Federated Learning with Only Positive Labels

Felix X. Yu [1] **Ankit Singh Rawat** [1] **Aditya Krishna Menon** [1] **Sanjiv Kumar** [1]

Abstract

We consider learning a multi-class classification model in the federated setting, where each user has access to the positive data associated with only a single class. As a result, during each federated learning round, the users need to locally update the classifier without having access to the features and the model parameters for the negative classes. Thus, naïvely employing conventional decentralized learning such as distributed SGD or Federated Averaging may lead to trivial or extremely poor classifiers. In particular, for embedding based classifiers, all the class embeddings might collapse to a single point. To address this problem, we propose a generic framework for training with only positive labels, namely *Federated Averaging with Spreadout* (FedAwS), where the server imposes a geometric regularizer after each round to encourage classes to be spreadout in the embedding space. We show, both theoretically and empirically, that FedAwS can almost match the performance of conventional learning where users have access to negative labels. We further extend the proposed method to settings with large output spaces.

1. Introduction

We consider learning a classification model in the federated learning (McMahan et al., 2017) setup, where each user has only access to a single class. The users are not allowed to communicate with each other, nor do they have access to the classification model parameters associated with other users' classes. Examples of such settings include decentralized training of face recognition models or speaker identification models, where in addition to the user specific facial images and voice samples, the classifiers of the users also constitute sensitive information that cannot be shared with other users.

[1]Google Research, New York. Correspondence to: Felix X. Yu <felixyu@google.com>, Ankit Singh Rawat <ankitsrawat@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

This setting can also be extended to the case where each user has access to data associated with a small number of classes. For example, one application is *deep retrieval* in the federated setting: training a query to document relevance model based on user interactions such as clicked documents after issuing a query, assuming that the clicks do not get recorded by a central server.

In this work, we assume that the classification models are "embedding-based" discriminative models (Krizhevsky et al., 2012; Vaswani et al., 2017; Konečnỳ et al., 2016): both the classes and the input instances are embedded into the same space, and the similarity between the class embedding and the input embedding (*i.e.* logit or score) captures the likelihood of the input belonging to the class. A popular example of this framework is neural network based classification. Here, given an input instance $\boldsymbol{x} \in \mathcal{X}$, a neural network $g_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^d$ (parameterized by $\boldsymbol{\theta}$) embeds the instance into a d dimensional vector $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. The class embeddings are learned as a matrix $W \in \mathbb{R}^{C \times d}$, commonly referred to as the classification matrix, where C denotes the number of classes. Finally, the logits for the instance $\boldsymbol{x}$ are computed as $W \cdot g_{\boldsymbol{\theta}}(\boldsymbol{x})$.

In the federated learning setup, one collaboratively learns the classification model with the help of a server which facilitates the iterative training process by keeping track of a global model. During each round of the training process:

- The server sends the current global model to a set of participating users.
- Each user updates the model with its local data, and sends the model delta to the server.
- The server averages ("Federated Averaging") the deltas collected from the participating users and updates the global model.

Notice that the conventional synchronized distributed SGD falls into the federated learning framework if each user runs a single step of SGD, and the data at different users is i.i.d. Federated learning has been widely studied in distributed training of neural networks due to its appealing characteristics such as leveraging the computational power of edge devices (Li et al., 2019), removing the necessity of sending user data to server (McMahan et al., 2017), and various improvements on trust/security (Bonawitz et al., 2016), privacy (Agarwal et al., 2018), and fairness (Mohri et al., 2019).

However, conventional federated learning algorithms are not directly applicable to the problem of learning with only positive labels due to two key reasons: First, the server cannot communicate the full model to each user. Besides sending the instance embedding model $g_{\boldsymbol{\theta}}(\cdot)$, for the i-th user, the server can communicate only the class embedding vector $\boldsymbol{w}_i$ associated with the positive class of the user. Note that, in various applications, the class embeddings constitute highly sensitive information as they can be potentially utilized to identify the users.

Second, when the i-th user updates the model using its local data, it only has access to a set of instances $\boldsymbol{x} \in \mathcal{X}_i$ from the i-th class along with the class embedding vector $\boldsymbol{w}_i$. While training a standard embedding-based multi-class classification models, the underlying loss function encourages two properties: i) similarity between an instance embedding and the positive class embedding should be as large as possible; and ii) similarity between the instance embedding and the negative class embeddings should be as small as possible. In our problem setting, the latter is not possible because the user does not have access to the negative class embeddings.

In other words, if we were to use the vanilla federated learning approach, we would essentially be minimizing a loss function that only encourages small distances between the instances and their positive classes in the embedding space. As a result, this approach would lead to a trivial optimal solution where all instances and classes collapse to a single point in the embedding space.

To address this problem, we propose *Federated Averaging with Spreadout* (FedAwS) framework, where in addition to Federated Averaging, the server applies a geometric regularization to make sure that the class embeddings are well separated (cf. Section 4). This prevents the model from collapsing to the aforementioned trivial solution. To the best of our knowledge, this is the first principled approach for learning in the federated setting without explicit access to negative classes. We further show that the underlying regularizer can be suitably modified to extend the FedAwS framework to settings with large number of classes. This extension is crucial for the real-world applications such as user identification models with a large number of users. Subsequently, we theoretically justify the FedAwS framework by showing that it approximates the conventional training settings with a loss function that has access to both positive and negative labels (cf. Section 5). We further confirm the effectiveness of the proposed framework on various standard datasets in Section 6. Before presenting our aforementioned contributions, we begin by discussing the related work and formally describing the problem setup in Section 2 and 3, respectively.

2. Related Works

To the best of our knowledge, this is the first work addressing the novel setting of distributed learning with only positive labels in the federated learning framework. The learning setting we are considering is related to the positive-unlabeled (PU) setting where one only has access to the positives and unlabeled data. Different from PU learning (Liu et al., 2002; Elkan & Noto, 2008; du Plessis et al., 2015; Hsieh et al., 2015), in the federated learning setting, the clients do not have access to unlabeled data for both positive and negative classes. The setting is also related to one-class classification (Moya & Hush, 1996; Manevitz & Yousef, 2001) used in applications such as outlier detection and novelty detection. Different from one-class classification, we are interested in collaboratively learning a multi-class classification model.

