도메인 시프트 환경에서 Shift-Aware 프로토타입 특징 증류를 활용한 Split Federated Learning

최현준, 이주형*
*가천대학교

hjc405@gachon.ac.kr, * j17.lee@gachon.ac.kr

A Study on the Mitigating Domain Shift in Split Federated Learning via Normalized Features and Prototype Distillation systems

Choi Hyun Jun, Lee JooHyung*
Department of Computing at Gachon University

요 약

Split Federated Learning(SFL)은 Raw data 와 라벨을 클라이언트에 유지하지만, Non-IID 환경에서 도메인별 smashed feature 분포 차이로 일반화 성능이 저하된다. 본 논문은 Laber-local SFL 에서 cut-layer 특징 정규화와 클래스 프로토타입 기반 특징 증류를 결합해 도메인 시프트를 완화하는 방법을 제안한다. 클라이언트는 로컬 라벨로 프로토타입을 계산하고, 서버는 이를 집계한 전역 프로토타입을 배포하여 정규화된 표현을 정렬한다. 또한 로컬-전역 프로토타입 거리로 시프트를 추정해 증류 강도를 적응적으로 조절함으로써 부정적 전이를 억제한다. 실험을 통해 제안 방법이 SFL baseline 대비 도메인 시프트 환경에서 일관된 성능 향상을 보임을 확인하였다.

Ⅰ. 서 론

개인정보 보호, 규제 준수, 보안 요구가 강화되면서 원천 데이터를 공유하지 않고 협력 학습을 수행하는 분산 학습의 중요성이 커지고 있다. Federated Learning(FL)은 각 클라이언트가 로컬 데이터로 학습한 뒤 업데이트를 서버에서 집계하는 방식(FedAvg)으로 이를 가능하게 한다 [1]. 그러나 실제 분산 데이터는 대개 Non-IID 로 클라이언트 간 분포 차이가 존재하며, 이는 전역 모델의 수렴 및 일반화 성능 저하로 이어질 수 있다. 또한 FL 은 클라이언트가 전체 모델을 학습해야 하므로 자원 제약이 큰 환경에서 부담이 크다 [1].

이를 완화하기 위해 Split Learning(SL)과 결합한 Split Federated Learning(SFL)이 제안되었다 [2,3]. SFL 에서는 클라이언트가 네트워크 앞단의 모델을 실행해 cut layer 의 출력인 smashed feature 를 서버로 전송하고, 서버는 나머지 모델로 예측을 수행한다. 하지만 Non-IID 환경에서는 클라이언트별 앞단의 모델이 도메인 특화 표현을 학습하기 쉬워, 동일 클래스라도 smashed feature 분포가 달라질 수 있다. 그 결과 서버 top 모델은 라운드마다 변하는 입력 분포(=cut-layer 표현 공간의 domain shift)를 학습하게 되어 학습 안정성과 성능이 악화될 수 있다 [3,4]. 특히 프라이버시를 위해 라벨이 클라이언트에만 존재하는 label-local SFL 에서는 서버가 지도 신호로 표현 정렬을 직접 수행하기 어렵다는 제약이 존재한다 [3,4].

이에 본 논문은 label-local SFL 에서 도메인 시프트에 강건한 학습을 위해, (i) cut-layer 특징 정규화로 smashed feature 의 통계적 변동을 완화하고, (ii) 클래스 프로토타입 기반 feature distillation 을 통해 샘플 단위 특징 공유 없이 의미 수준의 정렬을 유도하며, (iii) 로컬-전역 프로토타입 거리로 shift 수준을 추정해 증류 강도를 적응적으로 조절하는 방법을 제안한다. 제안 방식은 추가 통신을 클래스 요약 정보 수준으로 제한하면서도 Non-IID/다중 도메인 환경에서 학습 안정성과 전역 일반화 성능 향상을 목표로 한다.

Ⅱ. 본론

2.1. Shift-Aware 프로토타입 특징 증류 기반 SFL

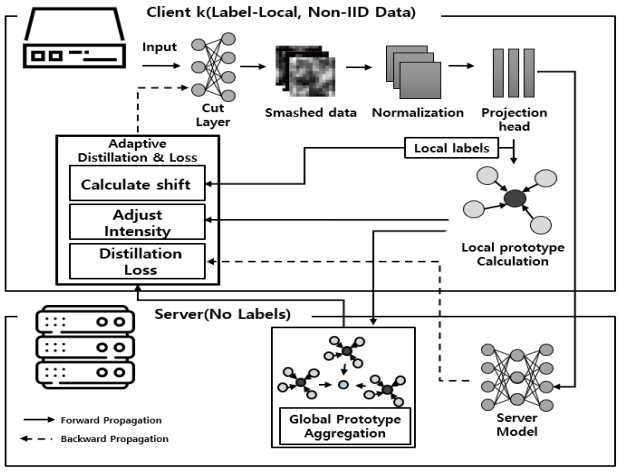

그림 1. Label-local SFL 에서 도메인 시프트를 완화하기 위한 본 방법의 개요.

