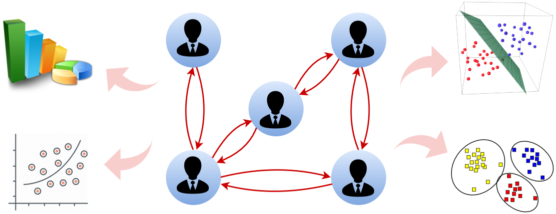
Recycled ADMM: Improve Privacy and Accuracy with Less Computation in Distributed Algorithms

Xueru Zhang, Mohammad Mahdi Khalili, Mingyan Liu

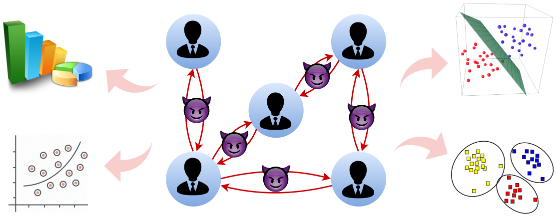
EECS Department, University of Michigan, Ann Arbor

Motivation



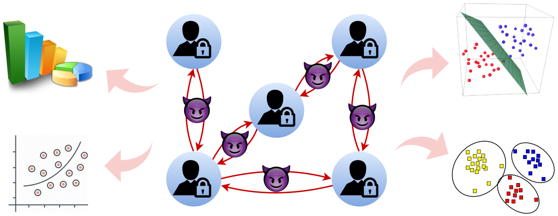
- Need to perform distributed learning tasks.
  - Data may have different owners, locality, etc.
  - Common computational objective.

Motivation



- Need to perform distributed learning tasks.
  - Data may have different owners, locality, etc.
  - Common computational objective.
- Individual entities have privacy concerns over sharing.

Motivation



- Need to perform distributed learning tasks.
  - Data may have different owners, locality, etc.
  - Common computational objective.
- Individual entities have privacy concerns over sharing.

How to accomplish the computational tasks without jeopardizing privacy?

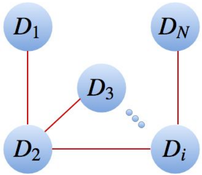
Problem Formulation

- Regularized Empirical Risk Minimization

$$\min_{f_c} O_{ERM}(f_c, \{D_i\}_{i=1}^N) = \sum_{i=1}^{N} O(f_c, D_i)$$

where

$$O(f_c, D_i) = \frac{C}{B_i} \sum_{n=1}^{B_i} \mathscr{L}(y_i^n f_c^T x_i^n) + \frac{\rho}{N} R(f_c)$$



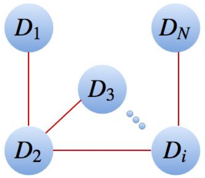
Problem Formulation

- Regularized Empirical Risk Minimization

$$\min_{f_c} O_{ERM}(f_c, \{D_i\}_{i=1}^N) = \sum_{i=1}^{N} O(f_c, D_i)$$

where

$$O(f_c, D_i) = \frac{C}{B_i} \sum_{n=1}^{B_i} \mathscr{L}(y_i^n f_c^T x_i^n) + \frac{\rho}{N} R(f_c)$$



- Distributed optimization
  - (Sub)gradient based
  - Alternating Direction Method of Multipliers (ADMM) based

Conventional ADMM

- Introduce local variables and auxiliary variables to decentralize:

$$\min_{\{f_i\},\{w_{ij}\}} \quad \tilde{O}_{ERM}(\{f_i\}_{i=1}^N, D_{all}) = \sum_{i=1}^{N} O(f_i, D_i)$$

$$\text{s.t.} \quad f_i = w_{ij}, \; w_{ij} = f_j, \quad i \in \mathscr{N}, j \in \mathscr{V}_i$$

Conventional ADMM

- Introduce local variables and auxiliary variables to decentralize:

$$\min_{\{f_i\},\{w_{ij}\}} \quad \tilde{O}_{ERM}(\{f_i\}_{i=1}^N, D_{all}) = \sum_{i=1}^{N} O(f_i, D_i)$$
$$\text{s.t.} \quad f_i = w_{ij}, \;\; w_{ij} = f_j, \quad i \in \mathscr{N}, j \in \mathscr{V}_i$$

- Dual variables: $\lambda_{ij}^a \sim (f_i = w_{ij})$, $\lambda_{ij}^b \sim (w_{ij} = f_j)$.