We consider the setting of learning a discriminative embedding-based classifier. Popular neural networks fall in this category. An alternative approach is to train generative models. For example, each user can learn a generative model based on its own data, and the server performs the MAP estimation during the inference time. We do not consider this approach because it does not fit into the federated learning framework, where the clients and server collaboratively train a model. In addition, training a good generative model is both data and computation consuming. Another possible generative approach is to use federated learning to train a GAN model to synthesize negative labels for each user possibly using the techniques proposed in (Augenstein et al., 2019) and therefore convert the problem into learning with both positives and negatives. Training a GAN model in the federated setting is a separate and expensive process. In this paper we consider the setting where the users do not have access to either true or synthesized negatives.

As mentioned in the introduction, a typical application of federated learning with only positive labels is to use this learning framework to train user identification models such as speaker/face recognition models. Although the proposed FedAwS algorithm promotes user privacy by not sharing the data among the users or with the server, FedAwS itself does not provide formal privacy guarantees. To show formal privacy guarantees, we notice that differential privacy methods for federated learning (Agarwal et al., 2018; Abadi et al., 2016) can be readily employed in FedAwS by adding noise to the updates sent from each user.

On the technical side, the proposed FedAwS can be seen as using stochastic negative mining to improve the spreadout regularizer. The stochastic negative mining method was first proposed in Reddi et al. (2019) to mine hard negative classes for each data point. Differently, we mine hard negative classes for each class. The spreadout regularization was first proposed to improve learning discriminative vi-

sual descriptors (Zhang et al., 2017) and further used in the extreme-multiclass classification setting (Guo et al., 2019). The spreadout regularization is related to the design of error-correcting output code (ECOC) matrix (Dietterich & Bakiri, 1991; Pujol et al., 2006). In order for the ECOC matrix to work, the class embeddings have to be well separated from each other. In particular, similar to Proposition 1, Yu et al. (2013) show that the classification error can be bounded by the distance between data and positive label in the embedding space, and a measure of spreadout of the classes. Differently, our result is on the true error instead of the empirical error.

3. Problem Setup

3.1. Federated learning of a classification model

Let us first consider the conventional federated learning of a classification model, when each client has access to data from multiple classes. Let the instance space be $\mathcal{X}$, and suppose there are C classes indexed by the set $[C]$. Let $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathbb{R}^C\}$ be a set of scorer functions, where each scorer, given an instance $\boldsymbol{x}$, assigns a score to each of the C classes. In particular, for $c \in [C]$, $f(\boldsymbol{x})_c$ represents the relevance of the c-th class for the instance $\boldsymbol{x}$, as measured by the scorer $f \in \mathcal{F}$. We consider scorers of the form

$$f(\boldsymbol{x}) = W g_{\boldsymbol{\theta}}(\boldsymbol{x}), \tag{1}$$

where $g_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^d$ maps the instance $\boldsymbol{x}$ to a d-dimensional embedding, and $W \in \mathbb{R}^{C \times d}$ uses this embedding to produce the scores (or logits) for C classes as $W g_{\boldsymbol{\theta}}(\boldsymbol{x})$. The c-th row of W, $\boldsymbol{w}_c$, is referred to as the *embedding vector* of the c-th class. The score of the c-th class is thus $\boldsymbol{w}_c^\top g_{\boldsymbol{\theta}}(\boldsymbol{x})$.

Let us assume a distributed setup with m clients. In the traditional federated learning setup, for $i \in [m]$, the i-th client has access to n_i instance and label pairs $\mathcal{S}^i = \{(\boldsymbol{x}_1^i, y_1^i), \ldots, (\boldsymbol{x}_{n_i}^i, y_{n_i}^i)\} \subset \mathcal{X} \times [C]$ distributed according to an unknown distribution $\mathrm{P}_{\mathrm{XY}}^i$, i.e., $(\boldsymbol{x}_j^i, y_j^i) \sim \mathrm{P}_{\mathrm{XY}}^i$. Let $\mathcal{S} = \cup_{i \in [m]} \mathcal{S}^i$ denote the set of $n = \sum_{i \in [m]} n_i$ instance and label pairs collectively available at all the clients. Our objective is to find a scorer in $\mathcal{F}$ that captures the true relevance of a class for a given instance.

Formally, let $\ell : \mathbb{R}^C \times [C] \to \mathbb{R}$ be a loss function such that $\ell(f(\boldsymbol{x}), y)$ measures the quality of the scorer f on $(\boldsymbol{x}, y)$ pair. The client minimizes an empirical estimate of the risk based on its local observations $\mathcal{S}^i$ as follows:

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} \hat{\mathcal{R}}(f; S^i) := \frac{1}{n_i} \sum_{j \in [n_i]} \ell\big(f(\boldsymbol{x}_j^i), y_j^i\big). \tag{2}$$

In the federated learning setting, the m clients are interested in collaboratively training a single classification model on their joint data. A coordinator server facilitates the joint iterative distributed training as follows:

- At the t-th round of training, the coordinator sends the current model parameters $\boldsymbol{\theta}_t$ and W_t to all clients.
- For $i \in [m]$, the i-th client updates the current model based on its *local* empirical estimate of the risk[1]:

$$\boldsymbol{\theta}_t^i = \boldsymbol{\theta}_t - \eta \cdot \nabla_{\boldsymbol{\theta}_t} \hat{\mathcal{R}}(f_t; \mathcal{S}^i). \tag{3}$$

$$W_t^i = W_t - \eta \cdot \nabla_{W_t} \hat{\mathcal{R}}(f_t; \mathcal{S}^i). \tag{4}$$

- The coordinator receives the updated model parameters from all clients $\{\boldsymbol{\theta}_t^i, W_t^i\}_{i \in [m]}$, and updates its estimate of the model parameters using *Federated Averaging*:

$$\boldsymbol{\theta}_{t+1} = \sum_{i \in [m]} \omega_i \cdot \boldsymbol{\theta}_t^i; \quad W_{t+1} = \sum_{i \in [m]} \omega_i \cdot W_t^i, \tag{5}$$

  where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$ denotes the weights that the coordinator assigns to the training samples of different clients. For example, $\omega_i = \frac{n_i}{n}$ assigns uniform importance to all the training samples across different clients[2].

In the above, assuming that each client has data of multiple classes, the loss function in (2) can take various forms such as the contrastive loss (Hadsell et al., 2006; Chopra et al., 2005), triplet loss (Chechik et al., 2010) and softmax cross-entropy. All such losses encourage two properties:

- The embedding vector $g(\boldsymbol{x}_j^i)$ and its positive class embedding $\boldsymbol{w}_{y_j^i}$ are close. In other words, one wants large logits or scores for positives instance and label pairs.
- The embedding vector $g(\boldsymbol{x}_j^i)$ and its negative class class embeddings $\boldsymbol{w}_c$, $c \neq y_j^i$ are far away. In other words, one wants small logits or scores for negatives instance and label pairs.

