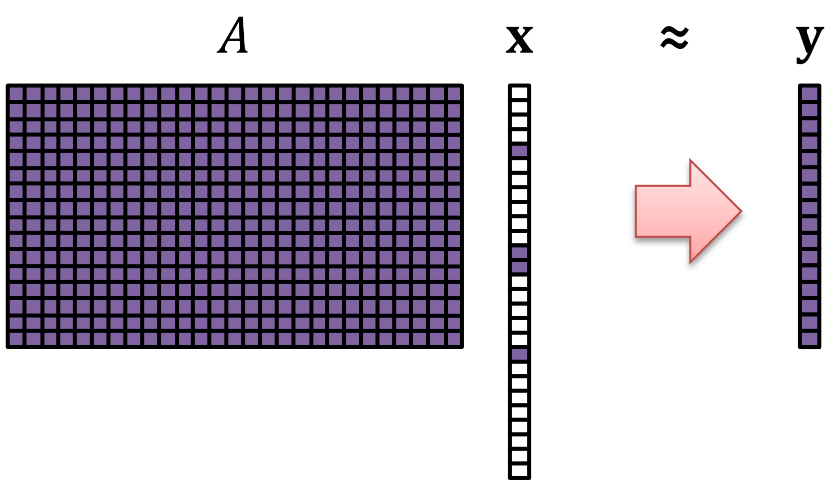
A MULTILEVEL ITERATED SHRINKAGE APPROACH TO l_1 PENALIZED LEAST-SQUARES

ABSTRACT. The area of sparse representation of signals is drawing tremendous attention in recent years. The sparse representation of a signal is often achieved by minimizing l_1 penalized least squares functional. Our new method takes advantage of the typically sparse representation of the signal. At each iteration it adaptively creates and processes a hierarchy of lower-dimensional problems, based on non-zero sparsity patterns, employing well-known iterated shrinkage methods, and gradually ignoring irrelevant data from the over-complete dictionary. In addition, we examine another way of analyzing the dictionary, in order to improve performance and reduce the runtime complexity. This new approach may significantly enhance the performance of existing iterative shrinkage algorithms in cases where the dictionary is explicit matrix.

A MULTILEVEL APPROACH FOR L1 PENALIZED LEAST-SQUARES MINIMIZATION

A signal $y \in \mathbb{R}^n$ can be approximated as a linear combination of a few columns (often called "atoms") from an over-complete matrix (often called "dictionary") $A \in \mathbb{R}^{n \times m}$, where $m > n$, A is redundant. That is, $y \approx Ax^*$, where the *representation vector* $x^* \in \mathbb{R}^m$ is sparse, containing few non-zero elements. The signal y represented by only a few columns of A.



One approach for applying such reconstructions uses the well-known l_1, which has somewhat similar "sparsity properties".

One common approach features an l_1 penalized least-squares minimization:

$$(1)\ \min_{x \in \mathbb{R}^m} f(x) = \min_{x \in \mathbb{R}^m} \frac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_1,$$

with $\lambda > 0$ a scalar parameter that balances between sparsity and adherence to the data.

$$(2)\ \frac{1}{2}\|Ax - y\|_2^2 = \cdots = \frac{1}{2}x^T A^T A x - x^T A^T y + \frac{1}{2}y^T y$$

$M = \frac{1}{2}A^T A \in \mathbb{R}^{m\times m}$, $c = A^T y$, $b = \frac{1}{2}y^T y$. M is a symmetric matrix.

(3) $\frac{1}{2}\|Ax - y\|_2^2 = \cdots = \frac{1}{2}x^T A^T Ax - x^T A^T y + \frac{1}{2}y^T y = x^T Mx - x^T c + b$

COORDINATE DESCENT (CD)

In each iteration of CD, we update the elements of x one by one on some prescribed order. Updating the element x_i requires minimizing the following functional for the scalar variable z, which then replaces x_i:

$\min_z f(z) = \frac{1}{2}\|Ax - a_i x_i + a_i z - y\|_2^2 + \lambda(\|x\|_1 - |x_i| + |z|)$, $\|x\|_1 = \sum_{j=1}^m |x_j|$

Where a_i is the ith column of A, and m_i is the ith column of M.

