Algorithms for Group Lasso

Yihang Chen
Department of Mathematics
Peking University
1700010780

Contents

1 Problem formulation

Lasso

Consider the ℓ_1-regularized problem

$$\min_x \quad f_\mu(x) = g(x) + \mu h(x) := \frac{1}{2}\|Ax - b\|_F^2 + \mu\|x\|_{1,2} \tag{1.1}$$

where $A \in \mathbb{R}^{m\times l}, b \in \mathbb{R}^{n\times l}$ and $\mu > 0$ are given.

2 Commercial Solvers

We can obtain and solutions directly from cvx mosek and cvx gurobi.

Denote $\hat{x} = \mathrm{vec}(x), \hat{b} = \mathrm{vec}(b)$, and $\hat{A} = \mathrm{diag}\underbrace{\{A, \cdots, A\}}_{l\times A}$. The indicator matrix of group p is a diagnoal matrix whose jj-th element is defined as 1 if $n|j-p$ and 0 otherwise.

The group lasso problem (1.1) is equivalent to the following cone optimization problem

$$\begin{aligned} \min \quad & t + \mu \sum_{p=1}^{n} t_n \\ \text{s.t.} \quad & \|[2s, t-2]\|_2 \le t+2 \\ & \|\hat{A}\hat{x} - \hat{b}\|_2 \le s \\ & \|I_p\hat{x}\|_2 \le t_p, \quad 1 \le p \le n \end{aligned} \tag{2.1}$$

by with we use the fact that $x^\top x \le yz$ iff $\|[2x; y-z]\|_2 \le y+z$.

The problem (2.2) can be solved by mosek. The variable is organized as $[\hat{x}^\top, s, t, t_1, \cdots, t_n] \in \mathbb{R}^{nl+n+2}$.

For gurobi, the variable is organized as $[\hat{y}^\top, \hat{x}^\top, s, t, t_1, \cdots, t_n] \in \mathbb{R}^{(m+n)l+n+2}$.. Since gurobi only support the QCQP problem, if we directly transform SOCP (2.2) into QCQP, it might encounter a general type of non-convex optimization ($\hat{A}^\top\hat{A} - I_s I_s^\top$ is indefinite, where I_s is the indicator of variable s) and is very inefficient. Note that the gurobi solver can solve the non-convex problem in the form $x^\top x \le y^2$ or $x^\top x \le yz$ efficiently, we introduce a new variable $\hat{y} = \hat{A}\hat{x} - \hat{b}$ to deal with the situation.

Empirically, it is much more efficient to introduce $\hat{y}$ than use non-convex optimization (require 'params.NonConvex=2', only supported in Gurobi 9.1).

$$\begin{aligned} \min \quad & t + \mu \sum_{p=1}^{n} t_n \\ \text{s.t.} \quad & \hat{A}\hat{x} - \hat{b} = \hat{y} \\ & \|[2s, t-2]\|_2 \le t+2 \\ & \|\hat{y}\|_2 \le s \\ & \|I_p\hat{x}\|_2 \le t_p, \quad 1 \le p \le n \end{aligned} \tag{2.2}$$

Specifically, we use the sparse matrix in MATLAB to accelerate the codes. A good reference of Mosek and Gurobi on cone optimization can be found at their websites. [12]

[1]https://docs.mosek.com/9.2/toolbox/case-studies-regression.html
[2]https://www.gurobi.com/documentation/9.0/examples/qcp_m.html

3 Various algorithms

3.1 Subgradient method for the primal problem

The subgradient of f_μ is $\partial f_\mu(x) = A^\top(Ax-b) + \mu \cdot \mathrm{Diag}(xx^\top)^{\otimes -\frac{1}{2}} x$ in the following way:

Define $b = \mathrm{diag}(xx^\top), B = \mathrm{Diag}(b)$, where "diag" outputs the diagonal vector while "Diag"" outputs the diagonal matrix, then $\|x\|_{1,2} = \mathbb{1}^\top b^{\otimes \frac{1}{2}}$

$$d\|x\|_{1,2} = \frac{1}{2}\mathbb{1}^\top b^{\otimes -\frac{1}{2}} \otimes db = \frac{1}{2}(db)^\top b^{\otimes -\frac{1}{2}} = \frac{1}{2}(dB)^\top B^{\otimes -\frac{1}{2}} = (B^{\otimes -\frac{1}{2}} x)^\top dx \tag{3.1}$$

The subgradient method can be summarized in Algorithm 1.

