# Distributed Statistical Learning for Wireless: Architectures, Algorithms and Information-theoretic View

**Romain Chor**

Laboratoire d'Informatique Gaspard Monge, Université Gustave Eiffel
Advanced Wireless Technology Lab, Huawei Paris Research Center

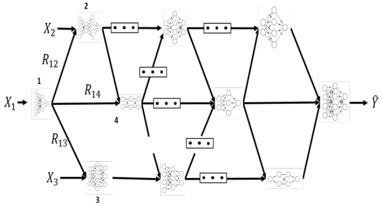
Thesis supervisors: Abdellatif Zaidi & Abderrezak Rachedi

June 7th, 2024

# Table of Contents

# A Network of Intelligent Devices for Wireless Networks

Modern "AI" in the era of Big Data = Devices collecting data + Communication via wireless networks

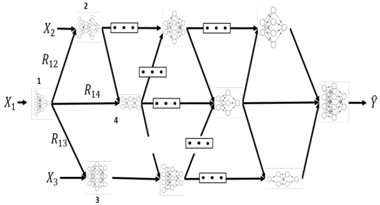



# A Network of Intelligent Devices for Wireless Networks

Modern "AI" in the era of Big Data = Devices collecting data + Communication via wireless networks

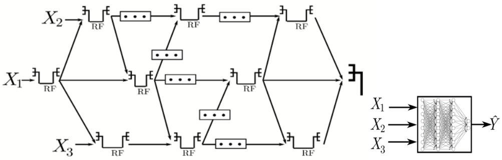
## Challenges of AI for wireless networks

- **Spatial data distributedness**
  - Useful data is distributed over multiple sites/nodes by nature.
  - Every part of the data may not be enough by its own.
- **Heterogeneity**
  - Devices may possess each a small amount of data.
  - Data is heterogeneous across devices, especially for sensory signals.
- **Privacy**
  - Devices not allowed and/or not desiring to share raw data.
  - Privacy/GDPR issues
- **Communication**
  - Bandwidth/power constraints
  - Devices mobility



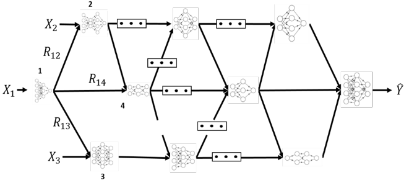

# Distributed Learning Solutions for Wireless Networks

**Centralized Learning**




Single node, multiple processing units

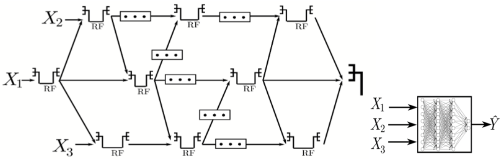
**Distributed Learning**




Devices in a wireless network

# Distributed Learning Solutions for Wireless Networks

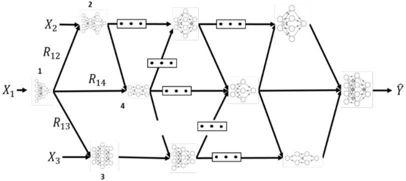
## Centralized Learning




Single node, multiple processing units

- Easy design, *e.g.*, use SOTA ML neural networks) but…
- requires large bandwidth.
- No privacy.

## Distributed Learning

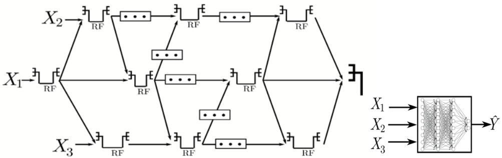



Devices in a wireless network

- Saves bandwidth.
- Preserves privacy (no raw data exchange).
- But *a priori* possible degradation of performance.

# Distributed Learning Solutions for Wireless Networks

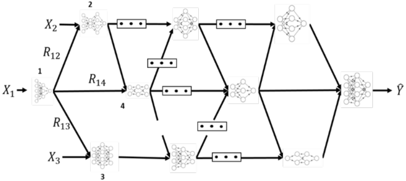
**Centralized Learning**




Single node, multiple processing units

- Easy design, *e.g.*, use SOTA ML neural networks) but...
- requires large bandwidth.
- No privacy.

**Distributed Learning**




Devices in a wireless network

- Saves bandwidth.
- Preserves privacy (no raw data exchange).
- But *a priori* possible degradation of performance.

This talk: generalization error as performance measure

# Preliminaries - Generalization Error

**Data:** For an unknown distribution $\mu$ on $\mathcal{Z}$,

- Input data $Z \sim \mu$
- Dataset: $n$ i.i.d. samples $S = \{Z_i\}_{i=1}^n \sim \mu^{\otimes n}$

**Learning algorithm:** mapping $\mathcal{A}$ from $Z \in \mathcal{Z}$ to a hypothesis $W \in \mathcal{W}$.

**Loss function:** $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}_+$.
E.g., for binary classification:
$z = (x, y),\ x \in \mathbb{R}^d,\ y \in \{-1, 1\}$, $\ell(z, w) \coloneqq \mathbb{1}_{yf(x,w)<0}$
(0-1 loss) with $f$ decision function.

---

# Preliminaries - Generalization Error

**Data:** For an unknown distribution $\mu$ on $\mathcal{Z}$,

- Input data $Z \sim \mu$
- Dataset: $n$ i.i.d. samples $S = \{Z_i\}_{i=1}^n \sim \mu^{\otimes n}$

**Learning algorithm:** mapping $\mathcal{A}$ from $Z \in \mathcal{Z}$ to a hypothesis $W \in \mathcal{W}$.

**Loss function:** $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}_+$.
E.g., for binary classification:
$z = (x, y),\ x \in \mathbb{R}^d,\ y \in \{-1, 1\},\ \ell(z, w) \coloneqq \mathbb{1}_{yf(x,w)<0}$
(0-1 loss) with $f$ decision function.

---

**Population risk:**

$$\mathcal{L}(w) \coloneqq \mathbb{E}_{Z\sim\mu}[\ell(Z, \mu)]$$

**Empirical risk:**

$$\hat{\mathcal{L}}(S, w) \coloneqq \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell(Z_i, w)$$

**Generalization error:**

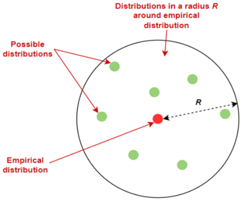
$$\operatorname{gen}(S, w) \coloneqq \mathcal{L}(w) - \hat{\mathcal{L}}(S, w)$$

Generalization error depends on:

- Loss function
- Data distribution
- Size of training dataset
- Learning algorithm

Related to algorithmic stability & robustness:

- Uniform stability
- Average stability




Exact analysis out of reach - resort to bounds:

- Tail bounds
- In-expectation bounds

# Distributed Learning: Problem Setup (1/2)

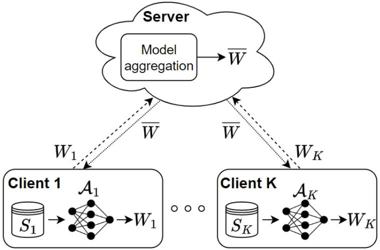
$K$ **clients,** each with a (local) dataset of size $n$.
**Datasets:** for a distribution $\mu$ on $\mathcal{Z}$,

- Input data $Z \sim \mu$.
- Client $\#i$: $n$ i.i.d. samples $S_i = \{Z_{i,j}\}_{j=1}^{n} \sim \mu^{\otimes n}$.
- Notation: $S = S_{1:K} := \cup_{i=1}^{K} S_i$.

All clients equipped with a same NN.
**(Local) learning algorithms:** Client $\#i$ learns model $W_i$ using algorithm $\mathcal{A}_i : S_i \in \mathcal{Z}^{\otimes n} \to \mathcal{W}$. Induces a distribution $P_{W_i|S_i}$.
**Central server:** aggregates the models $W_{1:K} := (W_i)_{i=1}^{K}$ as $\overline{W}$ according to

$$P_{S,W_{1:K},\overline{W}} := P_{\overline{W}|W_{1:K}} \prod_{i\in[K]} P_{S_i,W_i}.$$

---



# Distributed Learning: Problem Setup (1/2)

$K$ **clients,** each with a (local) dataset of size $n$.
**Datasets:** for a distribution $\mu$ on $\mathcal{Z}$,

- Input data $Z \sim \mu$.
- Client $\#i$: $n$ i.i.d. samples $S_i = \{Z_{i,j}\}_{j=1}^{n} \sim \mu^{\otimes n}$.
- Notation: $S = S_{1:K} \coloneqq \cup_{i=1}^{K} S_i$.

