used to speed convergence
- a Barzilai-Borwein heuristic is used for the step size $\alpha$ (instead of using a pessimistic bound like the Lipschitz constant)

FPC Active set (Wen et al. ’09) (FPC-AS) for BPDN:

- extend FPC into the two-part algorithm FPC Active Set
- in the first stage, calls FPC
- in the second stage, use conjugate gradients (CG) or quasi-Newton methods (e.g. L-BFGS or L-BFGS-B), for solving reduced (active set) bound constrained problem
- two-step process is repeated for a smaller value of $\mu$ in a continuation scheme

Coordinate Gradient Descent (Yun and Toh '09) (CGD) for BPDN:

- similar to IST
- but use certain coordinate block to update:

$$d^k = \operatorname*{arg\,min}_{d, d_i = 0, i \notin J} (A^T(Ax^k - b))^T d + d^T H^k d + \mu \|x^k + d\|_1,$$

Bregman (Yin et al. '09) for $\ell_1$-min:

- outer iteration, updated observation vector b
- inner iteration, solves BPDN
- typically, only a few outer iterations are needed, but each iteration requires a solve of BPDN, which is costly.
- a version of the Bregman algorithm, known as the Linearized Bregman algorithm, takes only one step of the inner iteration per outer iteration; consequently, many outer iterations are taken; linearized Bregman is equivalent to gradient ascent on the dual problem.

Spectral Projected Gradient for $\ell_1$ (van den Berg et al. '09) (SPGL1) for $\ell_1$ QC:

- adapted the spectral projection gradient algorithm.
- interestingly, they introduced a clever root finding procedure such that solving a few instances of LASSO for different values of $\tau$ enables them to equivalently solve $\ell_1$ QC.

Fast Iterative Soft-Thresholding Algorithm (Beck and Teboulle '08) (FISTA) for BPDN:

- belongs to the class of accelerated proximal gradient (APG) algorithms studied earlier by Nesterov, Nemirovski, and others.
- theoretical rate of convergence is $O(1/k^2)$ (or $O(1/\sqrt{\epsilon})$).

Nestrov's Algorithm (Becker et al '09) (NESTA) for $\ell_1$ QC:

- based on Nesterov's smoothing technique like FISTA
- one of the key ideas is an averaging of sequences of iterates, which has been shown to improve the convergence properties of standard gradient-descent algorithms

## Example on image reconstruction

Show some images for image deblurring, inpainting, decomposition by balanced approach

$$\min_{x_1,x_2} \ \frac{1}{2}\|A(\sum_{i=1}^{2} W_i^T x_i) - b\|_D^2 + \sum_{i=1}^{2} \frac{\kappa_i}{2}\|(I - W_i W_i^T)x_i\|^2 + \sum_{i=1}^{2} \lambda_i^T |x_i|,$$

where, for $i = 1, 2$, $W_i^T W_i = I$, $\kappa_i > 0$, $\lambda_i$ is a given positive weight vector, and $D$ is a given symmetric positive definite matrix.

# Extensions

$\ell_1$-regularized logistic regression ($\ell_1$LR) problem

Proposed as a promising method for feature selection in classification problems.

$$\min_{w \in \Re^{n-1}, v \in \Re} \ \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-(w^T a_i + v b_i))) + \mu \|w\|_1,$$

where $a_i = b_i z_i$ and $(z_i, b_i) \in \Re^{n-1} \times \{-1, 1\}, \ i = 1, ..., m$ are a given set of (observed or training) examples.

## Matrix Completion

imagine that we only observe a few entries of a data matrix. Then is it possible to accurately guess the entries that we have not seen?

Netflix problem: Given a sparse matrix where $M_{ij}$ is the rating given by user $i$ on movie $j$, predict the rating a user would assign to a movie he has not seen, i.e., we would like to infer users preference for unrated movies. (impossible! in general)

The problem is ill-posed. Intuitively, users' preferences depend only on a few factors, i.e., $\mathrm{rank}(M)$ is small.

Thus consider the low-rank matrix completion problem:

$$\min_{X \in \Re^{m\times n}} \left\{ \mathrm{rank}(X) \mid X_{ij} = M_{ij},\ (i,j) \in \Omega \right\}, \quad \text{(NP hard!)}$$

where $\Omega =$ index set of $p$ observed entries.

We assume $m \leq n$ w.l.o.g.

