DEC used in a federated setting

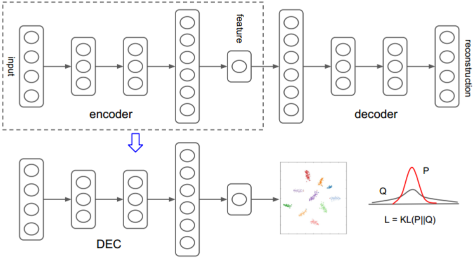
Lorenzo Sani

Università degli Studi di Bologna

September 28, 2021

Deep Embedding for Clustering (DEC)[3]

- Aims:
  - Learning feature representations of data ($\mathbb{B}^{40}$ data space $\Rightarrow$ $\mathbb{R}^f$ feature space)
  - Learning cluster assignments (number of clusters f is an hyperparameter)



- Stacked denoising Autoencoder (SAE)
- DNN optimized for the clustering objective

Stacked Denoising Autoencoder (SAE)

- Random Flipping Noise for binary data, respecting the frequencies of the whole dataset
- 5 layer depth Dense Neural Network with random dropout, shapes: 40-26-26-100-f
- Tied Dense layers is used for the decoder part
- Constraints and regularizers typical for SAE were tested
  - Unit Norm: constrains the weights incident to each hidden unit to have unit norm
- Binary Cross Entropy (BCE) loss function for reconstruction of binary data

Training of SAE

- Pretrain
  - 2500, or 5000 epochs
  - dropout rate was set equal to random flipping rate, many values were tested, 20%, 10%, 5%, 1%
  - minimizing BCE optimized by Stochastic Gradient Descent, decaying learning rate, 0.1 then divided by 10 every 1000, or 2000 epochs
- Finetune
  - 5000, or 10000 epochs, double the pretrain epochs
  - without dropout and random flipping, 0%
  - minimizing BCE optimized by Stochastic Gradient Descent, decaying learning rate, same used for pretraining

Clustering Model

- Extract finetuned Encoder part from SAE, without dropout and random flipping
- Attach a statistical clustering layer
  - clusters' centroids in feature space initialized by k-means
  - auxiliary probability distribution, Student's t, used as a kernel to measure the similary between embedded points and clusters' centroids
- Kullback-Leibler Divergence (KLD) loss is used
  - measures how many information we expect to lose approximating the the actual probaility distribution with the one obtained with the soft assignment

Clustering Model training

- Initialize clusters' centroids using k-means algorithm (clients)
  - 25 different random initializations
  - max 300 epochs
  - look for f centroids
- Aggregate initial centroids between clients (server)
  - random sampling of one client's first centroid
  - add iteratively to the list of aggregated centroids the farest centroid available from the list of all clients' centroids until f centroids are collected
- Initialize the clustering layer with the aggregated centroids, these are "weights" to optimize
- Train for 10000 epochs, updating every 100 epochs the auxiliary distribution and the soft assignments
- Optimize using SGD with a fixed learning rate of 0.1

Federated training

- FedAvg algorithm[2] is used for aggregating SAE and Clustering model weights
- The entire dataset of 2043 sample patients is divided in equal number between clients
  - 2 clients set up, client 1 has 1021 samples and client 2 has 1022 samples
  - 4 clients set up, client 1,2,3 have 510 samples and client 4 has 513 samples
  - 6 clients set up, client 1,2,3,4,5 have 340 samples and client 6 has 343 samples
  - 8 clients set up, client 1,2,3,4,5,6,7 have 255 samples and client 8 has 258 samples
- Every client is weighted only by the number of samples
- FLOWER framework[1] was used to implement Federated Learning (FL)

Metrics

- SAE
    - Accuracy between original data and reconstructed data
    - Rounded accuracy between original data and rounded reconstructed data
    - G metric, ratio between train loss and evaluation loss
- Clustering Model
    - Cycle accuracy of labels assignment:
      accuracy between predicted assignment of real data and predicted assignment of reconstructed data
    - Number of samples whose label prediction change w.r.t. the previous epoch
    - G metric, ratio between train loss and evaluation loss

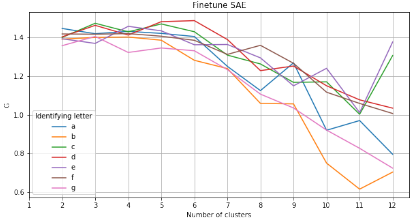
Comparison between different models

Identifying letter	AE epochs	Dropout rate	Random Flip rate	Unit Norm
a	2500	0.2	0.2	No
b	2500	0.05	0.05	No
c	2500	0.2	0.2	Yes
d	2500	0.1	0.1	Yes
e	2500	0.05	0.05	Yes
f	2500	0.01	0.01	Yes
g	5000	0.01	0.01	Yes

- Different features space dimension (f) were tested:
  2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
  for every model
- Different numbers of clients were tested:
  2, 4, 6, 8, for only some models

