J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

연합학습을 위한 패턴 및 그룹 기반 효율적인 분산 합의 최적화☆

Efficient distributed consensus optimization based on patterns and groups for federated learning

강 승 주[1] 천 지 영[2] 노 건 태[2] 정 익 래[1*]
Seung Ju Kang Ji Young Chun Geontae Noh Ik Rae Jeong

요 약

인공지능으로 자동화와 연결성이 극대화되는 4차 산업혁명 시대를 맞이하여 모델의 업데이트를 위한 데이터 수집과 활용의 중요성이 점차 높아지고 있다. 인공지능 기술을 사용하여 모델을 생성하기 위해서는 일반적으로 데이터를 한곳에 모아야 업데이트할 수 있으나, 이런 경우 사용자의 개인정보를 침해할 수 있다. 본 논문에서는 분산 저장된 데이터를 직접 공유하지 않으면서 서로 협력하여 모델을 업데이트할 수 있는 분산형 기계학습 방법인 연합학습을 소개하며, 기존의 서버 없이 참여자들 간의 분산 합의 최적화를 이루는 연구를 소개한다. 또한, Kirkman Triple System을 기반으로 한 패턴 및 그룹을 생성하는 알고리즘을 이용하며, 병렬적인 업데이트 및 통신을 하는 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 제안한다. 이러한 알고리즘은 기존의 분산 합의 최적화 알고리즘 이상의 프라이버시를 보장하며, 모델이 수렴할 때까지의 통신시간을 감소시킨다.

☞ 주제어 : 연합학습, 최적화, 가중치 모델, 통신시간, 프라이버시, ADMM

ABSTRACT

In the era of the 4th industrial revolution, where automation and connectivity are maximized with artificial intelligence, the importance of data collection and utilization for model update is increasing. In order to create a model using artificial intelligence technology, it is usually necessary to gather data in one place so that it can be updated, but this can infringe users' privacy. In this paper, we introduce federated learning, a distributed machine learning method that can update models in cooperation without directly sharing distributed stored data, and introduce a study to optimize distributed consensus among participants without an existing server. In addition, we propose a pattern and group-based distributed consensus optimization algorithm that uses an algorithm for generating patterns and groups based on the Kirkman Triple System, and performs parallel updates and communication. This algorithm guarantees more privacy than the existing distributed consensus optimization algorithm and reduces the communication time until the model converges.

☞ keyword : Federated learning, Optimization, Weight model, Communication time, Privacy, ADMM

1. 서 론

인공지능을 이용한 데이터 활용에 대한 요구가 증가하면서 프라이버시 문제가 다수 발생하고 있다. 인공지능을 이용하기 위해서는 데이터를 한곳에 모아야 학습이 진행되기 때문인데, 이러한 경우 한곳에 모인 데이터에서 개인정보가 누출될 수 있다. 이를 해결하고자 하는 방안으로 연합학습(Federated Learning)이 대두되고 있다[1,2,3]. 특히 구글이 차세대 인공지능 학습으로 연합학습을 선택하였는데, 이러한 소식이 알려지면서 연합학습에 관한 관심이 점차 높아지고 있다. 기존의 기계학습은 학습하고자 하는 데이터를 중앙 서버에 모으고, 서버에서 가중치 모델을 일괄적으로 업데이트한다. 반면, 연합학습에서는 참여자의 개별 데이터를 중앙 서버로 전달하지 않고, 중앙 서버의 가중치 모델을 클라이언트로 보내 각각의 참여자가 가진 데이터로 업데이트한다. 데이터를 한곳에 집중시키지 않아 서버의 저장 공간, 연산량 등을 줄일 수 있어 비용 측면에서 더욱 효율적이고, 참여자의 데이터가 직접 노출되지 않아 프라이버시를 일정 수준 보장한다.

디지털 기술이 발달함에 따라 개별 기기와 기관에서

1 Department of Information Security, Korea University, Seoul, 02841, Korea.
2 Department of Bigdata & Information Security, Seoul Cyber University, Seoul, 01133, Korea.
* Corresponding author (irjeong@korea.ac.kr)
[Received 30 June 2022, Reviewed 9 July 2022(R2 11 August 2022), Accepted 19 August 2022]
☆ 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1F1A1063992).

독립적으로 생산, 수집, 저장하는 분산형 데이터양이 증가하고 있다. 분산형 데이터의 증가는 서로 다른 기관에서 동일 목적을 가지고 진행되는 데이터를 통합하여 의미있는 결과를 도출할 수 있다. 현재의 방식은 각 기관에서 분산형 데이터를 중앙 서버에 집중시켜 수집하고 있으며, 이러한 방식을 토대로 업데이트가 진행되고 있다. 이때 데이터를 익명으로 처리했다 할지라도, 서버에서 업데이트한 가중치 모델이 노출되면, 개인정보가 누출될 위험이 존재하기 때문에 프라이버시 문제가 발생할 수 있다[4].

참여자의 개별 데이터를 직접적으로 노출하지 않는 연합학습의 도입으로 업데이트 및 통신 과정 중 발생하는 프라이버시 문제를 일정 수준 보장할 수 있다. 하지만 중앙 서버에서 참여자들의 가중치 모델을 집계하는 한, 가중치 모델의 집계 결과인 글로벌 가중치 모델을 편향시켜 모델 업데이트를 조작하는 모델 중독 공격[5], 방어 메커니즘이 약한 집계 알고리즘을 조작하여 글로벌 가중치 모델이 비정상적으로 작동하도록 하는 집계 공격[6] 등 이 발생할 수 있다. 이러한 공격으로부터 참여자의 개별 데이터가 직접 노출되지는 않지만, 글로벌 가중치 모델이 노출되어 프라이버시 문제가 발생할 수 있다. 이러한 문제점 때문에 연합학습에서 중앙 서버가 존재하지 않는 탈 중앙 연합학습(Decentralized Federated Learning)이 필요하다.

중앙 서버를 중심으로 통신하는 기존의 연합학습 네트워크 환경은 클라이언트-서버 모델이다[7]. 이와 달리 탈 중앙 연합학습은 P2P(Peer-to-Peer) 모델 네트워크 환경이며, 모든 참여자가 클라이언트와 서버의 역할을 동시에 수행한다[8]. 이러한 네트워크 환경으로 연합학습을 진행한다면, 중앙 서버를 준비할 필요가 없어 비용이 감소하고, 각 참여자가 자원을 할당해 중앙 서버의 연산 능력에서 발생하는 오버헤드를 분산시킴으로써, 참여자가 증가하더라도 연합학습 시스템을 유지할 수 있는 높은 확장성을 가질 수 있다.

현재 연합학습이 가장 활발히 활용되고 있는 분야는 의료산업이며, 연합학습을 사용하면 서로 다른 기관에서 보유하고 있는 의료 데이터를 직접 공유하지 않고도 통합된 가중치 모델을 생성할 수 있다. 해당 산업 외에 개인이 정보 관리의 주체가 되는 마이데이터 사업 분야에도 연합학습의 활용이 가능하다. 앞으로 스마트폰, 웨어러블 기기, 스마트홈 장치(AI 스피커, 지능형 전자제품), 자동차 등 개인화된 제품과 장치는 더욱 많아질 것이다. 이러한 환경에서는 서버가 존재하기 힘들고, 신용카드 정보와 같은 개인의 정보를 보호하면서 업데이트 및 통신을 해야 하므로 탈 중앙 연합학습을 사용해야 한다.

