Vertical Federated Kernel Learning

Heng Huang

Department of Electrical & Computer Engineering, University of Pittsburgh, PA
JD Finance America Corporation, Mountain View, CA

Collaborate with Bin Gu, Zhiyuan Dang, Xiang Li

RSEML AAAI2021
Motivations

Vertically Partitioned Federated Kernel Learning
- Problem Statement
- Brief Review of Doubly Stochastic Kernel Methods
- Federated Doubly Stochastic Kernel Learning Algorithm

Algorithm Analysis
- Convergence Analysis
- Security Analysis
- Complexity Analysis

Validations

Summary
Motivations

Widespread existence of Vertically Partitioned Data

Samples
- Data from digital finance company
- Data from bank
- Data from E-commerce company

Features
- Loan
  - $6,120
  - $11,560
  - $89,478
- Repayment
  - $6,120
  - $10,000
  - $0

- Monthly deposit
  - $10,100
  - $0
- Account balance
  - $78,456
  - $11,120
  - $561

- Online Shopping
  - $14,256
  - $4,842
  - $189

A loan application to the digital finance company.

Heng Huang
VFKL
RSEML AAAI2021
Direct access to the data in other providers or sharing of the data may be prohibited due to legal and commercial reasons.
Challenges

Existing Vertically Federated Learning Algorithms

- Cooperative statistical analysis
- Linear regression
- Association rule-mining
- \(K\)-means clustering
- Logistic regression
- XGBoost

Weakness of existing vertically federated learning algorithms

- Assume the models are implicitly linearly separable, i.e.,
  \[ f(x) = g \circ h(x) = g \circ \sum_{\ell=1}^{q} h^\ell(w_{G_\ell}, x_{G_\ell}). \]
- Kernel models usually take the form of
  \[ f(x) = \sum_{i}^{N} \alpha_i K(x_i, x) \]
  which do not satisfy the assumption of implicitly linear separability.

It is still an open question to train the vertically partitioned data efficiently and scalably by kernel methods while keeping data privacy.
Challenges

Existing Vertically Federated Learning Algorithms
- Cooperative statistical analysis
- Linear regression
- Association rule-mining
- $K$-means clustering
- Logistic regression
- XGBoost

Weakness of existing vertically federated learning algorithms
- Assume the models are implicitly linearly separable, i.e.,
  \[ f(x) = g \circ h(x) = g \circ \sum_{\ell=1}^{q} h^\ell(w_G, x_G). \]
- Kernel models usually take the form of \( f(x) = \sum_{i}^{N} \alpha_i K(x_i, x) \) which do not satisfy the assumption of implicitly linear separability.

It is still an open question to train the vertically partitioned data efficiently and scalably by kernel methods while keeping data privacy.
A training set \( S = \{(x_i, y_i)\}_{i=1}^{N} \), where \( x_i \in \mathbb{R}^d \) and \( y_i \in \{+1, -1\} \) for binary classification or \( y_i \in \mathbb{R} \) for regression.

- \( x = [x_{G_1}, x_{G_2}, \ldots, x_{G_q}] \), and \( x_{G_\ell} \in \mathbb{R}^{d_\ell} \) is stored on the \( \ell \)-th worker and \( \sum_{\ell=1}^{q} d_{\ell} = d \).
- Let \( L(u, y) \) be a scalar loss function which is convex with respect to \( u \in \mathbb{R} \).
- A positive definite kernel function \( K(x', x) \) and the associated reproducing kernel Hilbert spaces (RKHS) \( \mathcal{H} \). We have \( \langle f(\cdot), K(x, \cdot) \rangle_{\mathcal{H}} = f(x) \).