Algorithm 1 Subgradient method for the primal problem with continuation method

---

1: **Input:** initial value x_0, step size α, continuation parameter γ, N, maximum iteration number for each stage M.
2: **for** $i = 1, \cdots, N$ **do**
3: $\quad \mu_i = \gamma^{N-i}\mu$.
4: $\quad$ **for** $j = 1, \cdots, M$ **do**
5: $\quad\quad x \leftarrow x - \alpha \partial f_\mu(x)$.
6: $\quad$ **end for**
7: **end for**
8: **Output:** x.

3.2 Gradient method for the smoothed primal problem

For a compact convex subset K if a finite dimensional Hilbert space X and consider the σ_K is the support of K, defined by $\sigma_K(z) := \sup_{y\in K}\langle z, y\rangle, \forall z \in X$. Then a class of smoothing approximations defined by $\sigma_\lambda(z) := \sup_{y\in K}\langle z, y\rangle - \frac{1}{2}\lambda\|y\|_F^2$. Using Danskin's Theorem, one can show that σ_λ is smooth with $\frac{1}{\lambda}$-Lipschitz gradient given by $\sigma'_\lambda(x) = \mathrm{Proj}_K(\frac{x}{\lambda})$.

We take $X = \mathbb{R}^l$, and $K := \{z \in X | \|z\|_2 \le 1\}$. Then we could separate the problem (1.1) into separate rows:

$$\frac{1}{2}\|\sum_{i=1}^n A(:,i)x(i,:) - b\|_F^2 + \mu\sum_{i=1}^n \|x(i,:)\|_2 \tag{3.2}$$

Then the smoothed gradient is, for the i-th row

$$\nabla f_\lambda(x)(i,:) = A^\top(Ax-b)(i,:) + \mu \mathrm{Proj}_{\|z\|\le 1}(\frac{x(i,:)}{\lambda}) \tag{3.3}$$

In $(k+1)$-th iteration, if $k = 0$, we use the initial step size α. Otherwise, we use the BB step size α_k:

$$\alpha_k = \frac{(x_k - x_{k-1})^\top(x_k - x_{k-1})}{(x_k - x_{k-1})^\top(\nabla f_{i,j}(x_k) - \nabla f_{i,j}(x_{k-1}))} \tag{3.4}$$

where x and gradient is vectorized. Then we can update x_{k+1} by $x_{k+1} = x_k - \alpha_k \nabla f_{i,j}(x_k)$.

Similarly,we use the continuation strategy. We have three parameters γ, M_1, M_2 for continuation, and set $\mu_0 = \mu_{\max} = \max\{\gamma\|A^\top b\|_\infty, \mu\}$. While $\mu_i > \mu$ or $\lambda_j > \lambda$, we update μ_{i+1}, λ_{i+1} by

$$\mu_{i+1} = \max\{\mu, \gamma\min\{\|\nabla g(x_k)\|_\infty, \mu_i\}\}, \quad \lambda_{j+1} = \max\{\beta\lambda_j, \lambda\} \tag{3.5}$$

Algorithm 2 Gradient method for smoothed primal problem with continuation strategy

Input: initial value x_0, step size α, continuation parameter γ, M_1, M_2, λ decay parameter β.
$\mu_0 = \mu_{\max} = \max\{\gamma\|A^\top b\|_\infty, \mu\}, \alpha_0 = \alpha, k = 0.$
while $\mu_i > \mu$ or $\lambda_j > \lambda$ **do**
  for $l = 1, 2, \cdots, M_1$ **do**
    Update x_{k+1} by BB stepsize.
  end for
  $\mu_{i+1} = \max\{\mu, \gamma \min\{\|\nabla g(x_k)\|_\infty, \mu_i\}\}, \lambda_{j+1} = \max\{\beta\lambda_j, \lambda\}, i = i+1, j = j+1.$
  Set $x_0 := x_k$ and $k = 0$. Update $\alpha_k = \min\{\alpha, \lambda_j\}$.
end while
for $l = 1, 2, \cdots, M_2$ **do**
  Update x_{k+1}, by BB stepsize.
end for