탈 중앙 연합학습은 서버 없이 가중치 모델을 송·수신하여 업데이트하므로 참여자 간의 분산 합의 최적화 방법이 필요하다. 또한, 서버와 같은 중간자가 없으므로 추가적인 영지식 증명(Zero-Knowledge Proof), 동형 암호(Homormorphic Encryption), 다자간 연산(Multi-Party Computation)과 같은 보안기술이 필요하다. 하지만, 이러한 보안기술은 많은 연산량이 요구되어 통신시간 지연을 발생시킨다. Yu Ye 등[9]은 참여자 간의 순서를 정한 후, 분산 합의를 이루어 참여자 개인의 가중치 모델을 업데이트한다. 이러한 업데이트 시 최적화 방법의 하나인 ADMM(Alternating Direction Method of Multipliers)[10]을 사용한다.

본 논문에서는 Kirkman Triple System을 기반으로 한 패턴 및 그룹을 생성하는 알고리즘을 이용하며, 가중치 모델의 병렬적인 업데이트 및 통신을 하는 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 제안한다. 이러한 알고리즘은 Yu Ye 등의 분산 합의 최적화 알고리즘 이상의 프라이버시를 보장하며, 모든 참여자의 가중치 모델이 수렴할 때까지의 업데이트 및 통신시간을 감소시킨다. 또한, 실험을 통해 Yu Ye 등의 분산 합의 최적화 알고리즘보다 본 논문에서 제안한 패턴 및 그룹 기반 분산 합의 최적화 알고리즘이 가중치 모델 수렴까지의 업데이트 및 통신시간이 짧다는 결과를 내, 업데이트 및 통신시간 감소를 검증한다. 연합학습은 의료산업, 마이데이터 사업뿐만 아니라, 미국이 정부-민간 공동으로 시행한 반도체 분야의 SEMATECH 공동연구개발과 같이 정부-민간 간의 공동연구를 진행하는 사업에도 활용될 수 있다. 본 논문에서 제안한 알고리즘을 이용하여 연합학습을 진행하면 서버가 필요 없으므로, 정부로 데이터가 집중되는 현상을 방지할 수 있다. 또한, Kirkman Triple System을 기반으로 한 패턴 및 그룹을 생성하는 알고리즘을 이용하여 민간 데이터의 프라이버시를 보장할 수 있다.

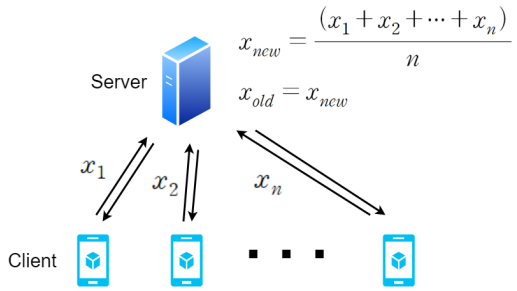
2. 배경 지식

2.1 연합학습

현재 대부분의 인공지능 모델 학습 방법은 스마트폰에서 생산되는 개인의 활동, 앱 사용 데이터뿐만 아니라 자동차의 주행 정보, 개인의 병원 진료 기록 등 모든 학습데이터를 한곳에 통합하고 이를 이용하는 방법을 사용하고 있다. 하지만 이러한 방법은 해당 데이터를 서버로 전송하는 도중 공격자가 탈취할 수 있어, 프라이버시 문제가

발생한다. 연합학습은 구글에서 제안된 기법으로 개인 모바일 기기에 저장된 데이터를 이용하여 가중치 모델을 업데이트하고, 이를 취합하여 범용적인 가중치 모델을 만드는 기법이다[1,2,3]. 이러한 기법은 기존 분산 학습 방법과 유사하지만 분산된 데이터 자체를 보호하면서 협업 모델을 생성할 수 있다는 점에서 큰 차이가 존재한다.

일반적인 연합학습 프로토콜은 그림 1과 같다. 이러한 프로토콜은 클라이언트와 서버로 구성되며 작동 방법은 다음과 같다. 먼저, 각 클라이언트는 자신이 소유한 개별 데이터 집합을 이용하여 가중치 모델($x_1 \ldots x_n$)을 업데이트한 후 서버에 보낸다. 서버는 클라이언트들의 가중치 모델을 수신하고 평균을 구해 기존의 글로벌 가중치 모델을 업데이트한 뒤, 다시 클라이언트에게 업데이트된 가중치 모델을 송신한다.



(그림 1) 연합학습 프로토콜

(Figure 1) Federated Learning Protocol

2.2 최적화

최적화는 일반적으로 어떤 제약조건이 있을 수도 있는 상황에서 함수의 최대치와 최소치를 찾거나, 효율적인 실행속도와 주파수대역폭(Bandwidth)을 증가시키고, 메모리 요구량을 감소시키는 방향으로 어떤 시스템을 개선하는 과정이다. 연합학습에서의 최적화는 주어진 가중치 모델 $x \in S$ 중에서 목적함수(Objective Function) $f(x)$를 최소화 (혹은 최대화) 하는 최적해(Optimum)를 찾는 것을 의미한다[11]. 이를 수학적으로 간단히 표현하면 다음과 같다.

$$\min_{x \in S} f(x) \ \ (\text{or} \ \max_{x \in S} f(x))$$

최소화하는 과정과 최대화하는 과정은 서로 대칭되기 때문에, 목적함수를 최소화하여 최적 해를 찾는 방법에 대해서만 나타낸다. 가중치 모델의 값 범위에 따라 최적화를 통해 지역 최소해(Local Minimum)가 도출될 수도 있고, 전역 최소해(Global Minimum)가 도출될 수도 있다. 전역 최소 해를 찾는 것이 일반적으로 어려우므로 지역 최소 해를 찾는 것만으로도 때때로 충분히 의미 있게 생각하기도 하지만, 전역 최소 해를 찾는 것이 최적화의 최종목적이다.

2.3 ADMM

ADMM은 원래의 문제보다 최적화가 쉬운 부분문제로 분할하고, 이를 취합함으로써 복잡한 원 문제를 해결하는 방식의 볼록 최적화(Convex Optimization) 알고리즘이다[10]. 이러한 알고리즘은 선형 제약조건을 효과적으로 처리할 수 있다. 또한, 부드럽지 않거나 복합적인(Composite) 목적함수를 최적화할 때 유용하며, 쌍대이론을 토대로 체계적으로 알고리즘을 구성할 수 있다.

두 개의 함수 $f(x)$와 $g(x)$로 구성된 목적함수에 대한 다음의 최적화 문제를 고려해본다.

$$\min_{x} \{f(x) + g(x)\} \tag{2.1}$$

이 문제와 동일한 문제는 다음과 같이 구성할 수 있다.

$$\min_{x,z} \{f(x) + g(z)\} \ \ subject \ to \ \ x = z \tag{2.2}$$

원 문제인 (2.1)과 달리 문제 (2.2)에서는 새로운 변수 z가 추가로 도입되었다. 여기서, z를 보조변수라 하고, 원 변수 x의 복제변수로써 문제 (2.1) 목적함수의 최적화 문제를 나누는 역할을 한다. 문제 (2.2)에 대한 라그랑지안 함수(Lagrangian Function)는 다음과 같다.

$$L(x,z,y) = f(x) + g(z) + y^T(x - z)\ ,$$

여기서 y는 라그랑지안 승수(Lagrangian Multipliers)이며, (2.2)를 원시(Primal) 문제라 하고, x, z를 원시변수(Primal Parameter)라 한다. 원시 문제에서 x, z에 대한 최솟값을 구하기 위해 다음의 두 단계를 수렴할 때까지 반복적으로 실행하는 기울기 상승알고리즘을 실행한다.

$$(x^{t+1}, z^{t+1}) = \arg\min_{x,z} L(x,z,y^t),$$
$$y^{t+1} = y^t + \rho(x^{t+1} - z^{t+1}),$$