Conventional ADMM

- Introduce local variables and auxiliary variables to decentralize:

$$\min_{\{f_i\},\{w_{ij}\}} \quad \tilde{O}_{ERM}(\{f_i\}_{i=1}^N, D_{all}) = \sum_{i=1}^N O(f_i, D_i)$$

$$\text{s.t.} \quad f_i = w_{ij}, \;\; w_{ij} = f_j, \quad i \in \mathscr{N}, j \in \mathscr{V}_i$$

- Dual variables: $\lambda_{ij}^a \sim (f_i = w_{ij})$, $\lambda_{ij}^b \sim (w_{ij} = f_j)$.
- Augmented Lagrangian:

$$\begin{aligned} L_\eta(\{f_i\},\{w_{ij},\lambda_{ij}^k\}) &= \sum_{i=1}^N O(f_i, D_i) + \sum_{i=1}^N \sum_{j \in \mathscr{V}_i} (\lambda_{ij}^a)^T (f_i - w_{ij}) \\ &+ \sum_{i=1}^N \sum_{j \in \mathscr{V}_i} (\lambda_{ij}^b)^T (w_{ij} - f_j) \\ &+ \sum_{i=1}^N \sum_{j \in \mathscr{V}_i} \frac{\eta}{2}(||f_i - w_{ij}||_2^2 + ||w_{ij} - f_j||_2^2) \end{aligned}$$

Conventional ADMM

In the $(t+1)$-th iteration, the ADMM updates consist of the following:

primal updates:

$$f_i(t+1) = \operatorname*{argmin}_{f_i} L_\eta(\{f_i\}, \{w_{ij}(t), \lambda_{ij}^k(t)\}) ;$$

$$w_{ij}(t+1) = \operatorname*{argmin}_{w_{ij}} L_\eta(\{f_i(t+1)\}, \{w_{ij}, \lambda_{ij}^k(t)\}) ;$$

dual updates:

$$\lambda_{ij}^a(t+1) = \lambda_{ij}^a(t) + \eta(f_i(t+1) - w_{ij}(t+1)) ;$$

$$\lambda_{ij}^b(t+1) = \lambda_{ij}^b(t) + \eta(w_{ij}(t+1) - f_j(t+1)) .$$

Simplified ADMM

- Initialize $\lambda_{ij}^{a}(0) = \lambda_{ij}^{b}(0) = 0$

Simplified ADMM

- Initialize $\lambda_{ij}^{a}(0) = \lambda_{ij}^{b}(0) = \mathbf{0}$
- Then $\lambda_{ij}^{a}(t) = \lambda_{ij}^{b}(t)$ and $\lambda_{ij}^{k}(t) = -\lambda_{ji}^{k}(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.

Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$

Simplified ADMM

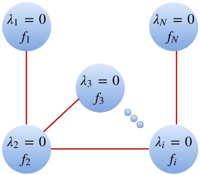
- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

$$
\begin{aligned}
f_i(t+1) &= \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i \\
&\quad + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \}\ ; \\
\lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(t+1) - f_j(t+1))\ .
\end{aligned}
$$

Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

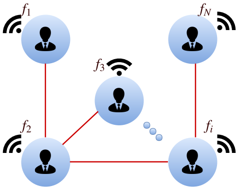
$$
\begin{aligned}
f_i(t+1) &= \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i \\
&\quad + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \ \} \ ; \\
\lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(t+1) - f_j(t+1)) \ .
\end{aligned}
$$



Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

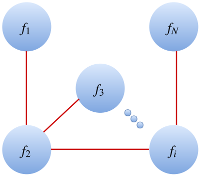
$$
\begin{aligned}
f_i(t+1) &= \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i \\
&\quad + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \} ; \\
\lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(t+1) - f_j(t+1)) .
\end{aligned}
$$



Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

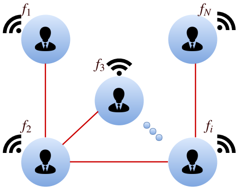
$$
\begin{aligned}
f_i(t+1) &= \underset{f_i}{\text{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i \\
&\quad + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \} ; \\
\lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(t+1) - f_j(t+1)) .
\end{aligned}
$$



Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

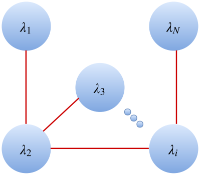
$$\begin{aligned} f_i(t+1) &= \underset{f_i}{\text{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i \\ &\quad + \eta \sum_{j \in \mathscr{V}_i} \|\frac{1}{2}(f_i(t) + f_j(t)) - f_i\|_2^2 \}; \\ \lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(t+1) - f_j(t+1)). \end{aligned}$$



Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathcal{N}, j \in \mathcal{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathcal{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathcal{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

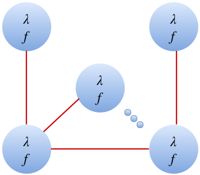
$$
\begin{aligned}
f_i(t+1) &= \underset{f_i}{\operatorname{argmin}} \{ O(f_i, D_i) + 2\lambda_i(t)^T f_i \\
&\quad + \eta \sum_{j \in \mathcal{V}_i} || \frac{1}{2}(f_i(t) + f_j(t)) - f_i ||_2^2 \} ; \\
\lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathcal{V}_i} (f_i(t+1) - f_j(t+1)) .
\end{aligned}
$$



Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a, b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j \in \mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

$$
\begin{aligned}
f_i(t+1) &= \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i \\
&\quad + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \} ; \\
\lambda_i(t+1) &= \lambda_i(t) + \frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(t+1) - f_j(t+1)) .
\end{aligned}
$$

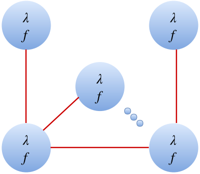


Simplified ADMM

- Initialize $\lambda_{ij}^a(0) = \lambda_{ij}^b(0) = 0$
- Then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ $k \in \{a,b\}, i \in \mathscr{N}, j \in \mathscr{V}_i$.
- Let $\lambda_i(t) = \sum_{j\in\mathscr{V}_i} \lambda_{ij}^a(t) = \sum_{j\in\mathscr{V}_i} \lambda_{ij}^b(t)$
- Then the ADMM updates can be simplified as:

$$f_i(t+1) = \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i + \eta \sum_{j\in\mathscr{V}_i} \|\frac{1}{2}(f_i(t) + f_j(t)) - f_i\|_2^2\};$$

$$\lambda_i(t+1) = \lambda_i(t) + \frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(t+1) - f_j(t+1)).$$

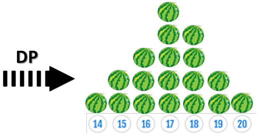


How to make this procedure "private"?

Differential Privacy

- Obtain almost the same conclusion regardless of participation

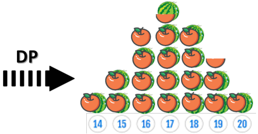
Student ID	Last Name	Initial	Age	Program
ST348-245	Walton	L.	21	Drafting
ST348-246	Wilson	R.	19	Science
ST348-247	Thompson	G.	18	Business
ST348-248	James	L.	23	Nursing
ST348-249	Peterson	M.	37	Science
ST348-250	Graham	J.	20	Arts
ST348-251	Smith	F.	26	Business
ST348-252	Nash	S.	22	Arts



Differential Privacy

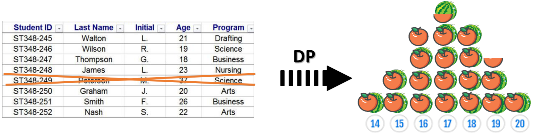
- Obtain almost the same conclusion regardless of participation

Student ID	Last Name	Initial	Age	Program
ST348-245	Walton	L.	21	Drafting
ST348-246	Wilson	R.	19	Science
ST348-247	Thompson	G.	18	Business
ST348-248	James	L.	23	Nursing
~~ST348-249~~	~~Peterson~~	~~M.~~	~~37~~	~~Science~~
ST348-250	Graham	J.	20	Arts
ST348-251	Smith	F.	26	Business
ST348-252	Nash	S.	22	Arts



Differential Privacy

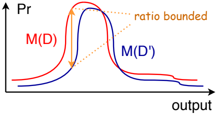
- Obtain almost the same conclusion regardless of participation



