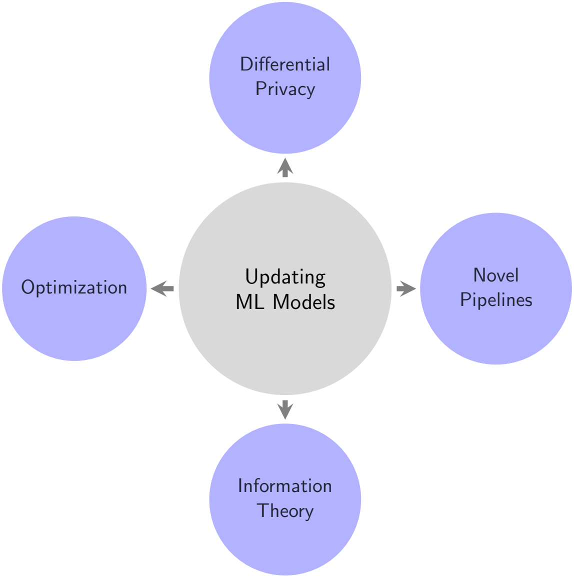
# Updating ML Models

Ananth Mahadevan

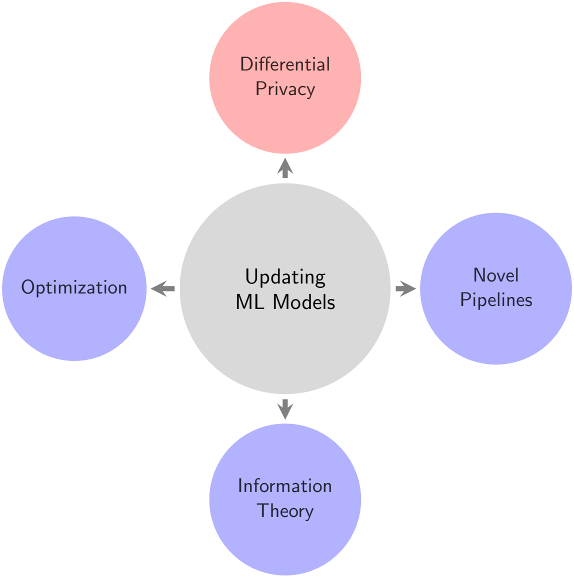
November 8, 2020

# Overview


Differential Privacy
Optimization
Updating ML Models
Novel Pipelines
Information Theory

# Common Terminology

- Fixed training Dataset $\mathcal{D}$
- Learning Algorithm $A$ (can be randomized)
- Datapoints to be remove $\mathcal{D}_{\mathcal{R}}$, where $|\mathcal{D}_{\mathcal{R}}| = r$, remaining dataset $\mathcal{D}' = \mathcal{D} - \mathcal{D}_{\mathcal{R}}$
- Naive approach is retraining from scratch, i.e, $A(\mathcal{D}')$
- Mechanism $M$ which offers an efficient way to update the model


Differential
Privacy
Optimization
Updating
ML Models
Novel
Pipelines
Information
Theory

# Certified Data Removal [Guo et al., 2020]

- $A$ outputs a model in hypothesis space $\mathcal{H}$
- Defines $\epsilon$-*certified removal*, $\forall \mathcal{T} \subseteq \mathcal{H}$

$$e^{-\epsilon} \leq \frac{P(M(A(\mathcal{D}), \mathcal{D}_{\mathcal{R}}) \in \mathcal{T})}{P(A(\mathcal{D}') \in \mathcal{T})} \leq e^{\epsilon}$$

- Insufficiency of Parametric indistinguishability
  - Approximate removal processes leaves a gradient residual
  - Residuals can reveal the prior presence of that training sample