여기서 t는 업데이트 횟수이며, ρ는 step-size의 역할을 한다. 해를 구하는 과정에서 기울기에만 근거한 방법은

일반적으로 불안정하다. 이러한 문제를 개선할 수 있는 여러 방법 중 증강(Augmented) 라그랑지안 방법이 있다. 이 방법은 $L(x,z,y)$에 제약조건 $x-z=0$에 대한 l_2 규제를 다음과 같이 추가적으로 고려한다.

$$L_\rho(x,z,y)=L(x,z,y)+\frac{\rho}{2}\|x-z\|_2^2,$$

여기서 ρ는 규제 항의 가중치를 조율하는 변수(Tuning Parameter)로서의 역할을 한다. 기울기 상승알고리즘에 증강 라그랑지안 방법을 적용하면 다음의 두 단계로 정리할 수 있다.

$$(x^{t+1},z^{t+1})=\arg\min_{x,z}L_\rho(x,z,y^t),$$
$$y^{t+1}=y^t+\rho(x^{t+1}-z^{t+1}).$$

ADMM 알고리즘에서 Alternating Direction 용어는 x^{t+1}와 z^{t+1}을 동시에 최적화하여 구하기보다 x^{t+1}와 z^{t+1}을 교차 분할 하여 구하는 과정이라는 의미가 있다. 따라서, ADMM 알고리즘의 표준적인 세 단계를 나타내면 다음과 같다.

$$x^{t+1}=\arg\min_{x}L_\rho(x,z^t,y^t),$$
$$z^{t+1}=\arg\min_{z}L_\rho(x^{t+1},z,y^t),$$
$$y^{t+1}=y^t+\rho(x^{t+1}-z^{t+1}).$$

2.4 Kirkman Triple System

1850년에 Thomas Penyngton Kirkman 목사가 제안한 Kirkman's Schoolgirl Problem에 대한 해결책이 Kirkman Triple System이다[12]. Kirkman's Schoolgirl Problem이란 한 학교의 15명의 젊은 여성이 7일 동안 3명씩 나란히 걸으며, 같은 사람 2명이 나란히 두 번 걷지 않도록 매일 나란히 걷는 사람에 대해 배치를 해야 하는 문제이다. 이러한 문제를 해결하기 위해서는 사람 수와 며칠 동안 걸어야 하는지 등을 고려하여 사람을 배치하는 조합론이 필요하다. Kirkman Triple System은 15명이 3명씩 나란히 걷는 경우뿐만 아니라 고려해야 할 총인원이 6으로 나누었을 때 나머지가 3인 경우 적용이 가능하며, 같은 사람 2명이 나란히 두 번 걷지 않도록 매일 나란히 걷는 사람에 대한 배치를 만들 수 있는 조합론이다.

3. 관련 연구

연합학습의 일반적인 아이디어는 중앙 서버에서 개별 업데이트한 가중치 모델을 수신한 뒤, 글로벌 가중치 모델로 업데이트하는 방법으로 진행된다. 하지만, 중앙 서버의 존재는 때때로 문제가 되기도 하는데, 해당 서버는 단일 서버 모델이기 때문에 단일 장애 지점(Single Point of Failure)[13] 문제가 존재하며, 항상 서비스 가능한 상태가 아니므로 장기적인 관점에서 문제가 발생할 수 있다. 서버가 없는 탈 중앙 연합학습은 이러한 문제를 해결하기 위해 네트워크의 통신 방식을 Peer-to-Peer 네트워크 형태로 변경한다. 네트워크의 모든 참여자는 그래프 형태로 연결되어 있으며, 연결된 이웃 참여자와의 통신으로 분산 합의 최적화를 진행해 개별 가중치 모델을 업데이트한다.

이 절에서는 서버가 없는 연합학습을 위한 분산 합의 최적화와 관련된 연구인 Yu Ye 등[9]과 Beomyeol Jeon 등[14]에 관한 설명을 한다.

3.1 Yu Ye 등

Yu Ye 등은 Peer-to-Peer 네트워크에서 연결된 모든 참여자와 통신을 하여 가중치 모델을 업데이트하는 방법인 기존의 탈 중앙 연합학습을 위한 분산 합의 최적화와는 달리, 참여자 간의 순서를 미리 정하고 통신을 한다. 이러한 방법을 이용해 Yu Ye 등은 ADMM 기반의 분산 합의 최적화 알고리즘인 I-ADMM(Incremental ADMM) 알고리즘을 제안하며, 해당 알고리즘의 시나리오는 다음과 같다.

- (통신 순서 고정) : 1) 해밀턴(Hamiltonian) 사이클 형태로 형성된 그래프에 각 참여자를 배정해 통신 순서를 고정한다.
- (초기화 단계) : 2) 참여자 각자 자신의 가중치 모델, 쌍대변수, 보조변수를 초기화한다.
- (업데이트 단계) : 3) 정해진 순서에 따라 이전 참여자에게 전달받은 보조변수와 참여자 자신의 개별 데이터 집합을 기반으로 4.2장의 (4.1), (4.2), (4.3)을 실행하여 자신의 가중치 모델, 쌍대변수, 보조변수를 업데이트한다.
- (전송단계) : 4) 업데이트한 보조변수를 다음 순서 참여자에게 송신한다.

- (업데이트 및 전송단계 반복) : 5) 업데이트 단계와 전송단계를 모든 참여자의 가중치 모델이 수렴할 때까지 반복한다.

I-ADMM 에서는 외부 공격자가 참여자 사이에 송·수신되는 보조변수, 참여자 각자의 초깃값, 4.2장의 (4.1), (4.2), (4.3)을 알 수 있는데, 이러한 정보를 이용하면 모든 참여자의 가중치 모델을 알아낼 수 있다. 분산 합의 최적화 과정에서 악의적인 공격자가 참여자 사이의 통신을 도청할 경우, 해당 참여자의 개별 데이터가 도출될 수 있어 프라이버시 문제가 발생한다.

3.2 Beomyeol Jeon 등

Beomyeol Jeon 등은 연합학습의 통신참여자들 모두 연결되어 통신할 수 있는 기존의 탈 중앙 연합학습을 위한 분산 합의 최적화와는 달리, 참여자들을 그룹화하여 그룹 안의 참여자만 연결한 후 그룹 기반으로 통신을 진행한다. 또한, 참여자들끼리 그룹을 형성할 때 한 그룹에 속하는 참여자는 3명으로 고정되어있으며, 다른 패턴을 사용할 동안 같은 그룹에 중복되는 사람과 통신하지 않도록 하는 조합법인 Kirkman Triple System을 사용한다[15]. 이러한 조합법은 참여자 수가 6으로 나누었을 때 나머지가 3인 경우에만 가능하다.

Beomyeol Jeon 등은 이러한 그룹화와 패턴을 이용하여, 기존의 탈 중앙 연합학습을 위한 분산 합의 최적화에서 발생하는 프라이버시 문제를 일정수준 해결하는 ADMM 기반의 분산 합의 최적화 알고리즘을 제안한다. 이러한 알고리즘을 이용하여 참여자 자신의 가중치 모델, 쌍대변수, 보조변수를 업데이트할 때, SGD(Stochastic Gradient Descent) 방법과 4.2장의 (4.4), (4.5), (4.6)을 사용한다.

4. 분산 합의 최적화 알고리즘

Yu Ye 등이 제안한 분산 합의 최적화 알고리즘은 참여자 개인의 가중치 모델을 업데이트한 후, 다음 참여자에게 업데이트된 보조변수를 송신하는 단방향 통신 방법을 사용하기 때문에 통신시간이 다수 발생한다. 또한, 4.2장의 (4.1), (4.2), (4.3)이 공개되어 있으므로 참여자가 보조변수를 송신할 때 악의적인 참여자나 외부 공격자가 이것을 도청한다면 해당 참여자의 개별 데이터가 노출될 수 있으므로, 프라이버시 문제가 발생한다.