All clients equipped with a same NN.
**(Local) learning algorithms:** Client $\#i$ learns model $W_i$ using algorithm $\mathcal{A}_i : S_i \in \mathcal{Z}^{\otimes n} \to \mathcal{W}$. Induces a distribution $P_{W_i|S_i}$.
**Central server:** aggregates the models $W_{1:K} \coloneqq (W_i)_{i=1}^{K}$ as $\overline{W}$ according to

$$P_{S,W_{1:K},\overline{W}} \coloneqq P_{\overline{W}|W_{1:K}} \prod_{i\in[K]} P_{S_i,W_i}.$$

---

**Example 1:**

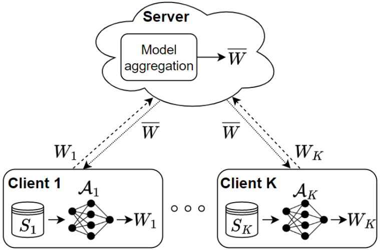
- $\mathcal{A}_i = \mathcal{A}, \forall i \in [K]$.
- $\mathcal{A} =$ SGD or $\mathcal{A} =$ ADAM.

**Example 2:**

- $\mathcal{A}_1 =$ SGD with learning rate $0.01$.
- $\mathcal{A}_2 =$ SGD with learning rate $0.002$.
- Etc.

**Example 3:**

- $\mathcal{A}_1 =$ SGD.
- $\mathcal{A}_2 =$ ADAM.
- Etc.



# Distributed Learning: Problem Setup (2/2)

For a hypothesis $\overline{w}$,

**Loss function:**

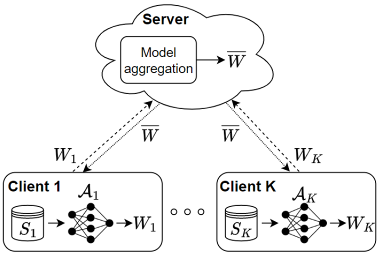
$$\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}_+$$

**Population risk:**

$$\mathcal{L}(\overline{w}) \coloneqq \mathbb{E}_{Z\sim\mu}[\ell(Z,\overline{w})]$$

**Empirical risk:**

$$\hat{\mathcal{L}}(S,\overline{w}) \coloneqq \frac{1}{nK}\sum_{i=1}^{K}\sum_{j=1}^{n}\ell(Z_{i,j},w)$$




**Generalization error**

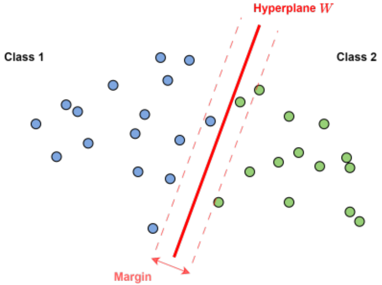
$$\mathrm{gen}(S,w) \coloneqq \mathcal{L}(\overline{w}) - \hat{\mathcal{L}}(S,\overline{w})$$

How to characterize generalization error of Distributed Learning?

# Case study: Distributed Support Vector Machines (DSVM)

Consider a binary classification problem.
Local learning algorithms $\mathcal{A}_i$: Support Vector Machines (SVM).
Hypothesis $W_i$: hyperplane coefficients.

**(Centralized) SVM**

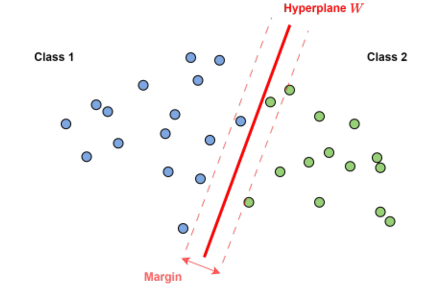



$W$ aims at classifying correctly all points, while maximizing a margin between them.
For a point $x$ with class label $y$, the SVM prediction is $\hat{y} = sign(W^\top x)$.

# Case study: Distributed Support Vector Machines (DSVM)

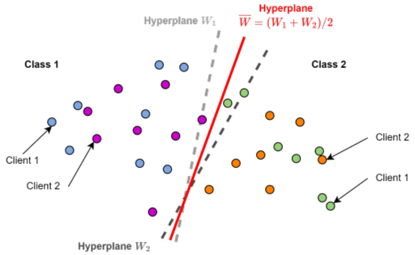
Consider a binary classification problem.
Local learning algorithms $\mathcal{A}_i$: Support Vector Machines (SVM).
Hypothesis $W_i$: hyperplane coefficients.

**(Centralized) SVM**




$W$ aims at classifying correctly all points, while maximizing a margin between them.
For a point $x$ with class label $y$, the SVM prediction is $\hat{y} = sign(W^\top x)$.

**Distributed SVM, $K = 2$ clients**



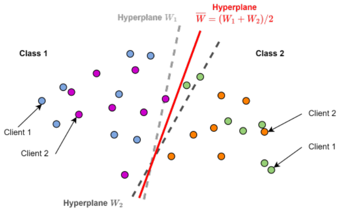

- $W_1$ can not classify all purple points errorless.
- $W_2$ can not classify all green points errorless.
- $\overline{W}$ classifies all points errorless!

# Generalization error of DSVM

**Theorem [SCZ22]**

Consider DSVM with $K$ clients. Then,

$$\mathbb{E}\left[\mathrm{gen}_\theta\left(S_{1:K}, \overline{W}\right)\right] \leq \mathcal{O}\left(\frac{\sqrt{\log(nK)\log(K)}}{\sqrt{nK^2}}\right).$$




[SCZ22] Rate-distortion Theoretic Bounds on Generalization Error for Distributed Learning. Sefidgaran M., Chor R. and Zaidi A., 2022

# Generalization error of DSVM

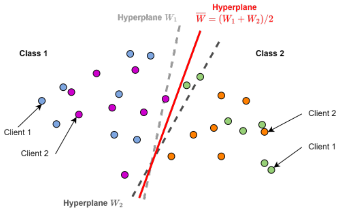
**Theorem [SCZ22]**

Consider DSVM with $K$ clients. Then,

$$\mathbb{E}\left[\mathrm{gen}_{\theta}(S_{1:K}, \overline{W})\right] \leq \mathcal{O}\left(\frac{\sqrt{\log(nK)\log(K)}}{\sqrt{nK^2}}\right).$$

1. Bound for distributed SVM decreases faster than that of the centralized one (with $nK$ samples) with a factor of order $\sqrt{\log(K)/K}$.
2. Similar behavior showed in high probability with a tail bound.

$\Longrightarrow$ SVM generalizes better (is more robust) when applied distributedly than in a centralized manner.




[SCZ22] Rate-distortion Theoretic Bounds on Generalization Error for Distributed Learning. Sefidgaran M., Chor R. and Zaidi A., 2022

# Elements of proof

1. Rate-distortion in standard lossy source coding.
2. Rate-distortion for lossy algorithm compression.
3. Rate-distortion bound for generalization error in centralized learning.
4. Dimensionality reduction: Johnson-Lindenstrauss transformation

# Elements of proof (1/3)

## Rate-distortion function and sochastic learning algorithms

(In standard source coding) Quantifies fundamental compression rate of a source within a fixed average distortion level $\epsilon$. If $W$ is obtained given $S$, the compression of $W$ into $\hat{W}$ within average distortion $\epsilon$ has minimum rate

$$\mathfrak{RD}(Q,\epsilon) := \inf_{P_{\hat{W}|S}} I(S,\hat{W}) \quad \text{s.t.} \quad \mathbb{E}[d(W,\hat{W})] \leq \epsilon$$

where $Q$ is a joint distribution over $\mathcal{S} \times \mathcal{W}$.

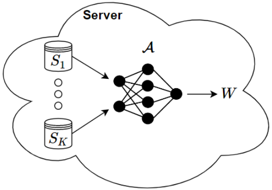
(For statistical learning) Measures lossy compressibility of an algorithm *i.e.*, smallest compressed hypothesis space that can be found s.t. distortion given by $d(W,\hat{W}) := \text{gen}(S,W) - \text{gen}(S,\hat{W})$ smaller than $\epsilon$ (in average):

$$\mathfrak{RD}(Q,\epsilon) := \inf_{P_{\hat{W}|S}} I(S,\hat{W}) \quad \text{s.t.} \quad \mathbb{E}[\text{gen}(S,W) - \text{gen}(S,\hat{W})] \leq \epsilon$$

**Theorem (Centralized learning)**

Consider the centralized learning setup with dataset $S$ of $nK$ i.i.d. samples and learning algorithm $\mathcal{A} : \mathcal{Z}^{\otimes nK} \to \mathcal{W}$. Suppose that for all $\hat{w} \in \mathcal{W}$, $\ell(Z,\hat{w})$ is $\sigma$-subgaussian *i.e.*, $\forall t \in \mathbb{R}$, $\mathbb{E}[\exp(t(\ell(Z,\hat{w}) - \mathbb{E}[\ell(Z,\hat{w})]))] \leq \exp(\sigma^2 t^2/2)$. Then, for any $\epsilon \in \mathbb{R}$,

$$\mathbb{E}[\text{gen}(S,W)] \leq \sqrt{\frac{2\sigma^2}{nK}\mathfrak{RD}(P_{S,W},\epsilon)}$$