**Problem Statement**

- A kernel method tries to find a function \( f \in \mathcal{H} \)

\[
\arg \min_{f \in \mathcal{H}} \mathcal{R}(f) = \mathbb{E}_{(x, y) \in S} L(f(x), y) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2
\]  

(1)

- \( \lambda > 0 \) is a regularization parameter.
Active and Passive workers

According to whether the label is included in a worker, we divide the workers into two types:

- active worker, is the data provider who holds the label of a sample.
- passive worker, only has the input of a sample.

The active worker would be a dominating server in federated learning, while passive workers play the role of clients.

Goal of Vertically Partitioned Federated Kernel Learning

Make active workers to cooperate with passive workers to solve the nonlinear learning problem (1) on the vertically partitioned data \( \{D^\ell\}_{\ell=1}^q \) while keeping the vertically partitioned data private.
Random Feature Approximation

Theorem 1 [Rudin 1962]

A continuous, real-valued, symmetric and shift-invariant function \( K(x, x') = K(x - x') \) on \( \mathbb{R}^d \) is a positive definite kernel if and only if there is a finite non-negative measure \( \mathbb{P}(\omega) \) on \( \mathbb{R}^d \), such that \( K(x - x') = \int_{\mathbb{R}^d} e^{i\omega^T (x-x')} d\mathbb{P}(\omega) = \int_{\mathbb{R}^d \times [0,2\pi]} 2 \cos(\omega^T x + b) \cos(\omega^T x' + b) d(\mathbb{P}(\omega) \times \mathbb{P}(b)) \), where \( \mathbb{P}(b) \) is a uniform distribution on \([0, 2\pi]\), and \( \phi_\omega(x) = \sqrt{2} \cos(\omega^T x + b) \).

Random Feature Approximation

\[
K(x, x') \approx \frac{1}{m} \sum_{i=1}^{m} \phi_{\omega_i}(x)\phi_{\omega_i}(x')
\]  

- \( m \) is the number of random features
- \( \omega_i \) are drawn from \( \mathbb{P}(\omega) \).
- Gaussian RBF kernel \( K(x, x') = \exp(-||x - x'||^2 / 2\sigma^2) \), \( \mathbb{P}(\omega) \) is a Gaussian distribution with density proportional to \( \exp(-\sigma^2 ||\omega||^2 / 2) \).
Doubly Stochastic Kernel Methods

Doubly Stochastic Gradient

- the doubly stochastic gradient of loss function $L(f(x_i), y_i)$ w.r.t. the sampled instance $(x, y)$ and the random direction $\omega$ is
  
  $$\zeta(\cdot) = L'(f(x_i), y_i) \phi_\omega(x_i) \phi_\omega(\cdot).$$

- the stochastic gradient of $R(f)$ can be formulated as follows.
  
  $$\hat{\zeta}(\cdot) = \zeta(\cdot) + \lambda f(\cdot) = L'(f(x_i), y_i) \phi_{\omega_i}(x_i) \phi_{\omega_i}(\cdot) + \lambda f(\cdot)$$

- The doubly stochastic gradient is unbiased: $E_{(x,y)} E_{\omega} \hat{\zeta}(\cdot) = \nabla R(f)$

Updating rule

Given stepsize $\gamma_t$, let $f_1(\cdot) = 0$, the updating rule is

$$f_{t+1}(\cdot) = f_t(\cdot) - \gamma_t (\zeta(\cdot) + \lambda f(\cdot)) = \sum_{i=1}^{t} -\gamma_i \prod_{j=i+1}^{t} (1 - \gamma_j \lambda) \zeta_i(\cdot)$$

$$= \sum_{i=1}^{t} -\gamma_i \prod_{j=i+1}^{t} (1 - \gamma_j \lambda) L'(f(x_i), y_i) \phi_{\omega_i}(x_i) \phi_{\omega_i}(\cdot)$$

$$= \alpha'_i$$
The main idea behind VFKL's parallelism is to vertically divide the computation of the random features $\phi_\omega(x) = \sqrt{2} \cos(\omega^T x + b)$ and $f(x)$. 
Data Privacy

Divide the computation of $\phi_{\omega_i}(x_i) = \sqrt{2} \cos(\omega_i^T x_i + b)$ to avoid transferring the local data $(x_i)_{g_\ell}$ to other workers. i.e.,

1. send a random seed to the $\ell$-th worker.
2. generate the random direction $\omega_i$ uniquely according to the random seed.
3. locally compute $(\omega_i)^T_{g_\ell} (x_i)_{g_\ell} + b$ which avoids directly transferring the local data.

