Byzantine-tolerant federated Gaussian process regression for streaming data

Xu Zhang, Zhenyuan Yuan, Minghui Zhu

School of Electrical Engineering and Computer Science
Pennsylvania State University

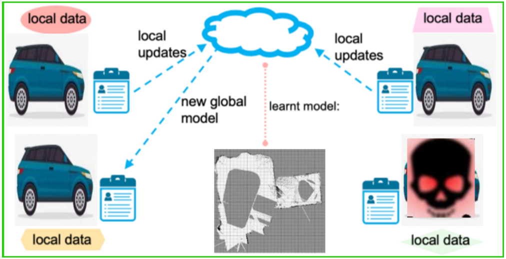
November 28, 2022

2022 Conference on Neural Information Processing Systems
New Orleans, LA

Problem formulation

Network model:

- Cloud can communicate with agents
- Agents cannot communicate with each other
- Byzantine agents send arbitrary model parameters to the cloud



Observation model:

$$y^{[i]}(t) = \eta(\boldsymbol{z}^{[i]}(t)) + e^{[i]}(t)$$

- Training data $(\boldsymbol{z}^{[i]}(t), y^{[i]}(t))$ arrive sequentially

Objective: Design a Byzantine-tolerant algorithm which

- Correctly learns the function η
- Does not require to share local streaming data $(z^{[i]}(t), y^{[i]}(t))$

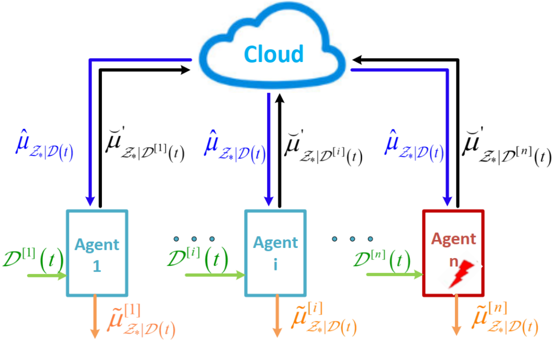
Byzantine-tolerant federated GPR

Contribution

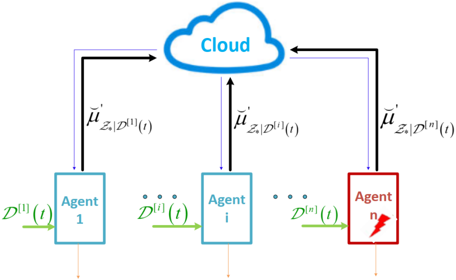
Design a Byzantine-tolerant federated Gaussian process regression (GPR) algorithm which

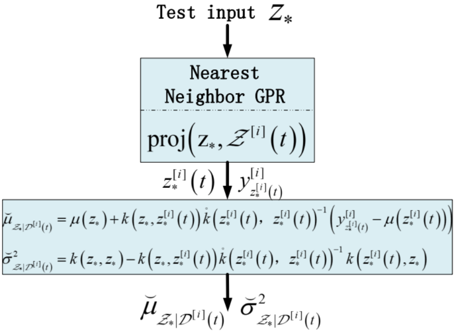
- Can guarantee the correct predictions and tolerate less than one quarter Byzantine agents
- Can deal with streaming data and perform on-line learning