이 절에서는 Yu Ye 등이 제안한 분산 합의 최적화 알고리즘보다 모든 참여자의 가중치 모델이 수렴할 때까지의 업데이트 및 통신시간은 줄이며, 프라이버시도 고려한 새로운 알고리즘을 제안한다.

4.1 통신 패턴 생성

$n = 9,27,45,\cdots$와 같이 참여자 수가 6으로 나누었을 때 나머지가 3이면서 9의 배수인 경우의 일반적인 Kirkman Triple System 생성은 표 1와 같다. Kirkman Triple System 생성에서 패턴의 개수는 $4+\frac{n-9}{2}$이며, 그룹의 수는 k이다. 그룹 내 첫 번째 참여자 C_i에서 $i > k$일 경우, 두 번째 참여자 C_i가 $i > 2k$일 경우 그리고, 세 번째 참여자 C_i에서 $i > 3k$일 경우 i를 k만큼 감소시킨다.

본 논문에서는 9의 배수이면서 6으로 나누었을 때 나

(표 1) Kirkman Triple System 생성
(Table 1) Kirkman Triple System Generation

row \ col	0	1	⋯	$k-1$
0	$\{C_1, C_{k+1}, C_{2k+1}\}$	$\{C_2, C_{k+2}, C_{2k+2}\}$	⋯	$\{C_k, C_{2k}, C_{3k}\}$
1	$\{C_1, C_{k+2}, C_{2k+3}\}$	$\{C_2, C_{k+3}, C_{2k+4}\}$	⋯	$\{C_k, C_{k+1}, C_{2k+2}\}$
⋯	⋯	⋯	⋯	⋯
$k-1$	$\{C_1, C_{\frac{(3k+1)}{2}+1}, C_{2k+2}\}$	$\{C_2, C_{\frac{(3k+1)}{2}+2}, C_{2k+3}\}$	⋯	$\{C_k, C_{\frac{(3k+1)}{2}}, C_{2k+1}\}$
k	$\{C_1, C_2, C_3\}$	$\{C_4, C_5, C_6\}$	⋯	$\{C_{3k-2}, C_{3k-1}, C_{3k}\}$
⋯	⋯	⋯	⋯	⋯

(표 2) 표기법
(Table 2) Notation

Notation	
C_i	Participant i
k	$k=\frac{n}{3}$
n	Total Number of Participants
col	Column of Table 2
row	Row of Table 2
N	Set of Participants
Π	Set of Patterns
π	Set of Groups
$L_{row,col}$	(row,col) th Group
x_j^{t+1}	$(t+1)$ th Weight Model for Participants j
y_j^{t+1}	$(t+1)$ th Dual Variable for Participant j
z_j^{t+1}	$(t+1)$ th Assist Variable for Participant j
t	Number of Updates for Each Participant
r,g,p	Temporary Variable
ρ	Hyper Parameter

머지가 3이 되는 수 중, 참여자 수가 9일 때로 국한하여 통신 패턴 생성을 나타낸다. 참여자가 9명 일 때의 Kirkman Triple System은 표 3과 같다. 또한, Kirkman Triple System 생성을 기반으로 한 통신 패턴 생성 알고리즘을 제안한다. 이러한 알고리즘은 그림 2와 같으며, 시나리오는 다음과 같다.

(표 3) 9명 Kirkman Triple System
(Table 3) 9Participants Kirkman Triple System

row \ col	0	1	2
0	$\{C_1,C_4,C_7\}$	$\{C_2,C_5,C_8\}$	$\{C_3,C_6,C_9\}$
1	$\{C_1,C_5,C_9\}$	$\{C_2,C_6,C_7\}$	$\{C_3,C_4,C_8\}$
2	$\{C_1,C_6,C_8\}$	$\{C_2,C_4,C_9\}$	$\{C_3,C_5,C_7\}$
3	$\{C_1,C_2,C_3\}$	$\{C_4,C_5,C_6\}$	$\{C_7,C_8,C_9\}$

- (초기화 단계) : 1) $\Pi,\pi,L_{row,col}$를 초기화한다.
- (그룹 생성 및 저장) : 2) Kirkman Triple System 생성에서 하나의 그룹($L_{row,col}$)을 선택해, 해당 그룹을 π에 추가하고, $L_{row,col}$를 초기화한다.
- (그룹 생성 및 저장 반복) : 3) 그룹 생성 및 저장 단계를 $k-1$번 반복한다.
- (패턴 생성 및 저장) : 4) 그룹 생성 및 저장 단계, 그룹 생성 및 저장 반복단계를 거쳐 생성된 패턴 π를 Π에 추가하여 저장하며, π를 초기화한다.
- (반복) : 5) 그룹 생성 및 저장, 그룹 생성 및 저장 반복 그리고, 패턴 생성 및 저장 단계를 $3+\frac{n-9}{2}$번 반복한다.
- (반환) : 6) 생성된 Π를 반환한다.

Algorithm 1: Comm Pattern Generation Algorithm

N : $\{C_1,C_2,...,C_n\}$
Initialize $\Pi=\varnothing$, $\pi=\varnothing$, $L_{row,col}=\varnothing$
for $row=0,1,...,(3+\frac{n-9}{2})$ do
  for $col=0,1,...,k-1$ do
    $L_{row,col}$=sample one group in Table 2
    $\pi=\pi\cup L_{row,col}$
    $L_{row,col}=\varnothing$
  end for
  $\Pi=\Pi\cup\pi$
  $\pi=\varnothing$
for end
return Π

(그림 2) 알고리즘 1
(Figure 2) Algorithm 1

4.2 패턴 및 그룹 기반 분산 합의 최적화

본 논문에서는 Yu Ye 등의 분산 합의 최적화 알고리즘보다 모든 참여자의 가중치 모델이 수렴하기까지의 업데이트 및 통신시간을 줄이면서 프라이버시까지 고려한 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 제안한다. 이러한 알고리즘은 한 참여자가 다른 한 명의 참여자에게만 보조변수를 송신하는 단방향 통신 방법이 아니라 그룹을 만들어 그룹 내에서는 단방향으로 통신을 하고, 그룹끼리는 양방향으로 통신을 한다. 모든 단방향 통신은 그룹끼리 병렬적으로 진행하기 때문에 통신시간을 줄일 수 있다. 본 논문에서 제안하는 패턴 및 그룹 기반 분산 합의 최적화 알고리즘은 그림 3과 같으며, 일반적인 시나리오는 다음과 같다.

Algorithm 2: Pattern and Group-based Distributed Consensus Optimization Algorithm

N : $\{C_1, C_2, ..., C_n\}$
$\Pi =$ by Comm Pattern Generation Algorithm
Initialize $z^0 = 0, x^0 = random, y^0 = 0, t = 0$
for π in Π do
  for $L_{row,col}$ in π parallel do
    $C_r, C_g, C_p \in L_{row,col}$ (random sample)
    for $j = r, g, p$ do
      update x_j^{t+1} according to (4.1)
      update y_j^{t+1} according to (4.2)
      update z_j^{t+1} according to (4.3)
      if $j = r$
        send z_j^{t+1} to C_g
      elif $j = g$
        send z_j^{t+1} to C_p
      elif $j = p$
        send z_j^{t+1} to any groups
        receive z^{t+1} from any groups
        $z =$ average z^{t+1}
        update x_j^{t+1} according to (4.4)
        update z_j^{t+1} according to (4.5)
        update y_j^{t+1} according to (4.6)
      each participants t update
    for end
  for end
for end