Modal Privacy

1. The model coefficients $\alpha_i$ are stored in different workers separately and privately.
2. To compute $f(x)$, we locally compute $f^\ell(x) = \sum_{i \in \Lambda^\ell} \alpha_i \phi_{\omega_i}(x)$ and transfer it to other worker, and $f(x)$ can be reconstructed by summing over all the $f^\ell(x)$.
System Structure

Tree-Structured Communication

(a) Tree structure $T_1$ on workers $\{1, \ldots, 4\}$

(b) Tree structure $T_2$ on workers $\{1, \ldots, 3\}$

Figure: Illustration of tree-structured communication with two totally different tree structures $T_1$ and $T_2$.

Two totally different tree structures are used in the computation of $\phi_{\omega_i}(x_i) = \sqrt{2} \cos(\omega_i^T x_i + b)$ to protect the data privacy (see Alg. 3).
VFKL Algorithm 1: Main Procedure

**Input:** \( \mathcal{P}(\omega) \), local normalized data \( D^\ell \), regularization parameter \( \lambda \), constant learning rate \( \gamma \).

1: **keep doing in parallel**

2: Pick up an instance \((x_i)_\ell\) from the local data \( D^\ell \) with index \( i \).

3: Send \( i \) to other workers using a reverse-order tree structure \( T_0 \).

4: Sample \( \omega_i \sim \mathcal{P}(\omega) \) with the random seed \( i \) for all workers.

5: Use Algorithm 3 to compute \( \omega_i^T x_i + b \) and locally save it.

6: Compute \( f^{\ell'}(x_i) \) for \( \ell' = 1, \ldots, q \) by calling Algorithm 2.

7: Use tree-structured communication scheme based on \( T_0 \) to compute \( f(x_i) = \sum_{\ell=1}^{q} f^{\ell}(x_i) \).

8: Compute \( \phi_{\omega_i}(x_i) \) according to \( \omega_i^T x_i + b \).

9: Compute \( \alpha_i = -\gamma \left( L'(f(x_i), y_i) \phi_{\omega_i}(x_i) \right) \) and locally save \( \alpha_i \).

10: Update \( \alpha_j = (1 - \gamma \lambda) \alpha_j \) for all previous \( j \) in the \( \ell \)-th worker and other workers.

11: **end parallel loop**

**Output:** \( \alpha_\Lambda^\ell \).
VFKL Algorithm 2: Computing $f^\ell(x)$

**Input:** $\mathbb{P}(\omega)$, $\alpha_{\Lambda \ell}$, $\Lambda^\ell$, $x$.
1: Set $f^\ell(x) = 0$.
2: for each $i \in \Lambda^\ell$ do
3: Sample $\omega_i \sim \mathbb{P}(\omega)$ with the random seed $i$ for all workers.
4: Obtain $\omega_i^T x + b$ if it is locally saved, otherwise compute $\omega_i^T x + b$ by using Algorithm 3.
5: Compute $\phi_{\omega_i}(x)$ according to $\omega_i^T x + b$.
6: $f^\ell(x) = f^\ell(x) + \alpha_i \phi_{\omega_i}(x)$
7: end for