# Removal Mechanism for Linear Classfiers

- $A$ empirical risk $L(\mathbf{w}; \mathcal{D})$ with a convex loss function $\ell(\mathbf{w}^T\mathbf{x}, y)$
- $\mathbf{w}^* = A(\mathcal{D}) = \text{argmin}_w L(\mathbf{w}; \mathcal{D})$
- To remove a single point $\mathcal{D}_{\mathcal{R}} = \{(\mathbf{x}_n, y_n)\}$
- Newton Update Step: $\mathbf{w}^- = M(\mathbf{w}^*, (\mathbf{x}_n, y_n)) = \mathbf{w}^* - H_{\mathbf{w}^*}^{-1}\nabla$
- Where $H_{\mathbf{w}^*} = \nabla^2 L(\mathbf{w}^*, \mathcal{D}')$ and $\nabla = \lambda\mathbf{w}^* + \nabla\ell((\mathbf{w}^*)^T\mathbf{x}_n, y_n)$
- $H_{\mathbf{w}^*}^{-1}\nabla$ is from *influence function* literature

# Influence Function

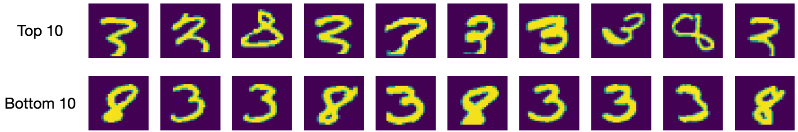



*Figure 3.* **MNIST training digits sorted by norm of the removal update $\|\mathbf{H}_{\mathbf{w}^*}^{-1}\boldsymbol{\Delta}\|_2$.** The samples with the highest norm (**top**) appear to be atypical, making it harder to undo their effect on the model. The samples with the lowest norm (**bottom**) are prototypical 3s and 8s, and hence are much easier to remove.

# Certifing Removal

- $\mathbf{w}^-$ is approximate close to minimizer of $L(\mathbf{w};\mathcal{D}')$
- $\nabla L(\mathbf{w}^-;\mathcal{D}')$ is gradient residual and if non-zero, reveals Information
- Even a small $\|\nabla L(\mathbf{w}^-;\mathcal{D}')\|_2$ doesn't guarantee certifiable removal
- Therefore, perturb loss at training time

$$L_b(\mathbf{w};\mathcal{D}) = \sum_{i=1}^{n} \ell(\mathbf{w}^T\mathbf{x}_i, y_i) + \frac{\lambda n}{2}\|\mathbf{w}\|_2^2 + \mathbf{b}^T\mathbf{w}$$

Where $\mathbf{b} \in \mathbb{R}^d$ drawn randomly from some distribution

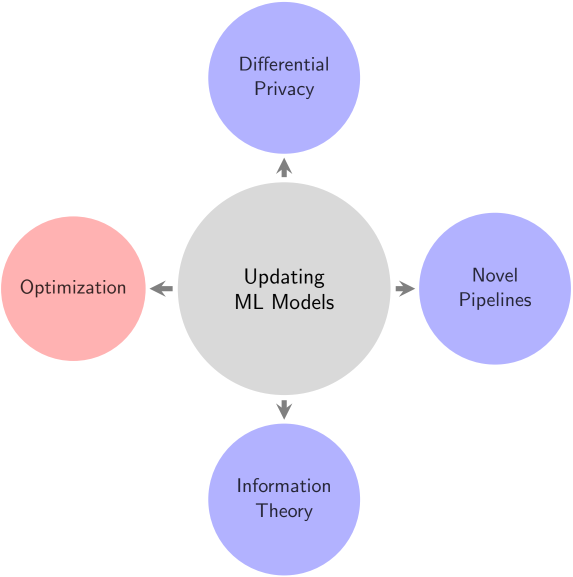
# Benefits and Drawbacks

## Benefits

- Provides formal guarantee of statistical indistinguishability
- Works well with Differentially Private trained networks
- Uses influence functions to approximate data removal

## Limitations

- Requires inverting a Hessian matrix
- Non-convex loss functions not supported
- Adding noise during training hurts model performance
- Very strict notion of removal