- Agent-based local GPR
- Cloud-based aggregated GPR
- Agent-based fused GPR



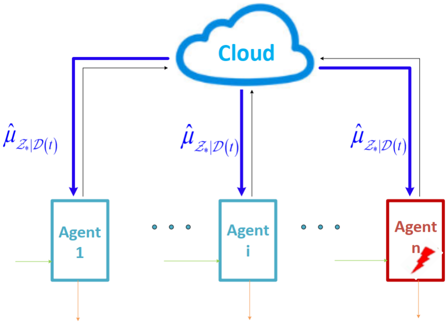
Agent-based local GPR

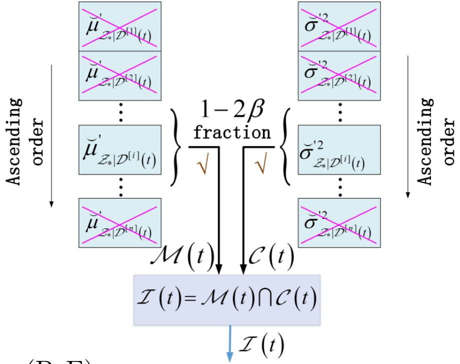




- $\breve{\mu}'_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \leftarrow \begin{cases} \breve{\mu}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} & \text{Benign agent,} \\ \star & \text{Byzantine agent} \end{cases}$
- $\breve{\sigma}'^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \leftarrow \begin{cases} \breve{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} & \text{Benign agent,} \\ \star & \text{Byzantine agent} \end{cases}$

Cloud-based aggregated GPR



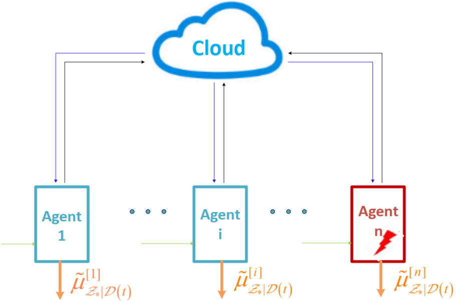


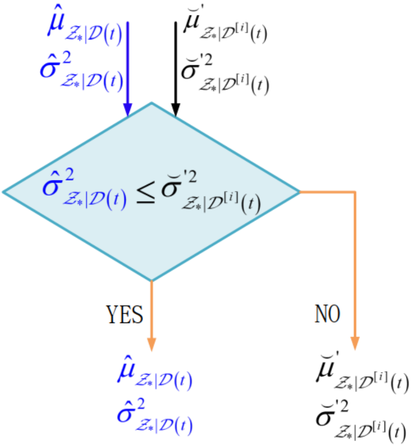
- Byzantine-tolerant product of experts (PoE):

$$\hat{\mu}_{\boldsymbol{z}_*|\mathcal{D}(t)} = \frac{\hat{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}(t)}}{|\mathcal{I}(t)|} \sum_{i \in \mathcal{I}(t)} \breve{\mu}'_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)} \breve{\sigma}'^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)},$$

$$\hat{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}(t)} = \frac{|\mathcal{I}(t)|}{\sum_{i \in \mathcal{I}(t)} \breve{\sigma}'^{-2}_{\boldsymbol{z}_*|\mathcal{D}^{[i]}(t)}}.$$

Agent-based fused GPR





- Output: $\tilde{\mu}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)}$, $(\tilde{\sigma}^{[i]}_{\boldsymbol{z}_*|\mathcal{D}(t)})^2$

Robustness of cloud-based aggregated GPR

Assumption

Less than one quarter of the agents are Byzantine.

Dispersion: $d^{[i]}(t) \triangleq \sup_{\boldsymbol{z}\in\boldsymbol{\mathcal{Z}}} D(\boldsymbol{z}, \mathcal{Z}^{[i]}(t))$

Theorem (Cloud-based aggregated GPR: Mean)

For any $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$ *and* $0 < \delta < 1$, *with probability at least* $1-\delta$, *it holds that*
$\left|\hat{\mu}_{\boldsymbol{z}_*|\mathcal{D}(t)} - \eta(\boldsymbol{z}_*)\right| \leq$
$(1 - \frac{\kappa(d^{\max}(t))}{\sigma_f^2+(\sigma_e^{\max})^2})\|\eta\|_\infty + \frac{\sigma_f^2\ell_\eta d^{\max}(t)}{\sigma_f^2+(\sigma_e^{\min})^2} + \sqrt{2\sigma^2(\ln 2 - \ln\delta)} + \Delta(d^{\max}(t))$ *where*

