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

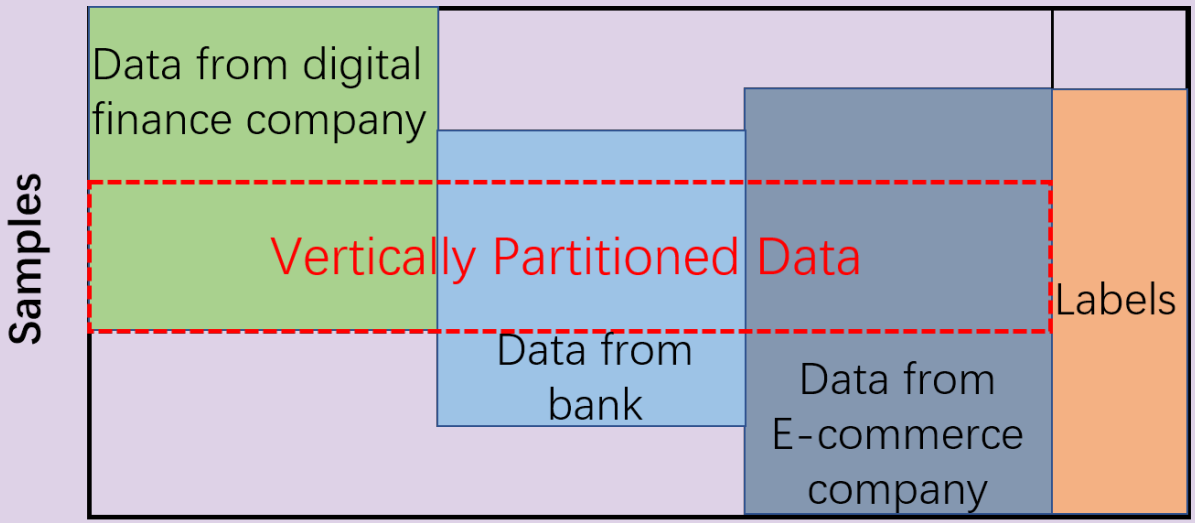
RSEML AAAI2021

Outline

- 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



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

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

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

A loan application to the digital finance company.

Motivations

General Data Protection Regulation (GDPR)

- 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., $f(x) = g \circ h(x) = g \circ \sum_{\ell=1}^{q} h^{\ell}(w_{\mathcal{G}_\ell}, x_{\mathcal{G}_\ell})$.
- 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., $f(x) = g \circ h(x) = g \circ \sum_{\ell=1}^{q} h^{\ell}(w_{\mathcal{G}_\ell}, x_{\mathcal{G}_\ell})$.
- 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.

Vertically Partitioned Federated Kernel Learning

Preliminaries

- A training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$ for binary classification or $y_i \in \mathbb{R}$ for regression.
  - $x = [x_{\mathcal{G}_1}, x_{\mathcal{G}_2}, \ldots, x_{\mathcal{G}_q}]$, and $x_{\mathcal{G}_\ell} \in \mathbb{R}^{d_\ell}$ is stored on the ℓ-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 $f \in \mathcal{H}$

$$\arg\min_{f \in \mathcal{H}} \mathcal{R}(f) = \mathbb{E}_{(x,y)\in\mathcal{S}} L(f(x), y) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \tag{1}$$

- $\lambda > 0$ is a regularization parameter.

Vertically Partitioned Federated Kernel Learning

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

$$K(x, x') \approx \frac{1}{m} \sum_{i=1}^{m} \phi_{\omega_i}(x) \phi_{\omega_i}(x') \tag{2}$$

- m is the number of random features
- ω_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 ω is $\zeta(\cdot) = L'(f(x_i), y_i)\phi_\omega(x_i)\phi_\omega(\cdot)$.
- the stochastic gradient of $\mathcal{R}(f)$ can be formulated as follows.

$$\widehat{\zeta}(\cdot) = \zeta(\cdot) + \lambda f(\cdot) = L'(f(x_i), y_i)\phi_{\omega_i}(x_i)\phi_{\omega_i}(\cdot) + \lambda f(\cdot) \tag{3}$$