3.3 Fast (Nesterov/accelerated) gradient method for the smoothed primal problem

We still apply the continuation strategy with only a slight modification of the Algorithm 2. Specifically, we set $x_{-1} = x_0$. In $(k+1)$-th iteration, we update x_{k+1} by

$$\begin{cases} y & = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}) \\ x_{k+1} & = y - \alpha_k \nabla f_\lambda(x_k) \end{cases} \tag{3.6}$$

Algorithm 3 Fast gradient method for smoothed primal problem with continuation strategy

Input: initial value x_0, step size α, continuation parameter γ, M_1, M_2, λ decay parameter β.
$\mu_0 = \mu_{\max} = \max\{\gamma\|A^\top b\|_\infty, \mu\}, \alpha_0 = \alpha, k = 0.$
while $\mu_i > \mu$ or $\lambda_j > \lambda$ **do**
  for $l = 1, 2, \cdots, M_1$ **do**
    Update x_{k+1} by (3.6), $\alpha_{k+1} = \alpha_k, k = k+1$.
  end for
  $\mu_{i+1} = \max\{\mu, \gamma \min\{\|\nabla g(x_k)\|_\infty, \mu_i\}\}, \lambda_{j+1} = \max\{\beta\lambda_j, \lambda\}, i = i+1, j = j+1.$
  Set $x_{=1} = x_0 := x_k$ and $k = 0$. Update $\alpha_k = \min\{\alpha, \lambda_j\}$.
end while
for $l = 1, 2, \cdots, M_2$ **do**
  Update x_{k+1} by (3.6), $\alpha_{k+1} = \alpha_k, k = k+1$.
end for

3.4 Proximal gradient method for the primal problem

Define the proximal operator $\mathrm{prox}_{\mu h}(x) = \arg\min_z \frac{1}{2}\|z - x\|_F^2 + \mu h(z)$. When $h(x) = \|x\|_{1,2}$, the proximal operator can be computed explicitly as

$$z(i,:) - x(i,:) + \mu\partial\|z(i,:)\|_2 = 0 \tag{3.7}$$

Hence, we have

$$z(i,:) = \begin{cases} 0 & \textbf{if } \|x(i,:)\|_2 \le \mu \\ \frac{x(i,:)}{\|x(i,:)\|_2}(\|x(i,:)\|_2 - \mu) & \textbf{if } \|x(i,:)\|_2 > \mu \end{cases} \tag{3.8}$$

We use this definition of proximal operator in the following parts.

Define $f_i = g + \mu_i h$, in $(k+1)$-th iteration, we use the BB step size

$$\alpha_k = \frac{(x_k - x_{k-1})^\top (x_k - x_{k-1})}{(x_k - x_{k-1})^\top (\nabla g(x_k) - \nabla g(x_{k-1}))} \tag{3.9}$$

Then, we update x_{k+1} by

$$x_{k+1} = \mathrm{prox}_{\alpha_k \mu_i h}(x_k - \alpha_k \nabla g(x_k)) \tag{3.10}$$

Algorithm 4 Proximal gradient method with continuation strategy

Input: initial value x_0, step size α, continuation parameter $\gamma, \varepsilon_1, \varepsilon_2$.
$\mu_0 = \mu_{\max} = \max\{\gamma\|A^\top b\|_\infty, \mu\}, \alpha_0 = \alpha, i = k = 0.$
Update x_{k+1} by (3.10), $k = k + 1$.
while $\mu_i > \mu$ **do**
  for $k = 1, 2, \cdots, M_1$ **do**
    Calculate BB step size s_k by (3.9), update x_{k+1} by (3.10).
  end for
  $\mu_{i+1} = \max\{\mu, \gamma \min\{\|\nabla g(x_k)\|_\infty, \mu_i\}\}, i = i + 1.$
  Set $\alpha_k = \alpha$, update x_{k+1} by (3.10), $k = k + 1$.
end while
for $k = 1, 2, \cdots, M_2$ **do**
  Calculate BB step size s_k by (3.9), update x_{k+1} by (3.10).
end for