For example, given a distance measure $\boldsymbol{d}(\cdot, \cdot)$, the contrastive loss is expressible as

$$\ell_{\mathrm{cl}}\big(f(\boldsymbol{x}), y\big) = \underbrace{\alpha \cdot \big(\boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y)\big)^2}_{\ell_{\mathrm{cl}}^{\mathrm{pos}}(f(\boldsymbol{x}), y)} + \underbrace{\beta \cdot \sum_{c \neq y} \big(\max\big\{0, \nu - \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_c)\big\}\big)^2}_{\ell_{\mathrm{cl}}^{\mathrm{neg}}(f(\boldsymbol{x}), y)}, \tag{6}$$

where $\alpha, \beta \in \mathbb{R}$ are some predefined constants. In (6), $\ell_{\mathrm{cl}}^{\mathrm{pos}}(\cdot)$ encourages high logit for the positive instance and label pairs. Similarly, $\ell_{\mathrm{cl}}^{\mathrm{neg}}(\cdot)$ aims to decrese the logit for the negative instance and label pairs.

[1] In the federated learning setup, the client may also update the model with a few steps, not just a single step.

[2] Recently, Mohri et al. (2019) proposed the *agnostic federated learning* framework to account for the heterogeneous data distribution across the clients, which crucially relies on the selecting the non-uniform weights. In this paper, for the ease of exposition, we restrict ourselves to the uniform weights, i.e., $\omega_i = \frac{n_i}{n}$.

3.2. Federated Learning with only positive labels

In this work, we consider the case where each client has access to only the data belonging to a single class. To simplify the notation, we assume that there are $m = C$ clients and the i-th client has access of the data of the i-th class. The algorithm and analysis also applies to the setting where multiple clients have the same class.

The clients are not allowed to share their data with other clients, nor can they access the label embeddings associated with other clients. Formally, in each communication round, the i-th client has access to

- n_i instance and label pairs with the same label i: $\mathcal{S}^i = \{(\boldsymbol{x}_1^i, i), \ldots, (\boldsymbol{x}_{n_i}^i, i)\} \subset \mathcal{X} \times [C]$
- Its own class embedding $\boldsymbol{w}_i$.
- The current instance embedding model parameter $\boldsymbol{\theta}$.

Without access to the negative instance and label pairs, the loss function can only encourage the instances embedding and the positive class embedding to be close to each other. For example, with the contrastive loss in (6), in the absence of negative labels, one can only employ $\ell_{\mathrm{cl}}^{\mathrm{pos}}(\cdot)$ part of the loss function. Since $\ell_{\mathrm{cl}}^{\mathrm{pos}}(\cdot)$ is a monotonically decreasing function of the distance between the instance and the positive label, this approach would quickly lead to a trivial solution with small risk where all the users and the classes have an identical embedding. Regardless of the underlying loss function, training with only positive instance and label pairs will result in this degenerate solution. We propose an algorithm to address this problem in the next section.

4. Algorithm

To prevent all the class embeddings $\{\boldsymbol{w}_i\}_{i=1}^C$ from collapsing into a single point in the optimization process, we propose Federated Averaging with Spreadout (FedAwS).

4.1. Federated Averaging with Spreadout (FedAwS)

In addition to Federated Averaging, the server performs an additional optimization step on the class embedding matrix $W \in \mathbb{R}^{C \times d}$ to ensure that different class embeddings are separated from each other by at least a margin of ν. In particular, in each round of training, the server employs a geometric regularization, namely *spreadout regularizer*, which takes the following form.

$$\mathrm{reg}_{\mathrm{sp}}(W) = \sum_{c \in [C]} \sum_{c' \neq c} \left(\max\left\{0, \nu - \boldsymbol{d}(\boldsymbol{w}_c, \boldsymbol{w}_{c'})\right\}\right)^2. \quad (7)$$

A similar objective was first proposed as a regularizer to improve learning discriminative visual descriptors (Zhang et al., 2017) and then used in extreme-multiclass classification (Guo et al., 2019). There, it was shown that the spreadout regularization can improve the quality and stability of the learned models. In this work, we argue that the spreadout regularizer along with the positive part of the underlying loss function (e.g., $\ell_{\mathrm{cl}}^{\mathrm{pos}}(\cdot)$ in (6)) constitutes a valid loss function that takes the similarity of the instance from both positive and negative labels into account (cf. Section 5). This proves critical in allowing for meaningful training in the federated setting with only positive labels.

Algorithm 1 Federated averaging with spreadout (FedAwS)

1: **Input.** For C clients and C classes indexed by $[C]$, n_i examples $\mathcal{S}_i$ at the i-th client.
2: Server initializes model parameters $\boldsymbol{\theta}^0, W^0$.
3: **for** $t = 0, 1, \ldots, T-1$ **do**
4: The server communicates $\boldsymbol{\theta}^t, \boldsymbol{w}_i^t$ to the i-th client.
5: **for** $i = 1, 2, \ldots, C$ **do**
6: The i-th client updates the model based on $\mathcal{S}_i$:
7: $(\boldsymbol{\theta}^{t,i}, \boldsymbol{w}_i^{t,i}) \leftarrow (\boldsymbol{\theta}^t, \boldsymbol{w}_i^t) - \eta \nabla_{(\boldsymbol{\theta}^t, \boldsymbol{w}_i^t)} \hat{\mathcal{R}}_{\mathrm{pos}}(\mathcal{S}^i)$,
8: where $\hat{\mathcal{R}}_{\mathrm{pos}}(\mathcal{S}^i) = \frac{1}{n_i} \sum_{j \in [n_i]} \ell_{\mathrm{cl}}^{\mathrm{pos}}(f(\boldsymbol{x}), y)$.
9: The i-th client sends $(\boldsymbol{\theta}^{t,i}, \boldsymbol{w}_i^{t,i})$ to the server.
10: **end for**
11: Server updates the model parameters:
12: $\boldsymbol{\theta}^{t+1} = \frac{1}{C} \sum_{i \in [C]} \boldsymbol{\theta}^{t,i}$.
13: $\tilde{W}^{t+1} = [\boldsymbol{w}_i^{t,i}, \ldots, \boldsymbol{w}_C^{t,C}]^\top$.
14: $W^{t+1} \leftarrow \tilde{W}^{t+1} - \lambda \eta \nabla_{\tilde{W}^{t+1}} \mathrm{reg}_{\mathrm{sp}}(\tilde{W}^{t+1})$.
15: **end for**
16: **Output:** $\boldsymbol{\theta}^\top$ and $W^\top$.