  - The doubly stochastic gradient is unbiased: $\mathbb{E}_{(x,y)}\mathbb{E}_\omega \widehat{\zeta}(\cdot) = \nabla\mathcal{R}(f)$

Updating rule

Given stepsize γ_t, let $f_1(\cdot) = \mathbf{0}$, the updating rule is

$$f_{t+1}(\cdot) = f_t(\cdot) - \gamma_t\left(\zeta(\cdot) + \lambda f(\cdot)\right) = \sum_{i=1}^{t} -\gamma_i \prod_{j=i+1}^{t}(1-\gamma_j\lambda)\zeta_i(\cdot) \tag{4}$$

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

Federated Doubly Stochastic Kernel Learning

System Structure

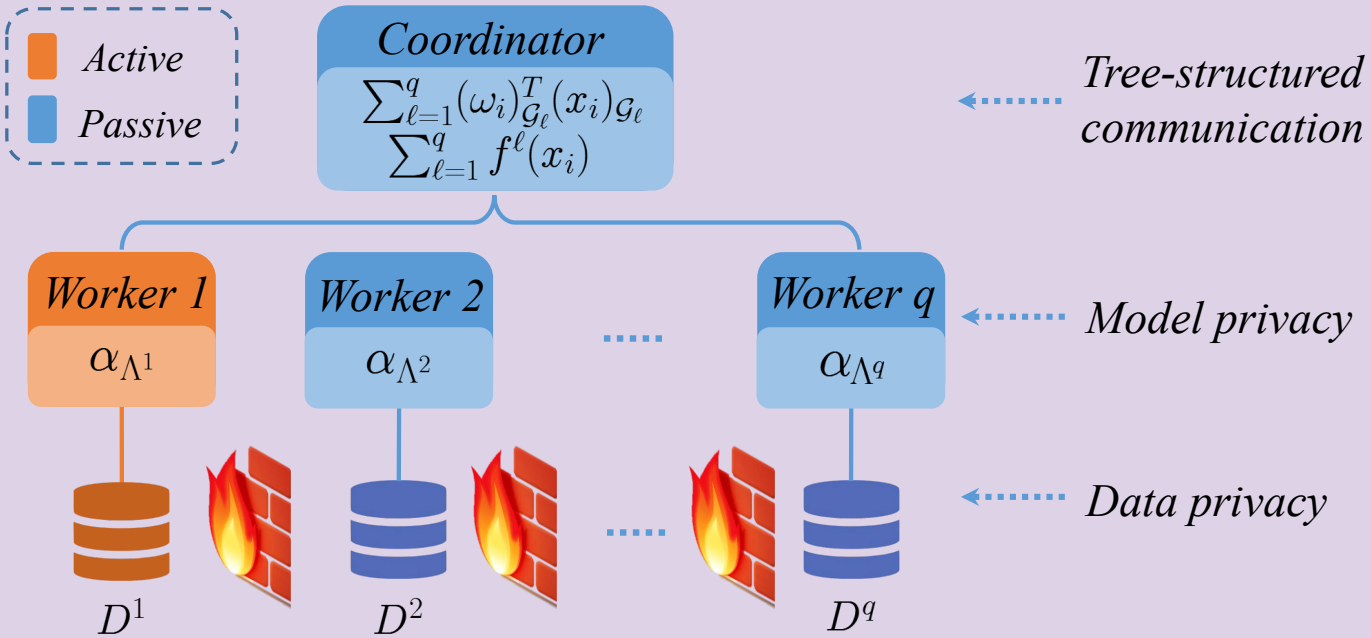


Figure: System structure of VFKL.

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 $f(x)$.