Differential Privacy
Optimization
Updating ML Models
Novel Pipelines
Information Theory

# DeltaGrad [Wu et al., 2020]

- $M$ targets the Gradient Descent (GD) algorithm
- Naive retraining $A(\mathcal{D}')$ recomputed gradients over all remaining points

$$\mathbf{w}^U_{t+1} \leftarrow \mathbf{w}^U_t - \frac{\eta_t}{n-r} \sum_{i \in \mathcal{D}'} \nabla L_i(\mathbf{w}^U_t)$$

- Instead rewrite it as a *leave-r-out* formula

$$\mathbf{w}^I_{t+1} = \mathbf{w}^I_t - \frac{\eta_t}{n-r} \left[ \sum_{i \in \mathcal{D}} \nabla L_i(\mathbf{w}^I_t) - \sum_{i \in \mathcal{D}_\mathcal{R}} \nabla L_i(\mathbf{w}^I_t) \right]$$

- Much cheaper to compute $r$ gradients, when $r \ll n$

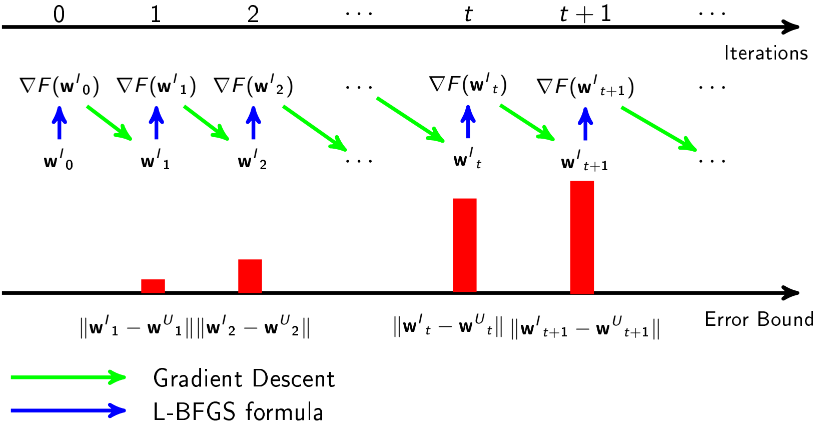
# Approximating $\sum_{i \in \mathcal{D}} \nabla L_i(\mathbf{w}_t^I)$

- Need to use historical $\nabla L(\mathbf{w}_t)$ to approximate $\nabla L(\mathbf{w}_t^I)$
- Taylor expansion around $\mathbf{w}_t^I$ gives the following

$$\nabla L(\mathbf{w}_t^I) = \nabla L(\mathbf{w}_t) + \mathbf{H}_t \cdot (\mathbf{w}_t^I - \mathbf{w}_t)$$

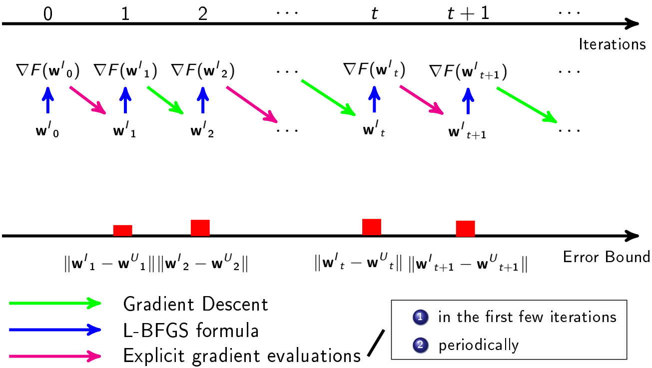
  Where $\mathbf{H}_t = \int_0^1 \mathbf{H}(\mathbf{w}_t + x(\mathbf{w}_t^I - \mathbf{w}))dx$
- Maintaining a Hessian matrix is expensive, so leverage the L-BFGS algorithm to compute a Hessian-vector product
- This leads to issues in error bounds of the approximation