The FedAwS algorithm which modifies the Federated Averaging using the spreadout regularizer is summarized in Algorithm 1. Note that in Step 7, the local objective at each client is define by the positive part $\ell^{\mathrm{pos}}(\cdot)$ of the the underling loss (cf. (6)). The algorithm differs from the conventional Federated Averaging in two ways. First, averaging of W is replaced by updating the class embeddings received from each client (Step 13). Second, an additional optimization step is performed on server to encourage the separation of the class embeddings (Step 14). Here, we also introduce a learning rate multiplier λ which controls the effect of the spreadout regularization term on the trained model.

Remark 1. In Algorithm 1, we assumed all clients participate in each communication round for the ease of exposition. However, the algorithm easily extends to the practical setting, where only a subset of clients are involved in each round: Let $\mathcal{C}^t$ denote the set of clients participating the t-th round. Then, the server performs the updates in Step 12 and Step 13 with the help of the information received from the clients indexed by $\mathcal{C}^t$. Note that the optimization in Step 7 and Step 14 can employ multiple steps of SGD steps or be based on other optimizers.

4.2. FedAwS with stochastic negative mining

There are two unique challenges that arise when we perform optimization w.r.t. (7). First, the best ν is problem dependent and therefore hard to choose. Second, when C is large (also known as the extreme multiclass classification setting), even computing the spreadout regularizer becomes expensive. To this end we propose the following modification of (7):

$$\mathrm{reg}_{\mathrm{sp}}^{\mathrm{top}}(W) = \sum_{c\in\mathcal{C}^t} \sum_{\substack{y\in\mathcal{C}',\\ y\neq c}} -\boldsymbol{d}^2(\boldsymbol{w}_c, \boldsymbol{w}_y) \cdot [\![y \in \mathcal{N}_k(c)]\!], \quad (8)$$

where $\mathcal{C}'$ is a subset of classes, and $\mathcal{N}_k(c)$ denotes the set of k classes that are closest to the class c in the embedding space. The regularizer in (8) can be viewed as an adaptive approximator of the spreadout regularizer in (7), where, for each class c, we adaptively set ν to be the distance between $\boldsymbol{w}_c$ and its $(k+1)$-th closest class embedding. Intuitively, we only need to make sure that, in the embedding space, each class is as far away as possible from its close classes.

This approach of adaptively picking ν is motivated by the stochastic negative mining method first proposed in (Reddi et al., 2019), where for each instance, they consider only the positive label and a small set of most confusing ('hard') negative labels to define the underlying loss function. On the contrary, we are picking the most confusing classes based on only the class embeddings. Furthermore, the method is applied at the server as a regularizer as opposed to defining the underlying loss function for an individual instance. As we demonstrate in Section 6, the stochastic negative mining is crucial to improve the quality of FedAwS.

Before presenting these empirical results, we provide a theoretical justification for this in the following section.

5. Analysis

To justify our FedAwS technique, we will:

(i) relate the classification error to the separation of the class embeddings

(ii) introduce a particular *cosine contrastive loss*, which we show to be *consistent* for classification

(iii) relate the FedAwS objective to empirical risk minimization using the cosine contrastive loss, despite the latter requiring both positive *and* negative labels.

Put together, this justifies why the FedAwS classifier can be close in performance to that of a consistent classifier, despite only being trained with positive labels.

We first state a simple result arguing that *small* distance between the instance embedding and the *true* class embedding, and *large* distance between the class embeddings, imply low classification error.

Proposition 1. *Let the minimum distance between the class embeddings be $\rho := \inf_{i\neq j} \boldsymbol{d}(\boldsymbol{w}_i, \boldsymbol{w}_j)$, and the expected distance between the embeddings of an instance $\boldsymbol{x}$ and its true class y be $\epsilon = \mathbb{E}_{(\boldsymbol{x},y)\sim \mathrm{P_{XY}}} \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y)$. Then, the probability of misclassification satisfies*

$$\mathrm{P}\big(\exists z \neq y \; s.t. \; \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y) \geq \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_z)\big) \leq 2\epsilon/\rho.$$

Proof. Note that, if there exists any $z \neq y$ such that $\boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y) \geq \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_z)$, then

$$\begin{aligned} \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y) &\geq \frac{1}{2}\big(\boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y) + \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_z)\big) \\ &\overset{(i)}{\geq} \frac{\boldsymbol{d}(\boldsymbol{w}_y, \boldsymbol{w}_z)}{2} \overset{(ii)}{\geq} \frac{\rho}{2}, \end{aligned} \quad (9)$$

where (i) and (ii) follow from the triangle inequality and the definition of ρ, respectively. Next, by combing (9) with Markov's inequality, we obtain that

$$\begin{aligned} &\mathrm{P}\big(\exists z \neq y \text{ s.t. } \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y) \geq \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_z)\big) \\ &\quad\leq \mathrm{P}\big(\boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y) \geq \frac{\rho}{2}\big) \\ &\quad\leq \frac{2\mathbb{E}_{(\boldsymbol{x},y)\sim \mathrm{P_{XY}}} \boldsymbol{d}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y)}{\rho} = \frac{2\epsilon}{\rho}. \end{aligned}$$

□

To relate the FedAwS objective to a contrastive loss, without loss of generality, we work with normalized embeddings; i.e., we assume that the rows of the matrix W as well as the instance embeddings generated by $g_{\boldsymbol{\theta}}(\cdot)$ have unit Euclidean norm[3]. We can then adopt the cosine distance:

$$\boldsymbol{d}_{\cos}(\boldsymbol{u}, \boldsymbol{u}') = 1 - \boldsymbol{u}^\top \boldsymbol{u}' \quad \forall\, \boldsymbol{u},\, \boldsymbol{u}' \in \mathbb{R}^d. \quad (10)$$

Specializing the contrastive loss in (6) to the cosine distance measure gives us the *cosine contrastive loss*.