System Structure

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 $(x_i)_{\mathcal{G}_\ell}$ to other workers. i.e.,

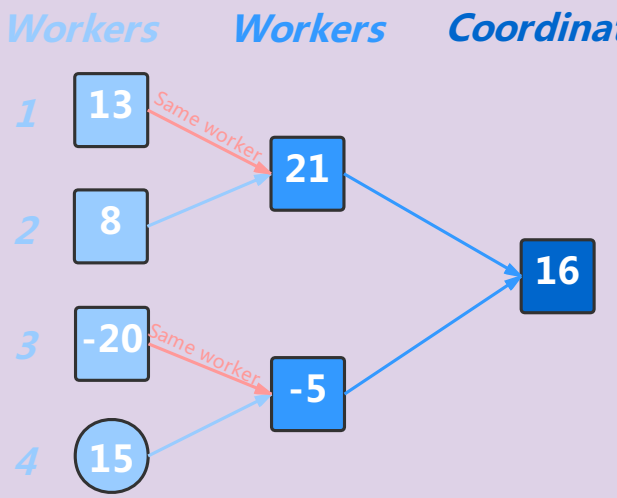
1. send a random seed to the ℓ-th worker.
2. generate the random direction ω_i uniquely according to the random seed.
3. locally compute $(\omega_i)_{\mathcal{G}_\ell}^T (x_i)_{\mathcal{G}_\ell} + b$ which avoids directly transferring the local data.

Modal Privacy

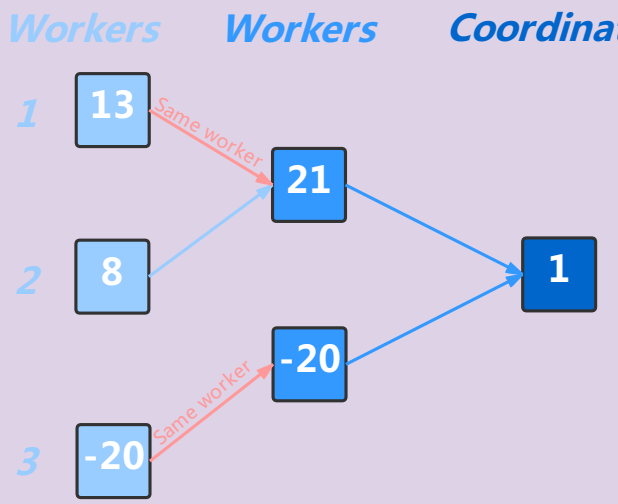
1. The model coefficients α_i are stored in different workers separately and privately.
2. To compute $f(x)$, we locally compute $f^\ell(x) = \sum_{i \in \Lambda^\ell} \alpha_i \phi_{\omega_i}(x)$ and transfer it to other worker, and $f(x)$ can be reconstructed by summing over all the $f^\ell(x)$.

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: $\mathbb{P}(\omega)$, local normalized data D^{ℓ}, regularization parameter λ, constant learning rate γ.

1: **keep doing in parallel**

2: Pick up an instance $(x_i)_{\mathcal{G}_\ell}$ from the local data D^{ℓ} with index i.

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

4: Sample $\omega_i \sim \mathbb{P}(\omega)$ 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 α_i.

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

11: **end parallel loop**

Output: $\alpha_{\Lambda^{\ell}}$.

VFKL Algorithm 2: Computing $f^{\ell}(x)$

Input: $\mathbb{P}(\omega)$, $\alpha_{\Lambda^{\ell}}$, Λ^{ℓ}, 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: ω_i, x_i

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

1: **for** $\hat{\ell} = 1, \ldots, q$ **do**

2: Compute $(\omega_i)^T_{\mathcal{G}_{\hat{\ell}}}(x_i)_{\mathcal{G}_{\hat{\ell}}}$ and randomly generate a uniform number $b^{\hat{\ell}}$ from $[0, 2\pi]$ with the seed $\sigma^{\hat{\ell}}(i)$.

3: Calculate $(\omega_i)^T_{\mathcal{G}_{\hat{\ell}}}(x_i)_{\mathcal{G}_{\hat{\ell}}} + b^{\hat{\ell}}$.

4: **end for**

5: Use tree-structured communication scheme based on the tree structure T_1 for workers $\{1, \ldots, q\}$ to compute $\xi = \sum_{\hat{\ell}=1}^{q} \left((\omega_i)^T_{\mathcal{G}_{\hat{\ell}}}(x_i)_{\mathcal{G}_{\hat{\ell}}} + b^{\hat{\ell}} \right)$.