$r = y - Ax, \tilde{r} = r + a_i x_i$

$$\min_z f(z) = \frac{1}{2}\|a_i z - \tilde{r}\|_2^2 + \lambda(\|x\|_1 - |x_i| + |z|) = \cdots$$

$$= \frac{1}{2}z^2 a_i^T a_i - z a_i^T \tilde{r} + \tilde{r}^T\tilde{r} + \lambda(\|x\|_1 - |x_i| + |z|) = z^2 M_{ii} - z\left(c_i - m_i^T x + M_{ii}x\right) + \cdots$$

$$f'(z) = z a_i^T a_i - a_i^T \tilde{r} + \lambda * sign(z) = zM_{ii} - c_i + m_i^T x - M_{ii}x + \lambda * sign(z) = 0$$

$$z = \frac{a_i^T\tilde{r} - \lambda}{a_i^T a_i} = \frac{a_i^T(r + a_i x_i) - \lambda}{a_i^T a_i} = \frac{a_i^T r - \lambda}{a_i^T a_i} + x_i = \frac{c_i - m_i^T x - \lambda}{M_{ii}} + x_i$$

Note:

$$z = \frac{a_i^T\tilde{r} - \lambda * sign(z)}{a_i^T a_i}, \lambda > 0$$

- Assume that $a_i^T\tilde{r} < \lambda, sign(z) == + \rightarrow\ z = \frac{a_i^T\tilde{r} - \lambda * sign(z)}{a_i^T a_i} = \frac{a_i^T\tilde{r} - \lambda}{a_i^T a_i} < 0 \rightarrow contradiction.$
- Assume that $a_i^T\tilde{r} > \lambda, sign(z) == + \rightarrow\ z = \frac{a_i^T\tilde{r} - \lambda * sign(z)}{a_i^T a_i} = \frac{a_i^T\tilde{r} - \lambda}{a_i^T a_i} > 0 \rightarrow OK.$
- Assume that $a_i^T\tilde{r} > -\lambda, sign(z) == - \rightarrow\ z = \frac{a_i^T\tilde{r} - \lambda * sign(z)}{a_i^T a_i} = \frac{a_i^T\tilde{r} + \lambda}{a_i^T a_i} > 0 \rightarrow contradiction.$
- Assume that $a_i^T\tilde{r} < -\lambda, sign(z) == - \rightarrow\ z = \frac{a_i^T\tilde{r} - \lambda * sign(z)}{a_i^T a_i} = \frac{a_i^T\tilde{r} + \lambda}{a_i^T a_i} < 0 \rightarrow OK.$

In conclusion the following options are possible:

1. $a_i^T\tilde{r} < -\lambda$ or,
2. $a_i^T\tilde{r} > \lambda$

We define a “shrinkage” function:

$$\mathcal{S}_q(t) \equiv sign(t) \cdot \max(0, |t| - q)$$

Therefore, we can write the solution as follows:

$$z_{opt} = \mathcal{S}_{\frac{\lambda}{a_i^T a_i}}\left(\frac{a_i^T r}{a_i^T a_i} + x_i\right) = \mathcal{S}_{\frac{\lambda}{M_{ii}}}\left(\frac{c_i - m_i^T x}{M_{ii}} + x_i\right).$$

From these equations we get the following algorithm for Coordinate Descent (CD):

$input: r = y - Ax, \lambda, x, A$

For each x_i do minimization in one variable:

$for\ i = 1 \dots m$:

$$x_i^{new} \leftarrow \mathcal{S}_{\frac{\lambda}{a_i^T a_i}}\left(\frac{a_i^T r}{a_i^T a_i} + x_i^{old}\right) \rightarrow O(n)$$

$$if\ \left(\left|x_i^{new} - x_i^{old}\right| > 0\right)\ then: r \leftarrow r - a_i\left(x_i^{new} - x_i^{old}\right) \rightarrow O(n)$$

end

$output: A^T r, x, r$

Complexity: $O(m) * O(n) + k * O(n), k \ll n < m \rightarrow O(nm). \left(k = \left|\left\{i: \left|x_i^{new} - x_i^{old}\right| > 0\right\}\right|\right)$