**Output:** $f^\ell(x)$
VFKL Algorithm 3: Computing $\omega_i^T x_i + b$

**Input:** $\omega_i, x_i$

{ // This loop asks multiple workers running in parallel. }

1: **for** $\hat{\ell} = 1, \ldots, q$ **do**
2: Compute $(\omega_i)_{G_{\hat{\ell}}}^T (x_i)_{G_{\hat{\ell}}}$ and randomly generate a uniform number $b_{\hat{\ell}}$ from $[0, 2\pi]$ with the seed $\sigma_{\hat{\ell}}(i)$.
3: Calculate $(\omega_i)_{G_{\hat{\ell}}}^T (x_i)_{G_{\hat{\ell}}} + b_{\hat{\ell}}$.
4: **end for**
5: Use tree-structured communication scheme based on the tree structure $T_1$ for workers $\{1, \ldots, q\}$ to compute $\xi = \sum_{\hat{\ell} = 1}^q \left((\omega_i)_{G_{\hat{\ell}}}^T (x_i)_{G_{\hat{\ell}}} + b_{\hat{\ell}}\right)$.
6: Pick up $\ell' \in \{1, \ldots, q\} - \{\ell\}$ uniformly at random.
7: Use tree-structured communication scheme based on the totally different tree structure $T_2$ for workers $\{1, \ldots, q\} - \{\ell'\}$ to compute $\overline{b}_{\ell'} = \sum_{\hat{\ell} \neq \ell'} b_{\hat{\ell}}$.

**Output:** $\xi - \overline{b}_{\ell'}$. 
Lemma 1

The output of Algorithm 3 (i.e., \( \sum_{\ell=1}^{q} \left( (\omega_i)^T g_{\hat{\ell}}(x) g_{\hat{\ell}} + b_{\hat{\ell}} \right) - b_{\ell'} \)) is equal to \( \omega_i^T x + b \), where each \( b_{\hat{\ell}} \) and \( b \) are drawn from a uniform distribution on \([0, 2\pi]\), \( b_{\ell'} = \sum_{\ell \neq \ell'} b_{\hat{\ell}} \), and \( \ell' \in \{1, \ldots, q\} - \{\ell\} \).

VFKL can produce the same doubly stochastic gradients as that of a DSG algorithm.
Convergence Analysis

Assumption 1

1. There exists an optimal solution, denoted as $f_*$, to the problem (1).
2. We have an upper bound for the derivative of $L(u, y)$ w.r.t. its 1st argument, i.e., $|L'(u, y)| < M$.
3. The loss function $L(u, y)$ and its first-order derivative are $\mathcal{L}$-Lipschitz continuous in terms of the first argument.
4. We have an upper bound $\kappa$ for the kernel value, i.e., $K(x, x') \leq \kappa$. We have an upper bound $\phi$ for random feature mapping, i.e., $|\phi_\omega(x)\phi_\omega(x')| \leq \phi$. 
Convergence Analysis

**Theorem 2**

Set $\epsilon > 0$, $\min \left\{ \frac{1}{\lambda}, \frac{\epsilon \lambda}{4M^2(\sqrt{\kappa} + \sqrt{\phi})^2} \right\} > \gamma > 0$, for Algorithm 1, with $\gamma = \frac{\epsilon \vartheta}{8\kappa B}$ for $\vartheta \in (0, 1]$, under Assumption 1, we will reach $\mathbb{E} \left[ |f_t(x) - f_*(x)|^2 \right] \leq \epsilon$ after

$$t \geq \frac{8\kappa B \log(8\kappa e_1/\epsilon)}{\vartheta \epsilon \lambda}$$

(5)

iterations, where $B = \left[ \sqrt{G_2^2 + G_1 + G_2} \right]^2$, $G_1 = \frac{2\kappa M^2}{\lambda}$, $G_2 = \frac{\kappa^{1/2}M(\sqrt{\kappa} + \sqrt{\phi})}{2\lambda^{3/2}}$ and $e_1 = \mathbb{E}[\|h_1 - f_*\|^2_{\mathcal{H}}]$.