(그림 3) 알고리즘 2
(Figure 3) Algorithm 2

- (통신 패턴 생성) : 1) 통신 패턴 생성 알고리즘에 의해 그룹 안에서 중복되는 사람과의 통신을 최소화할 수 있는 Kirkman Triple System 통신 패턴을 생성한다.
- (초기화 단계) : 2) y^0, z^0, t 모두 0으로 초기화 하며, x^0은 무작위 값으로 초기화한다.
- (통신 패턴 선택) : 3) 패턴 집합 Π에서 하나의 패턴을 선택한다.
- (그룹 내 참여자 순서 설정) : 4) 각 그룹 내 참여자 중 무작위로 한 명씩 선택하여 선택된 순서대로 $R = \{C_{r_i} | \ i \in \{1,...,k\}\}$, $G = \{C_{g_i} | \ i \in \{1,...,k\}\}$, $P = \{C_{p_i} | \ i \in \{1,...,k\}\}$로 설정한다.
- (그룹 내 합의 업데이트) : 5) R에 있는 참여자들은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 x_j^{t+1}, y_j^{t+1}, z_j^{t+1}를 업데이트한다.
- (그룹 내 단방향 통신) : 6) R에 있는 참여자들은 같은 그룹에 속한 다음 참여자 C_{g_i}에게 z_j^{t+1}를 송신하며, t를 업데이트한다.
- (그룹 내 합의 업데이트) : 7) G에 있는 참여자들은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 x_j^{t+1}, y_j^{t+1}, z_j^{t+1}를 업데이트한다.
- (그룹 내 단방향 통신) : 8) G에 있는 참여자들은 같은 그룹에 속한 다음 참여자 C_{p_i}에게 z_j^{t+1}를 송신하며, t를 업데이트한다.
- (그룹 내 합의 업데이트) : 9) P에 있는 참여자들은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 x_j^{t+1}, y_j^{t+1}, z_j^{t+1}를 업데이트한다.
- (그룹 간 양방향 통신 및 업데이트) : 10) P에 있는 참여자들은 다른 그룹의 참여자 C_{p_i}들과 z_j^{t+1}를 공유한 뒤, 평균을 구해 z를 업데이트한다.
- (그룹별 집계 업데이트) : 11) P에 있는 참여자들은 병렬적으로 (4.4), (4.5), (4.6)을 실행하여 x_j^{t+1}, y_j^{t+1}, z_j^{t+1}를 업데이트하며, t를 업데이트한다.
- (반복) : 12) 통신 패턴 선택 단계부터 그룹별 집계 업데이트 단계까지 모든 참여자의 가중치 모델이 수렴할 때까지 반복한다.

$$x_j^{t+1} = \arg\min_{x_j} f_j(x_j^t) + \frac{\rho}{2}\|z_j^t - x_j^t + \frac{y_j^t}{\rho}\|^2 \quad (4.1)$$

$$y_j^{t+1} = y_j^t + \rho(z_j^t - x_j^{t+1}) \quad (4.2)$$

$$z_j^{t+1} = z_j^t + \frac{1}{n}(x_j^{t+1} - \frac{y_j^{t+1}}{\rho} - x_j^t + \frac{y_j^t}{\rho}) \quad (4.3)$$

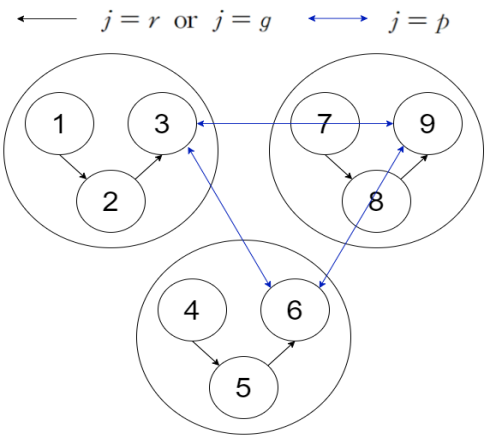
$$x_j^{t+1} = \frac{1}{2+\rho}(2x_j^{t+1} - y_j^{t+1} + \rho z_j^{t+1}) \quad (4.4)$$

$$z_j^{t+1} = x_j^{t+1} + \frac{1}{\rho}y_j^{t+1} \quad (4.5)$$

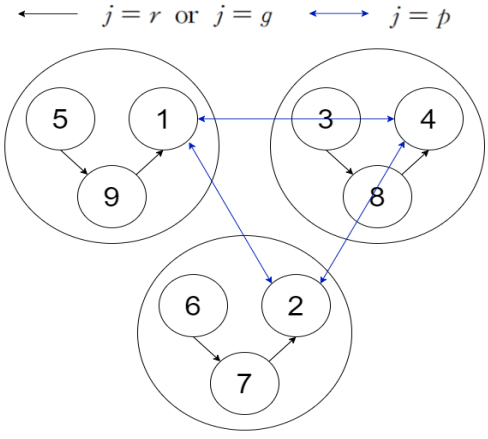
$$y_j^{t+1} = y_j^{t+1} + \rho(x_j^{t+1} - z) \quad (4.6)$$

패턴 및 그룹 기반 분산 합의 최적화 알고리즘의 이해를 돕기 위해 표 3을 이용하여 참여자가 9명일 때의 시나리오를 그림 4, 그림 5, 그림 6, 그림 7로 나타낸다.

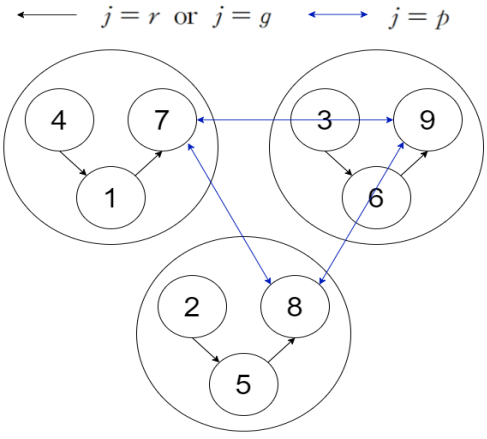
첫 번째로, 표 3의 3번 행 패턴을 사용하고, 그룹 내 참여자의 업데이트 및 통신 순서는 무작위로 설정하여 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 실행한다. 실행과정은 그림 4와 같으며, R은 참여자 1, 4, 7을 의미하고, G는 참여자 2, 5, 8을 의미한다. 그리고, P는 참여자 3, 6, 9를 의미한다.



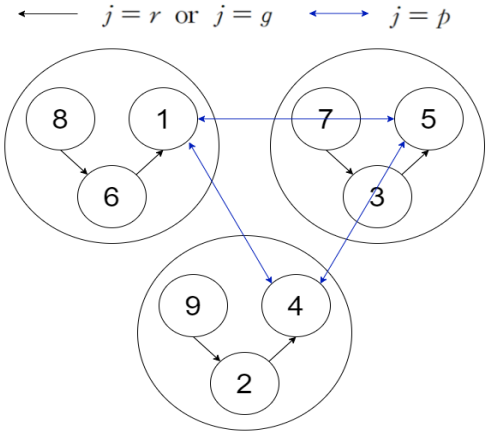
(그림 4) 패턴 1
(Figure 4) Pattern 1



(그림 6) 패턴 3
(Figure 6) Pattern 3



(그림 5) 패턴 2
(Figure 5) Pattern 2



(그림 7) 패턴 4
(Figure 7) Pattern 4

먼저, 참여자 1, 4, 7은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 자신의 가중치 모델, 쌍대 변수, 보조변수인 (x_1^1,y_1^1,z_1^1), (x_4^1,y_4^1,z_4^1), (x_7^1,y_7^1,z_7^1)을 업데이트한 뒤, 참여자 2, 5, 8에게 z_1^1, z_4^1, z_7^1를 송신하며 t를 업데이트한다. 이후, 참여자 2, 5, 8은 이전 참여자의 z_1^1, z_4^1, z_7^1을 수신하고, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_2^1,y_2^1,z_2^1), (x_5^1,y_5^1,z_5^1), (x_8^1,y_8^1,z_8^1)을 업데이트한 뒤, 참여자 3, 6, 9에게 z_2^1, z_5^1, z_8^1을 송신하며 t를 업데이트한다. 참여자 3, 6, 9는 이전 참여자의 z_2^1, z_5^1, z_8^1을 수신하고, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_3^1,y_3^1,z_3^1), (x_6^1,y_6^1,z_6^1), (x_9^1,y_9^1,z_9^1)을 업데이트한다. 이후, 참여자 3, 6, 9는 양방향 통신으로 z_3^1, z_6^1, z_9^1을 송·수신하고 평균을 구해 z를 업데이트한다. 또한, 병렬적으로 (4.4), (4.5), (4.6)을 실행해 (x_3^1,y_3^1,z_3^1), (x_6^1,y_6^1,z_6^1), (x_9^1,y_9^1,z_9^1)을 업데이트하며 t를 업데이트한다.