위의 그림과 같이 서버 입력 분포의 통계적 변동을 완화하기 위해, 클라이언트는 smashed feature h 를 서버로 보내기 전 정규화 함수 $g(\cdot)$를 적용한다 [그림 1].

$$\tilde{h} = g(h)$$

여기서 $g(\cdot)$ 는 ℓ_2 -정규화이며, 도메인별 스케일/분산 차이를 줄여 서버가 받는 입력의 분포 드리프트를 완화하는 것을 목표로 한다. 정규화는 추가 통신 없이 클라이언트 로컬에서 수행되므로 경량이며, 서버 측 모델 변경 없이도 적용 가능하다.

정규화만으로는 클래스 의미 수준의 정렬이 충분하지 않을 수 있으므로, 본 논문은 샘플 단위 특징 노출 없이 클래스 요약 정보만을 교환하는 프로토타입 정렬을 도입한다. 클라이언트는 정규화된 특징 $\tilde{h}$에 대해 간단한 projection head $\phi(\cdot)$를 적용해

$$z = \phi(\tilde{h})$$

를 얻고, 클래스별 로컬 프로토타입을 다음과 같이 계산한다.

$$p_{k,c} = \frac{1}{|D_{k,c}|} \sum_{(x,y)\in D_{k,c}} z, c \in \mathcal{C}$$

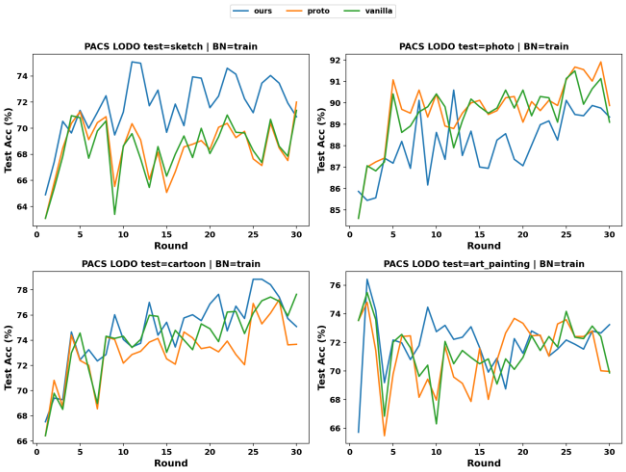

그림 X. PACS LODO(BN=train)에서 도메인별 라운드-테스트 정확도(%) 곡선.

$D_{k,c}$ 는 라운드 t에서 클라이언트 k의 클래스 c 샘플 집합이며, 클라이언트는 라운드마다 $\{(p_{k,c}^t, n_{k,c}^t)\}$를 서버로 전송한다($n_{k,c}^t = |D_{k,c}^t|$). 여기서 $\{(p_{k,c}^t, n_{k,c}^t)\}$는 클라이언트 k가 관측한 클래스 집합 $\mathcal{C}_k^t = \{c \mid n_{k,c}^t > 0\}$에 대해 구성한 클래스 요약 메시지이며, $p_{k,c}^t \in \mathbb{R}^d$는 클래스 c의 로컬 프로토타입(해당 클래스 샘플들의 z 평균), $n_{k,c}^t \in \mathbb{N}$ 는 그 샘플 수이다. 즉 서버로 전송되는 메시지는 $M_k^t = \{(p_{k,c}^t, n_{k,c}^t)\}_{c \in \mathcal{C}_k^t}$이다. 서버는 이를 집계하여 전역 프로토타입 P_c를 업데이트한다.

$$\tilde{p}_c = \frac{\sum_k n_{k,c}\, p_{k,c}}{\sum_k n_{k,c}}, P_c \leftarrow (1-\beta)P_c + \beta \tilde{p}_c.$$

이후 서버는 $\{P_c\}$를 클라이언트에 배포하며, 클라이언트는 각 샘플의 표현 z가 정답 클래스의 전역 프로토타입 P_y에 가까워지도록 distillation 손실을 추가한다.

$$L_{\text{proto}} = \mathbb{E}_{(x,y)\sim D_k}[d(z, \text{sg}(P_y))]$$

여기서 $d(\cdot,\cdot)$ 는 ℓ_2 또는 코사인 거리, sg$(\cdot)$ 는 stop-gradient 이다. 이 과정은 서버가 라벨을 보지 못해도 클라이언트 로컬 라벨을 통해 클래스 조건부 정렬을 유도할 수 있으며, 추가 통신은 클래스 수 $|\mathcal{C}|$에 비례하는 프로토타입만 전송하므로 샘플 특징 공유 대비 노출과 비용을 크게 줄인다.