3.5 Fast proximal gradient method for the primal problem

In this part, we update x_{k+1} by

$$\begin{cases} y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1}) \\ x_{k+1} &= \text{prox}_{\alpha_k \mu_i h}(y_k - \alpha_k \nabla g(y_k)) \end{cases} \tag{3.11}$$

Algorithm 5 Fast proximal gradient method with continuation strategy

Input: initial value x_0, step size α, continuation parameter $\gamma, \varepsilon_1, \varepsilon_2$.
$\mu_0 = \mu_{\max} = \max\{\gamma\|A^\top b\|_\infty, \mu\}, \alpha_0 = \alpha, i = k = 0.$
Update x_{k+1} by (3.11), $k = k + 1$.
while $\mu_i > \mu$ **do**
  for $k = 1, 2, \cdots, M_1$ **do**
    Calculate BB step size s_k by (3.9), update x_{k+1} by (3.11).
  end for
  $\mu_{i+1} = \max\{\mu, \gamma \min\{\|\nabla g(x_k)\|_\infty, \mu_i\}\}, i = i + 1.$
  Set $\alpha_k = \alpha$, update x_{k+1} by (3.11), $k = k + 1$.
end while
for $k = 1, 2, \cdots, M_2$ **do**
  Calculate BB step size s_k by (3.9), update x_{k+1} by (3.11).
end for

3.6 Augmented Lagrangian method for the dual problem

The original problem (1.1) is equivalent to the following problem:

$$\min_x \quad \frac{1}{2}\|y\|_F^2 + \mu\|x\|_{1,2} \quad \text{s.t.} \quad Ax - b = y \tag{3.12}$$

The corresponding Lagrangian is

$$L(x, y, z) = \frac{1}{2}\|y\|_F^2 + \mu\|x\|_{1,2} + \langle z, Ax - b - y\rangle \tag{3.13}$$

where $z \in \mathbb{R}^m$, $\langle x, y\rangle := \text{tr}(x^\top y)$. By minimizing L, we have

$$\begin{aligned} \min_{x,y} L(x, y, z) &= -\langle z, b\rangle + \min_y(\frac{1}{2}\|y\|_F^2 - \langle z, y\rangle) + \min_x(\mu h(x) + \langle A^\top z, x\rangle) \\ &= -\langle z, b\rangle - g_0^\star(z) + \mu h^\star(-A^\top z/\mu) \end{aligned} \tag{3.14}$$

where the $g_0^\star$ and $h^\star$ are the conjugate of the function $g_0 = \frac{1}{2}\|\cdot\|_F^2$ and h, which can be directly computed by $g_0^\star(z) = \frac{1}{2}\|z\|_F^2$, $h^\star(z) = \begin{cases} 0, & \|z\|_{\infty,2} \le 1 \\ -\infty, & \|z\|_{\infty,2} > 1 \end{cases}$, where $\|z\|_{\infty,2} := \max_i \|z(i,:)\|_2$.

Hence the dual problem for problem (1.1) is

$$\min \frac{1}{2}\|z\|_F^2 + \langle z, b\rangle, \quad \text{s.t.} \quad A^\top z = w, \quad \|w\|_{\infty,2} \le \mu. \tag{3.15}$$

whose augmented Lagrangian is

$$L_a(z, w, \lambda) = \frac{1}{2}\|z\|_F^2 + \langle z, b\rangle + \langle \lambda, A^\top z - w\rangle + \frac{a}{2}\|A^\top z - w\|_F^2. \tag{3.16}$$

If we set $z^0 = 0, w^0, \lambda^0 = 0$. Given (z^k, w^k, λ^k), the relationship between w^{k+1} and z^{k+1}

$$w^{k+1} = \lambda^k/a + A^\top z^{k+1} - \text{prox}_{\mu h}(\lambda^k/a + A^\top z^{k+1}). \tag{3.17}$$