# Elements of proof (2/3)

## Definitions

- For each client $i \in [K]$:
  - Dataset of $n$ i.i.d. samples: $S_i = \{Z_{i,j}\}_{j\in[n]} \subseteq \mathcal{Z}^n$, $S_i \sim \mu^{\otimes n}$
  - Learning algorithm: $A_i(S_i) = W_i \in \mathcal{W}$, $W_i$ local model.
  - Aggregated/Global model: $\overline{W} = (\sum_{i\in[K]} W_i)/K$
- $\mathcal{A}_i$ induces the distribution $P_{W_i|S_i}$, which together with $\mu$ induce the joint distribution $P_{S_i,W_i} = \mu^{\otimes n} P_{W_i|S_i}$. Thus, for $S = \cup_i S_i$, $W_{1:K} = (W_i)_i$, the distributed learning algorithm $\mathcal{A}(S)$ induces

$$P_{S,W_{1:K},\overline{W}} = P_{\overline{W}|W_{1:K}} \prod_{i\in[K]} P_{S_i,W_i}$$

- Compression of $W_i$ with $W_j$, $\forall j \neq i$ fixed: let $W_{1:K\setminus i} = (W_1, \ldots, W_{i-1}, W_{i+1}, \ldots, W_K)$, $\hat{W}_i$ compression of $W_i$. Hence, $\hat{\overline{W}}_i = (\hat{W}_i + W_{1:K\setminus i})/K$ is the resulting compressed global hypothesis.
  - Depends on $S_i$ and $W_{1:K\setminus i}$.
  - Rate-distortion function: for every distribution $Q$ over $\mathcal{W} \times (\mathcal{Z}^{\otimes n} \times \mathcal{W})^{\otimes K}$,

$$\mathfrak{RD}_i(Q,\epsilon) := \inf_{P_{\hat{\overline{W}}_i|S_i,W_{1:K\setminus i}}} I\Big(S_i; \hat{\overline{W}}_i | W_{1:K\setminus i}\Big) \quad \text{s.t.} \quad \mathbb{E}\Big[\mathrm{gen}(S_i,\overline{W}) - \mathrm{gen}(S_i,\hat{\overline{W}}_i)\Big] \leq \varepsilon$$

# Elements of proof (3/3)

Block-coding: we use the previously introduced compression scheme to extend the centralized learning generalization bound to our distributed learning setup.
Doing the compression of the hypothesis $W_i$ of client $\#$i in a lower-dimensional space using the Johnson-Lindenstrauss transformation gives the following.

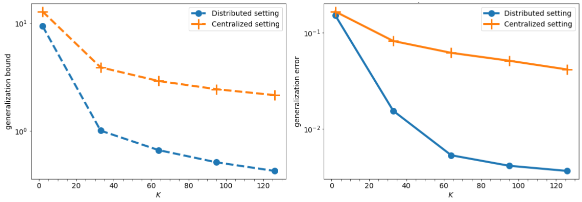
**Lemma**

For every $m \in \mathbb{N}^*$ and every non-negative triplet $(c_1, c_2, \nu)$, it holds that

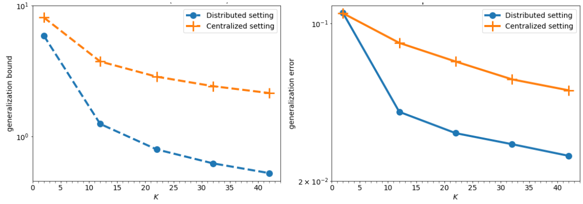
$$\mathfrak{RD}_i(Q, \epsilon) \leq m \log((c_2 + \nu)/\nu)$$

where $\epsilon$ depends on $m, c_1, c_2$ and $\nu$.

# Experiments: Results




(a) $n = 100$




(b) $n = 300$

Figure: Theoretical bound (left plots) and generalization error (right plots) for distributed and centralized learning settings versus $K$.

## Experimental setup

- Dataset: MNIST, 2 classes
- Model: SVM with Gaussian kernel, SGD training
- Hyperparameters:
  - Initial learning rate: 0.01
  - Regularization parameter: 0.00001
  - Kernel parameter: 0.01
  - Kernel feature space's dimension: 2000

## Interpretation

1. Generalization error of DSVM is smaller than for centralized SVM, for any $K$!
2. In-expectation bound follows the behavior of the generalization error.

# Distributed Learning: Problem Setup (recall)

$K$ **clients,** each with a (local) dataset of size $n$.
**Datasets:** for a distribution $\mu$ on $\mathcal{Z}$,

- Input data $Z \sim \mu$.
- Client $\#i$: $n$ i.i.d. samples $S_i = \{Z_{i,j}\}_{j=1}^{n} \sim \mu^{\otimes n}$.
- Notation: $S = S_{1:K} := \cup_{i=1}^{K} S_i$.

**(Local) learning algorithms:** mapping $\mathcal{A}$ from $S_i \in \mathcal{Z}^{\otimes n}$ to $W_i \in \mathcal{W}$. For client $\#i$, induces a distribution $P_{W_i|S_i}$.

**Central server/Fusion center:** aggregates the models $W_{1:K} := (W_i)_{i=1}^{K}$ according to

$$P_{S,W_{1:K},\overline{W}} := P_{\overline{W}|W_{1:K}} \prod_{i\in[K]} P_{S_i,W_i}.$$

**Population risk:**

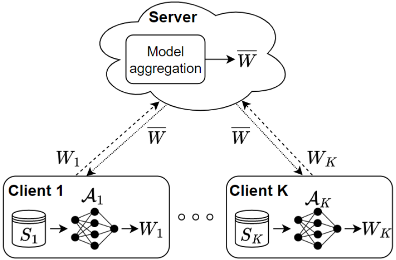
$$\mathcal{L}(\overline{w}) := \mathbb{E}_{Z\sim\mu}[\ell(Z,\overline{w})]$$

**Empirical risk:**

$$\hat{\mathcal{L}}(S,\overline{w}) := \frac{1}{nK}\sum_{i=1}^{K}\sum_{j=1}^{n}\ell(Z_{i,j},w)$$

**Generalization error**

$$\operatorname{gen}(S,w) := \mathcal{L}(\overline{w}) - \hat{\mathcal{L}}(S,\overline{w})$$



# Generalization Error of Distributed Learning Algorithms

**Theorem [SCZ22]**

If the loss is $\sigma$-subgaussian, then $\forall \epsilon \in \mathbb{R}$,

$$\mathbb{E}\left[\operatorname{gen}(S_{1:K}, \overline{W})\right] \leq \sqrt{\frac{2\sigma^2}{n} \max_{i \in [K]} \mathfrak{RD}_i(P_{S_i, W_{1:K}, \overline{W}}, \epsilon)} + \epsilon.$$

where $\mathfrak{RD}_i(P_{S_i, W_{1:K}, \overline{W}}, \epsilon)$ is the rate-distortion function, measuring the fundamental local algorithm compressibility of client $i$ within $\epsilon$ distortion, conditioned on other $W_j$, $j \neq i$.

# Generalization Error of Distributed Learning Algorithms

**Theorem [SCZ22]**

If the loss is $\sigma$-subgaussian, then $\forall \epsilon \in \mathbb{R}$,

$$\mathbb{E}[\mathrm{gen}(S_{1:K}, \overline{W})] \leq \sqrt{\frac{2\sigma^2}{n} \max_{i \in [K]} \mathfrak{RD}_i(P_{S_i, W_{1:K}, \overline{W}}, \epsilon)} + \epsilon.$$

where $\mathfrak{RD}_i(P_{S_i, W_{1:K}, \overline{W}}, \epsilon)$ is the rate-distortion function, measuring the fundamental local algorithm compressibility of client $i$ within $\epsilon$ distortion, conditioned on other $W_j$, $j \neq i$.

- Intuitively, each $W_i$ has effect of $1/K$ on $\overline{W}$; hence with Lipschitz loss, separate compression allows for local distortion of order $K\epsilon$.
- Bound reduces to mutual-information based bounds for $\epsilon = 0$ (Xu and Raginsky, 2017).
- Multiple extensions and similar tail bounds.

# Summary

- **Problem:** Generalization error of distributed stochastic learning algorithms.

# Summary

- **Problem:** Generalization error of distributed stochastic learning algorithms.

## Results

1. General tail bounds and in-expectation bounds on the generalization error.
   - Improves over the prior arts [YDP20, BDP22].
2. Generalization error bound decreases as number of clients increases, for
   - Distributed SVM
   - Federated SGLD
   - Locally deterministic algorithms with Lipschitz loss
3. Experimentally verified the findings.

# Summary

- **Problem:** Generalization error of distributed stochastic learning algorithms.

## Results

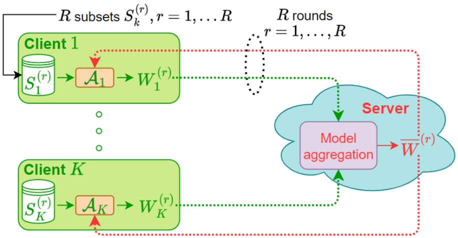
1. General tail bounds and in-expectation bounds on the generalization error.
   - Improves over the prior arts [YDP20, BDP22].
2. Generalization error bound decreases as number of clients increases, for
   - Distributed SVM
   - Federated SGLD
   - Locally deterministic algorithms with Lipschitz loss
3. Experimentally verified the findings.

