Analysis of Total Variation Minimization for Clustered Federated Learning

Alexander Jung

Aalto University, Finland

LinkedIn: https://www.linkedin.com/in/aljung/
YouTube: alexjung111

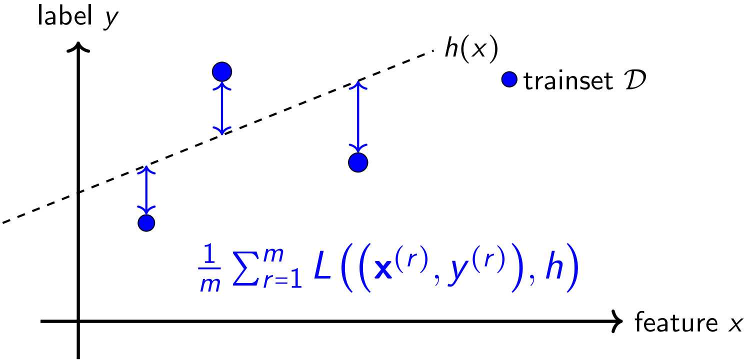
What is it all About?

How can we identify - in a distributed and privacy-preserving fashion - useful chunks of data that can be pooled together to train a big personalized machine learning model ?

Table of Contents

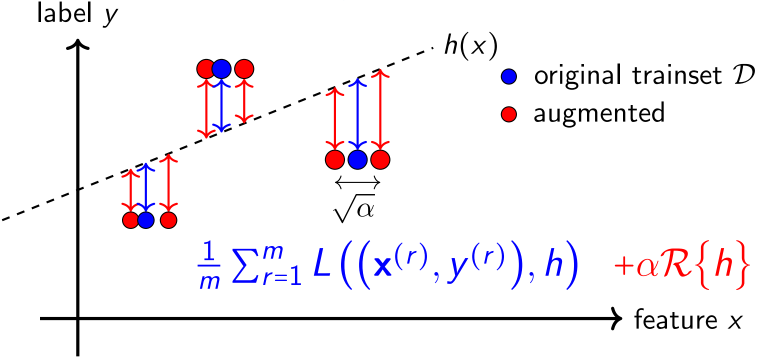
Table of Contents

Machine Learning



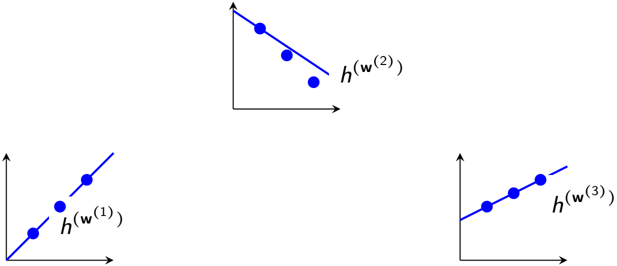
- find $\hat{h}$ with smallest risk $\mathbb{E}L\left((\mathbf{x}, y), h\right)$, $(\mathbf{x}, y) \sim p\Big(\,(\mathbf{x}, y)\,\Big)$
- approximate risk by average loss $\frac{1}{m}\sum_{r=1}^{m} L\Big(\big(\mathbf{x}^{(r)}, y^{(r)}\big), h\Big)$
- works only if $\mathrm{effective\,dim}\big(\mathcal{H}\big) < m$

Applied Machine Learning



- what if effective $\dim(\mathcal{H}) \geq m$?
- either increase m by augmentation,
- or decrease $\dim(\mathcal{H})$ by model pruning
- e.g., via adding penalty to loss function (Lagrangian duality)

Federated Learning (FL)

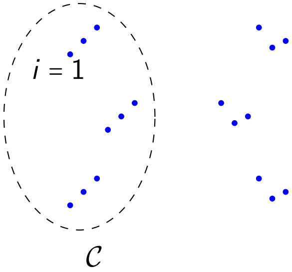


- FL ≈ ML with distributed data and computers
- collection of data generators $i = 1, \ldots, n$
- each i generates local dataset $\mathcal{D}^{(i)}$
- each i learns local model params. $\mathbf{w}^{(i)}$ (**personalization**)
- quality of $\mathbf{w}^{(i)}$ measured by local loss $L_i\left(\mathbf{w}^{(i)}\right)$