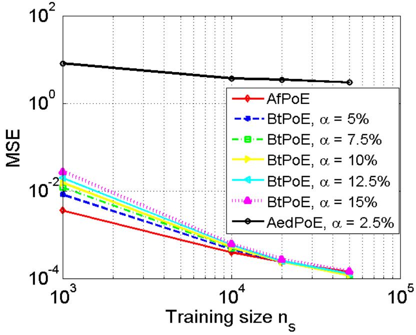
$$\Delta(s) \triangleq \frac{2\alpha(\sqrt{2\sigma^2(\ln(2n)-\ln\delta)}+\frac{\sigma_f^2\|\eta\|_\infty}{\sigma_f^2+(\sigma_e^{\min})^2})}{1-4\beta}\frac{\sigma_f^4+\sigma_f^2(\sigma_e^{\max})^2-\kappa(s)^2}{\sigma_f^2(\sigma_e^{\min})^2}.$$

Theorem (Cloud-based aggregated GPR: Variance)

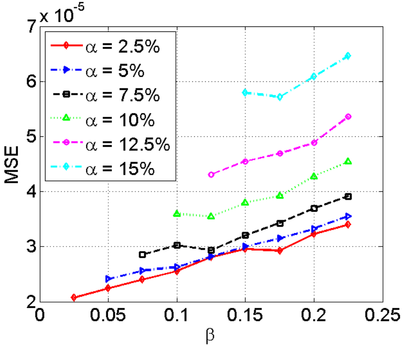
For any $\boldsymbol{z}_* \in \boldsymbol{\mathcal{Z}}$, *it holds that* $\frac{\sigma_f^2(\sigma_e^{\min})^2}{\sigma_f^2+(\sigma_e^{\max})^2} \leq \hat{\sigma}^2_{\boldsymbol{z}_*|\mathcal{D}(t)} \leq \sigma_f^2 - \frac{\kappa(d^{\max}(t))^2}{\sigma_f^2+(\sigma_e^{\max})^2}$.

Experiments (Synthetic dataset)

Experiment 1: Prediction performance in terms of consistency and different α, β



(a) Consistency evaluation



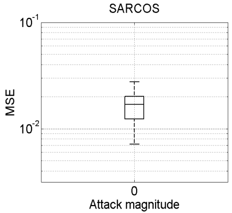
(b) Prediction performance on different β

Experiment 2: Prediction performance on different functions

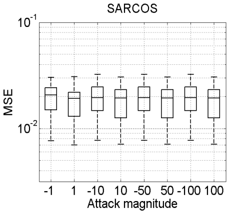
Algorithm	AfPoE	BtPoE	AedPoE
MSE ($\times 10^{-3}$)	0.0049 ± 0.007	0.0236 ± 0.172	26.5339 ± 0.019

Experiments (Real-world datasets)

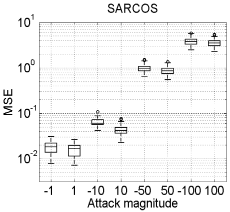
Experiment 3: Performance on different attack magnitudes



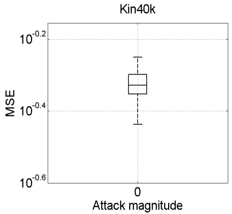
(c) Attack-free standard PoE



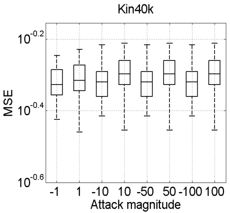
(d) Byzantine-tolerant PoE



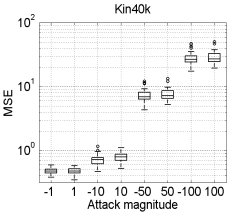
(e) Attacked standard PoE



(f) Attack-free standard PoE



(g) Byzantine-tolerant PoE



(h) Attacked standard PoE

Conclusion

- Design a Byzantine-tolerant federated GPR algorithm
- Derive the upper bounds on the prediction errors and the lower and upper bounds of the predictive variances
- Demonstrate the robustness of Byzantine-tolerant GPR algorithm through experiments

Thank you