- **Approach:** Rate-distortion theoretic framework, adapted for algorithm compressibility
- **Intuition:** Distributed algorithms reduce the variance of the model!

[YDP20] Information-theoretic bounds on the generalization error and privacy leakage in federated learning. Yagli S., Dytso A., Poor H.V., 2020

[BDP22] Improved Information Theoretic Generalization Bounds for Distributed and Federated Learning. Barnes L.P., Dytso A., and Poor H.V., 2022

# From "one-shot" to multi-round Federated Learning




For a model $\overline{w} = \overline{w}^{(R)}$,

**Empirical risk:**

$$\hat{\mathcal{L}}(S, \overline{w}) = \frac{1}{nK} \sum_{k \in [K]} \sum_{i \in [n]} \ell(Z_{k,i}, \overline{w})$$

**Population risk:**

$$\mathcal{L}(\overline{w}) = \mathbb{E}_Z[\ell(Z, w)]$$

**Generalization error:**

$$\mathrm{gen}(S, w) = \mathcal{L}(\overline{w}) - \hat{\mathcal{L}}(S, \overline{w})$$

### $R$-rounds FL algorithm

$K$ clients, each equipped with a dataset $S_k = \{Z_{k,i}\}_{i=1}^n \sim \mu^{\otimes n}$.

1. Round $r = 0$: every client $k \in [K]$ initializes its model $W_k^{(0)}$ with some $\overline{W}^{(0)} = W_0$.
2. Rounds $r \in [1; R]$: every client $k \in [K]$ learns a local model $W_k^{(r)}$ with their algorithm $\mathcal{A}_k$, using samples $S_k^{(r)}$ and initialization $\overline{W}^{(r-1)}$:
$$W_k^{(r)} := \mathcal{A}_k(S_k^{(r)}, \overline{W}^{(r-1)})$$
Local models $\{W_k^{(r)}\}_{k \in [K]}$ are sent to the server.
3. Server aggregates local models as $\overline{W}^{(r)}$ and send this global model back to the clients.

Final global model (after $R$ rounds): $\overline{W}^{(R)}$

### Technical problem

What is the effect of the # of communication rounds $R$ in Federated Learning?

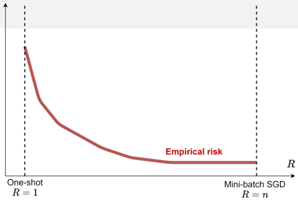
### Practical value

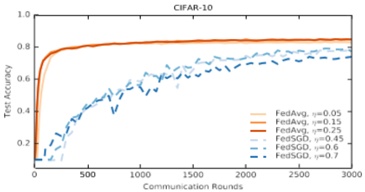
Saving in # of communication rounds $R$ would translate into bandwidth savings!

# What do we know about this problem?

## Result 1: Empirical risk of LocalSGD

When $\mathcal{A}_k = SGD, \forall k \in [K]$ and $\overline{w}$ is the arithmetic average of clients' local models, empirical risk $\hat{\mathcal{L}}(S, \overline{w})$ decreases with # of communication rounds $R$. [McMahan+17]
$\implies$ More communication with the parameter server helps for optimization in Federated Learning (FL).




[McMahan+17]

[McMahan+17] Communication-efficient Learning of Deep Networks from Decentralized Data. McMahan B.H. *et al.*, 2017

# What do we know about this problem?

## Result 1: Empirical risk of LocalSGD

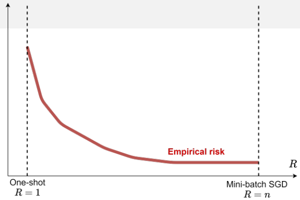
When $\mathcal{A}_k = SGD, \forall k \in [K]$ and $\overline{w}$ is the arithmetic average of clients' local models, empirical risk $\hat{\mathcal{L}}(S, \overline{w})$ decreases with # of communication rounds $R$. [McMahan+17]
$\Longrightarrow$ More communication with the parameter server helps for optimization in Federated Learning (FL).

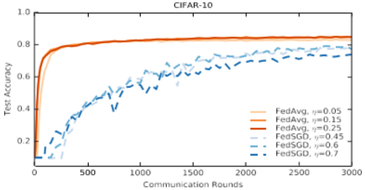
## Technical Problem

- As previously said, empirical risk does not reflect the true performance of the model $\overline{w}$.
- What matters is how the population risk or the generalization error evolve w.r.t. $R$!

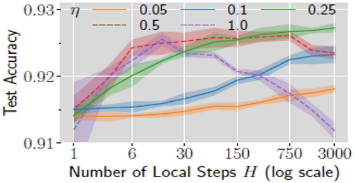
## Result 2

In some cases, it was observed experimentally that LocalSGD with $R < n$ has smaller population risk than ParallelSGD $(R = n)$.




[McMahan+17]




[GLHA23]

[McMahan+17] Communication-efficient Learning of Deep Networks from Decentralized Data. McMahan B.H. *et al.*, 2017

[GLHA23] Why (and When) Does LocalSGD Generalize Better Than SGD? Gu X. *et al.*, 2023

# Only a few theoretical results on generalization error in FL

## Prior art

- Rate-distortion theoretic bounds for "one-shot" FL algorithms (previous section) [SCZ22]
  - Centralized learning: all datasets collected at one point *i.e.*, $\cup_{k=1}^{K} S_k = S$ to train a model $W$, $\mathbb{E}_{S,W}[\mathrm{gen}(S,W)] = \mathcal{O}(1/\sqrt{nK})$
  - One-shot FL: $\mathbb{E}_{S,W}[\mathrm{gen}(S,\overline{W})] = \mathcal{O}(1/\sqrt{nK^2})$

    $\Longrightarrow$ More clients = smaller generalization error than centralized setting!

# Only a few theoretical results on generalization error in FL

## Prior art

- Rate-distortion theoretic bounds for "one-shot" FL algorithms (previous section) [SCZ22]
  - Centralized learning: all datasets collected at one point *i.e.*, $\cup_{k=1}^{K} S_k = S$ to train a model $W$, $\mathbb{E}_{S,W}[\text{gen}(S,W)] = \mathcal{O}(1/\sqrt{nK})$
  - One-shot FL: $\mathbb{E}_{S,W}[\text{gen}(S,\overline{W})] = \mathcal{O}(1/\sqrt{nK^2})$

  $\implies$ More clients = smaller generalization error than centralized setting!
- For multi-round FL:
  - Bound on a proxy to generalization error. [BDP22]
  - Bound for generalization error for specific loss functions and learning algorithms, suggesting that generalization error increases with $R$. [CSZ23]

[CSZ23] More Communication Does Not Result in Smaller Generalization Error in Federated Learning. Chor R., Sefidgaran M. and Zaidi A., 2023

# Only a few theoretical results on generalization error in FL

## Prior art

- Rate-distortion theoretic bounds for "one-shot" FL algorithms (previous section) [SCZ22]
    - Centralized learning: all datasets collected at one point *i.e.*, $\cup_{k=1}^{K} S_k = S$ to train a model $W$, $\mathbb{E}_{S,W}[\text{gen}(S,W)] = \mathcal{O}(1/\sqrt{nK})$
    - One-shot FL: $\mathbb{E}_{S,W}[\text{gen}(S,\overline{W})] = \mathcal{O}(1/\sqrt{nK^2})$
    
    $\implies$ More clients = smaller generalization error than centralized setting!
- For multi-round FL:
    - Bound on a proxy to generalization error. [BDP22]
    - Bound for generalization error for specific loss functions and learning algorithms, suggesting that generalization error increases with $R$. [CSZ23]

(In more general settings) Does generalization error increases with the number of communication rounds $R$ in FL?

[CSZ23] More Communication Does Not Result in Smaller Generalization Error in Federated Learning. Chor R., Sefidgaran M. and Zaidi A., 2023

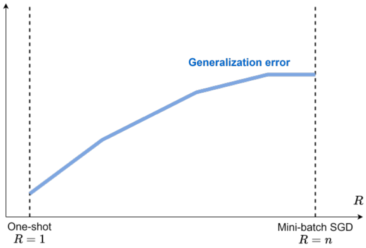
# What can we expect?

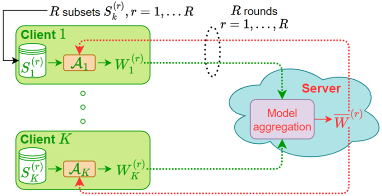
## Intuition

- FL helps to make individual models extract features that are in other clients' data when $R$ is larger.
- Works as if every client "sees" more data locally *i.e.,* "virtually" larger training dataset.
- Generalization error is shown to decrease with the dataset size $n$ [XR2017]: for a model $W$ trained on $S$,

$$|\operatorname{gen}(S,W)| \leq \sqrt{\frac{2\sigma^2}{n} I(S;W)}$$

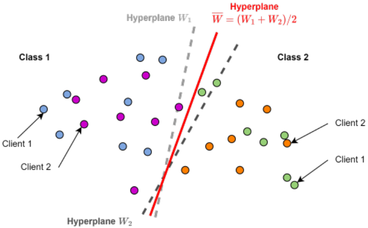
$\Longrightarrow$ Generalization error of FL should decrease with $R$.