Then, we have the following problem:

$$\arg\min_z \frac{1}{2}\|z\|_F^2 + b^\top z + \frac{a}{2}\|\text{prox}_{\mu h}(\lambda^k/t + A^\top z)\|_F^2 \tag{3.18}$$

We consider to use the Newton's method to solve the minimization (3.18). We define $z^{k,0} = z^k$, the update can be written as

$$\begin{aligned} z^{k,j+1} &= z^{k,j} - H(z^{k,j})^{-1} d(z^{k,j}) \\ &= z^{k,j} - (I + a \sum_{\|v^{k,j}(i,:)\|_2 > \mu} A_i A_i^\top)^{-1}(z^{k,j} + b + a \sum_{\|v^{k,j}(i,:)\|_2 > \mu} A_i \text{prox}_{\mu h}(v^{k,j})_i) \end{aligned} \tag{3.19}$$

where $v^{k,j} = \lambda^k/a + A^\top z^{k,j}$. We perform the update until $\|d(z^{k,j})\|_2/\|d(z^{k,0})\|_2 \le \epsilon_3$, assuming we terminate the iteration at the M_3-th step.

Since the computational cost of solving $H(z^{k,j})^{-1}d(z^{k,j})$ is large when $H(z^{k,j})$ varies, we approximate $H(z^{k,j}) \approx I + aAA^\top = LDL^\top$ in advance. Empirically, we find approximate $d(z^{k,j}) \approx z^{k,j} + b + aA\text{prox}_{\mu h}(v^{k,j})$ does not impair the performance and improve the efficiency.

In all, we can update $(z^{k+1}, w^{k+1}, \lambda^{k+1})$:

$$\begin{cases} z^{k+1} = z^{k,M_3}. \\ w^{k+1} = \lambda^k/a + A^\top z^{k+1} - \text{prox}_{\mu h}(\lambda^k/a + A^\top z^{k+1}) \\ \lambda^{k+1} = \lambda^k + a(A^\top z^{k+1} - w^{k+1}) \end{cases} \tag{3.20}$$

3.7 Alternating direction method of multipliers for the dual problem

Similarity we obtain the augmented Lagrangian (3.16), while we minimize this Lagrangian with alternating direction strategy. First we minimize $L_a(z^k, w, \lambda^k)$ w.r.t. w, we have $W^{k+1} = \lambda^k/a + A^\top z^k - \text{prox}_{\mu h}(\lambda^k/a + A^\top z^k)$. Then we minimize $L_a(w^{k+1}, z, \lambda^k)$ w.r.t. z. Therefore we can update $(z^{k+1}, w^{k+1}, \lambda^{k+1})$:

$$\begin{cases} w^{k+1} = \lambda^k/a + A^\top z^k - \text{prox}_{\mu h}(\lambda^k/a + A^\top z^k) \\ z^{k+1} = (I + aAA^\top)^{-1}(-b - A\lambda^k + aAw^{k+1}) \\ \lambda^{k+1} = \lambda^k + a(A^\top z^{k+1} - w^{k+1}) \end{cases} \tag{3.21}$$

3.8 Alternating direction method of multipliers with linearization for the primal problem

The primal problem can be reformulated as

$$\min \frac{1}{2}\|Ax - b\|_F^2 + \mu\|y\|_{1,2} \quad \text{s.t.} \quad x = y \tag{3.22}$$

Algorithm 6 ADMM for the dual problem with continuation strategy

Input: Augmented Lagragian parameter a, continuation parameter γ, M_1, M_2. Calculate $\mu_0 = \max\{\gamma\|A^\top b\|_\infty, \mu\}$. Initialize variables $i = k = 0, z^0 = 0, \lambda^0 = 0$.
while $\mu_i > \mu$ **do**
    for $k = 1, 2, \cdots, M_1$ **do**
        Update $(z^{k+1}, w^{k+1}, \lambda^{k+1})$ by (3.21).
    end for
    $\mu_{i+1} = \max\{\mu, \gamma\mu_i\}, i = i + 1, z^0 = z^k, \lambda^0 = \lambda^k, k = 0.$
end while
for $k = 1, 2, \cdots, M_2$ **do**
    Update $(z^{k+1}, w^{k+1}, \lambda^{k+1})$ by (3.21).
end for
$x = -\lambda^k$.