Definition 1 (Cosine contrastive loss)**.** *Given an instance and label pair $(\boldsymbol{x}, y)$ and the scorer $f(\boldsymbol{x})$ in (1), the cosine contrastive loss takes the following form.*

$$\begin{aligned} \ell_{\mathrm{ccl}}\big(f(\boldsymbol{x}), y\big) = &\big(\boldsymbol{d}_{\cos}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_y)\big)^2 + \\ &\sum_{c\neq y} \big(\max\big\{0, \nu - \boldsymbol{d}_{\cos}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{w}_c)\big\}\big)^2. \end{aligned} \quad (11)$$

Further, by using $s_c = g_{\boldsymbol{\theta}}^\top(\boldsymbol{x})\boldsymbol{w}_c$ to denote the logit for class c, the cosine contrastive loss can be expressed as

$$\begin{aligned} &\ell_{\mathrm{ccl}}\big(f(\boldsymbol{x}), y\big) = \\ &\quad (1 - s_y)^2 + \sum_{c\neq y} \big(\max\big\{0, \nu - 1 + s_c\big\}\big)^2 \end{aligned} \quad (12)$$

[3]The analysis in this section easily extends to unnormalized embeddings. However, the restriction to normalized embeddings slightly improves performance empirically.

Note that, besides utilizing the cosine distance, we have used $\alpha = 1$ and $\beta = 1$ in (6) to obtain (11). The following result states that cosine contrastive loss is a valid *surrogate loss* (Bartlett et al., 2006) for the misclassification error.

Lemma 1. *Let $\nu \in (1, 2)$. The cosine contrastive loss in (12) is a surrogate-loss of the misclassification error, i.e.,*

$$\ell_{\text{ccl}}\big(f(\boldsymbol{x}), y\big) \geq 2(\nu - 1) \cdot [\![y \notin \text{Top}_1(f(\boldsymbol{x}))]\!], \quad (13)$$

where $\text{Top}_1(f(\boldsymbol{x}))$ denotes the indices of the classes that $f(\cdot)$ assigns the highest score for the instance $\boldsymbol{x}$.

Proof. If $y \in \text{Top}_1(f(\boldsymbol{x}))$, then $[\![y \notin \text{Top}_1(f(\boldsymbol{x}))]\!] = 0$. Since $\ell_{\text{ccl}}\big(f(\boldsymbol{x}), y\big) \geq 0$, we have

$$\ell_{\text{ccl}}\big(f(\boldsymbol{x}), y\big) \geq 2(\nu - 1) \cdot [\![y \notin \text{Top}_1(f(\boldsymbol{x}))]\!] \quad (14)$$

in this case. Now, let's consider the case when $y \notin \text{Top}_1(f(\boldsymbol{x}))$. For $a \in \mathbb{R}$, let $\phi(a) = (1 - a)^2$ and $\tilde{\phi}(a) = (\max\{0, \nu - 1 - a\})^2$. We have

$$\begin{aligned}
\ell_{\text{ccl}}\big(f(\boldsymbol{x}), y\big) &= \phi(s_y) + \sum_{c \neq y} \tilde{\phi}(-s_c) \\
&\geq \phi(s_y) + \tilde{\phi}(-\max_{c \neq y} s_c) \overset{(i)}{\geq} \tilde{\phi}(s_y) + \tilde{\phi}(-\max_{c \neq y} s_c) \\
&\overset{(ii)}{\geq} 2 \cdot \tilde{\phi}\Big(\big(s_y - \max_{c \neq y} s_c\big)/2\Big) \overset{(iii)}{\geq} 2 \cdot (\nu - 1) \\
&= 2(\nu - 1) \cdot [\![y \notin \text{Top}_1(f(\boldsymbol{x}))]\!], \quad (15)
\end{aligned}$$

where (i) follows as we have $\phi(a) \geq \tilde{\phi}(a), \forall\ a$ and (ii) utilizes the convexity of $\tilde{\phi}$. (iii) follows as we have $\tilde{\phi}(a) > \nu - 1$ for $a < 0$, and

$$y \notin \text{Top}_1(f(\boldsymbol{x})) \quad \Longleftrightarrow \quad s_y - \max_{c \neq y} s_c < 0.$$

The statement of the lemma follows from (14) and (15). □

Lemma 1 established that the cosine contrastive loss is a valid surrogate for the misclassfication error in the sense of (Bartlett et al., 2006). We note here the bound is tight. For example, for $\epsilon > 0$, suppose $\nu = 2 - \epsilon$, $s_y = -\epsilon$, $s_{y'} = \epsilon$ for some $y' \neq y$, and $s_c = -1 + \epsilon/2$, $c \neq \{y, y'\}$. Then, $\ell_{\text{ccl}}\big(f(\boldsymbol{x}), y\big) = 2(1 - \epsilon) + \epsilon^2 + 4\epsilon = 2(\nu - 1)[\![y \notin \text{Top}_1(f(\boldsymbol{x}))]\!] + \epsilon^2 + 4\epsilon$, which tends to $2(\nu - 1)[\![y \notin \text{Top}_1(f(\boldsymbol{x}))]\!]$ as ϵ goes to 0. One may follow similar analysis as in Reddi et al. (2019, Theorem 4) to show the *statistical consistency* (Zhang, 2004) of minimizing this loss.

We now explicate a connection between the classification-consistent cosine contrastive loss and the objective underlying the FedAwS algorithm. To do so, we assume that $n_1 = \cdots = n_C = \frac{n}{C}$, and note that FedAwS effectively seeks to collaboratively minimize

$$\mathcal{R}_{\text{sp}}(f) = \sum_{i \in [C]} \frac{n_i}{n} \cdot \hat{\mathcal{R}}_{\text{pos}}(\mathcal{S}^i) + \lambda \cdot \text{reg}_{\text{sp}}(W), \quad (16)$$

with the regularizer $\text{reg}_{\text{sp}}(W)$ from (7). Now we observe:

Proposition 2. *Suppose $\lambda = \frac{1}{C}$ and $n_1 = \cdots = n_C = \frac{n}{C}$. Then, FedAwS objective equals the empirical risk with respect to the loss function*

$$\ell_{\text{sp}}(f(\boldsymbol{x}), y) = (1 - s_y)^2 + \sum_{c \neq y} \big(\max\big\{0, \nu - 1 + \boldsymbol{w}_y^\top \boldsymbol{w}_c\big\}\big)^2, \quad (17)$$

i.e., $\mathcal{R}_{\text{sp}}(f) = \frac{1}{n} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \ell_{\text{sp}}(f(\boldsymbol{x}), y)$.