VFKL converges to the optimal solution almost at a rate of $O(1/t)$. 
Assumption 2: Semi-honest Security

All workers will follow the protocol or algorithm to perform the correct computations. However, they may retain records of the intermediate computation results which they may use later to infer the data of other workers.

Inference Attack

An inference attack on the $\ell$-th worker is to infer a certain feature group $G$ of sample $x_i$ which belongs to other workers without directly accessing it.

Theorem 3

Under the semi-honest assumption, the VFKL algorithm can prevent inference attack.
Complexity Analysis

Computational Complexities
1. The computational complexity for one iteration of VFKL is $O(dqt)$.
2. The total computational complexity of VFKL is $O(dqt^2)$.

Communication Complexities
1. The communication cost for one iteration of VFKL is $O(qt)$.
2. The total communication cost of VFKL is $O(qt^2)$.

- $d$ is the dimension of the samples
- $q$ is the number of the workers
- $t$ is the total iteration number.
Validations

Comparison Methods

- PP-SVMV [Yu et al. (2006)] SOTA in Kernel federated learning field.
- SecureBoost [Cheng et al. (2019)] recently proposed to generalize the gradient tree-boosting algorithm to federated scenarios.
- SOTA kernel classification solvers that can access the whole data samples without the federated learning constraint: LIBSVM [Chang and Lin (2011)] and DSG [Dai et al. (2014)].
- FD-SVRG [Wan et al. (2007)], which uses a linear model to comparatively verify the accuracy of VFKL.
Validations

Datasets: eight benchmark binary classification datasets and two real-world financial datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Features</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>gisette</td>
<td>5,000</td>
<td>6,000</td>
</tr>
<tr>
<td>phishing</td>
<td>68</td>
<td>11,055</td>
</tr>
<tr>
<td>a9a</td>
<td>123</td>
<td>48,842</td>
</tr>
<tr>
<td>ijcnn1</td>
<td>22</td>
<td>49,990</td>
</tr>
<tr>
<td>cod-rna</td>
<td>8</td>
<td>59,535</td>
</tr>
<tr>
<td>w8a</td>
<td>300</td>
<td>64,700</td>
</tr>
<tr>
<td>real-sim</td>
<td>20,958</td>
<td>72,309</td>
</tr>
<tr>
<td>epsilon</td>
<td>2,000</td>
<td>400,000</td>
</tr>
<tr>
<td>defaultcredit</td>
<td>23</td>
<td>30,000</td>
</tr>
<tr>
<td>givemecredit</td>
<td>10</td>
<td>150,000</td>
</tr>
</tbody>
</table>
Validation Results

Figure: The results of binary classification above the comparison methods.
Validation Results

Figure: The results of binary classification above the comparison methods.

(a) deafultcredit

(b) givemecredit

Heng Huang
Validation Results

(a) gisette  (b) phishing  (c) a9a  (d) ijcnn1

Figure: The elapsed time of different structures on four datasets.
Validation Results

(a) phishing  
(b) a9a  
(c) cod-rna  
(d) w8a

Figure: The change of training time when increasing the number of training instances.
Figure: The boxplot of test errors of three state-of-the-art kernel methods, tree-boosting method (SecureBoost), linear method (FD-SVRG) and our VFKL.
Asynchronous Communication

Worker 1
Worker 2
Worker 3

Synchronous Communication

Worker 1
Worker 2
Worker 3

Asynchronous Communication

Heng Huang

VFKL

RSEML AAAI2021

28/30
Summary

1. Introduce a federated doubly stochastic kernel learning (i.e., VFKL) algorithm
   - Effectively handle vertically partitioned data
   - Produce a sublinear convergence rate $O(1/t)$
   - Guarantee data security under the semi-honest assumption
   - First efficient and scalable privacy-preservation federated kernel method

2. Validations
   - Confirm the effectiveness of our VFKL
   - Show the superiority of our VFKL compared with the existing SOTA kernel, tree-boosting and linear algorithm
Thank You!