[XR17] Information-theoretic analysis of generalization capability of learning algorithms. Xu A. and Raginsky M., 2017

# Generalization error of Federated SVM (FSVM)



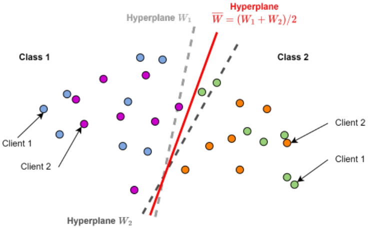

**Theorem [SCZW24]**

For FSVM optimized using $(K, R, n, e, b)$-FL-SGD with $\mathcal{W} = \mathcal{B}_d(1)$, $\mathcal{X} = \mathcal{B}_d(B)$ and $\theta \in \mathbb{R}^+$, under some assumptions and for some constants $q_{e,b}$ and $\alpha$,

$$\mathbb{E}\left[\operatorname{gen}_\theta\left(\mathbf{S}, \overline{W}^{(R)}\right)\right] = \mathcal{O}\left(\sqrt{\frac{B^2 \log(nK\sqrt{K}) \sum_{r\in[R]} L_r}{nK^2\theta^2}}\right),$$

where $L_r \leq q_{e,b}^{2(R-r)} \log\max\left(\frac{K\theta}{Bq_{e,b}^{(R-r)}}, 2\right)$.

# Generalization error of Federated SVM (FSVM)



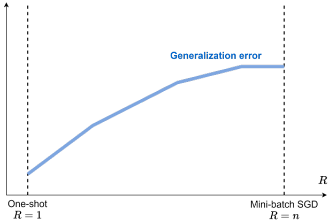

**Theorem [SCZW24]**

For FSVM optimized using $(K, R, n, e, b)$-FL-SGD with $\mathcal{W} = \mathcal{B}_d(1)$, $\mathcal{X} = \mathcal{B}_d(B)$ and $\theta \in \mathbb{R}^+$, under some assumptions and for some constants $q_{e,b}$ and $\alpha$,

$$\mathbb{E}\left[\operatorname{gen}_\theta\left(\mathbf{S}, \overline{W}^{(R)}\right)\right] = \mathcal{O}\left(\sqrt{\frac{B^2 \log(nK\sqrt{K}) \sum_{r\in[R]} L_r}{nK^2\theta^2}}\right),$$

where $L_r \leq q_{e,b}^{2(R-r)} \log \max\left(\frac{K\theta}{Bq_{e,b}^{(R-r)}}, 2\right)$.

1. Explicit bound that depends on $\#$ of communication rounds $R$, $\#$ of clients $K$ and local dataset size $n$.
2. Bound increases with $R$, for fixed $(n, K) \implies$ More communication may hurts for generalization!




[SCZW24] Lessons from Generalization Error Analysis of Federated Learning : You May Communicate Less Often!. Sefidgaran M., Chor R., Zaidi. A and Wan Y., 2024

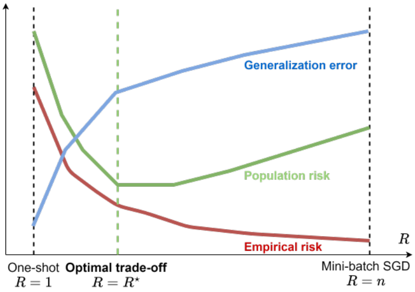
# Towards optimal communication?

Recall that generalization error is given by

$$\mathrm{gen}(S, \overline{w}) := \mathcal{L}(\overline{w}) - \hat{\mathcal{L}}(S, \overline{w})$$

with empirical risk $\hat{\mathcal{L}}(S, \overline{w})$ and population risk $\mathcal{L}(\overline{w})$.

1. Population risk = Empirical risk + Generalization Error.
2. Empirical risk decreases with $R$.
3. Generalization error increases with $R$ (previous results).




## Consequences

$\Longrightarrow$ Population risk may have a minimum for $R^* < R_{max}$!
$\Longrightarrow$ Less communication can be beneficial for the true performance of an FL algorithm.

## Implications

- Choice of # of communication rounds directly related to the required system bandwidth!
- Should be designed on the true system performance indicators, not the empirical risk (often considered for simplicity).

# Experiments: FSVM

## Experimental setup

- Dataset: MNIST, 2 classes, $n = 100$
- Model: SVM with Gaussian kernel, SGD training
- Learning rate $0.01$, batch size $1$, # of epochs $40$

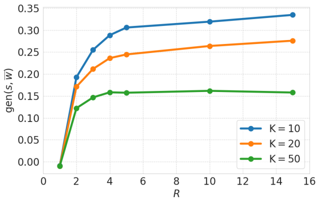
## Interpretation

(a) Generalization error increases with $R$ for different fixed $K$. Similar results for other values of $n$.

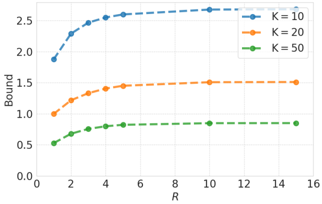
(b) In-expectation bound follows the behavior of the generalization error.

(c) Empirical risk decreases with $R$.

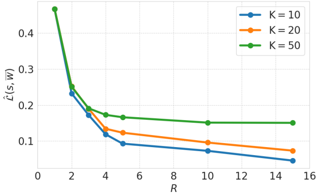
(d) Population risk quickly converges to a value



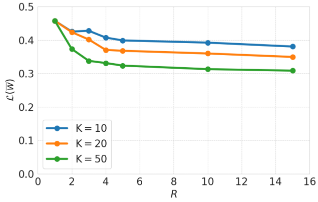

(a) Generalization error




(b) Bound




(a) Empirical risk




(b) Population risk

# Generalization error of FL for general algorithms & models (1/2)

**Theorem [SCZW24]**

For any $(P_{\mathbf{W}|\mathbf{S}}, K, R, n)$-FL model with distributed dataset $\mathbf{S} \sim P_{\mathbf{S}}$, if the loss $\ell(Z_k, w)$ is $\sigma$-subgaussian for every $w \in \mathcal{W}$ and any $k \in [K]$, then for every $\epsilon \in \mathbb{R}$ it holds that

$$\mathbb{E}_{\mathbf{S},\mathbf{W}\sim P_{\mathbf{S},\mathbf{W}}}\left[\operatorname{gen}(\mathbf{S}, \overline{W}^{(R)})\right] \leq \sqrt{2\sigma^2 \sum_{k\in[K],r\in[R]} \mathfrak{RD}(P_{\mathbf{S},\mathbf{W}}, k, r, \epsilon_{k,r})/(nK)} + \epsilon.$$

for any set of parameters $\{\epsilon_{k,r}\}_{k\in[K],r\in[R]} \subset \mathbb{R}$ which satisfy $\frac{1}{KR}\sum_{k\in[K]}\sum_{r\in[R]} \epsilon_{k,r} \leq \epsilon$.

# Generalization error of FL for general algorithms & models (1/2)

**Theorem [SCZW24]**

For any $(P_{\mathbf{W}|\mathbf{S}}, K, R, n)$-FL model with distributed dataset $\mathbf{S} \sim P_{\mathbf{S}}$, if the loss $\ell(Z_k, w)$ is $\sigma$-subgaussian for every $w \in \mathcal{W}$ and any $k \in [K]$, then for every $\epsilon \in \mathbb{R}$ it holds that

$$\mathbb{E}_{\mathbf{S},\mathbf{W}\sim P_{\mathbf{S},\mathbf{W}}}\left[\operatorname{gen}(\mathbf{S}, \overline{W}^{(R)})\right] \leq \sqrt{2\sigma^2 \sum_{k\in[K],r\in[R]} \mathfrak{RD}(P_{\mathbf{S},\mathbf{W}}, k, r, \epsilon_{k,r})/(nK)} + \epsilon.$$

for any set of parameters $\{\epsilon_{k,r}\}_{k\in[K],r\in[R]} \subset \mathbb{R}$ which satisfy $\frac{1}{KR}\sum_{k\in[K]}\sum_{r\in[R]} \epsilon_{k,r} \leq \epsilon$.

- Captures "contribution" of each client's local model during each round to the global model through rate-distortion functions.
- Same observation in high probability with a tail bound.