두 번째로, 표 3의 0번 행 패턴을 사용하고, 그룹 내 참여자의 업데이트 및 통신 순서는 무작위로 설정하여 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 실행한다. 실행과정은 그림 5와 같으며, R은 참여자 4, 2, 3을

의미하고, G는 참여자 1, 5, 6을 의미한다. 그리고, P는 참여자 7, 8, 9를 의미한다.

먼저, 참여자 4, 2, 3은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 자신의 가중치 모델, 쌍대 변수, 보조변수인 (x_4^2,y_4^2,z_4^2), (x_2^2,y_2^2,z_2^2), (x_3^2,y_3^2,z_3^2)을 업데이트한 뒤, 참여자 1, 5, 6에게 z_4^2, z_2^2, z_3^2를 송신하며 t를 업데이트한다. 이후, 참여자 1, 5, 6은 이전 참여자의 z_4^2, z_2^2, z_3^2을 수신하고, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_1^2,y_1^2,z_1^2), (x_5^2,y_5^2,z_5^2), (x_6^2,y_6^2,z_6^2)을 업데이트한 뒤, 참여자 7, 8, 9에게 z_1^2, z_5^2, z_6^2을 송신하며 t를 업데이트한다. 참여자 7, 8, 9는 이전 참여자의 z_1^2, z_5^2, z_6^2을 수신하며, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_7^2,y_7^2,z_7^2), (x_8^2,y_8^2,z_8^2), (x_9^2,y_9^2,z_9^2)을 업데이트한다. 이후, 참여자 7, 8, 9는 양방향 통신으로 z_7^2, z_8^2, z_9^2을 송·수신하고 평균을 구해 z를 업데이트한다. 또한, 병렬적으로 (4.4), (4.5), (4.6)을 실행하여 (x_7^2,y_7^2,z_7^2), (x_8^2,y_8^2,z_8^2), (x_9^2,y_9^2,z_9^2)을 업데이트하며 t를 업데이트한다.

세 번째로, 표 3의 1번 행 패턴을 사용하고, 그룹 내 참여자의 업데이트 및 통신 순서는 무작위로 설정하여 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 실행한다. 실행과정은 그림 6과 같으며, R은 참여자 5, 6, 3을 의미하고, G는 참여자 9, 7, 8을 의미한다. 그리고, P는 참여자 1, 2, 4를 의미한다.

먼저, 참여자 5, 6, 3은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 자신의 가중치 모델, 쌍대 변수, 보조변수인 (x_5^3,y_5^3,z_5^3), (x_6^3,y_6^3,z_6^3), (x_3^3,y_3^3,z_3^3)을 업데이트한 뒤, 참여자 9, 7, 8에게 z_5^3, z_6^3, z_3^3를 송신하며 t를 업데이트한다. 이후, 참여자 9, 7, 8은 이전 참여자의 z_5^3, z_6^3, z_3^3 을 수신하고, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_9^3,y_9^3,z_9^3), (x_7^3,y_7^3,z_7^3), (x_8^3,y_8^3,z_8^3)을 업데이트한 뒤, 참여자 1, 2, 4에게 z_9^3, z_7^3, z_8^3을 송신하며 t를 업데이트한다. 참여자 1, 2, 4는 이전 참여자의 z_9^3, z_7^3, z_8^3을 수신하며, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_1^3,y_1^3,z_1^3), (x_2^3,y_2^3,z_2^3), (x_4^3,y_4^3,z_4^3)을 업데이트한다. 이후, 참여자 1, 2, 4는 양방향 통신으로 z_1^3, z_2^3, z_4^3을 송·수신하고 평균을 구해 z를 업데이트한다. 또한, 병렬적으로 (4.4), (4.5), (4.6)을 실행하여 (x_1^3,y_1^3,z_1^3), (x_2^3,y_2^3,z_2^3), (x_4^3,y_4^3,z_4^3)을 업데이트하며 t를 업데이트한다.

네 번째로, 표 3의 2번 행 패턴을 사용하고, 그룹 내 참여자의 업데이트 및 통신 순서는 무작위로 설정하여 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 실행한다. 실행과정은 그림 7과 같으며, R은 참여자 8, 9, 7을 의미하고, G는 참여자 6, 2, 3을 의미한다. 그리고, P는 참여자 1, 4, 5를 의미한다.

먼저, 참여자 8, 9, 7은 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 자신의 가중치 모델, 쌍대 변수, 보조변수인 (x_8^4,y_8^4,z_8^4), (x_9^4,y_9^4,z_9^4), (x_7^4,y_7^4,z_7^4)을 업데이트한 뒤, 참여자 6, 2, 3에게 z_8^4, z_9^4, z_7^4을 송신하며 t를 업데이트한다. 이후, 참여자 6, 2, 3은 이전 참여자의 z_8^4, z_9^4, z_7^4을 수신하고, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_6^4,y_6^4,z_6^4), (x_2^4,y_2^4,z_2^4), (x_3^4,y_3^4,z_3^4)을 업데이트한 뒤, 참여자 1, 4, 5에게 z_6^4, z_2^4, z_3^4을 송신하며 t를 업데이트한다. 참여자 1, 4, 5는 이전 참여자의 z_6^4, z_2^4, z_3^4을 수신하며, 이를 이용해 병렬적으로 (4.1), (4.2), (4.3)을 실행하여 (x_1^4,y_1^4,z_1^4), (x_4^4,y_4^4,z_4^4), (x_5^4,y_5^4,z_5^4)을 업데이트한다. 이후, 참여자 1, 4, 5는 양방향 통신으로 z_1^4, z_4^4, z_5^4을 송·수신하고 평균을 구해 z를 업데이트한다. 또한, 병렬적으로 (4.4), (4.5), (4.6)을 실행하여 (x_1^4,y_1^4,z_1^4), (x_4^4,y_4^4,z_4^4), (x_5^4,y_5^4,z_5^4)을 업데이트하며 t를 업데이트한다.

네 번째까지 진행한 결과, 모든 참여자의 가중치 모델이 수렴하지 않는다면, 패턴 및 그룹 기반 분산 합의 최적화 알고리즘의 통신 패턴 선택 단계부터 그룹별 집계 업데이트 단계까지 모든 참여자의 가중치 모델이 수렴할 때까지 반복한다.