And we can also solve the problem as suggested with different parameters, the algorithm is:

$input: x, M, Mx, c, \lambda$

$for\ i = 1 \dots m$:

$$x_i^{new} \leftarrow \mathcal{S}_{\frac{\lambda}{M_{ii}}}\left(\frac{c_i - m_i^T x}{M_{ii}} + x_i^{old}\right) \rightarrow O(1).$$

// Notice that $m_i^T x = Mx[i]$, and therefore $O(1)$ accessing it.

$$if\ \left(\left|x_i^{new} - x_i^{old}\right| > 0\right)\ then: Mx \leftarrow Mx + m_i\left(x_i^{new} - x_i^{old}\right) \rightarrow O(m)$$

end

$output: x, Mx$

Complexity: $O(m) + k * O(m) = O(km). \left(k = \left|\left\{i: \left|x_i^{new} - x_i^{old}\right| > 0\right\}\right|\right)$

We would like to use that algorithm when the calculation should take place many times, for example - for different signals (since we use the same dictionary).

Linesearch

We use line-search so that $f\left(x^{k+1^*}\right) = f\left(x^k + \alpha(x^{k+1} - x^k)\right), \alpha \in \mathbb{R}^+$ is a line-search scalar. This way we accelerate the convergence to x^* [1].

Multilevel Iterated Shrinkage

As was demonstrated earlier, the solution is sparse – most columns will not end up in the support. Therefore, we introduce a multilevel method for (1); we accelerate the convergence of simple iterative method for (1) using a nested hierarchy of smaller versions of the problem. At each iteration, called a “V-cycle”, our algorithm reduces the dimension of the problem and creates a multilevel hierarchy of smaller and smaller problems, low-level problems, involving a lower dimensional dictionary at each “level”. We take advantage of the typical sparsity of x and reduce the dimension of the problem (1) by ignoring ostensibly irrelevant columns from A. That is, each low-level problem restricted to a specially chosen subset of the columns of A, resulting in a nested hierarchy of sub-dictionaries. It then performs shrinkage sweeps over each of the low dimensional problems in turn, that aim to activate the atoms which comprise the support of a true minimizer. We iteratively repeat these V-cycle until some convergence criterion is satisfied. Under suitable conditions our algorithm converges to a global minimizer of (1).

Definition of the low-level problem

We now define the reduced problem given its designated subset of atoms, $C \subset \{1, \dots, m\}$[1]. We define a zero-filling prolongation matrix $P \in \mathbb{R}^{m \times |C|}$, such that $x = Px_c$ retains the values of x_c for the elements in C, and sets the other elements of x to zero.

We get the new problem:

$$\|x\|_1 = \|Px_c\|_1 = \|x_c\|_1$$

$$\min_{x \in \mathbb{R}^m} f(x) = \min_{x_c \in \mathbb{R}^{|C|}} f(Px_c) = \min_{x_c \in \mathbb{R}^{|C|}} f_c(x_c) = \min_{x_c \in \mathbb{R}^{|C|}} \frac{1}{2}\|A_c x_c - y\|_2^2 + \lambda \|x_c\|_1$$

Where A_c is the reduced sub-dictionary of the upper-level dictionary A, with columns given by the columns of A corresponding to the indices in C. We can recursively extend this two-level framework to multi levels.

And for the second approach for the problem:

$$\min_{x \in \mathbb{R}^m} f(x) = \min_{x_c \in \mathbb{R}^{|C|}} f(Px_c) = \min_{x_c \in \mathbb{R}^{|C|}} f_c(x_c) = x_c^T M_c x_c - x_c^T c + b + \lambda \|x_c\|_1$$

Where M_c is the sub-matrix of M, and $\forall i, j \in C, M_{ij} = a_i^T a_j$ corresponds to entry in M_c.

Choosing the low-level variables

Our low-level functional definition above suggests that we need to select a subset of low-level variables C, that is likely as possible to contain the support of the true minimizer [1].

$$supp(x) = \{i: x_i \neq 0\}$$

If $|supp(x)| < \lceil m/2 \rceil$, we add atoms corresponding to the largest values of in

$|A^T(Ax - y)| = |Mx - c|$. We choose these atoms for the support because they contribute significantly to the solution and for the true minimizer x^* [1].