# Problem with Error Bound



# Controlling the Errors

- Do explicit evaluations for $j_0$ "burn-in" iterations and then periodically every $T_0$ iterations



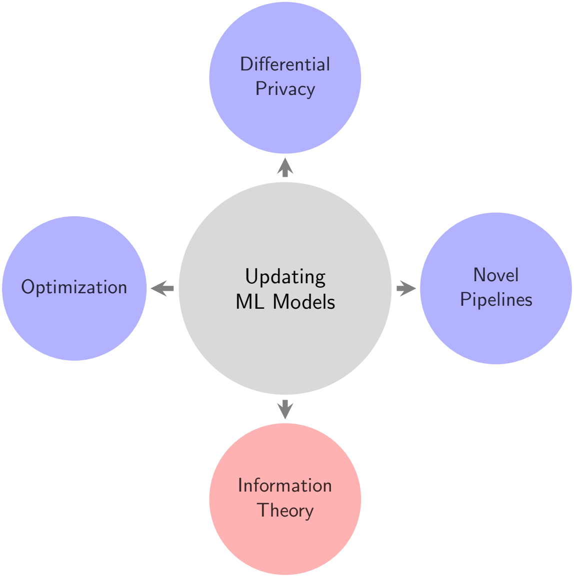
# Benefits and Limitations

## Benefits

- Handles both additions nad deletions of datapoints
- Can be applied to any ML model trained using Stochastic Gradient Descent
- Approximation guarantees and empirical results on

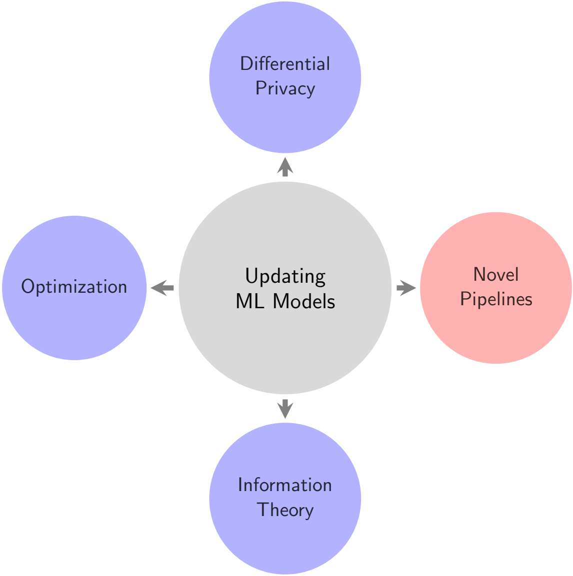
## Limitaitons

- Needs to cache all weights $\mathbf{w}_t$ and gradients $\nabla L(\mathbf{w}_t)$ during training
- Requires tuning of $T_0$ and $j_0$ based on dataset
- For SGD, only works with large batch sizes ($> 10000$), which hurts model performance


Differential Privacy
Optimization
Updating ML Models
Novel Pipelines
Information Theory

# Eternal Sunshine of the Spotless Net [Golatkar et al., 2020]

- 


Differential Privacy
Optimization
Updating ML Models
Novel Pipelines
Information Theory

# Machine Unlearning: SISA [Bourtoule et al., 2020]

-

# References I

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. (2020).
Machine Unlearning.
*arXiv:1912.03817 [cs].*

Golatkar, A., Achille, A., and Soatto, S. (2020).
Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks.
*arXiv:1911.04933 [cs, stat].*

# References II

Guo, C., Goldstein, T., Hannun, A., and van der Maaten, L. (2020).
Certified Data Removal from Machine Learning Models.
*arXiv:1911.03030 [cs, stat].*

Wu, Y., Dobriban, E., and Davidson, S. B. (2020).
DeltaGrad: Rapid retraining of machine learning models.
*arXiv:2006.14755 [cs, stat].*