Algorithm 1 ADMM (x₀, β₀, t) do
1. Input: x₀ ∈ Ω, β₀, t
2. Initialize: x₁ = x₀, y₁ = Ax₀, λ₁ = 0, γ = β₀∥A∥₁ and G = γI − β₀AᵀA or G = 0.
3. for r = 1, 2, . . . do
4. Update x₂ = arg min x∈Ω f(x) + γ∥x − y₁∥₂ (depends on unknown x₁), after t = O(1/t²) iterations, ADMM ensures that F(x₂) − F(x₁) ≤ γ.
5. end for
6. Output: x₂ = x₂ + y₁.

Lemma 1 [1]. By setting β = σmin(G)ζ (ζ depends on unknown x), after t = O(1/ζ²) iterations, ADMM ensures that F(x₂) − F(x₁) ≤ c.

(2) The stochastic ADMM updates x₁, y₁, and λ₁, the same to above (6) and (7), but updates x₂ as

\[ x₂ = \arg \min_{x \in \Omega} f(x) + \gamma∥x − y₁∥_2 \quad \text{where} \quad y₁ = \arg \min_{y \in \mathbb{R}^m} \{ f(x₁) + \frac{1}{2}∥Ax − y∥_2^2 \} \] (9)

where ζ is a stepsize and \( y₁ = \arg \min_{y \in \mathbb{R}^m} \{ f(x₁) + \frac{1}{2}∥Ax − y∥_2^2 \} \).

Algorithm 2 SADM (x₀, η, β₀, t) do
1. Input: x₀ ∈ Ω, η, β₀, t
2. Initialize: x₁ = x₀, y₁ = Ax₀, λ₁ = 0, γ = β₀∥A∥₁ and G = γI − β₀AᵀA ≥ I or G₁ = I.
3. for r = 1, 2, . . . do
4. Update x₂ = arg min x∈Ω f(x) + γ∥x − y₁∥₂ (depends on unknown x₁), after t = O(1/ζ²) iterations, with high probability, SADM ensures that F(x₂) − F(x₁) ≤ c.
5. end for
6. Output: x₂ = x₂ + y₁.

Lemma 2 [2]. By setting \( \beta = \frac{1}{\min(\sigma_{\min}(\Omega_{\ell+1}), \sigma_{\min}(\Omega_{\ell}))} \) (ζ depends on unknown x₁), after t = O(1/ζ²) iterations, with high probability, SADM ensures that F(x₂) − F(x₁) ≤ c.

Local Error Bound and Global Error Inequality

Definition 1. A function F(x) is said to satisfy a local error bound condition on \( \epsilon \)-sublevel set if there exist \( \epsilon \geq 0 \) such that for any \( x \in S \),

\[ d_{\text{sublevel}}(x, S) \leq \epsilon \rightarrow \Omega(\epsilon F(x) - F(x)) \; \text{(10)} \]

Lemma 3 [3]. For any \( x \in \Omega_{\ell+1} \), \( \epsilon > 0 \), we have

\[ x - x_{\ell+1} \leq \frac{1}{\epsilon} \left( d_{\text{sublevel}}(x, S) \right)^{\frac{1}{2}} \] (11)

where \( x_{\ell+1} \in S \) is the closest point in the \( \epsilon \)-sublevel to \( x \).

Locally Adaptive ADMM (LA-ADMM)

Algorithm 3 LA-ADMM (x₀, β₀, A, Ω) do
1. Input: x₀ ∈ Ω, A, initial β₀
2. for k = 1, 2, . . . do
3. Let xₖ = ADMM(xₖ₋₁, βₖ₋₁, tₖ₋₁)
4. Update βₖ = 2/βₖ₋₁, Dₖ = Dₖ₋₁/2.
5. end for

Theorem 1. Assume F(x) obeys the local error bound condition. Let LA-ADMM run with \( t = O\left( \frac{\epsilon_{\min}(\Omega_{\ell}))}{\epsilon} \right) \) for each stage and \( K = \log(\epsilon/\eta) \) iterations for each stage and \( K = \log(\epsilon/\eta) \).

Main Result 2

Theorem 2. Assume F(x) obeys the local error bound condition. Given \( \delta \in (0, 1) \) and \( \delta < K \), let LA-ADMM run with \( t = \frac{\epsilon_{\min}(\Omega_{\ell})}{\epsilon} \) for each stage and \( K = \log(\epsilon/\eta) \) iterations for each stage and \( K = \log(\epsilon/\eta) \).

Algorithm 4 LA-SADM (x₀, η, β₀, t) do
1. Input: x₀ ∈ Ω, η, initial β₀, and D₀.
2. for k = 1, 2, . . . do
3. Let xₖ = SADM(xₖ₋₁, η, βₖ₋₁, tₖ₋₁).
4. Update ηₖ = 2η, βₖ = βₖ₋₁/2, Dₖ = Dₖ₋₁/2.
5. end for

Main Result 1

Theorem 1. Assume F(x) obeys the local error bound condition. Let LA-ADMM run with \( t = O\left( \frac{\epsilon_{\min}(\Omega_{\ell}))}{\epsilon} \right) \) for each stage and \( K = \log(\epsilon/\eta) \) iterations for each stage and \( K = \log(\epsilon/\eta) \).

Remark 1. The number of iteration \( t \) depends on the unknown parameter. This dependence can be relaxed by using another level of restarting and increasing sequence of \( t \). We refer readers to our paper for more details.

Applications and Experiments

1. Generalized LASSO: \( \min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} ∥Ax − b∥_2^2 + \frac{1}{2} ∥x∥_1 \).
2. Fused LASSO penalizes \( ∥x∥_1 \) and \( ∥x∥_2 \) simultaneously.
3. Graph-guided fused LASSO (GGLASSO): A ∈ ℝⁿ×m encodes graph information.
4. Sparse graph-guided fused LASSO (SG-GGLASSO): \( ∥x∥_1 + ∥x∥_2 + |x|_1 \).
5. Piecewise linear loss:
   - hinge loss \( (z - h) \geq 0 \):
   - absolute loss \( (z - h)_+ \):
   - squared hinge loss \( (z - h)_+^2 \):
   - squared absolute loss \( |x|_1^2 \): 2 robust regression with a Low-rank Regularizer: \( F(X) = \frac{1}{2} ∥XAX − C∥₂ \).
7. Low-rank representation: \( F(X) = \frac{1}{2} ∥XAX − C∥₂ \).