Multilevel V-cycle

For solving the minimization problem, we repeat $x^{k+1} \leftarrow Vcycle(A, x^k, y, v)$ iteratively, until some convergence criterion is satisfied. The multilevel V-cycle procedure, along with its parameters, is defined in $Algorithm 1$. The multilevel V-cycle procedure for the second approach with M dictionary is defined in $Algorithm 2$.

The algorithms use CD iterated shrinkage methods as "relaxations".

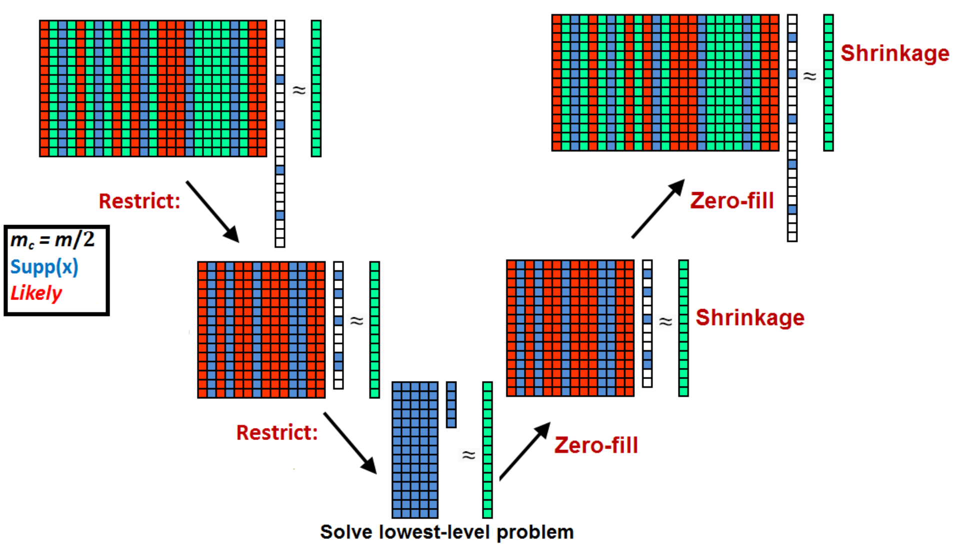
$Algorithm 1: x \leftarrow Vcycle(A, x, y, v)$

$\%\ Iterative\ Shrinkage\ method: CD(A, x, y, \lambda).$

$\%\ number\ of\ relaxations\ at\ each\ level: v.$

1) Choose low-level variables C.
2) If $C = supp(x)$ or $|C| < 2m_{\min}$,
    a. Solve the lowest-level problem (4).

   Else $x_c \leftarrow Vcycle(A_c, x_c, y, v)$
3) Prolong solution: $x \leftarrow Px_c$
4) Apply v relaxations: $x \leftarrow CD(A, x, y, \lambda)$



Complexity:

$$O\left(nm + \frac{1}{2}nm + \frac{1}{4}nm + \cdots + n|C| = nm\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) + n|C|\right) = O(2nm + n|C|) = 2T_{CD(A,y,\lambda)}$$