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 $\bar{b}^{\ell'} = \sum_{\hat{\ell} \neq \ell'} b^{\hat{\ell}}$.

Output: $\xi - \bar{b}^{\ell'}$.

Algorithm Analysis

Lemma 1

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

1. 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 κ for the kernel value, *i.e.*, $K(x, x') \leq \kappa$. We have an upper bound ϕ for random feature mapping, *i.e.*, $|\phi_\omega(x)\phi_\omega(x')| \leq \phi$.

Convergence Analysis

Theorem 2

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

$$t \geq \frac{8\kappa B \log(8\kappa e_1/\epsilon)}{\vartheta\epsilon\lambda} \tag{5}$$

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 $e_1 = \mathbb{E}[\|h_1 - f_*\|_{\mathcal{H}}^2]$.

VFKL converges to the optimal solution almost at a rate of $\mathcal{O}(1/t)$.

Security Analysis

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 ℓ-th worker is to infer a certain feature group $\mathcal{G}$ 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 $\mathcal{O}(dqt)$.
2. The total computational complexity of VFKL is $\mathcal{O}(dqt^2)$.

Communication Complexities

1. The communication cost for one iteration of VFKL is $\mathcal{O}(qt)$
2. The total communication cost of VFKL is $\mathcal{O}(qt^2)$.

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

Validations

Comparison Methods

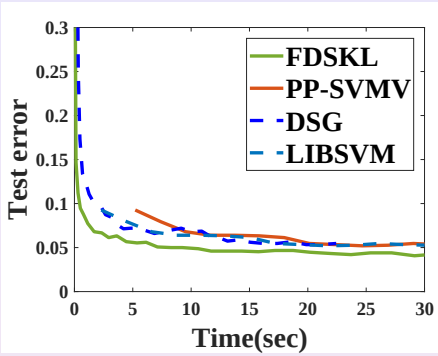
- PP-SVMV [Yu et al.(2006)] SOTA in Kernel federated learning feild.
- SecureBoost [Cheng et al.(2019)] recently proposed to generalize the gradient tree-boosting algorithm to federated scenarios.
- SOTA kernel classification solvers that can access the whole data samples without the federated learning 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

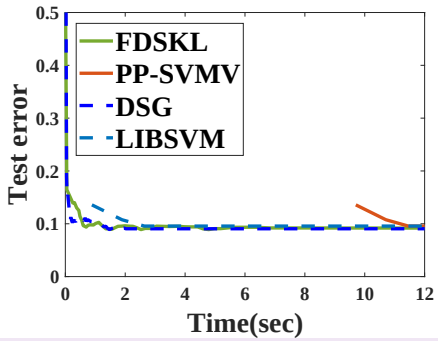
Datasets: eight benchmark binary classification datasets and two real-world financial datasets.