4.3 프라이버시 분석

Yu Ye 등의 분산 합의 최적화 알고리즘은 초기화 과정에서 초기 가중치 모델을 0으로 초기화한다. 이러한 경우 외부 공격자는 ρ, n, z_j^0, z_j^1, x_j^0, y_j^0를 알 수 있고 4.2장의 (4.2), (4.3)을 이용하면, 모르는 변수가 2개(x_j^1, y_j^1), 식이 2개이기 때문에 x_j^1, y_j^1을 알아낼 수 있다. 같은 방법으로 외부 공격자는 ρ, n, z_j^1, z_j^2, x_j^1, y_j^1를 알 수 있고 4.2장의 (4.2), (4.3)을 이용하면, 모르는 변수가 2개 (x_j^2,y_j^2), 식이 2개이기 때문에 x_j^2, y_j^2을 알아낼 수 있다. 이러한 방법으로 계속 진행하다 보면, 공격자가 알아내고자 하는 특정 참여자의 가중치 모델(x_j^t)을 알아낼 수

있다.

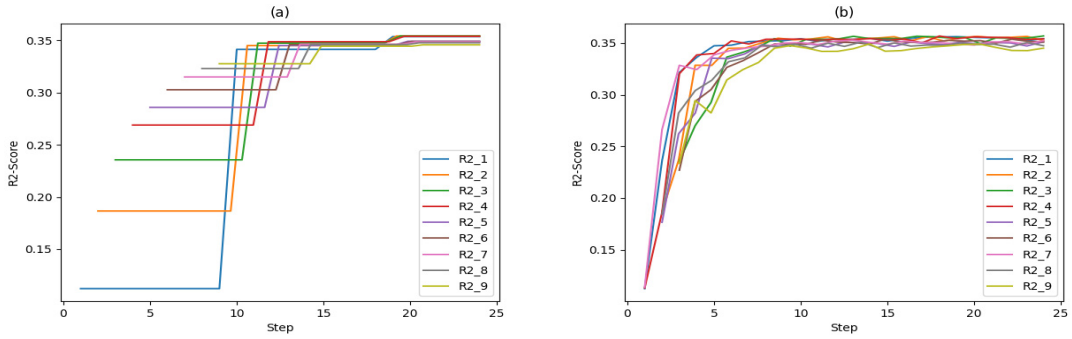
본 논문에서 제안한 패턴 및 그룹 기반 분산 합의 최적화 알고리즘에서는 참여자 각자의 초기 가중치 모델을 무작위 값으로 설정하여, 외부 공격자는 ρ, n, z_j^0, z_j^1, y_j^0를 알 수 있고 4.2장의 (4.2), (4.3)을 이용하면, 모르는 변수가 3개(x_j^0,x_j^1,y_j^0), 식이 2개이기 때문에 x_j^1, y_j^1을 알아낼 수 없다. 따라서 특정 참여자의 가중치 모델(x_j^t)을 알아내기 힘들게 하여 프라이버시를 향상한다.

또한, 업데이트 및 통신 순서에 Kirkman Triple System을 기반으로 한 패턴을 적용하고, 그룹 안에서의 참여자 순서는 무작위로 선택한다. 이러한 방법은 외부 공격자가 알아낸 보조변수를 송신한 참여자가 r인지, g인지, p인지 알아내기 어렵게 한다. 참여자 p는 참여자 r,g와 달리 4.2장의 (4.4), (4.5), (4.6)을 추가로 실행하여 가중치 모델을 업데이트한다. 이러한 이유로 외부 공격자가 알아낸 보조변수를 송신한 참여자가 r인지, g인지, p인지 알아내지 못하면, 4.2장의 (4.4), (4.5), (4.6)의 추가 실행 여부를 모르기 때문에, 특정 참여자의 가중치 모델(x_j^t)을 알아내기 힘들게 하여 프라이버시를 향상한다.

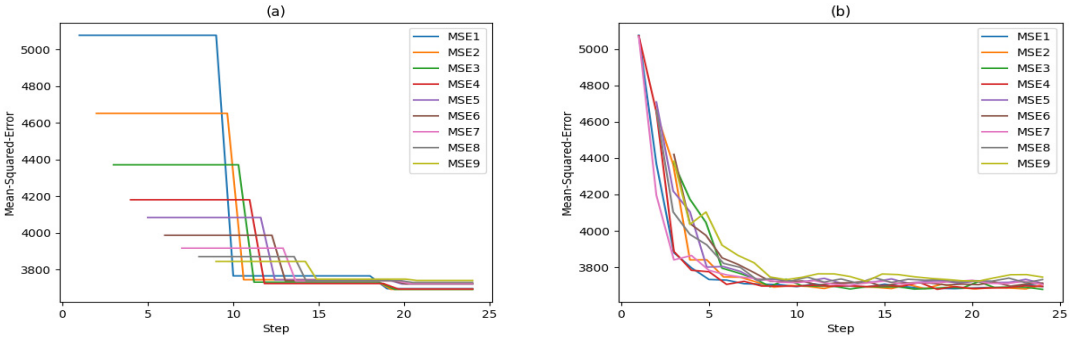
5. 실 험

이 절에서는 본 논문에서 제안한 패턴 및 그룹 기반 분산 합의 최적화 알고리즘의 수렴성과 업데이트 및 통신시간을 평가하기 위한 수치 실험결과를 보여준다. Yu Ye 등은 실험을 위해 해밀턴 주기를 보장하는 그래프를 생성한 후 4.2장의 (4.1), (4.2), (4.3)을 실행한다. 이처럼 본 논문의 실험에서도 가중치 모델의 수렴성과 업데이트 및 통신시간을 평가하는 실험을 진행하기 전에 Comm Pattern Generation 알고리즘으로 패턴을 생성한다.

가중치 모델의 수렴성을 평가하는 지표로써 예측값과 실제 값에 대한 분산으로 회귀 모델을 평가하는 R2-Score와 가중치 모델의 예측값과 실제값 차이의 합으로 회귀 모



(그림 8) R2-Score 비교
(Figure 8) R2-Score Comparison



(그림 9) Mean-Squared-Error 비교
(Figure 9) Mean-Squared-Error Comparison

델을 평가하는 Mean-Squared-Error를 사용한다. 본 논문의 실험에서 목적함수를 Lasso Regression으로 설정하고, Scikit-Learn에서 지원하는 datasets.load_diabetes() 당뇨병 데이터 집합을 사용하여 1년 동안의 당뇨병 진행도를 예측하는 문제를 푼다. 또한, 이러한 환경에서 가중치 모델의 수렴성과 업데이트 및 통신시간을 평가한다. Yu Ye 등의 분산 합의 최적화 알고리즘과 본 논문에서 제안한 분산 합의 최적화 알고리즘을 실행했을 때의 R2-Score 비교는 그림 8과 같으며, Mean-Squared-Error 비교는 그림 9와 같다.

그림 8과 그림 9의 (a)는 Yu Ye 등의 분산 합의 최적화 알고리즘을 실행했을 때의 결과를 나타내며, 그림 8과 그림 9의 (b)는 본 논문의 분산 합의 최적화 알고리즘을 실행했을 때의 결과를 나타낸다. 그림 8의 (a)는 9명의 참여자 모두 R2-Score 임계점인 0.345에 도달하는데 21번의 Step이 필요한 데 반해, (b)는 7번의 Step이 필요하다. 또한, 그림 9의 (a)는 9명의 참여자 모두 Mean-Squared-Error 임계점인 3750에 도달하는데 21번의 Step이 필요한 데 반해, (b)는 7번의 Step이 필요하다. 가중치 모델의 성능이 임계점에 도달하고, 이후의 Step들에서도 성능이 임계점과 크게 차이 나지 않을 때 수렴한다고 한다. Yu Ye 등의 분산 합의 최적화 알고리즘 실험결과는 21번의 Step으로 참여자 모두의 수렴이 이루어지지만 본 논문에서 제안한 분산 합의 최적화 알고리즘의 실험결과는 7번의 Step만으로 참여자 모두의 수렴이 이루어진다.