G metric

SAE



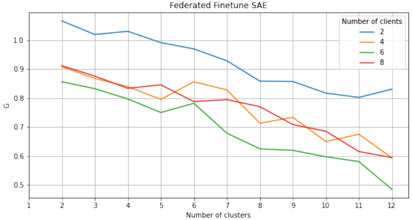
Clustering Model

Other Clustering Model metrics

Cycle Accuracy

Federated G metric

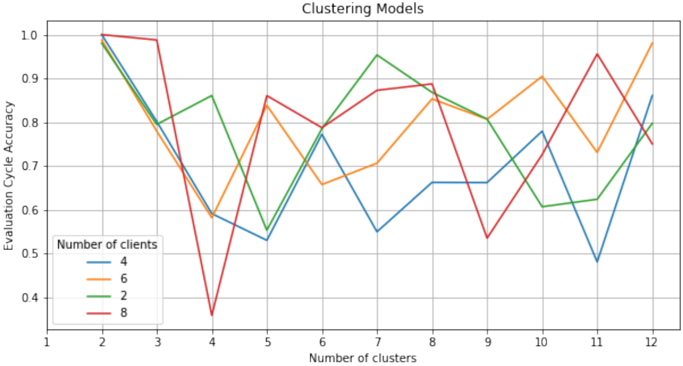
SAE



Clustering Model

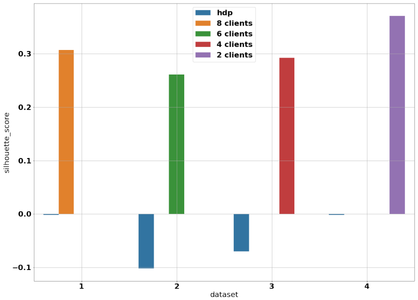
Other Federated Clustering Model metrics

Cycle Accuracy



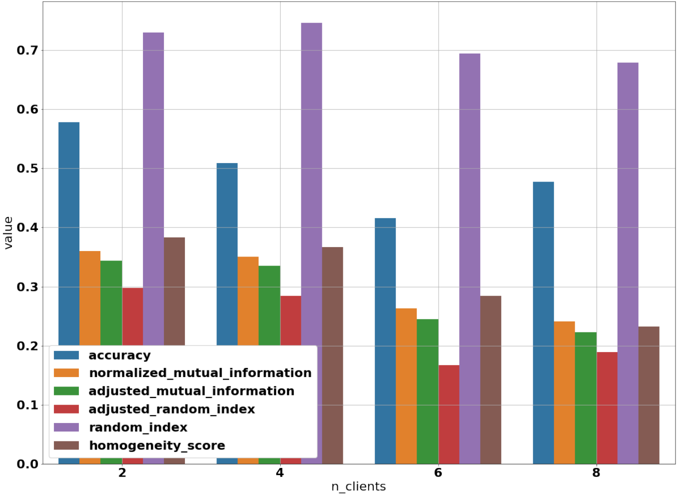
Objective Metrics for Comparison w.r.t. HDP

Silhouette Score

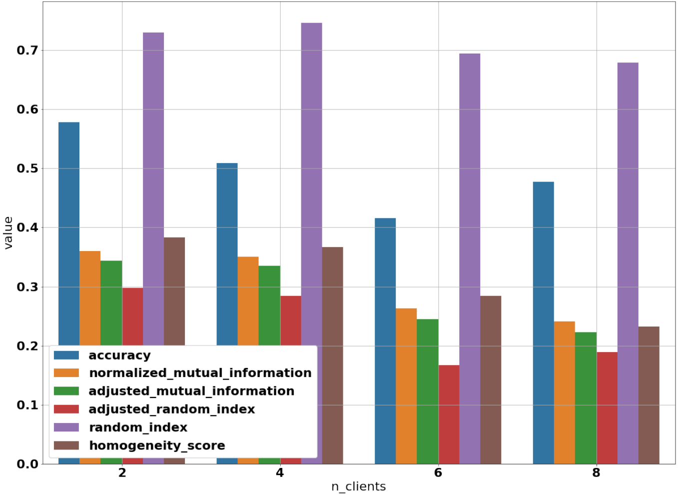


Metrics for comparison vs HDP

Clusters separation

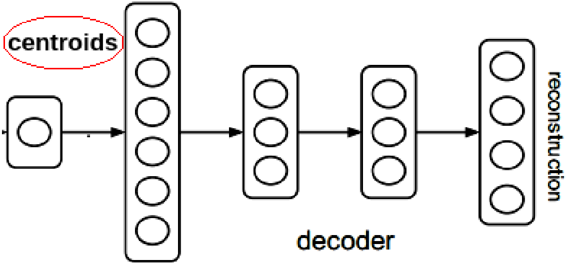


Clusters properties



Conclusions

- Metrics and representation suggest that federated DEC separates well clusters in the feature space
- Federated DEC is able to predict clusters' properties in the data space using the decoder part feeded with final centroids



Future work

- Tests on different federated set-ups with different aggregation algorithms
- Study proprieties of the clusters using the final DEC feature representation
- Test DEC on a wider, or at least different, slice of the dataset

Bibliography

Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro P. B. de Gusmão, and Nicholas D. Lane.
Flower: A friendly federated learning research framework, 2021.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas.
Communication-efficient learning of deep networks from decentralized data, 2017.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi.
Unsupervised deep embedding for clustering analysis.
CoRR, abs/1511.06335, 2015.

Thanks to
Gastone Castellani, Enrico Giampieri,
Daniele Dall'Olio, Tommaso Matteuzzi

The End,
Thanks for the attention