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

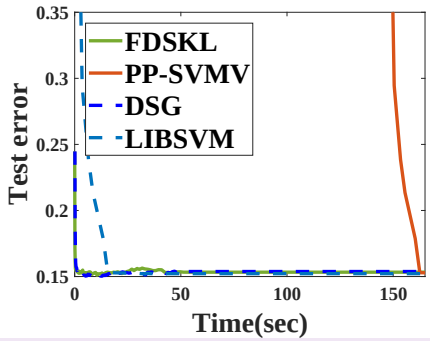
Validation Results



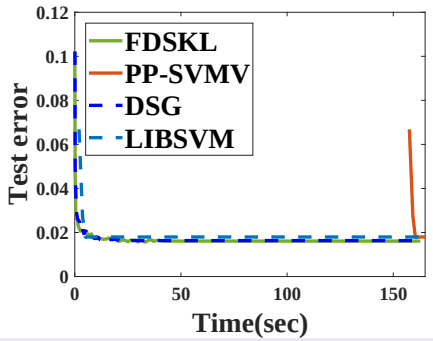
(a) gisette



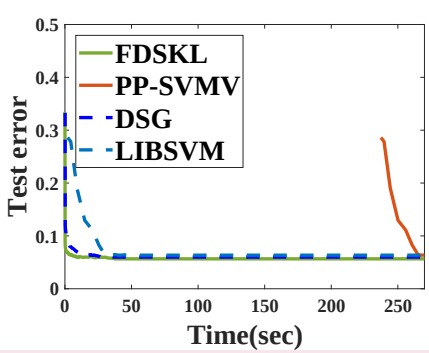
(b) phishing



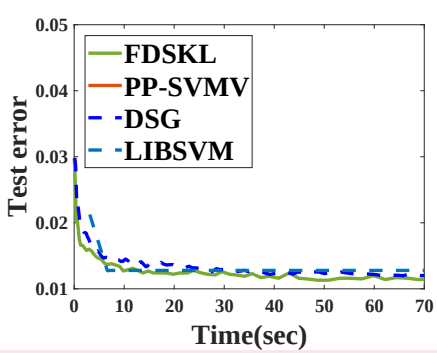
(c) a9a



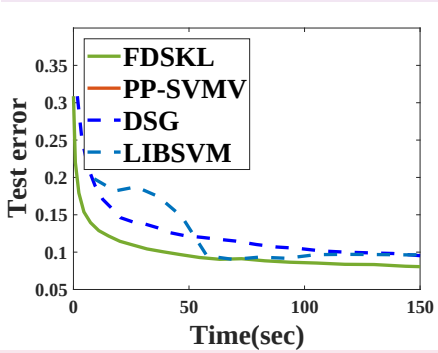
(d) ijcnn1



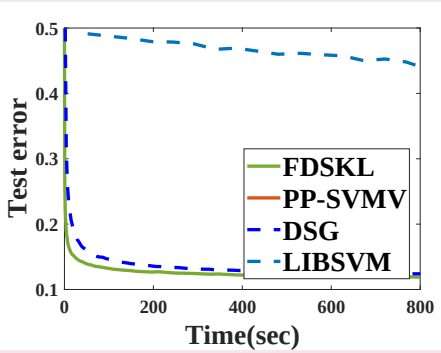
(e) cod-rna



(f) w8a