Opportunities and Challenges in FL

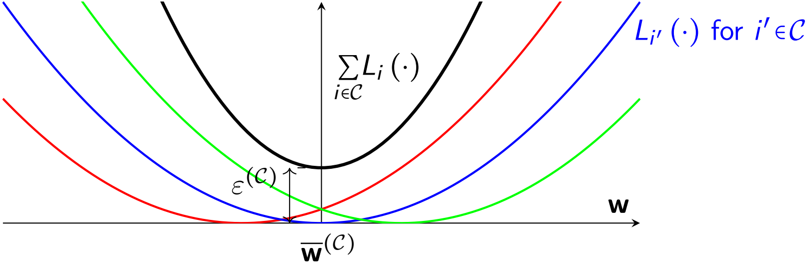
- can leverage information contained in other's data ☺
- can train tailored (personalized) model for individual ☺
- can leverage compute of other's devices ☺
- need to coordinate distributed computation ☹
- need to protect sensitive data ☹
- need to find out if other's data is useful ☹

Clustered FL



- local dataset i drawn i.i.d. from prob. dist. $p^{(i)}\big(\mathbf{x}, y\big)$
- cluster $\mathcal{C}$: a subset of i's with similar $p^{(i)}$
- we make this precise by a clustering assumption
- formulated in terms of the local loss functions

Clustering Assumption



Assumption

The data generators contain a cluster $\mathcal{C} \subseteq \{1, \ldots, n\}$ such that there is a common choice $\overline{\mathbf{w}}^{(\mathcal{C})}$ for the local model parameters for all $i \in \mathcal{C}$ satisfying

$$\sum_{i \in \mathcal{C}} L_i\left(\overline{\mathbf{w}}^{(\mathcal{C})}\right) \leq \varepsilon^{(\mathcal{C})}.$$

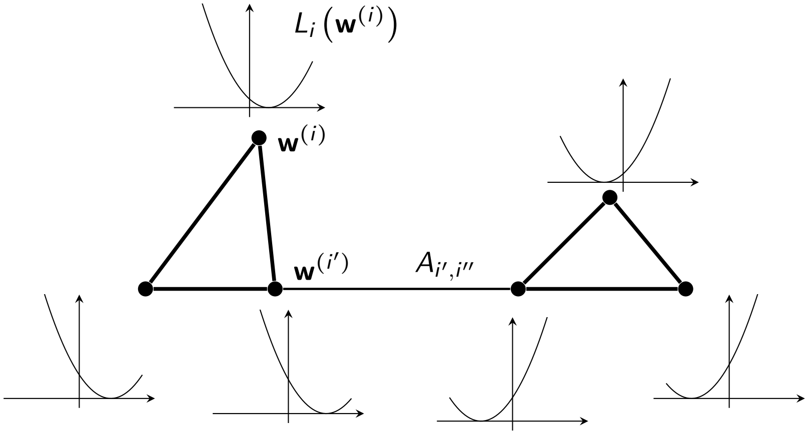
Note: Assumption parametrized by $\mathcal{C}, \varepsilon^{(\mathcal{C})}, \overline{\mathbf{w}}^{(\mathcal{C})}$

Table of Contents

The Empirical Graph

- consider data generators $p^{(i)}$ for $i = 1, \ldots, n$
- represent them as nodes $\mathcal{V} = \{1, \ldots, n\}$ of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- edges $\mathcal{E}$ represent **pair-wise similarities** between $p^{(i)}$
- edge weights $A_{i,i'} \geq 0$
- $A_{i,i'} = 0$ means no similarity, $\{i, i'\} \notin \mathcal{E}$
- $A_{i,i'} > 0$ indicates amount of similarity between $p^{(i)}, p^{(i')}$

Nodes Carry Local Loss Functions



Empirical Graph is Design Choice

- edge weights $A_{i,i'}$ are design choice for FL methods
- more edges $\Rightarrow$ more computation
- too few edges $\Rightarrow$ insufficient coupling within cluster
- avoid too many edges across clusters

Graph Learning Methods

- use statistical tests[1] for $p^{(i)} \stackrel{?}{=} p^{(i')}$
- choose $A_{i,i'}$ via (est.) KL-divergence[2] $D^{(\mathrm{KL})}\big(p^{(i)}, p^{(i')}\big)$
- compare gradients[3] $\nabla L_i(\mathbf{w}), \nabla L_{i'}(\mathbf{w})$
- compare vector representation (embedding)[4] $\mathbf{z}^{(i)}, \mathbf{z}^{(i')}$

[1] Schrab et.al., MMD Aggregated Two-Sample Test, JMLR, 2023

[2] Y. Bu et.al., "Estimation of KL Divergence: Optimal Minimax Rate," in IEEE Transactions on Information Theory, 2018

[3] Werner et.al.,Provably Personalized and Robust Federated Learning, TMLR, 2023.

