Addressing Statistical Heterogeneity in Federated Learning For Sea Ship Datasets

Serena Lin\(^1\), Emon Dey\(^2\), Anuradha Ravi\(^2\), Nirmalya Roy\(^2\)
\(^1\)Department of Electrical and Computer Engineering, Northeastern University
\(^2\)Department of Information Systems, University of Maryland in Baltimore County
We propose a method to homogenize heterogeneous datasets for training a federated learning model and determining necessary granularity for accurate model performance.
Machine Learning vs Federated Learning

Traditional Machine Learning

Federated Learning Architecture
Federated Learning (FedML)

- **Advantage**
  - Protects user privacy
    - Sends model weights

- **Disadvantage**
  - Slower weight updates
    - Slower convergence
      - Minimize the loss function
  - Sensitive to heterogeneity
    - Datasets
Statistical Heterogeneity

- **Causes**
  - Skewed label distribution
  - Skewed feature distribution
  - Granularity differences

- **Approach**
  - Ignore annotations and recluster based on images
  - Use annotations to confirm reclustering
  - Must determine number of classes
Methodology

- **Annotated Ship Datasets**
  - ABOShips, Seaships, VIS onshore and offshore
- **Python Scripts:**
  - Crop and Sort Images
  - Extract Features
  - Create T-SNE Plot
    - Determine Perplexity Value
  - Recluster Images & Reannotate
- **Future:** Use to train model
Dataset Preparation: Cropping

- ABOShips, Seaships, VIS offshore and onshore
- Annotations
  - Seaship boundaries
    - X min
    - X max
    - Y min
    - Y max
  - Boat Class
    - Ex: cargo ship, passenger ship, cruise-boat, bulk cargo carrier
- Crop and categorize into class folders
Dataset Preparation: Feature Extraction

- Convert images to vectors based on averaging feature vectors the algorithm recognizes and extracts
  - Stores the features as a numpy array
- Off the shelf resnet feature extractor (CNN)
  - Github repository: img2vec
  - Fixed classes
- Allows direct numerical image comparison
  - Similarity score csv files
- Customized Python Scripts

<table>
<thead>
<tr>
<th>Standard</th>
<th>Standard 2</th>
<th>Standard 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder 1</td>
<td>Folder 2</td>
<td>Similarity</td>
</tr>
<tr>
<td>Boat</td>
<td>Miscellaneous</td>
<td>0.97630334</td>
</tr>
<tr>
<td>Boat</td>
<td>Passengership</td>
<td>0.8921411</td>
</tr>
<tr>
<td>Boat</td>
<td>Motorboat</td>
<td>0.97385685</td>
</tr>
<tr>
<td>Boat</td>
<td>Ferry</td>
<td>0.8302561</td>
</tr>
<tr>
<td>Boat</td>
<td>Militaryship</td>
<td>0.7427224</td>
</tr>
<tr>
<td>Boat</td>
<td>Miscboat</td>
<td>0.9201295</td>
</tr>
<tr>
<td>Boat</td>
<td>Cruiseship</td>
<td>0.77466377</td>
</tr>
<tr>
<td>Boat</td>
<td>Sailboat</td>
<td>0.6982449</td>
</tr>
<tr>
<td>Boat</td>
<td>Seamark</td>
<td>0.9686793</td>
</tr>
<tr>
<td>Boat</td>
<td>Cargoship</td>
<td>0.764315</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Boat</td>
<td>0.97630334</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Passengership</td>
<td>0.82774088</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Motorboat</td>
<td>0.9299425</td>
</tr>
</tbody>
</table>

aboships_similarity.csv

img2vec simulation
T-SNE Plot

- t-distributed stochastic neighbor embedding (T-SNE)
- Nonlinear dimensionality reduction algorithm to reduce dimensionality
  - Clusters similar points together and distance between different clusters
- Perplexity value
  - If low, tendency is too many points together in a cluster & will not increase distance between different clusters
  - If high, opposite occurs
- Perplexity vs Divergence Graphs: pinpoint correct value
  - Divergence quantifies the difference between 2 probability distributions (ie clusters)
  - We find the minimum divergence before stabilization and take its perplexity value
We group ships based on features.

T-SNE allows cluster visualization of similarities/differences between classes.

Python Script

Set perplexity level to previously determined values.
Future Steps

● Cluster the datasets together
● Apply method to the federated learning setting
  ○ Integrate with FedML platform cross-silo edge devices
● Impact: can be applied to preparing many different types of image datasets
  ○ Is usable strategy for homogenization
Skills Learned Specific to Project

● Fundamentals of Machine & Federated Learning
  ○ Math behind the models: gradient descent algorithms, convolutional neural networks (cnn), loss functions. Back batch propagation, feature selection, unsupervised/supervised learning, bias-variance tradeoff

● Ubuntu Linux Terminal
  ○ Install and execute programs and code

● FedML Simulations and ML-ops Platform

● Github

● Python
  ○ Libraries: tensorflow, pytorch, sklearn, matplotlib
  ○ File image cropping, feature extraction, t-sne plot creation, perplexity scores, csv file read and write
Other Research Skills Learned

- Robot Operating System (ROS)
  - Fundamentals, writing publisher and subscribers in c++ and python
- Google Colab: keras machine learning model
- Semantic Segmentation Editor: Lidar
- Overleaf: LaTeX
  - Documentation & IEEE Paper Formatting
References


Acknowledgements

- Emon Dey
- Dr. Anuradha Ravi
- Dr. Nirmalya Roy
- PhD Students in the Lab
- Ms. Marjory Pineda
- Fellow REU students

This work has been partially supported by ONR Grant #N00014-23-1-2119, U.S. Army Grant #W911NF2120076, NSF CAREER Award #1750936, NSF REU Site Grant #2050999 and NSF CNS EAGER Grant #2233879.
Q & A