Proof. Note that

$$\begin{aligned}
\mathcal{R}_{\text{sp}}(f) &= \sum_{i \in [C]} \frac{n_i}{n} \cdot \hat{\mathcal{R}}_{\text{pos}}(\mathcal{S}^i) + \lambda \cdot \text{reg}_{\text{sp}}(W) \\
&= \frac{1}{n} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \ell_{\text{ccl}}^{\text{pos}}(f(\boldsymbol{x}), y) + \lambda \cdot \text{reg}_{\text{sp}}(W) \\
&= \frac{1}{n} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \ell_{\text{ccl}}^{\text{pos}}(f(\boldsymbol{x}), y) + \\
&\quad \lambda \sum_{y \in [C]} \sum_{c \neq y} \big(\max\big\{0, \nu - \boldsymbol{d}_{\cos}(\boldsymbol{w}_y, \boldsymbol{w}_c)\big\}\big)^2 \\
&\overset{(i)}{=} \frac{1}{n} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \Big(\ell_{\text{ccl}}^{\text{pos}}(f(\boldsymbol{x}), y) + \\
&\quad C\lambda \sum_{c \neq y} \big(\max\big\{0, \nu - \boldsymbol{d}_{\cos}(\boldsymbol{w}_y, \boldsymbol{w}_c)\big\}\big)^2\Big) \\
&\overset{(ii)}{=} \frac{1}{n} \sum_{(\boldsymbol{x}, y)} \Big((1 - s_y)^2 + \\
&\quad \sum_{c \neq y} \big(\max\big\{0, \nu - 1 + \boldsymbol{w}_y^\top \boldsymbol{w}_c\big\}\big)^2\Big) \\
&= \frac{1}{n} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \ell_{\text{sp}}(f(\boldsymbol{x}), y), \quad (18)
\end{aligned}$$

where (i) and (ii) follows from the assumptions that $n_1 = \cdots = n_C$ and $\lambda = \frac{1}{C}$, respectively. □

Note that the contribution of the negative labels in the loss function ℓ_{sp} is independent of the input embedding $g_{\boldsymbol{\theta}}(\boldsymbol{x})$.

Remark 2. Proposition 2 shows that the spreadout regularizer $\text{reg}_{\text{sp}}(W)$ can be viewed as the negative component of a conventional loss. The assumption of each user having equal number of examples makes this easy to see. Our analysis easily extends to unequal number of examples by using a weighted spreadout for each class.

Next, we utilize Proposition 2 to argue that the FedAwS objective $\mathcal{R}_{\text{sp}}(f)$ approximates the cosine contrastive loss – a valid surrogate for the misclassification error (cf. Lemma 1). Proposition 2 establishes that $\mathcal{R}_{\text{sp}}(f)$ corresponds to the empirical risk with respect to ℓ_{sp}. Note that $\ell_{\text{sp}}^{\text{pos}}(f(\boldsymbol{x}), y) =$

$\ell_{\rm ccl}^{\rm pos}(f(\boldsymbol{x}), y)$ (cf. (6)). Thus, the desired result follows by establishing that $\ell_{\rm sp}^{\rm neg}(f(\boldsymbol{x}), y)$ approximates $\ell_{\rm ccl}^{\rm neg}(f(\boldsymbol{x}), y)$. This approximation becomes better as the input embedding $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ gets closer to its class embedding $\boldsymbol{w}_y$, as encouraged by $\ell_{\rm sp}^{\rm pos}(f(\boldsymbol{x}), y)$.

Theorem 1. *Let $\nu \in (1, 2)$. Then, the loss $\ell_{\rm sp}$ in (17) satisfies*

$$\ell_{\rm ccl}(f(\boldsymbol{x}), y) - (1+2\nu) \cdot \sum_{c \neq y} |\boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}| \leq \ell_{\rm sp}(f(\boldsymbol{x}), y) \leq \ell_{\rm ccl}(f(\boldsymbol{x}), y) + (1+2\nu) \cdot \sum_{c \neq y} |\boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}|, \quad (19)$$

where $\boldsymbol{r}_{\boldsymbol{x},y} = \boldsymbol{w}_y - g_{\boldsymbol{\theta}}(\boldsymbol{x})$.

Proof. Note that $\boldsymbol{r}_{\boldsymbol{x},y} = \boldsymbol{w}_y - g_{\boldsymbol{\theta}}(\boldsymbol{x})$ denotes the mismatch between $\boldsymbol{w}_y$ and $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. Thus,

$$\boldsymbol{w}_y^\top \boldsymbol{w}_c = g_{\boldsymbol{\theta}}(\boldsymbol{x})^\top \boldsymbol{w}_c + \boldsymbol{r}_{\boldsymbol{x},y}^\top \boldsymbol{w}_c = s_c + \boldsymbol{r}_{\boldsymbol{x},y}^\top \boldsymbol{w}_c.$$

As a result $\ell_{\rm sp}$ in (17) can be written as

$$\begin{aligned}
&\ell_{\rm sp}(f(\boldsymbol{x}), y) \\
&= (1 - s_y)^2 + \sum_{c \neq y} \left(\max\left\{0, \nu - 1 + s_c + \boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}\right\}\right)^2 \\
&= (1 - s_y)^2 + \sum_{c \neq y} \left(\max\left\{0, \nu - 1 + s_c\right\}\right)^2 + \sum_{c \neq y} \Delta_c \\
&= \ell_{\rm ccl}(f(\boldsymbol{x}), y) + \sum_{c \neq y} \Delta_c, \qquad (20)
\end{aligned}$$

where

$$\Delta_c := \left(\max\left\{0, \nu - 1 + s_c + \boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}\right\}\right)^2 - \left(\max\left\{0, \nu - 1 + s_c\right\}\right)^2. \quad (21)$$

The result follows from (20) and Claim 1 below. □

Claim 1. *Given an instance and label pair $(\boldsymbol{x}, y)$ and the scorer f, for $c \neq y$, let Δ_c be as defined in (21). Then,*

$$|\Delta_c| \leq 2(1+2\nu) \cdot \left|\boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}\right|. \quad (22)$$

Proof. Let $a = \nu - 1 + s_c$ and $b = \boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}$. Thus, we want to show that

$$\left|\left(\max\left\{0, a + b\right\}\right)^2 - \left(\max\left\{0, a\right\}\right)^2\right| \leq (1+2\nu) \cdot |b|.$$

Let us consider four possible cases.