Student ID	Last Name	Initial	Age	Program
ST348-245	Walton	L.	21	Drafting
ST348-246	Wilson	R.	19	Science
ST348-247	Thompson	G.	18	Business
ST348-248	James	L.	23	Nursing
ST348-249	Peterson	M.	37	Science
ST348-250	Graham	J.	20	Arts
ST348-251	Smith	F.	26	Business
ST348-252	Nash	S.	22	Arts

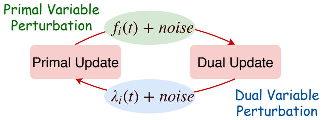
- A randomized algorithm $M(\cdot)$ is ϵ-differentially private if for any neighboring datasets D, D' and for any sets of possible outputs $S \subseteq \text{range}(M)$:

$$\frac{\Pr(M(D) \in S)}{\Pr(M(D') \in S)} \leq \exp(\epsilon)$$



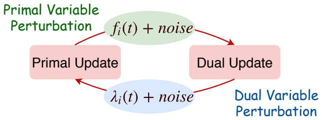
Existing work on differentially private ADMM

T. Zhang, et al. IEEE Trans. Inf. Forensic Secur. (2017)



Existing work on differentially private ADMM

T. Zhang, et al. IEEE Trans. Inf. Forensic Secur. (2017)

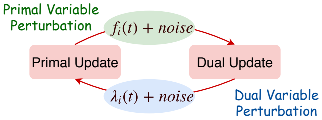


Issues:

- Privacy loss only evaluated for a single node for one iteration.
- Privacy loss accumulates over iterations; hard to balance privacy and utility simply by summing up privacy losses.

Existing work on differentially private ADMM

T. Zhang, et al. IEEE Trans. Inf. Forensic Secur. (2017)



Issues:

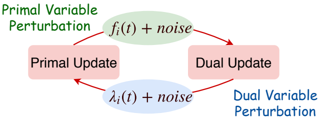
- Privacy loss only evaluated for a single node for one iteration.
- Privacy loss accumulates over iterations; hard to balance privacy and utility simply by summing up privacy losses.

X. Zhang, et al. ICML (2018)

- The total privacy loss of all nodes over the entire iterative process.
- M-ADMM to accommodate the private and varied penalty parameters for each node; increasing which can increase the algorithm's robustness and improve the privacy-utility tradeoff.

Existing work on differentially private ADMM

T. Zhang, et al. IEEE Trans. Inf. Forensic Secur. (2017)



Issues:

- Privacy loss only evaluated for a single node for one iteration.
- Privacy loss accumulates over iterations; hard to balance privacy and utility simply by summing up privacy losses.

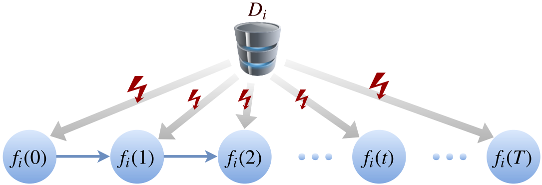
X. Zhang, et al. ICML (2018)

- The total privacy loss of all nodes over the entire iterative process.
- M-ADMM to accommodate the private and varied penalty parameters for each node; increasing which can increase the algorithm's robustness and improve the privacy-utility tradeoff.

Can we improve more?

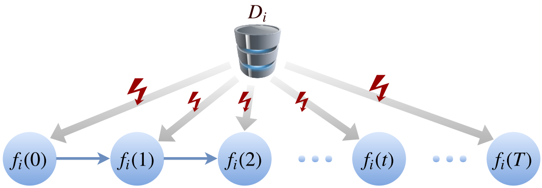
Make information recyclable

Exsiting work: raw data is used in every update

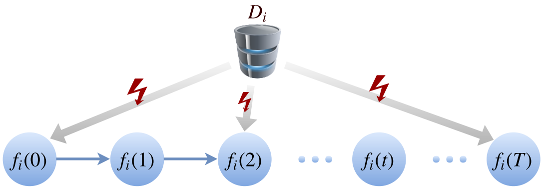


Make information recyclable

Exsiting work: raw data is used in every update



Our idea: make some updates with the existing computation instead of the raw data



Recycled ADMM

- Original primal updates:

$$f_i(t+1) = \underset{f_i}{\text{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \}$$