The augmented Lagrangian is

$$L_a^p(x, y, z) = \frac{1}{2}\|Ax - b\|_F^2 + \mu\|y\|_{1,2} + \langle z, x - y\rangle + \frac{a}{2}\|x - y\|_F^2. \tag{3.23}$$

We first update x^{k+1} by direct minimization $x^{k+1} = \arg\min_x L_a(x^k, y^k, z^k) = (A^\top A + aI)^{-1}(A^\top b + ay^k - z^k)$; then we update $y^{k+1} = \arg\min_y L_a(x^{k+1}, y, z^k) = \mathrm{prox}_{\frac{\mu h}{a}}(x^{k+1} + \frac{z^k}{t})$. The update can be summarized as

$$\begin{cases} x^{k+1} = (A^\top A + aI)^{-1}(A^\top b + ay^k - z^k) \\ y^{k+1} = \mathrm{prox}_{\frac{\mu h}{a}}(x^{k+1} + \frac{z^k}{a}) \\ z^{k+1} = z^k + a(x^{k+1} - y^{k+1}) \end{cases} \tag{3.24}$$

Algorithm 7 ADMM with linearization for the primal problem with continuation strategy

Input: Augmented Lagragian parameter a, continuation parameter $\gamma, \varepsilon_1, \varepsilon_2$. Calculate $\mu_0 = \max\{\gamma\|A^\top b\|_\infty, \mu\}$. Initialize variables $i = k = 0, x^0 = y^0 = x_0, z^0 = 0$.
while $\mu_i > \mu$ **do**
    for $k = 1, 2, \cdots, M_1$ **do**
        Update $(x^{k+1}, y^{k+1}, z^{k+1})$ by (3.24), $k = k + 1$.
    end for
    $\mu_{i+1} = \max\{\mu, \gamma\mu_i\}, i = i + 1, x^0 = z^k, y^0 = y^k, z^0 = z^k, k = 0.$
end while
for $k = 1, 2, \cdots, M_2$ **do**
    Update $(x^{k+1}, y^{k+1}, z^{k+1})$ by (3.24), $k = k + 1$.
end for
$x = x^k$.

4 Numerical results

Clearly, the gurobi is the most efficient commericial solver. All the solvers performs similarly in terms of the error to the true solution, and our solvers match the best commercial solver: gurobi. However, it is hard for the sub-gradient method to achieve high sparsity, since in the update, there is no truncation term.

The smoothed gradient performs similarly in BB step size and Nesterov acceleration. The proximal operator is more efficient, since it explicitly truncate the negative terms to zero. We find that BB step size requires the minimum number of steps. However, each round of calculation of BB step size is more computationally difficult.

For the last three algorithms, we find it obviously superior. In terms of efficiency and conciseness, the "ADMM_dual" seems to be the best solver.

We plot the exact solution below, the grouping effect can be readily visualized.

Table 1: Solvers.

	time	objval	err2cvx_mosek	err2_real	sparsity	iter
cvx_mosek	1.422165	0.538327	0	4.20e-05	0.114258	0
cvx_gurobi	2.889272	0.538328	5.57e-06	4.67e-05	0.124023	0
mosek	2.900205	0.538327	3.02e-07	4.22e-05	0.115234	0
gurobi	0.781531	0.538327	2.97e-07	4.23e-05	0.114258	0
subgrad	0.648926	0.538336	6.69e-06	4.62e-05	0.231445	2800
smooth_bb	0.66617	0.538328	2.01e-06	4.36e-05	0.113281	2000
smooth_nesterov	0.668355	0.538329	9.37e-06	3.67e-05	0.126953	2000
prox_bb	0.103632	0.538327	6.42e-08	4.20e-05	0.114258	377
prox_nesterov	0.258887	0.538327	2.02e-06	4.34e-05	0.118164	1177
alm_dual	0.158605	0.538327	6.24e-06	3.84e-05	0.102539	26
admm_dual	0.096789	0.53833	2.67e-06	4.11e-05	0.107422	75
admm_lprimal	0.112085	0.53833	2.71e-06	4.13e-05	0.108398	75