전역 프로토타입은 다양한 도메인의 평균적 표현을 반영하므로, 도메인 차이가 큰 클라이언트가 이를 과도하게 추종할 경우 성능 저하(negative transfer)가 발생할 수 있다. 이를 완화하기 위해 클라이언트 k는 로컬-전역 프로토타입 간 거리로 shift 수준을 추정한다.

$$d_k = \frac{1}{|\mathcal{C}_k|} \sum_{c \in \mathcal{C}_k} d(p_{k,c}, P_c)$$

$\mathcal{C}_k$는 클라이언트 k가 관측한 클래스 집합이고 증류 강도 λ_k를 d_k의 함수로 적응적으로 조절한다. 예를 들어,

$$\lambda_k = \lambda_0 \exp(-\alpha d_k)$$

로 두면 shift 가 클수록 증류를 약화하여 로컬 도메인에 필요한 표현 유지와 전역 정렬 간 균형을 맞출 수 있다.

클라이언트 k의 최종 학습 목표는 다음과 같다.

$$L = L_{\text{task}} + \lambda_k L_{\text{proto}}.$$

L_{task}는 label-local supervised loss(e.g. cross-entropy)이며, L_{proto}는 클래스 조건부 표현 정렬을 유도한다.

2.2. 실험 결과

본 연구는 도메인 시프트 환경에서 제안 기법의 효과를 검증하기 위해 PACS 데이터셋의 Leave-One-Domain-Out(LODO) 프로토콜을 사용하였다. 네 개 도메인(photo/art_painting/cartoon/sketch) 중 하나를 테스트 도메인으로 고정하고, 나머지 세 도메인을 각각 하나의 클라이언트로 구성하여 3-클라이언트 label-local SFL 을 수행하였다. 백본은 ResNet-18 이며 PACS 실험에서는 ImageNet pretrained 가중치를 사용하였다. 비교 대상은 정규화 및 프로토 타입 증류를 사용하지 않은 기본 SFL 인 Vanila 와 클래스 전역 프로토타입에 대한 특징 정렬만 적용한 Proto, 마지막으로 우리가 제안한 기법인 Ours 로 이루어져있다.

Test Domain	Vanila	Proto	Ours
Art Painting	75.49	74.80	**76.42**
Cartoon	77.65	77.22	**78.84**
Photo	91.50	**91.92**	90.60
Sketch	71.37	72.00	**75.08**
Avg.	79.00	78.99	**80.24**
Worst	71.37	72.00	**75.08**

표 1. PACS LODO(BN=train)에서 홀드아웃 도메인별 최고 정확도(%) 비교(vanilla/proto/ours, seed=0).

[그림 1]는 PACS LODO 환경(BN=train)에서 라운드에 따른 홀드아웃 도메인 테스트 정확도 변화를 나타낸다. 연합학습 특성상 정확도가 라운드별로 진동하는 경향이 있으나, 제안 방법(Ours)은 대부분의 도메인에서 더 높은 성능 구간을 형성한다. 도메인별 정량 비교는 다음 표 X 에서 상세히 설명한다. [표 1]에서 제안 방법(Ours)은 Art_painting, Cartoon, Sketch 에서 Vanilla 및 Proto 대비 더 높은 정확도를 보였으며, 특히 도메인 시프트가 큰 sketch 에서 Proto 대비 +3.08%p 향상을 보였다. 반면 photo 에서는 Proto/Vanilla 가 소폭 우세하였으나, 전체 평균 정확도는 Ours 가 80.24%로 가장 높았다 또한 최악 도메인 기준(Worst-domain) 정확도 역시 Ours 가 75.08 로 가장 높아 도메인 시프트 환경에서 강건성이 향상됨을 확인하였다.

Ⅲ. 결론

본 논문에서는 label-local Split Federated Learning 환경에서 도메인 시프트로 인한 성능 저하를 완화하기 위해, 정규화된 smashed feature 와 클래스 전역 프로토타입 기반 특징 증류를 결합하고, 클라이언트-전역 프로토타입 간 거리로 증류 강도를 조절하는 shift-aware 학습 방법을 제안하였다. PACS LODO 실험에서 제안 방법은 다수 도메인에서 기존 방법 대비 정확도를 향상시키고, 특히 도메인 시프트가 큰 조건에서 강건성을 개선함을 확인하였다.

ACKNOWLEDGMENT

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 2026 년도 SW 중심대학사업의 결과로 수행되었음"(2021-0-01389)

참 고 문 헌

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. AISTATS*, pp. 1273-1282, 2017.

[2] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split Learning for Health: Distributed Deep Learning without Sharing Raw Patient Data," *arXiv preprint* arXiv:1812.00564, 2018.

[3] C. Thapa, P. C. Mahawaga Arachchige, S. Camtepe, and L. Sun, "SplitFed: When Federated Learning Meets Split Learning," in *Proc. AAAI*, vol. 36, no. 8, pp. 8485-8493, 2022. (DOI: 10.1609/aaai.v36i8.20825)

[4] M. Arafeh, M. Wazzeh, H. Sami, H. Ould-Slimane, C. Talhi, A. Mourad, and H. Otrok, "Efficient privacy-preserving ML for IoT: Cluster-based split federated learning scheme for non-IID data," *Journal of Network and Computer Applications*, 2025. (DOI: 10.1016/j.jnca.2025.104105)