- Linearized approximation only in $2k$-th (even) updates:

$$O(f_i, D_i) \approx O(f_i(2k-1), D_i) + \nabla O(f_i(2k-1), D_i)^T (f_i - f_i(2k-1)) + \frac{\gamma}{2}||f_i - f_i(2k-1)||_2^2 \quad (\gamma \geq 0)$$

Recycled ADMM

- Original primal updates:

$$f_i(t+1) = \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \}$$

- Linearized approximation only in $2k$-th (even) updates:

$$O(f_i, D_i) \approx O(f_i(2k-1), D_i) + \nabla O(f_i(2k-1), D_i)^T (f_i - f_i(2k-1)) + \frac{\gamma}{2}||f_i - f_i(2k-1)||_2^2 \quad (\gamma \geq 0)$$

- Then the $2k$-th (even) primal updates becomes:

$$f_i(2k) = f_i(2k-1) - \frac{1}{2\eta V_i + \gamma}\{\nabla O(f_i(2k-1), D_i) + 2\lambda_i(2k-1) + \eta \sum_{j \in \mathscr{V}_i} (f_i(2k-1) - f_j(2k-1))\}$$

Recycled ADMM

- Original primal updates:

$$f_i(t+1) = \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(t)^T f_i + \eta \sum_{j \in \mathscr{V}_i} ||\frac{1}{2}(f_i(t) + f_j(t)) - f_i||_2^2 \}$$

- Linearized approximation only in $2k$-th (even) updates:

$$O(f_i, D_i) \approx O(f_i(2k-1), D_i) + \nabla O(f_i(2k-1), D_i)^T (f_i - f_i(2k-1)) + \frac{\gamma}{2}||f_i - f_i(2k-1)||_2^2 \quad (\gamma \geq 0)$$

- Then the $2k$-th (even) primal updates becomes:

$$f_i(2k) = f_i(2k-1) - \frac{1}{2\eta V_i + \gamma}\{\nabla O(f_i(2k-1), D_i) + 2\lambda_i(2k-1) + \eta \sum_{j \in \mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))\}$$

The information is "recycled"!

Recycled ADMM

- Odd updates: conventional ADMM

$$
\begin{aligned}
f_i(2k-1) &= \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(2k-2)^T f_i \\
&\quad + \eta \sum_{j\in\mathscr{V}_i} ||\frac{1}{2}(f_i(2k-2) + f_j(2k-2)) - f_i||_2^2 \} \; ; \\
\lambda_i(2k-1) &= \lambda_i(2k-2) + \frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1)) \; ;
\end{aligned}
$$

- Even updates: a variant of gradient descent

$$
\begin{aligned}
f_i(2k) &= f_i(2k-1) - \frac{1}{2\eta V_i + \gamma}\{\nabla O(f_i(2k-1), D_i) + 2\lambda_i(2k-1) \\
&\quad + \eta \sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))\} \; ; \\
\lambda_i(2k) &= \lambda_i(2k-1) \; .
\end{aligned}
$$

Differentially Private Recycled ADMM

- Odd updates: dual variable perturbation

$$
\begin{aligned}
f_i(2k-1) &= \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + (2\lambda_i(2k-2) + \epsilon_i(2k-1))^T f_i \\
&\quad + \eta \sum_{j\in\mathscr{V}_i} \|\frac{1}{2}(f_i(2k-2) + f_j(2k-2)) - f_i\|_2^2 \} ; \\
\lambda_i(2k-1) &= \lambda_i(2k-2) + \frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1)) ;
\end{aligned}
$$

- Even updates: sum operation over the existing stored information

$$
\begin{aligned}
f_i(2k) &= f_i(2k-1) - \frac{1}{2\eta V_i + \gamma}\{\eta \sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1)) \\
&\quad + 2\lambda_i(2k-1) + \underbrace{\epsilon_i(2k-1) + \nabla O(f_i(2k-1), D_i)}_{\text{the existing computation by KKT}}\} ; \\
\lambda_i(2k) &= \lambda_i(2k-1) .
\end{aligned}
$$

Theoretical Results

Convergence Analysis:

- A sufficient condition for the convergence of Recycled ADMM.

Theoretical Results

Convergence Analysis:

- A sufficient condition for the convergence of Recycled ADMM.