# Generalization error of FL for general algorithms & models (2/2)

**Theorem [SCZW23]**

Assume that $\ell(Z_k, w)$ is $\sigma$-subgaussian for every $w \in \mathcal{W}$ and any $k \in [K]$. Denote as

- $\mathbf{W}$ a concatenation of all local and global models at every round,
- For every $k \in [K]$ and $r \in [R]$, $\mathsf{P}_{k,r}$ a conditional prior on $W_k^{(r)}$ given $\overline{W}^{(r-1)}$

Then, with probability at least $(1-\delta)$ over $\mathbf{S}$, for all $P_{\mathbf{W}|\mathbf{S}}$, $\mathbb{E}_{\mathbf{W}\sim P_{\mathbf{W}|\mathbf{S}}}\left[\operatorname{gen}(\mathbf{S}, \overline{W}^{(R)})\right]$ is bounded by

$$\sqrt{\frac{\frac{1}{KR}\sum_{k\in[K],r\in[R]} \mathbb{E}_{\overline{W}^{(r-1)}\sim P_{\overline{W}^{(r-1)}|S_{[K]}^{[r-1]}}}\left[D_{KL}\left(P_{W_k^{(r)}|S_k^{(r)},\overline{W}^{(r-1)}}\|\mathsf{P}_{k,r}\right)\right]+\log(\frac{2n}{R\delta})}{(2n/R-1)/(4\sigma^2)}}.$$

# Generalization error of FL for general algorithms & models (2/2)

**Theorem [SCZW23]**

Assume that $\ell(Z_k, w)$ is $\sigma$-subgaussian for every $w \in \mathcal{W}$ and any $k \in [K]$. Denote as

- $\mathbf{W}$ a concatenation of all local and global models at every round,
- For every $k \in [K]$ and $r \in [R]$, $\mathsf{P}_{k,r}$ a conditional prior on $W_k^{(r)}$ given $\overline{W}^{(r-1)}$

Then, with probability at least $(1-\delta)$ over $\mathbf{S}$, for all $P_{\mathbf{W}|\mathbf{S}}$, $\mathbb{E}_{\mathbf{W}\sim P_{\mathbf{W}|\mathbf{S}}}\left[\operatorname{gen}(\mathbf{S}, \overline{W}^{(R)})\right]$ is bounded by

$$\sqrt{\frac{\frac{1}{KR}\sum_{k\in[K],r\in[R]}\mathbb{E}_{\overline{W}^{(r-1)}\sim P_{\overline{W}^{(r-1)}|S_{[K]}^{[r-1]}}}\left[D_{KL}\left(P_{W_k^{(r)}|S_k^{(r)},\overline{W}^{(r-1)}}\|\mathsf{P}_{k,r}\right)\right]+\log(\frac{2n}{R\delta})}{(2n/R-1)/(4\sigma^2)}}.$$

- Accounts explicitly for the effect of the number of rounds $R$ + number of participating clients $K$ and size of local datasets $n$.
- Captures "contribution" of each client's local model during each round to the global model: KL divergence terms.
- Same observations with a tail bound and bounds for lossy compression case.

# Experiments: Ordinary Least Squares

## Experimental setup

- Dataset: synthetic dataset with dimension $d = 10$, $n = 500$, $K = 10$
- Model: Ordinary Least Squares with SGD training.
- Hyperparameters:
  - Learning rate: 0.01
  - Client batch size: 1

## Interpretation

Generalization error increases with $R$, as in FSVM experiments.

Note: the shown bound is theoretically derived in [CSZ23].

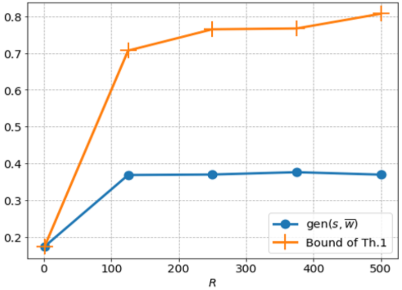



Figure: Generalization error vs. $R$

See [CSZ23]

# Experiments: CNNs

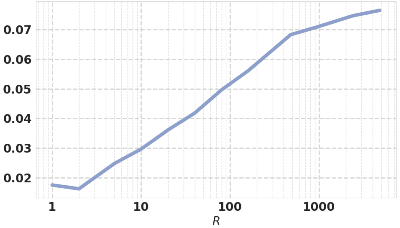
## Experimental setup

- Dataset: CIFAR-10 (50000 training images, 10000 test images) with $K = 16$.
- Model: ResNet-56 with SGD training
- Hyperpararamers:
  - Client batch size: 128
  - Learning rate: 1.0
  - Epochs: 100

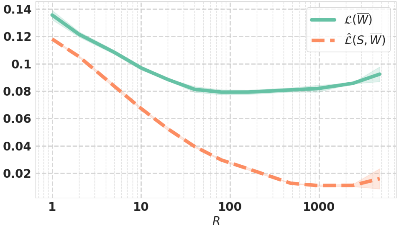
## Interpretation

1. As in FSVM experiments, similar observations for generalization error and empirical risk.
2. Explicit "U-shape" of the population risk; minimizer $R^* \simeq 100$.
3. $R^* << R_{max} = 3600$ hence huge savings in communication rounds are possible!

See [SCZW24]




(a) Generalization error vs. $R$




(b) Empirical and population risks vs. $R$

# Computation of Generalization Bounds for FL (1/2)

**Corollary**

Let $\hat{\mathcal{W}} = \mathcal{W}$ and $p_{\hat{W}_k^{(r)}|S_k^{(r)},\overline{W}^{(r-1)}} = P_{W_k^{(r)}|S_k^{(r)},\overline{W}^{(r-1)}}$. Then, the rate distortion function for $\epsilon = 0$ can be upper bounded as

$$\mathbb{E}_{S,\mathbf{W}\sim P_{\mathbf{S},\mathbf{W}}}\left[\mathrm{gen}(S,\overline{W}^{(R)})\right] \leq \sqrt{\frac{2\sigma^2 \sum_{k\in[K],r\in[R]} I(S_k^{(r)};W_k^{(r)}|\overline{W}^{(r-1)})}{nK}}.$$

# Computation of Generalization Bounds for FL (1/2)

**Corollary**

Let $\hat{\mathcal{W}} = \mathcal{W}$ and $p_{\hat{W}_k^{(r)}|S_k^{(r)},\overline{W}^{(r-1)}} = P_{W_k^{(r)}|S_k^{(r)},\overline{W}^{(r-1)}}$. Then, the rate distortion function for $\epsilon = 0$ can be upper bounded as

$$\mathbb{E}_{S,\mathbf{W}\sim P_{\mathbf{S},\mathbf{W}}}\left[\text{gen}(S,\overline{W}^{(R)})\right] \leq \sqrt{\frac{2\sigma^2 \sum_{k\in[K],r\in[R]} I(S_k^{(r)};W_k^{(r)}|\overline{W}^{(r-1)})}{nK}}.$$

- CMI term $I(S_k^{(r)};W_k^{(r)}|\overline{W}^{(r-1)})$ has no closed-form expression.
- Only access to a single instance of $(S,W,\overline{W}) \equiv (S_k^{(r)},W_k^{(r)},\overline{W}^{(r-1)})$.

Can we compute the CMI generalization bound in a "one-shot" manner?

[SCZ24] On the Effect of Communication on the Generalization Error in Federated Learning. Sefidgaran M., Chor R., Zaidi. A, 2024

# Computation of Generalization Bounds for FL (2/2)

- CMI reformulation:

$$I(S;W|\overline{W}) = \mathbb{E}_{P_{S,\overline{W}}}\left[D_{KL}\left(P_{W|S,\overline{W}} \| P_{W|\overline{W}}\right)\right] = \min_{\mathsf{P}_{W|\overline{W}}} E_{P_{S,\overline{W}}}\left[D_{KL}\left(P_{W|S,\overline{W}} \| \mathsf{P}_{W|\overline{W}}\right)\right], \quad (1)$$

where the minimum is achieved whenever the prior $\mathsf{P}_{W|\overline{W}}$ equals the marginal distribution $P^* := P_{W|\overline{W}} := \mathbb{E}_{P_S}[P_{W|S,\overline{W}}]$. Such a prior is often called "oracle".

- Assume that $\mathsf{P}_{W|\overline{W}} = \mathcal{N}(\mu_{\overline{W}}, \Sigma_{\overline{W}})$ and $P_{W|S,\overline{W}} = \mathcal{N}(\alpha_{S,\overline{W}}, C_{S,\overline{W}})$. Then,

$$I(S;W|\overline{W}) \propto \mathbb{E}_{P_{S,\overline{W}}}[(\alpha-\mu)^\top \Sigma^{-1}(\alpha-\mu)] = \mathbb{E}_{P_{S,\overline{W}}}[\alpha^\top \Sigma^{-1}\alpha] - \mathbb{E}_{P_{\overline{W}}}[\mu^\top \Sigma^{-1}\mu],$$

## 3 estimation steps

1. Oracle prior (inverse) covariance matrix $\Sigma^{-1}$: Use a *bootstrap* technique over the dataset distribution $P_S$.
2. Posterior and prior means $\alpha$ and $\mu$ for given $\overline{W}$ and $S$.
3. Expectation over $P_{S,\overline{W}}$: Monte-Carlo estimation methods naturally come to mind, but they rely on the generation of many i.i.d. samples from $P_{S,\overline{W}} = P_S P_{\overline{W}}$, which is not an option.