[4] Petukhova et.al, Text Clustering with LLM Embeddings, 2024.

Generalized Total Variation Minimization

learn local model parameters $\widehat{\mathbf{w}}^{(i)}$ by balancing their local loss with their variation across edges of the empirical graph, i.e.,

$$\left\{\widehat{\mathbf{w}}^{(i)}\right\}_{i=1}^{n} \in \underset{\mathbf{w}^{(i)}}{\operatorname{argmin}} \underbrace{\sum_{i \in \mathcal{V}} L_i\left(\mathbf{w}^{(i)}\right)}_{\text{average local loss}} + \alpha \underbrace{\sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} \phi(\mathbf{w}^{(i)} - \mathbf{w}^{(i')})}_{\text{variation across edges}}$$

- $\phi(\cdot)$ measures variation of model parameters
- $\phi(\mathbf{u})$ typically increasing with norm $\|\mathbf{u}\|$
- GTVMin parameter α controls preference for small variation
- comp./stat. of GTVMin depend crucially on $\phi(\cdot)$ and α

Special Cases of GTVMin

- graph sig. recovery:[1] $L_i\left(w^{(i)}\right) = \left(y^{(i)} - w^{(i)}\right)^2$, $\phi(\cdot) = (\cdot)^2$
- network Lasso:[2] $\phi(\cdot) = \|\cdot\|_2$
- convex clustering:[3] $L_i\left(\mathbf{w}^{(i)}\right) = \left\|\mathbf{w}^{(i)} - \mathbf{a}^{(i)}\right\|_2^2$, $\phi(\cdot) = \|\cdot\|_2$, fully connected empirical graph $\mathcal{G}$

---

[1] Chen et.al. Signal Recovery on Graphs: Variation Minimization. IEEE Trans. Sig. Proc. vol. 63, no. 17, 2015.

Puy et.al. Random sampling of bandlimited signals on graphs. Appl. Comp. Harm. Anal. 2018.

[2] D. Hallac, J. Leskovec, and S. Boyd, Network Lasso: Clustering and Optimization in Large Graphs,Proceedings SIGKDD, pages 387-396, 2015.

[3] D. Sun and K.-C. Toh and Y. Yuan; Convex Clustering: Model, Theoretical Guarantee and Efficient Algorithm, JMLR, 2021

Total Variation Minimization

GTVMin with $\phi(\mathbf{u}) := \|\mathbf{u}\|_2^2$

$$\left\{\widehat{\mathbf{w}}^{(i)}\right\}_{i=1}^{n} \in \underset{\mathbf{w}^{(i)}}{\operatorname{argmin}} \underbrace{\sum_{i \in \mathcal{V}} L_i\left(\mathbf{w}^{(i)}\right)}_{\text{average local loss}} + \alpha \underbrace{\sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} \left\|\mathbf{w}^{(i)} - \mathbf{w}^{(i')}\right\|_2^2}_{\text{variation across edges}}$$

can be implemented (computed) using

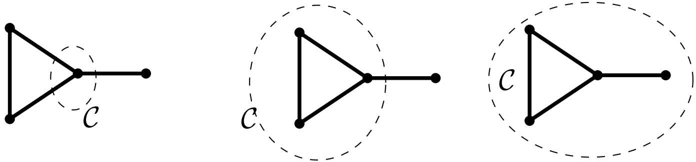
- gradient methods if $L_i\left(\right)$ diff.able [1]
- proximal methods if $L_i\left(\right)$ proximable [2]
- asynchronous distributed computers (smartphones)[3]

[1] J. Liu and C. Zhang, Distributed Learning Systems with First-Order Methods: An Introduction, 2020

[2] N. Parikh and S. Boyd, Proximal Algorithms, 2013

[3] D. Bertsekas and J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, 2015

Choose FL Flavour via Regularization Parameter



small α
personalized FL

moderate α
clustered FL

large α
global model FL

GTVMin solutions become increasingly clustered for increasing α

Table of Contents

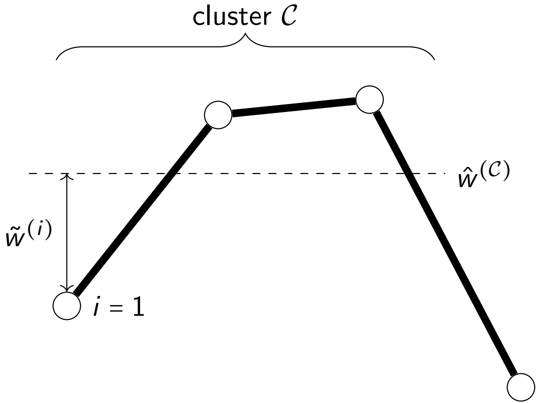
Error Analysis

- consider emp. graph $\mathcal{G}$ containing cluster $\mathcal{C}$
- learn model parameter $\widehat{\mathbf{w}}^{(i)}$ via GTVMin
- mainly interested if $\widehat{\mathbf{w}}^{(i)}$ captures cluster $\mathcal{C}$
- define clustering error