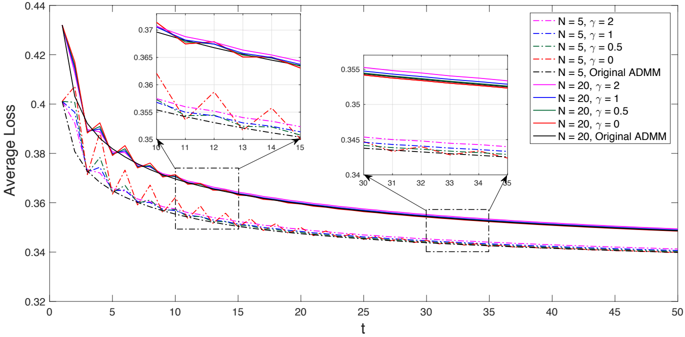
Privacy Analysis:

- The total privacy loss during $2K$ iterations.

$$\ln \frac{\Pr(\{\{f_i(t)\}_{i=1}^{N}\}_{t=0}^{2K} \in S|D_{all})}{\Pr(\{\{f_i(t)\}_{i=1}^{N}\}_{t=0}^{2K} \in S|\hat{D}_{all})} \leq \max_{i\in\mathcal{N}}\{\sum_{k=1}^{K} \frac{2C}{B_i}(\frac{1.4c_1}{(\frac{\rho}{N}+2\eta V_i)}+\alpha_i(k))\}$$

Numerical Results

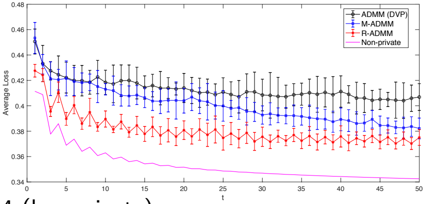
- Convergence of Recycled ADMM (non-private)

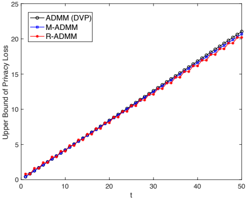


Numerical Results

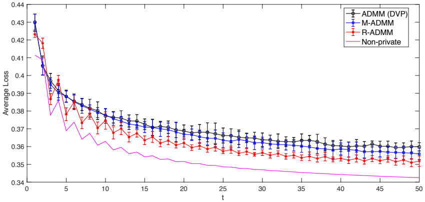
- Accuracy comparison under the same privacy guarantee

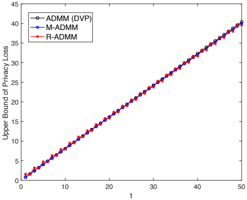
$\alpha_i = 2$ (more private)





$\alpha_i = 4$ (less private)

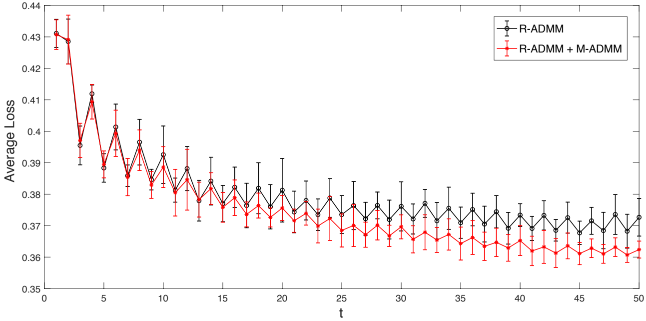




- ADMM (DVP): T. Zhang, et al. IEEE Trans. Inf. Forensic Secur. (2017)
- M-ADMM: X. Zhang, et al. ICML (2018)

Numerical Results

- Incorporate the idea from X. Zhang, et al. ICML (2018): Decrease the step-size (increase η and γ) over iterations to stabilize the algorithm.



Conclusions

- Recycled ADMM: improve the privacy-utility tradeoff significantly with less computation.
- Improvement is more significant with higher privacy requirement.
- Privacy-utility tradeoff can be further improved by controlling the step-size (η, γ).

Conclusions

- Recycled ADMM: improve the privacy-utility tradeoff significantly with less computation.
- Improvement is more significant with higher privacy requirement.
- Privacy-utility tradeoff can be further improved by controlling the step-size (η, γ).

Thank you!