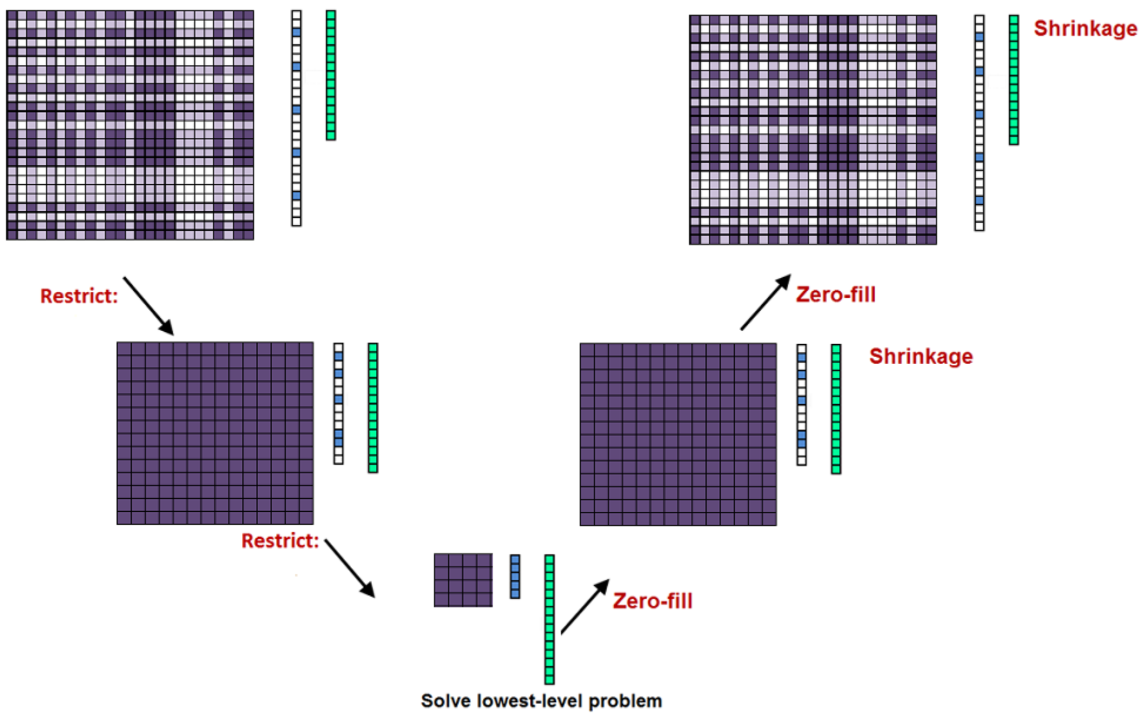
$Algorithm2: x \leftarrow Vcycle(M, x, c, \nu)$

% $Iterative\ Shrinkage\ method: CD(M, x, Mx, c, \lambda)$.

% $number\ of\ relaxations\ at\ each\ level: \nu$.

1) Choose low-level variables C.
2) If $C = supp(x)$ or $|C| < 2m_{\min}$,
   a. Solve the lowest-level problem (4).

   Else $x_c \leftarrow Vcycle(M_c, x_c, c, \nu)$
3) Prolong solution: $x \leftarrow Px_c$
4) Apply ν relaxations: $x \leftarrow CD(M, x, Mx, c, \lambda)$



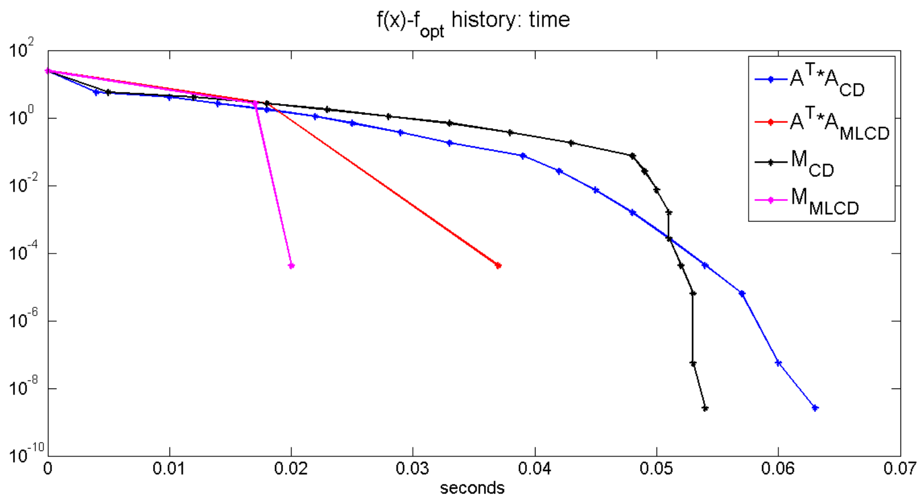
Complexity:

$$O\left(km + \frac{1}{2}km + \frac{1}{4}km + \cdots + k|C| = km\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) + k|C|\right) = O(2km + k|C|) = 2T_{CD(M,c,\lambda)}$$