# Experiments: Estimation of Generalization Error Bound for FL (1/2)

## Experimental setup

- Dataset: CIFAR-10 with $K = 16$.
- Model: ResNet-56 with Adam optimizer
- Hyperpararamers:
  - Client batch size: 128
  - Learning rate: 1e-3
  - Epochs: 100

## Interpretation

Computed (estimated) bound follows the behavior of the generalization error.

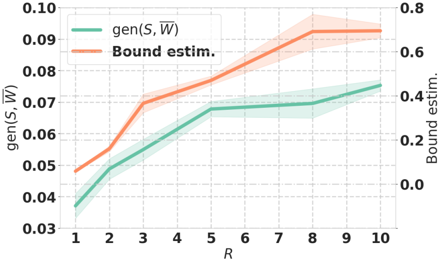



Figure: Generalization error & Computed bound vs. $R$

# Experiments: Estimation of Generalization Error Bound for FL (2/2)

## Experimental setup 2 (Fig. (b))

- Dataset: MNIST with $K = 16$.
- Model: 2-layer MLP (784-256-10) with Adam optimizer
- Hyperpararamers:
  - Client batch size: 128
  - Learning rate: 0.1
  - Epochs: 100

## Interpretation

Computed (estimated) bound follows the behavior of the generalization error.

See []

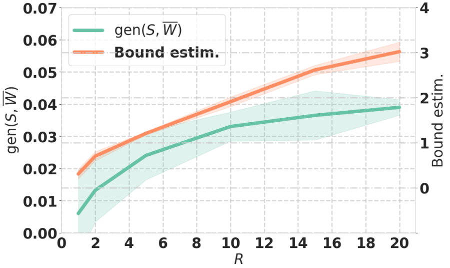



Figure: Generalization error & Computed bound vs. $R$

# Summary

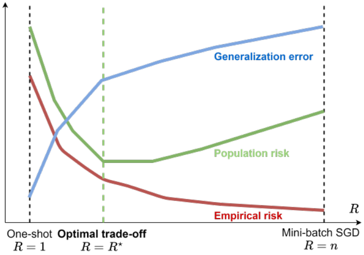
Problem: Generalization error of Federated Learning algorithms, beyond the one-shot case.

## Results

1. General tail bounds and in-expectation bounds on generalization error:
   - Novel in their kind.
   - Depend explicitly on number of communication rounds $R$, number of clients $K$ and local datasets size $n$.
   - Bounds decrease as number of communication rounds increases for Federated SVM.
   - Two different proof frameworks: PAC-Bayes and rate-distortion theory.
   - Method for estimating one generalization bound.

# Summary

Problem: Generalization error of Federated Learning algorithms, beyond the one-shot case.




## Results

1. General tail bounds and in-expectation bounds on generalization error:
   - Novel in their kind.
   - Depend explicitly on number of communication rounds $R$, number of clients $K$ and local datasets size $n$.
   - Bounds decrease as number of communication rounds increases for Federated SVM.
   - Two different proof frameworks: PAC-Bayes and rate-distortion theory.
   - Method for estimating one generalization bound.
2. Experimental observations:
   - Verification of theoretical findings for Federated SVM.
   - Similar observations for Convolutional Neural Networks (ResNet).
   - Observed that population risk has a minimizer $R^*$ much smaller than the maximum number of communication rounds.

## Implications

- Less bandwidth usage
- Better learning performance
- Smaller complexity

# Bibliography I

**For more information on the presented results, see:**

[SCZW24] **Lessons from Generalization Error Analysis of Federated Learning : You May Communicate Less Often!**
Sefidgaran M., Chor R., Zaidi A. and Wan Y., Accepted at *ICML 2024*
[CSZ23] **More Communication Does Not Result in Smaller Generalization Error in Federated Learning**
Chor R., Sefidgaran M. and Zaidi A., *ISIT 2023*
[SCZ22] **Rate-Distortion Theoretic Bounds on Generalization Error for Distributed Learning**
Sefidgaran M., Chor R. and Zaidi A., *NeurIPS 2022*

# Bibliography II


**General Distributed Learning**

- **A survey on distributed machine learning**, Verbraeken J. *et al.*, 2020
- **Communication-efficient learning of deep networks from decentralized data**, McMahan B.H. *et al.*, 2017
- **Advances and open problems in federated learning**, Kairouz P. *et al.*, 2019

**Privacy related**

- **A hybrid approach to privacy-preserving federated learning**, Truex S. *et al.*, 2019
- **Federated learning with differential privacy: Algorithms and performance analysis**, Wei K. *et al.*, 2020
- **A survey on security and privacy of federated learning**, Mothukuri V. *et al.*, 2021

**Computational power related**

- **Parallelized stochastic gradient descent**, Zinkevich M. *et al.*, 2010
- **Power allocation for wireless Federated Learning using graph neural networks**, 2021
- **Energy Efficient Federated Learning Over Wireless Communication Networks**, Yang Z. *et al.*, 2019

**Optimization results**

- **Communication-efficient learning of deep networks from decentralized data**, McMahan B.H. *et al.*, 2017
- **Local sgd converges fast and communicates little**, Stich S. *et al.*, 2019
- **Local sgd with periodic averaging: Tighter analysis and adaptive synchronization**, Haddadpour F. *et al.*, 2019

**Experimental results on generalization**

- **Communication-efficient learning of deep networks from decentralized data**, McMahan B.H. *et al.*, 2017
- **Why (and when) does local SGD generalize better than SGD?**, Gu X. *et al.*, 2023

# Statistical Learning through Examples

Statistical learning aims at solving a variety of problems using collected **data**.

# Statistical Learning through Examples

Statistical learning aims at solving a variety of problems using collected **data**. E.g.

1. Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack (and/or when this might happen). The prediction is to be based on demographic, diet and clinical measurements for that patient.

# Statistical Learning through Examples

Statistical learning aims at solving a variety of problems using collected **data**. E.g.

1. Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack (and/or when this might happen). The prediction is to be based on demographic, diet and clinical measurements for that patient.
2. Identify a handwritten number from a digitized image. This needs considering each pixel of the image.

# Statistical Learning through Examples

Statistical learning aims at solving a variety of problems using collected **data**. E.g.

1. Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack (and/or when this might happen). The prediction is to be based on demographic, diet and clinical measurements for that patient.
2. Identify a handwritten number from a digitized image. This needs considering each pixel of the image.
3. Classify an email as a spam based e.g. on the occurrence of specific keywords in the email body.

# Statistical Learning through Examples

Statistical learning aims at solving a variety of problems using collected **data**. E.g.

1. Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack (and/or when this might happen). The prediction is to be based on demographic, diet and clinical measurements for that patient.
2. Identify a handwritten number from a digitized image. This needs considering each pixel of the image.
3. Classify an email as a spam based e.g. on the occurrence of specific keywords in the email body.

Such problems are said to be **supervised** because there is a target variable $Y$, linked to some variables $X$ called features.

$$Y = f(X) + \varepsilon$$

where $f$ is the **target mapping**, $\varepsilon$ is some noise.

# Why Estimate $f$?

- Determine statistical correlations between the features.
- Determine statistical correlations between $X$ and $Y$ *i.e.*, understand which components of $X$ are helpful to explain $Y$.
- Accurately predict values of $Y$ given any features values $X$.

## Statistical Models

$f$ can be any mapping from $\mathcal{X}$ into $\mathcal{Y}$, which renders the estimation unaffordable. Statistical models restrict the considered mappings to a certain **hypothesis class** $\mathcal{H}$.

- **Statistical model:** $(\mathcal{Z}, \mathcal{H})$ where $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ is the data space and $\mathcal{H} = \{g : \mathcal{X} \to \mathcal{Y}\}$ is the hypothesis class/space.

# Statistical Models

$f$ can be any mapping from $\mathcal{X}$ into $\mathcal{Y}$, which renders the estimation unaffordable. Statistical models restrict the considered mappings to a certain **hypothesis class** $\mathcal{H}$.

- **Statistical model:** $(\mathcal{Z}, \mathcal{H})$ where $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ is the data space and $\mathcal{H} = \{g : \mathcal{X} \to \mathcal{Y}\}$ is the hypothesis class/space.
- **Parametric model:** When the hypothesis functions of the family $\mathcal{H}$ are entirely determined by parameters $W \in \mathcal{W}$ i.e. $\mathcal{H} = \{g \equiv g_W : \mathcal{X} \to \mathcal{Y} |\ W \in \mathcal{W}\}$.

# Statistical Models

$f$ can be any mapping from $\mathcal{X}$ into $\mathcal{Y}$, which renders the estimation unaffordable. Statistical models restrict the considered mappings to a certain **hypothesis class** $\mathcal{H}$.