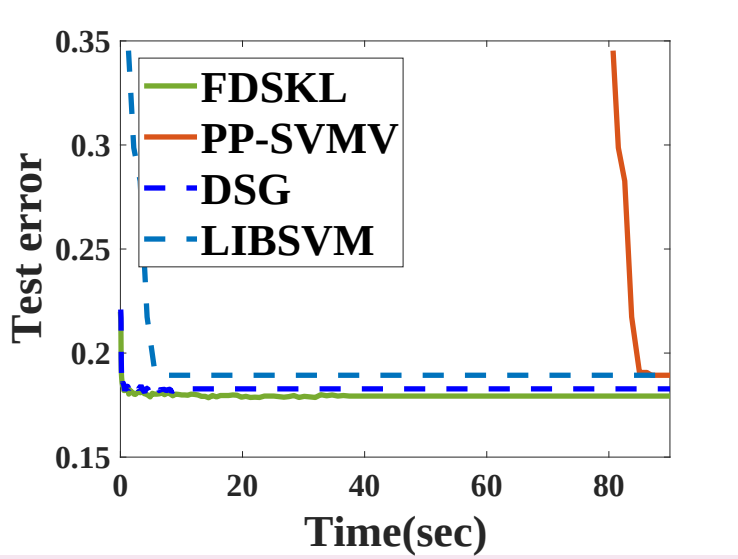
(g) real-sim

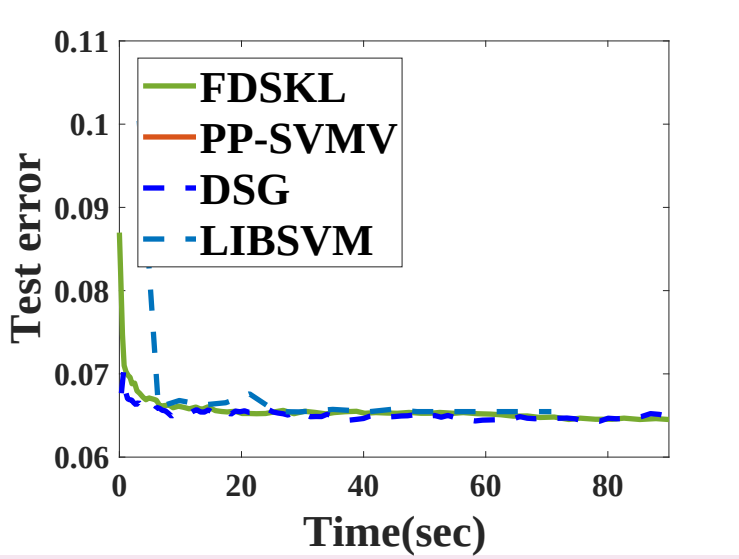


(h) epsilon

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

Validation Results

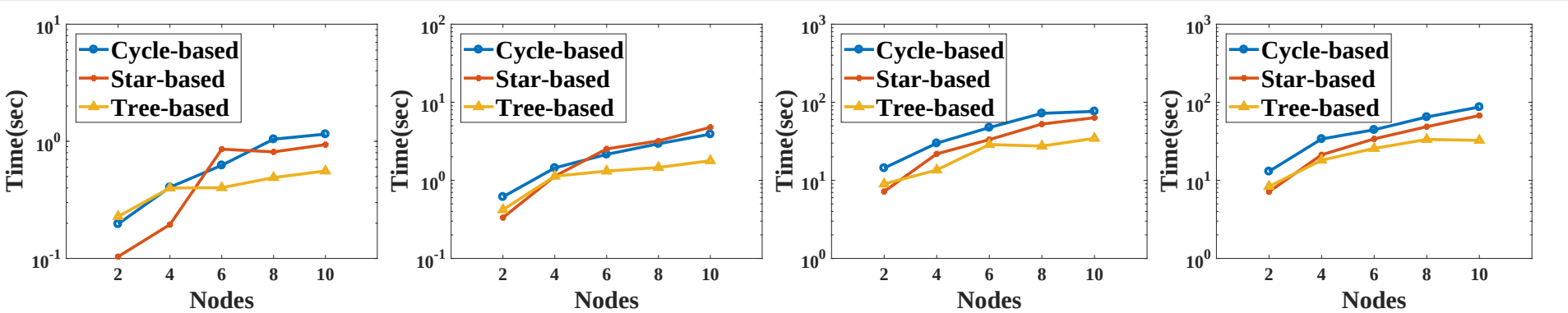




(a) deafultcredit (b) givemecredit

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

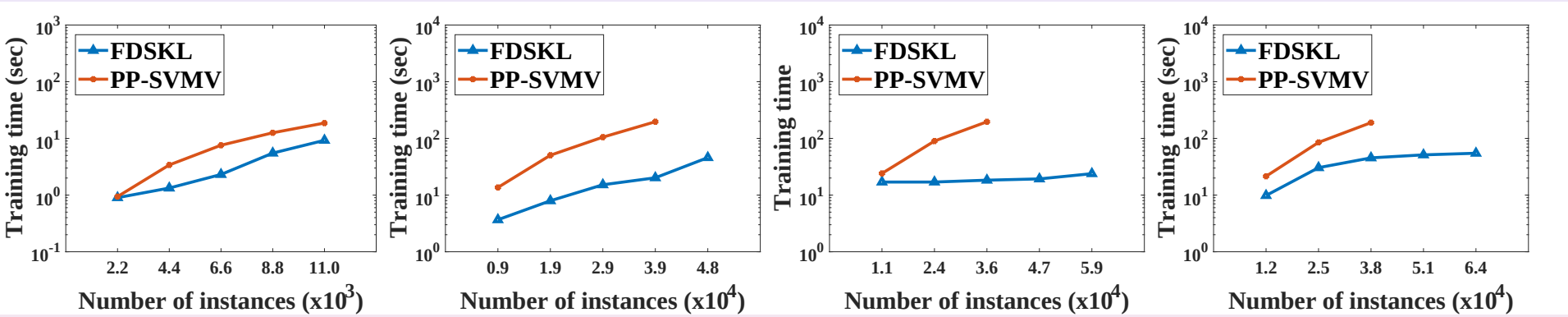
Validation Results



(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.