- **Case 1 ($a + b < 0$ and $a < 0$).** In this case, we have

$$\left|\left(\max\left\{0, a + b\right\}\right)^2 - \left(\max\left\{0, a\right\}\right)^2\right| = 0.$$

- **Case 2 ($a + b > 0$ and $a > 0$).** Note that

$$\begin{aligned}
&\left|\left(\max\left\{0, a + b\right\}\right)^2 - \left(\max\left\{0, a\right\}\right)^2\right| \\
&= |(a+b)^2 - a^2| = |b(b+2a)| \leq (1+2\nu) \cdot |b|,
\end{aligned}$$

  where the last inequality follows from the fact that $a = \nu - 1 + s_c \leq \nu$, since $s_c \leq 1$.

- **Case 3 ($a + b > 0$ and $a < 0$).** In this case,

$$\begin{aligned}
&\left|\left(\max\left\{0, a + b\right\}\right)^2 - \left(\max\left\{0, a\right\}\right)^2\right| \\
&= \left|\left(\max\left\{0, a + b\right\}\right)^2\right| \leq |b^2| \leq |b|,
\end{aligned}$$

  where the last equality follows as $|b| = |\boldsymbol{w}_c^\top \boldsymbol{r}_{\boldsymbol{x},y}| \leq 1$.

- **Case 4 ($a + b < 0$ and $a > 0$).** Note that

$$\begin{aligned}
&\left|\left(\max\left\{0, a + b\right\}\right)^2 - \left(\max\left\{0, a\right\}\right)^2\right| \\
&= \left|\left(\max\left\{0, a\right\}\right)^2\right| \leq |a|^2 \overset{(i)}{\leq} |b|^2 \leq |b|,
\end{aligned}$$

  where (i) follows as by combining $a > 0$ and $a + b < 0$ we obtain the order $b < -a < 0 < a$.

Now, by combining all the four case above and using the fact that $\nu \in (1, 2)$, we obtain the desired the result. □

As a final remark, our analysis above assumed that the cosine contrastive loss (11) uses *all* labels $c \neq y$ as "negatives" for the given label y. However, using similar ideas as in (Reddi et al., 2019), we may easily extend our analysis to the case where the loss uses the k *hardest labels* as negatives (cf. (8)).

6. Experiments

We empirically evaluate the proposed FedAwS method on benchmark image classification and extreme multi-class classification datasets. In all experiments, both the class embedding $\boldsymbol{w}_c$'s and instance embedding $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ are ℓ_2 normalized, as we found this slightly improves model quality.

For FedAwS, we use the squared hinge loss with cosine distance to define $\hat{\mathcal{R}}_{\rm pos}(\mathcal{S}^i)$ at the clients (cf. Algorithm 1):

$$\ell^{\rm pos}(f(\boldsymbol{x}), y) = \max\left(\left\{0, 0.9 - g_{\boldsymbol{\theta}}(\boldsymbol{x})^\top \boldsymbol{w}_y\right\}\right)^2. \quad (23)$$

This encourages all positive instance and label pairs $(\boldsymbol{x}, y)$ to have dot product larger than 0.9 in the embedding space.

We compare the following methods in our experiments.

- **Baseline-1**: Training with only positive squared hinge loss. As expected, we observe very low precision values because the model quickly collapses to a trivial solution.
- **Baseline-2**: Training with only positive squared hinge loss with the class embeddings fixed. This is a simple way of preventing the class embeddings from collapsing into a single point.

Dataset	Model	Baseline-1	Baseline-2	FedAwS	Softmax (Oracle)
CIFAR-10	RESNET-8	10.7	83.3	86.3	88.4
CIFAR-10	RESNET-32	9.8	92.1	92.4	92.4
CIFAR-100	RESNET-32	1.0	65.1	67.9	68.0
CIFAR-100	RESNET-56	1.1	67.5	69.6	70.0

Table 1: Precision@1 (%) on CIFAR-10 and CIFAR-100.

Dataset	#Features	#Labels	#TrainPoints	#TestPoints	Avg. #I/L	Avg. #L/I
AMAZONCAT	203,882	13,330	1,186,239	306,782	448.57	5.04
WIKILSHTC	1,617,899	325,056	1,778,351	587,084	17.46	3.19
AMAZON670K	135,909	670,091	490,449	153,025	3.99	5.45

Table 2: Summary of the datasets used in the paper. #I/L is the number of instances per label, and #L/I is the number of labels per instance.

- **FedAwS**: Our method with stochastic negative mining (cf. Section 4.2).
- **Softmax**: An oracle method of regular training with the softmax cross-entropy loss function that has access to both positive and negative labels.

6.1. Experiments on CIFAR

We first present results on the CIFAR-10 and CIFAR-100 datasets. We trained ResNets (RESNETS) (He et al., 2016a;b) with different number of layers as the underlying model. Specifically, we train RESNET-8 and RESNET-32 for CIFAR-10; and train RESNET-32 and RESNET-56 for CIFAR-100 with the larger number of classes.

From Table 1, we see that on both CIFAR-10 and CIFAR-100, FedAwS almost matches or comes very close to the performance of the oracle method which has access to all labels. The first baseline method, training with only positive squared hinge loss does not lead to any meaningful precision values. In this case, as discussed above the model collapses into a degenerate solution.

Interestingly, the naïve way of preventing the embeddings from collapsing by fixing the class embeddings as their random initialization gives a much better result. In fact, on CIFAR-10 with RESNET-32, Baseline-2 performs almost identically to the oracle and FedAwS. The reason behind this good performance is that with a smaller number of classes, at a random initialization in a high-dimensional space (64 in this case), the class embeddings are already well spread-out as they are almost orthogonal to each other. In addition, the 10 classes of CIFAR-10 are not related to each other. This makes the 10 nearly-orthogonal vectors ideal to be used as-is for class embeddings.

6.2. Experiments on extreme-multiclass classification

Datasets. We test the proposed approach on standard extreme multilabel classification datasets (Varma, 2018). These datasets have a large number of classes, and therefore are a good representatives of the applications of federated learning with only positive labels. Similar to Reddi et al. (2019), because these datasets are multi-label, we uniformly sample positive labels to obtain datasets corresponding to multi-class classification problems. The datasets and their statistics are summarized in Table 2.

Model architecture. We use a simple embedding-based classification model wherein an instance $\boldsymbol{x} \in \mathbb{R}^{d'}$, a high-dimensional sparse vector, is first embedded into $\mathbb{R}^{512}$ using a linear embedding lookup followed by averaging. The vector is then passed through a three-layer neural network with layer sizes 1024, 1024 and 512, respectively. The first two layers in the network apply a ReLU activation function. The output of the network is then normalized to obtain instance embeddings with unit ℓ_2-norm. Each class is represented as a 512-dimensional normalized vector.