$$\widetilde{\mathbf{w}}^{(i)} := \widehat{\mathbf{w}}^{(i)} - \underbrace{(1/|\mathcal{C}|) \sum_{i' \in \mathcal{C}} \widehat{\mathbf{w}}^{(i')}}_{=:\widehat{\mathbf{w}}^{(\mathcal{C})}}, \text{ for } i \in \mathcal{C},$$

between the learnt parameters $\widehat{\mathbf{w}}^{(i)}$ in the cluster $\mathcal{C}$ and their cluster-wide average $\widehat{\mathbf{w}}^{(\mathcal{C})}$.

Clustering Error of GTVMin



Upper Bound on Clustering Error

Theorem

The clustering error is upper bounded as

$$\sum_{i \in \mathcal{C}} \left\| \widetilde{\mathbf{w}}^{(i)} \right\|_2^2 \leq \frac{1}{\alpha \lambda_2\left(\mathbf{L}^{(\mathcal{C})}\right)} \left[\varepsilon^{(\mathcal{C})} + \alpha \left| \partial \mathcal{C} \right| 2 \left(\left\| \overline{\mathbf{w}}^{(\mathcal{C})} \right\|_2^2 + R^2 \right) \right]$$

Here, R denotes an upper bound on the Euclidean norm $\left\| \widehat{\mathbf{w}}^{(i)} \right\|_2$ outside the cluster, i.e., $\max_{i \in \mathcal{V} \setminus \mathcal{C}} \left\| \widehat{\mathbf{w}}^{(i)} \right\|_2 \leq R$.

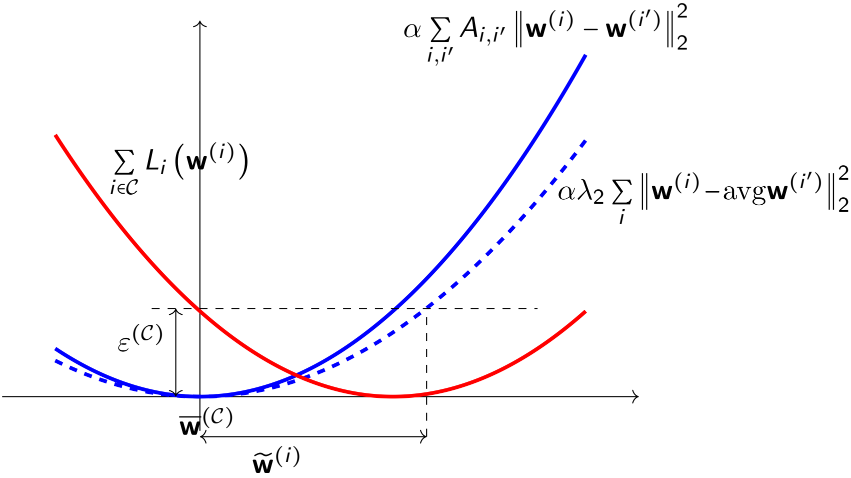
carefully note that:

- we only require clustering assumption
- allow for arbitrary loss functions (non-convex, non-smooth)
- need to ensure $\max_{i \in \mathcal{V} \setminus \mathcal{C}} \left\| \widehat{\mathbf{w}}^{(i)} \right\|_2 \leq R$