Validation Results

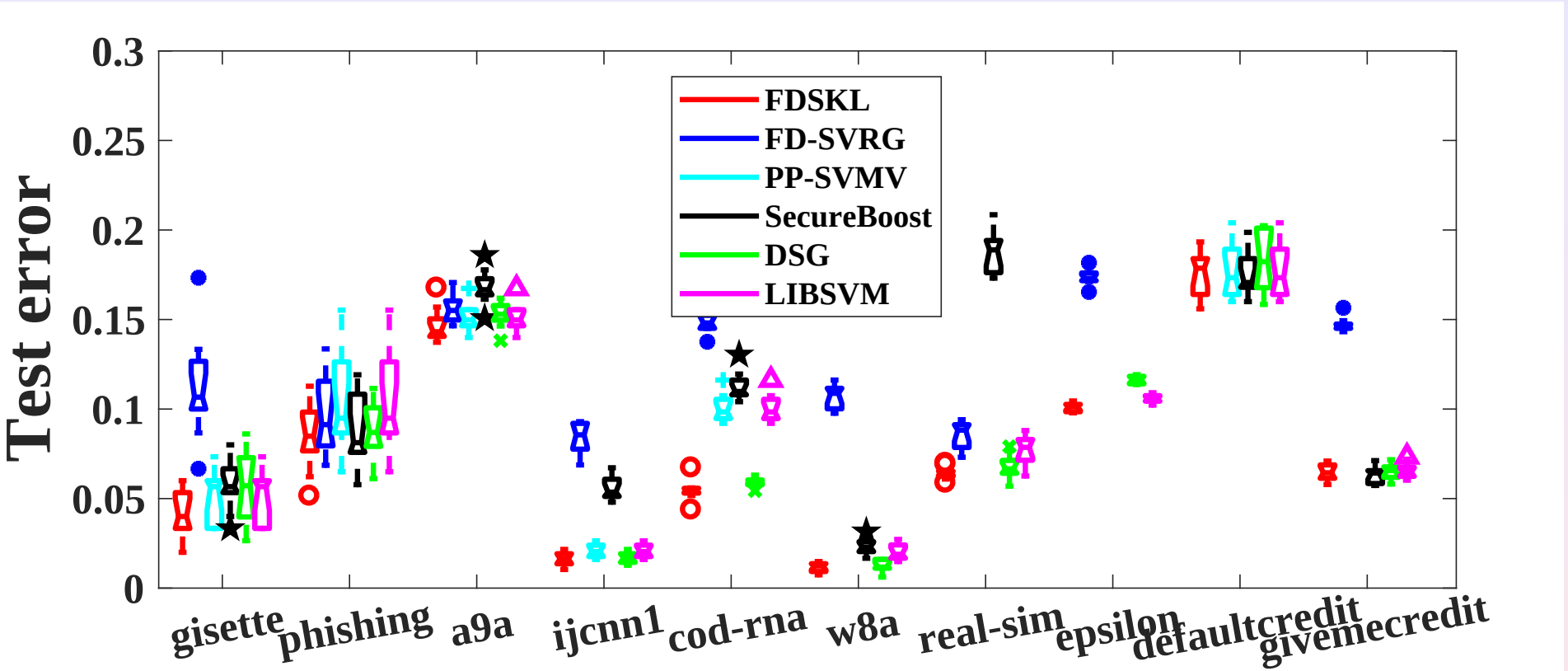
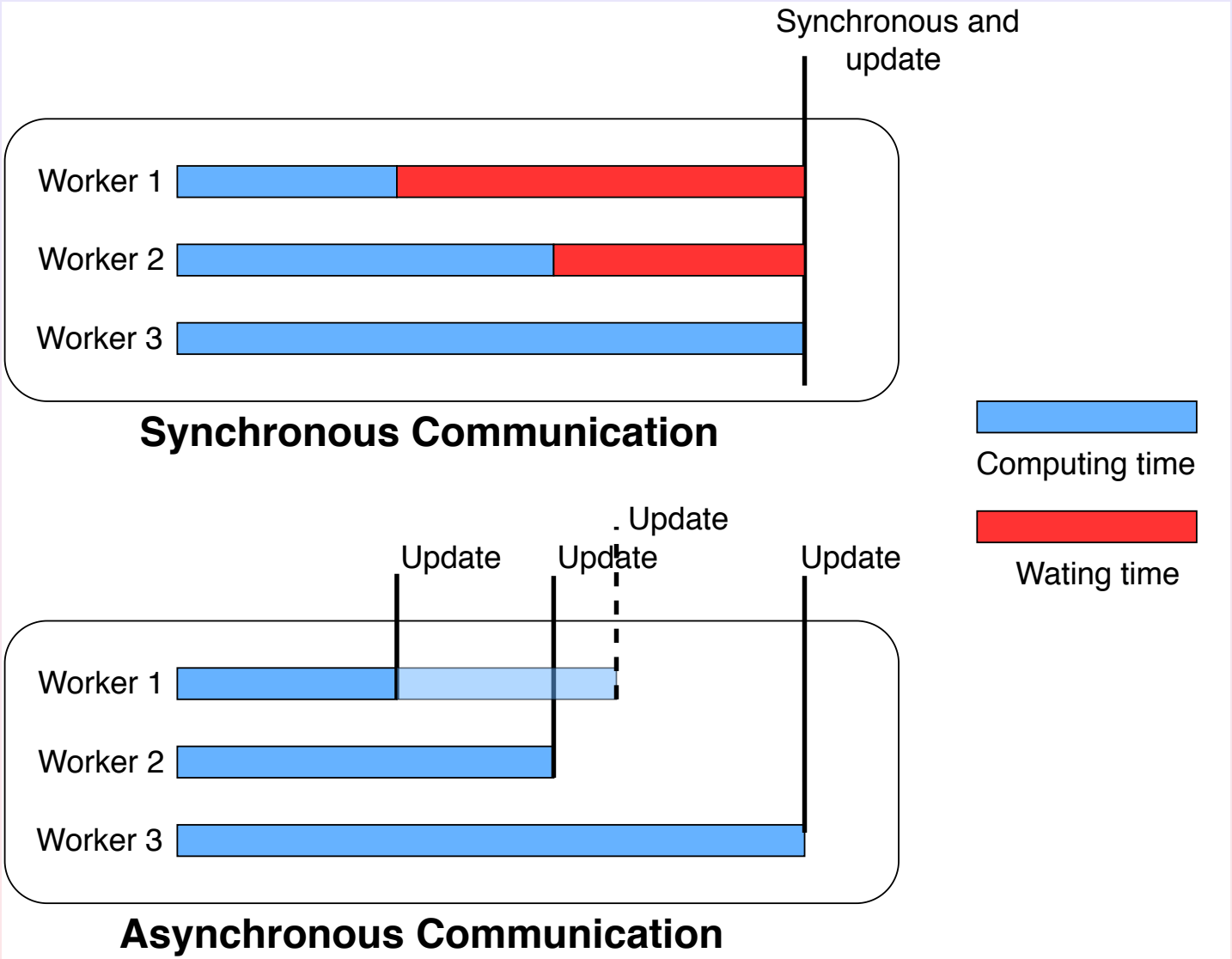


Figure: The boxplot of test errors of three state-of-the-art kernel methods, tree-boosting method (SecureBoost), linear method (FD-SVRG) and our VFKL.

Asynchronous Communication



Summary

1. Introduce a federated doubly stochastic kernel learning (*i.e.*, VFKL) algorithm
   - Effectively handle vertically partitioned data
   - Produce a sublinear convergence rate $\mathcal{O}(1/t)$
   - 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!