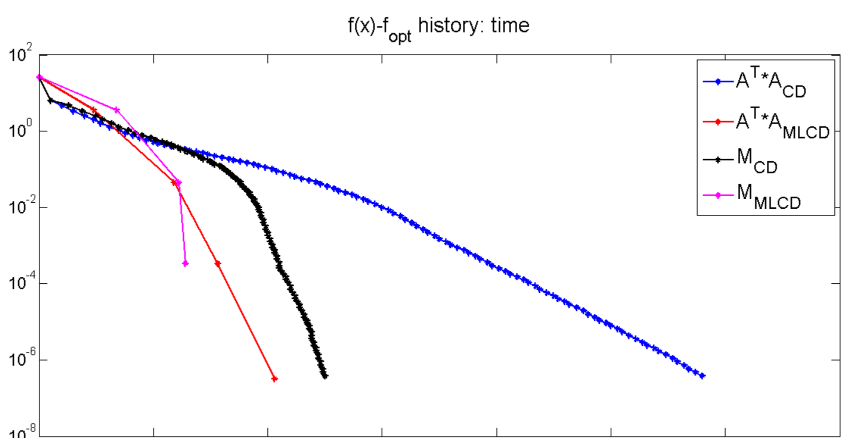
Numerical Results

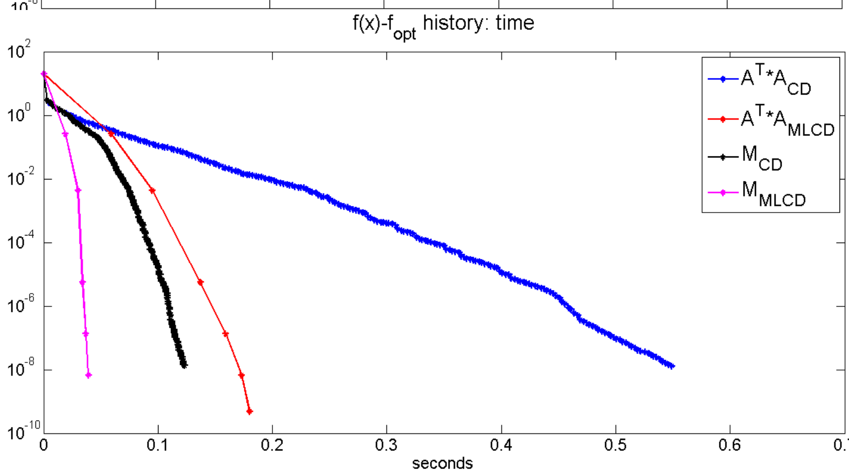
The following graphs show the convergence to $\min f(x)$. In each graph we see the run time of CD of the two representations: $CD(A, y, \lambda)$ (the blue graph) and $CD(M, c, \lambda)$ (the black graph). In addition, the graphs of the V-cycle, for both representations, are: $Vcycle(A, y, \lambda)$ (the red graph) and $Vcycle(M, c, \lambda)$ (the pink graph).

In all the experiments, the dictionary is $A \in \mathbb{R}^{512\times 2048}, and\ \lambda = 0.08$. Stopping criterion: reaching accuracy of 10^{-4}.



For ill conditioned dictionaries (in which converging to the minimum is harder task):





In the results, we can see that $CD(M, c, \lambda)$ is faster than $CD(A, y, \lambda)$, and $Vcycle(M, c, \lambda)$ is faster than $Vcycle(A, y, \lambda)$. Moreover, Vcycle is faster than CD, and takes less iterations until convergence is reached.

Conclusion

A multilevel approach is introduced for the solution of (1) when the matrix A is given explicitly. The new method takes advantage of the typically sparse representation of the signal by gradually ignoring ostensibly irrelevant data from the over-complete dictionary. This approach accelerates the performance of iterated shrinkage methods. Using matrix $M = A^T A$ causes further acceleration. However, computing M is very expensive un term of runtime, so we would like to use this approach when the calculation should take place many times, for example - for different signals.

Reference

[1] E. Treister and I. Yavneh, *A multilevel iterated-shrinkage approach to l_1 penalized least-squares minimization*, Signal Processing, IEEE Transactions on, 60 (2012), pp. 6319-6329.