This matrix completion problem has appeared in many applications of engineering and science including

- Collaborative Filtering
- System Identification
- Global Positioning
- Remote Sensing
- Machine Learning
- Computer Vision

By (Candés & Recht 08, Candés & Tao 09), a random rank-$r$ matrix can be recovered exactly with high probability from a uniform random sample of $p = O(rn\,\mathrm{polylog}(n))$ entries by solving the following convex relaxation:

$$\min_{X\in\Re^{m\times n}} \left\{\|X\|_* := \sum_{i=1}^{m} \sigma_i(X) \,|\, X_{ij} = M_{ij},\ (i,j)\in\Omega\right\}.$$

where $\sigma_i(X)$'s are singular values of $X$.
or the following nuclear norm regularized least squares problem:

$$\min_{X\in\Re^{m\times n}} \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 + \mu\|X\|_*.$$

where $\mu > 0$ is a given parameter.
$\mathcal{A}(X) = X_\Omega$, where $X_\Omega$ is the vector in $\Re^{|\Omega|}$ obtained from $X$ by selecting those elements whose indices are in $\Omega$.

## Sparse Covariance Selection

$$\min_{X \in \mathcal{S}^n_+} \; -\log\det X + \mathrm{tr}(XS) + \mu\|X\|_1$$

$S \in \mathcal{S}^n_+$ is an empirical covariance matrix.
$\|X\|_1 = \sum_{i,j=1}^n |X_{ij}|$, $\mu > 0$.

$-\log\det X + \mathrm{tr}(XS)$ is strictly convex, cont. diff. on its domain $\mathcal{S}^n_{++}$, $O(n^3)$ opers. to evaluate. $\|\cdot\|_1$ is convex, nonsmooth.
In applications, $n$ can exceed 5000.

The Fenchel dual problem is a bound-constrained convex program:

$$\begin{array}{cl} \min\limits_{W \in \mathcal{S}^n_+} & -\log\det W \\ \text{s.t.} & |(W-S)_{ij}| \le \mu,\ i,j = 1,...,n, \end{array}$$

IP method requires $O(n^7 \log(1/\epsilon))$ opers. to find $\epsilon$-optimal soln. Impractical! Nesterov's first-order smoothing method requires $O(n^4/\epsilon)$ opers. (Lu '07). Use coordinate descent method to solve the dual problem, cycling thru columns (and corresponding rows) $j = 1, ..., n$ of $W$. Each iteration reduces (via determinant property & duality) to

$$\min_{w \in \Re^{n-1}} \frac{1}{2} w^T W_{j^c j^c} w - S^T_{j^c j} w + \mu \|w\|_1.$$

Solve this using IP method $O(n^3)$ opers. or coordinate descent method (Friedman '07).

## Sparse Principle Component Analysis

Principal component analysis (PCA) is a widely used technique for data analysis and dimension reduction with numerous applications in science and engineering.

PCA seeks the linear combinations of the original variables such that the derived variables capture maximal variance. PCA can be done via singular value decomposition (SVD) of the data matrix.

$$X = UDV^T,$$

where $X \in \Re^{n \times p}$ with the number of observations $n$ and the number of variables $p$. $U(D)$ are the principal components (PCs) of unit length, and the columns of $V$ are the corresponding loadings of the principal components.

However, the standard PCA suffers from the fact that the principal components (PCs) are usually linear combinations of all the original variables, and it is thus often difficult to interpret the PCs.

consider sparse model

The success of PCA is due to the following three important optimal properties:
1. principal components sequentially capture the maximum variability among $X$, thus guaranteeing minimal information loss.
2. principal components are uncorrelated, so we can talk about one principal component without referring to others (the explained variance by different PCs has small overlap).
3. principal components point in orthogonal directions, that is, their loading vectors are orthogonal to each other.

## Suggested Problem Formulation

Finding sparse PCs by solving a sequence of semidefinite program relaxations of sparse PCA.

$$\max_V \langle \Sigma, V \rangle - \sum_{i,j=1,\dots,p} |V_{ij}| \quad \langle V \rangle = 1,\ V \succeq 0.$$

where $\Sigma = X^T X/(n-1)$.

$$\begin{array}{rl} \max\limits_{V} & \langle V^T \Sigma V \rangle - \rho \sum_{i,j=1}^{n} |V_{ij}| \\ \text{s.t.} & |V_i \Sigma V_j| \le \Delta_{ij} \ \forall i \ne j \\ & V^T V = I. \end{array}$$

where $\Delta_{ij} \ge 0$ $(i \ne j)$ are the parameters for controlling the correlation of the components corresponding to $V$.