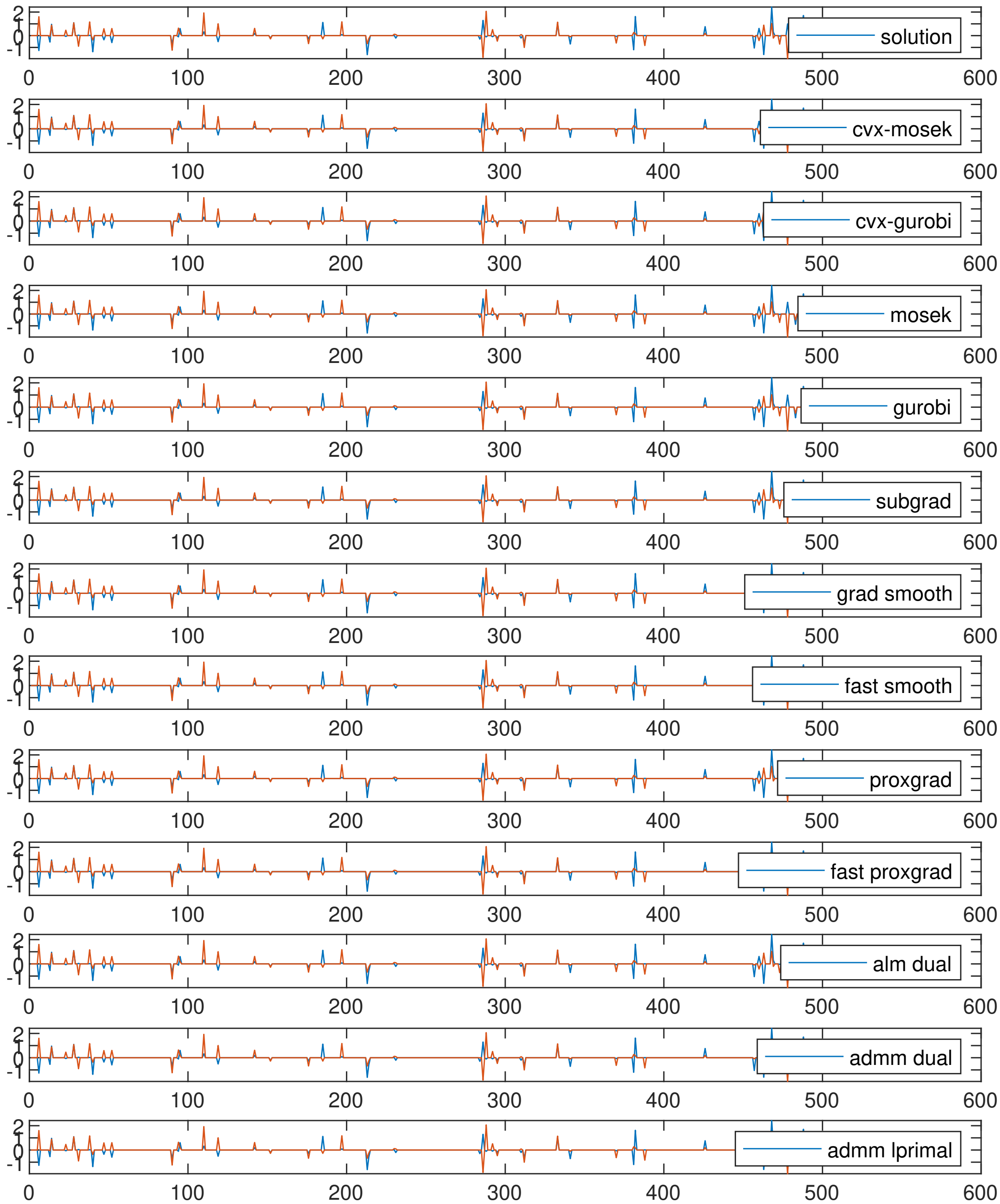

Figure 4.1: Visualization of solutions. $l = 2$.