- **Statistical model:** $(\mathcal{Z}, \mathcal{H})$ where $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ is the data space and $\mathcal{H} = \{g : \mathcal{X} \to \mathcal{Y}\}$ is the hypothesis class/space.
- **Parametric model:** When the hypothesis functions of the family $\mathcal{H}$ are entirely determined by parameters $W \in \mathcal{W}$ i.e. $\mathcal{H} = \{g \equiv g_W : \mathcal{X} \to \mathcal{Y} |\ W \in \mathcal{W}\}$.
- **Example (Linear regression):** $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \mathbb{R}$, and $\mathcal{H}$ is the class of linear functions i.e. $\forall w \in \mathcal{W}, g_w(x) = w^T x,\ x \in \mathcal{X}$.

# Statistical Models

$f$ can be any mapping from $\mathcal{X}$ into $\mathcal{Y}$, which renders the estimation unaffordable. Statistical models restrict the considered mappings to a certain **hypothesis class** $\mathcal{H}$.

- **Statistical model:** $(\mathcal{Z}, \mathcal{H})$ where $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ is the data space and $\mathcal{H} = \{g : \mathcal{X} \to \mathcal{Y}\}$ is the hypothesis class/space.
- **Parametric model:** When the hypothesis functions of the family $\mathcal{H}$ are entirely determined by parameters $W \in \mathcal{W}$ i.e. $\mathcal{H} = \{g \equiv g_W : \mathcal{X} \to \mathcal{Y} | \ W \in \mathcal{W}\}$.
- **Example (Linear regression):** $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \mathbb{R}$, and $\mathcal{H}$ is the class of linear functions i.e. $\forall w \in \mathcal{W}, g_w(x) = w^T x, \ x \in \mathcal{X}$.

In the following, we will consider only parametric models for ease of presentation. The parameters $W$ will be referred as hypothesis and $\mathcal{W}$ as hypothesis space.

# Model Evaluation

Once a model has been chosen, one needs to compute a hypothesis $W$, using $X$, which results in a good estimation of the target $Y$.

# Model Evaluation

Once a model has been chosen, one needs to compute a hypothesis $W$, using $X$, which results in a good estimation of the target $Y$.

- **Loss function:** A function $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}_+$ measuring how well a hypothesis $w \in \mathcal{W}$ predicts the target $y$ based on features $x$. Example (least squares regression): $\forall z = (x, y) \in \mathcal{Z},\ \forall w \in \mathcal{W},\ \ell(z, w) = (w^T x - y)^2$

# Model Evaluation

Once a model has been chosen, one needs to compute a hypothesis $W$, using $X$, which results in a good estimation of the target $Y$.

- **Loss function:** A function $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}_+$ measuring how well a hypothesis $w \in \mathcal{W}$ predicts the target $y$ based on features $x$. Example (least squares regression): $\forall z = (x, y) \in \mathcal{Z},\ \forall w \in \mathcal{W},\ \ell(z, w) = (w^T x - y)^2$
- **Population risk:** For a given hypothesis $w \in \mathcal{W}$,

$$\mathcal{L}(w) := \mathbb{E}_{Z \sim \mu}[\ell(Z, w)].$$

- **Objective:** Find the hypothesis $w^\star \in \mathcal{W}$ such that

$$w^\star \in \operatorname*{argmin}_{w \in \mathcal{W}} \mathcal{L}(w).$$

# Model Evaluation

Once a model has been chosen, one needs to compute a hypothesis $W$, using $X$, which results in a good estimation of the target $Y$.

- **Loss function:** A function $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}_+$ measuring how well a hypothesis $w \in \mathcal{W}$ predicts the target $y$ based on features $x$. Example (least squares regression): $\forall z = (x, y) \in \mathcal{Z},\ \forall w \in \mathcal{W},\ \ell(z, w) = (w^T x - y)^2$
- **Population risk:** For a given hypothesis $w \in \mathcal{W}$,

$$\mathcal{L}(w) := \mathbb{E}_{Z\sim\mu}[\ell(Z, w)].$$

- **Objective:** Find the hypothesis $w^\star \in \mathcal{W}$ such that

$$w^\star \in \underset{w\in\mathcal{W}}{\operatorname{argmin}}\ \mathcal{L}(w).$$

**Problem**

Data distribution $\mu$ unknown hence $\mathcal{L}(w)$ can not be computed!

# Empirical Risk Minimization (ERM)

- **Training dataset:** Let $S = \{Z_1, \ldots, Z_n\}$ be $n$ independent random variables distributed according to $\mu$, and independent of $Z = (X, Y)$.

# Empirical Risk Minimization (ERM)

- **Training dataset:** Let $S = \{Z_1, \ldots, Z_n\}$ be $n$ independent random variables distributed according to $\mu$, and independent of $Z = (X, Y)$.
- **Learning algorithm:** A (possibly stochastic) mapping $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ that inputs a training dataset $S$ and picks a hypothesis $W$.

# Empirical Risk Minimization (ERM)

- **Training dataset:** Let $S = \{Z_1, \dots, Z_n\}$ be $n$ independent random variables distributed according to $\mu$, and independent of $Z = (X, Y)$.
- **Learning algorithm:** A (possibly stochastic) mapping $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ that inputs a training dataset $S$ and picks a hypothesis $W$.
- $S$ is used by $\mathcal{A}$ to estimate the target function $f$ by finding the hypothesis $w$ minimizing the **empirical risk**:

$$\hat{\mathcal{L}}(s, w) := \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, w).$$

- It is an estimator of the population risk.

# Empirical Risk Minimization (ERM)

- **Training dataset:** Let $S = \{Z_1, \dots, Z_n\}$ be $n$ independent random variables distributed according to $\mu$, and independent of $Z = (X, Y)$.
- **Learning algorithm:** A (possibly stochastic) mapping $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ that inputs a training dataset $S$ and picks a hypothesis $W$.
- $S$ is used by $\mathcal{A}$ to estimate the target function $f$ by finding the hypothesis $w$ minimizing the **empirical risk**:

$$\hat{\mathcal{L}}(s, w) := \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, w).$$

- It is an estimator of the population risk.
- **Problem:** How to measure the quality of the estimation?

# Summary of the Statistical Learning Framework

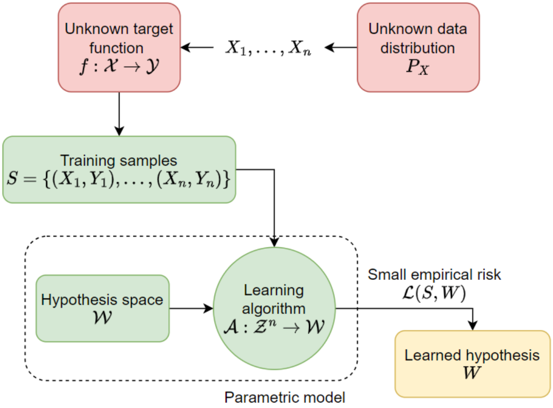



Figure: Illustration of statistical learning framework

# Generalization of a hypothesis

- Difference of performance of a given hypothesis $W$ for predicting any target value $Y$ given $X$, that is generated by the distribution $\mu$ ("ground truth" performance), as compared to the performance on a training dataset (empirical performance).

# Generalization of a hypothesis

- Difference of performance of a given hypothesis $W$ for predicting any target value $Y$ given $X$, that is generated by the distribution $\mu$ ("ground truth" performance), as compared to the performance on a training dataset (empirical performance).
- **Generalization Error:** Most common way to evaluate the generalization of a hypothesis,

$$\text{gen}(s, w) \coloneqq \mathcal{L}(w) - \hat{\mathcal{L}}(s, w)$$

# Generalization of a hypothesis

- Difference of performance of a given hypothesis $W$ for predicting any target value $Y$ given $X$, that is generated by the distribution $\mu$ ("ground truth" performance), as compared to the performance on a training dataset (empirical performance).
- **Generalization Error:** Most common way to evaluate the generalization of a hypothesis,

$$\mathrm{gen}(s, w) \coloneqq \mathcal{L}(w) - \hat{\mathcal{L}}(s, w)$$

- **Overfitting** can happen i.e. the hypothesis is really performant on $S$ but has poor accuracy on new data $Z_{n+1}, Z_{n+2}, \cdots \sim \mu$, resulting in large generalization error.

# Generalization of a hypothesis

- Difference of performance of a given hypothesis $W$ for predicting any target value $Y$ given $X$, that is generated by the distribution $\mu$ ("ground truth" performance), as compared to the performance on a training dataset (empirical performance).
- **Generalization Error:** Most common way to evaluate the generalization of a hypothesis,

$$\mathrm{gen}(s, w) \coloneqq \mathcal{L}(w) - \hat{\mathcal{L}}(s, w)$$

- **Overfitting** can happen i.e. the hypothesis is really performant on $S$ but has poor accuracy on new data $Z_{n+1}, Z_{n+2}, \cdots \sim \mu$, resulting in large generalization error.

**Problem with generalization error**

Exact analysis out of reach - resort to bounds:

- Tail bounds
- In-expectation bounds