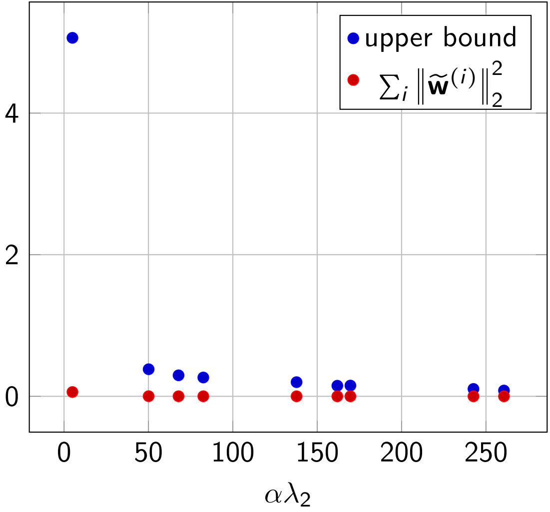
Ensure Upper Bound on Model Parameters

- we need good (enough) bound $R \geq \max_{i \in \mathcal{V} \setminus \mathcal{C}} \left\| \widehat{\mathbf{w}}^{(i)} \right\|_2$
- enforce bound by choosing $L_i\left(\mathbf{w}^{(i)}\right) = \infty$ for $\left\| \mathbf{w}^{(i)} \right\|_2 > R$
- place more restrictions on $L_i\left(\cdot\right)$, e.g.,
  - each $L_i\left(\cdot\right)$ differentiable with Lipschitz gradient
  - sum $\sum_i L_i\left(\cdot\right)$ is strongly convex

Proof Sketch



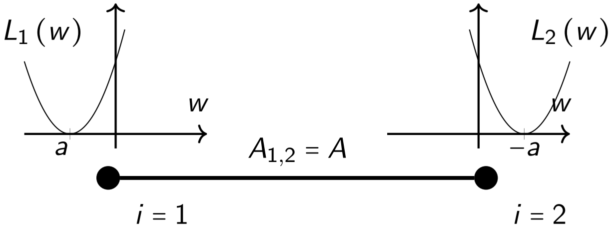
Numerical Test



Source code:
`https://github.com/alexjungaalto/ResearchPublic/`

Worst Case - Bound Becomes Tight(ish)

- $\mathcal{V} = \mathcal{C} = \{1, 2\}$, single edge $A_{1,2} = \lambda$
- local loss functions $L_1(w) = \rho(w-a)^2, L_2(w) = \rho(w+a)^2$.



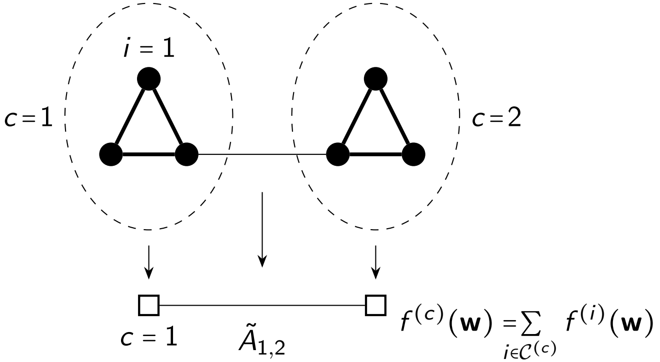
Error Analysis Beyond Clustering Error

- clustering asspt uses cluster-specific params $\overline{\mathbf{w}}^{(\mathcal{C})}$
- define estimation error $\Delta^{(i)} := \widehat{\mathbf{w}}^{(i)} - \overline{\mathbf{w}}^{(\mathcal{C})}$, for $i \in \mathcal{C}$
- we can decompose estimation error as

$$\Delta^{(i)} = \underbrace{\widetilde{\mathbf{w}}^{(i)}}_{\text{clustering error}} + \underbrace{\left(\widehat{\mathbf{w}}^{(\mathcal{C})} - \overline{\mathbf{w}}^{(\mathcal{C})}\right)}_{\text{constant across } i \in \mathcal{C}}$$

- our bound only covers first component
- how can we control $\widehat{\mathbf{w}}^{(\mathcal{C})} - \overline{\mathbf{w}}^{(\mathcal{C})}$?

Reduction to Cluster Graph



analyze GTVMin over cluster graph,[1]

$$\sum_{c} f^{(c)}(w^{(c)}) + \alpha \sum_{c,c'} \tilde{A}_{c,c'} \left\| w^{(c)} - w^{(c')} \right\|_2^2$$

[1] D. Sun and K.-C. Toh and Y. Yuan; Convex Clustering: Model, Theoretical Guarantee and Efficient Algorithm, JMLR, 2021

Table of Contents

Results

- derived upper bound on clustering error of GTVMin
- upper bound applies under mild clustering assumption
- bound is broadly applicable ☺
- can be very loose ☹

Follow Up

- how to make bounds tighter (average case analysis?)
- study graph constructions that optimize bound[1]
- guarantees for GTVMin over learnt $\mathcal{G}$

[1] Ying et.al., Exponential Graphs are Provably Efficient in Decentralized Deep Training, Neurips, 2021.

Questions?

Thank you!

Ping me if you are interested in Phd or Post-Doc positions !

LinkedIn: https://www.linkedin.com/in/aljung/
YouTube: alexjung111