한 번의 통신시간이 5초라 가정하여 실험을 진행하였으며 참여자 모두의 가중치 모델 수렴이 이루어지는 시간은 표 4와 같다. 이러한 실험결과에 따르면 본 논문에서 제안한 분산 합의 최적화 알고리즘은 Yu Ye가 제안한 분산 합의 최적화 알고리즘보다 가중치 모델의 성능이 떨어지지 않으며, 모든 참여자의 가중치 모델이 수렴할 때까지 필요한 업데이트 및 통신시간을 약 66.7% 단축할 수 있다.

(표 4) 수렴 통신시간

(Table 4) Convergence Communication Time

	Communication Time	Step
Yu Ye, et al.	105.002(s)	21
Ours	35.001(s)	7

6. 결론 및 향후 연구과제

본 논문에서 기존의 분산 합의 최적화 알고리즘보다 가중치 모델의 성능은 떨어지지 않으면서 업데이트 및 통신시간을 줄일 수 있고, 프라이버시 또한 뒤처지지 않는 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 제안한다. 또한, 해당 알고리즘을 실행하는 데 사용되는 Kirkman Triple System 기반의 통신 패턴 생성 알고리즘을 제안한다. 참여자가 9명이라 가정하고, 본 논문의 패턴 및 그룹 기반 분산 합의 최적화 알고리즘을 실행했을 때의 시나리오를 작성하였으며, 회귀 모델을 평가하는 R2-Score, Mean-Squared-Error를 사용하여 알고리즘의 수렴성과 업데이트 및 통신시간을 측정하였다.

향후 이어지는 연구에서는 가중치 모델의 성능 및 프라이버시 측정을 위해, 당뇨병 데이터 집합뿐만이 아니라 다양한 데이터 집합에서의 성능측정 및 가중치 모델의 예측값과 실제값의 차이를 측정하여 프라이버시가 얼마나 보장되는지 수치화하여 나타내는 노력이 이루어져야 한다. 또한, 보스턴 집값, 붓꽃, 병원 데이터 등 다양한 데이터 집합을 이용하며, Lasso Regression뿐만이 아닌 Ridge Regression, Logistic Regression 등 다양한 목적함수를 사용하여 가중치 모델의 성능 및 프라이버시 측정을 하여 금융, 제조, 물류 등 다양한 분야에서 활용하기 위한 노력이 이루어져야 할 것이다.

참고문헌(Reference)

[1] H. B. McMahan, et al, “Communication-efficient learning of deep networks from decentralized data”, Artificial intelligence and statistics. PMLR, pp. 1273-1282, 2017.
http://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com

[2] N. Rieke, et al, “The future of digital health with federated learning”, NPJ digital medicine, Vol 3.1, pp.1-7, 2020.
https://www.nature.com/articles/s41746-020-00323-1

[3] T. Li, et al, “Federated learning: Challenges, methods, and future directions”, IEEE Signal Processing Magazine, Vol 37.3, pp.50-60, 2020.
https://doi.org/10.1109/MSP.2020.2975749

[4] N. Bouacida, and P. Mohapatra, “Vulnerabilities in Federated Learning”, IEEE Access, Vol 9, pp.63229-63249, 2021.
https://doi.org/10.1109/ACCESS.2021.3075203

[5] V. Mothukuri, et al, “A survey on security and privacy of federated learning”, Future Generation Computer

Systems, Vol 115, pp.619-640, 2021.
https://doi.org/10.1016/j.future.2020.10.007

[6] S. Fu, C. Xie, B. Li, and Q. Chen, “Attack-resistant federated learning with residual-based reweighting”, arXiv:1912.11464, 2019.
https://doi.org/10.48550/arXiv.1912.11464

[7] H. S. Oluwatosin, “Client-Server Model”, IOSR Journal of Computer Engineering, Vol 16.1, pp.67-71, 2014.
https://www.researchgate.net/profile/Shakirat-Sulyman/publication/271295146_Client-Server_Model/links/5864e11308ae8fce490c1b01/Client-Server-Model.pdf

[8] I. Hegedus, G. Danner, M. Jelasity, “Gossip learning as a decentralized alternative to federated learning”, IFIP International Conference on Distributed Applications and Interoperable Systems. Springer, Cham, pp.74-90, 2019.
https://link.springer.com/chapter/10.1007/978-3-030-22496-7_5

[9] Y. Yu, et al, “Privacy-preserving incremental ADMM for decentralized consensus optimization”, IEEE Transactions on Signal Processing, Vol 68, pp.5842-5854, 2020.
https://doi.org/10.1109/TSP.2020.3027917

[10] S. Boyd, et al, “Distributed optimization and statistical learning via the alternating direction method of multipliers”, Foundations and Trends® in Machine learning, Vol 3.1, pp.1-122, 2011.
https://doi.org/10.1561/2200000016

[11] F. Archetti, and F. Schoen, “A survey on the global optimization problem: general theory and computational approaches”, Annals of Operations Research, Vol 1.2, pp.87-110, 1984.
https://link.springer.com/article/10.1007/BF01876141

[12] G. Falcone, and M. Pavone, “Kirkman’s Tetrahedron and the Fifteen Schoolgirl Problem”, The American Mathematical Monthly Vol 118.10, pp.887-900, 2011.
https://doi.org/10.4169/amer.math.monthly.118.10.887

[13] Noveck, B. Simone, “The single point of failure”, Innovating government TMC Asser Press, pp.77-99, 2011.

[14] B. Jeon, et al, “Privacy-preserving decentralized aggregation for federated learning”, IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops, pp.1-6, 2021.
https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484437

[15] X. Li, Z. Xu, and W. Chou, “A new method of constructing Kirkman triple system”, Chinese Control and Decision Conference, IEEE, pp.4237-4242, 2011.
https://doi.org/10.1109/CCDC.2011.5968970

◐ 저 자 소 개 ◑

강 승 주(Seung Ju Kang)
2021년 광운대학교 소프트웨어학과(공학사)
2021년~현재 고려대학교 정보보안학과 석사과정
관심분야 : 정보보호, 블록체인, 인공지능, 개인정보보호
E-mail : rkdtmdwn0923@korea.ac.kr

천 지 영(Ji Young Chun)
1997년 이화여자대학교 수학과(이학사)
2006년 고려대학교 정보보호학과(공학석사)
2011년 고려대학교 정보경영공학과(공학박사)
2021년~현재 서울사이버대학교 빅데이터·정보보호학과 조교수
2022년~현재 서울사이버대학교 빅데이터·AI센터 부센터장
관심분야 : 데이터 보안, 인공지능, 연합학습, 프라이버시 향상 기술
E-mail : jychun@iscu.ac.kr

노 건 태(Geontae Noh)
2008년 고려대학교 산업시스템정보공학과(공학사)
2010년 고려대학교 정보경영공학과(공학석사)
2014년 고려대학교 정보보호학과(공학박사)
2014년~2017년 고려대학교 정보보호연구원 박사후 연구원, 연구교수
2017년~현재 서울사이버대학교 빅데이터·정보보호학과 조교수
2020년~현재 서울사이버대학교 빅데이터·AI센터 센터장
관심분야 : 프라이버시 향상 기술, 데이터 보안, 블록체인, 인공지능
E-mail : gnoh@iscu.ac.kr

정 익 래(Ik Rae Jeong)
1998년 고려대학교 전산학과(공학사)
2000년 고려대학교 전산학과(공학석사)
2004년 고려대학교 정보보호학과(공학박사)
2006년~2008년 한국전자통신연구원 암호기술연구팀 선임연구원
2008년~현재 고려대학교 정보보호대학원 교수
관심분야 : 암호 이론, 프라이버시 향상 기술, 데이터베이스 보안, 생체인증
E-mail : irjeong@korea.ac.kr