Training setup. SGD with a large learning rate is used to optimize the embedding layers, and Adagrad is used to update other model parameters. In each round, we randomly select 4K clients associated with 4K labels.

In addition to the methods used in the CIFAR experiments, we also compare the FedAwS with SLEEC (Hadsell et al., 2006). This is an oracle method of regular training with access to both positive and negative labels.

Results. We report precision@k for $k \in \{1, 3, 4\}$ in Table 3. On all the datasets, FedAwS largely outperforms the two baseline methods of training with only positive labels. On both AMAZONCAT and AMAZON670K, it matches or comes very close to the performance of Softmax and SLEEC. Baseline-2 gives reasonable (although quite sub-optimal) performance on AMAZONCAT; but does not work on AMAZON670K and WIKILSHTC which have larger number of classes. Thus, randomly initialized class embeddings are not ideal in the situation of many classes, and it is crucial to train the class embeddings with the rest of the model.

Meta parameters. There are two meta parameters in the proposed method: the learning rate multiplier of the spread-out loss λ (cf. Algorithm 1), and the number top confusing labels considered in each round k (cf. (8)). To make a fair

		Federated Learning with Only Positives			Oracle	
		Baseline-1	Baseline-2	FedAwS	Softmax	SLEEC
AMAZONCAT	P@1	3.4	64.1	92.1	92.1	90.5
	P@3	3.2	46.8	70.8	77.9	76.3
	P@5	3.1	32.6	58.7	62.3	61.5
AMAZON670K	P@1	0.0	4.3	33.1	35.2	35.1
	P@3	0.0	2.8	29.6	31.6	31.3
	P@5	0.0	2.2	27.4	29.5	28.6
WIKILSHTC	P@1	7.6	7.9	37.2	54.1	54.8
	P@3	4.5	3.4	22.6	38.8	33.4
	P@5	2.8	2.6	16.2	29.9	23.9

Table 3: P@1,3,5 (%) of different methods on AMAZONCAT, AMAZON670K and WIKILSHTC.

	Baseline-1	Baseline-2	k = 10	k = 100	k = 500	k = all	$\lambda = 1$	$\lambda = 10$	$\lambda = 100$
P@1	3.4	64.1	26.3	92.1	86.9	87.7	73.2	92.1	92.2
P@3	3.2	46.8	21.5	70.8	66.1	69.7	50.2	70.8	71.7
P@5	3.1	32.6	18.2	58.7	49.3	52.2	40.4	58.7	57.9

Table 4: P@1,3,5 (%) of different meta parameters on AMAZONCAT.

comparison with other methods which do not have these meta parameters, in all of our other experiments in Table 3, we simply use $k = 10$ and $\lambda = 10$.

We perform an analysis of these two parameters in Table 4 on the AMAZONCAT dataset. The reason for the bad performance for a small k is that most of the picked labels are in fact positives in this setting (due to the inherent multi-label nature of the dataset), and over spreading the positive classes is not desirable. On the other hand, a very large k leads to sub-optimal performance, verifying the benefit and requirement of stochastic negative mining. That said, we would like to emphasize that once k is large enough, FedAwS is robust with respect to k. Even with $k = $ all (i.e., no SNM), FedAWS is still far better than the baselines.

Regarding λ, a relatively large value (10 or 100) is necessary to ensure the class embeddings are sufficiently spreadout.

7. Conclusion

We studied a novel learning setting, federated learning with only positive labels, and proposed an algorithm that can learn a high-quality classification model without requiring negative instance and label pairs. The idea is to impose a geometric regularization on the server side to make all class embeddings spreadout. We justified the proposed method both theoretically and empirically.

One can extend the identity-based class embeddings to the settings where the class embeddings are generated from class-level features. In addition, we notice that negative sampling techniques are crucial to make conventional extreme multiclass classification work. The proposed method is of independent interest in this setting because it replaces negative sampling altogether by imposing a strong geometric regularization. Further, even though our proposed method achieves near oracle performance on multiple datasets, exploring a fundamental trade-off between the performances in our novel setting and the oracle setting is another interesting future research direction.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Agarwal, N., Suresh, A. T., Yu, F. X. X., Kumar, S., and McMahan, B. cpSGD: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pp. 7564–7575, 2018.

Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., et al. Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*, 2019.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for federated learn-

ing on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Chechik, G., Sharma, V., Shalit, U., and Bengio, S. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition*, pp. 539–546, 2005.

Dietterich, T. G. and Bakiri, G. Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *AAAI*, pp. 572–577, 1991.

du Plessis, M., Niu, G., and Sugiyama, M. Convex formulation for learning from positive and unlabeled data. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 1386–1394, Lille, France, 07–09 Jul 2015. PMLR.

Elkan, C. and Noto, K. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–220, 2008.

Guo, C., Mousavi, A., Wu, X., Holtmann-Rice, D. N., Kale, S., Reddi, S., and Kumar, S. Breaking the glass ceiling for embedding-based classifiers for large output spaces. In *Advances in Neural Information Processing Systems*, pp. 4944–4954, 2019.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *Computer Vision and Pattern Recognition*, volume 2, pp. 1735–1742, 2006.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Hsieh, C.-J., Natarajan, N., and Dhillon, I. Pu learning for matrix completion. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 2445–2453. PMLR, 07–09 Jul 2015.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

Liu, B., Lee, W. S., Yu, P. S., and Li, X. Partially supervised classification of text documents. In *International Conference on Machine Learning*, volume 2, pp. 387–394, 2002.

Manevitz, L. M. and Yousef, M. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625, 2019.

Moya, M. M. and Hush, D. R. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.

Pujol, O., Radeva, P., and Vitria, J. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1007–1012, 2006.

Reddi, S. J., Kale, S., Yu, F., Holtmann-Rice, D., Chen, J., and Kumar, S. Stochastic negative mining for learning with large output spaces. *Artificial Intelligence and Statistics*, 2019.

Varma, M. Extreme classification repository. Website, 8 2018. http://manikvarma.org/downloads/XC/XMLRepository.html.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Yu, F. X., Cao, L., Feris, R. S., Smith, J. R., and Chang, S.-F. Designing category-level attributes for discriminative visual recognition. In *Computer Vision and Pattern Recognition*, pp. 771–778, 2013.

Zhang, T. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, 32(1):56–85, 02 2004.

Zhang, X., Yu, F. X., Kumar, S., and Chang, S.-F. Learning spread-out local feature descriptors. In *International Conference on Computer Vision*, pp. 4595–4603, 2017.