Summary on Transfer Learning via LDF to estimate $PM_{2.5}$ levels

Aman Patel

June 29, 2024

Introduction

PM2.5 pollution being a major public health concern, accurately predicting PM2.5 levels can help to reduce its adverse effects, especially for data poor regions where a good amount of training data is not available. This research paper addresses this problem by leveraging transfer learning. But existing TL methods transfer knowledge in a general manner and often rely on just alignment of features for both source and target domains, they fail to capture the intricate semantic and spatial dependencies which both locations might possess and have similar PM2.5 levels. So to efficiently capture this, they introduced a novel feature *LDF* which explicitly captures both the dependencies using a two-stage autoencoder leading to a better approach of transfer learning with a significant increase in accuracy.

Methodology

There are multiple sensors installed at several regions of both source and target locations to collect the data i.e. to capture and generate samples having values of geographical, topographical, meteorological and other related features. The significance of using auto-encoder model is feature extraction which basically captures both dependencies by reducing the dimension of input data to single dimension(here) called *LDF*. Below are steps used in extracting *LDF* and implementing *ITL*;

1 Cloud Formation : A cluster is formed where each cluster represents group of sensors whose samples have similarity in their feature vectors and importantly these sensors can be from either source or target domains and thereby capturing spatial proximity in source domain as well as semantic correlation with target domain. The neighborhood clouds selection can be done by selecting top m(some parameter) sensors based on calculated distance by formula(my assumption);

$$d_{ij} = \sqrt{\sum_k (x_{i,k} - x_{j,k})^2} \tag{1}$$

where $x_{i,k}$ and $x_{j,k}$ are the k-th features of sensors i and j, respectively.

2 Autoencoder Architecture : Assume we got n clouds containing sensors with similar features and PM2.5 values. The objective of this two-step process is to transform the input data to *LDF* and then reconstruct the original data from it, making *LDF* capable of feature extraction from each cloud to explicitly capture the dependencies in a much better way.

2.1 Encoder-Decoder : It takes the features of each sensor in the cloud C_i and compresses them into latent representation z, example

$$z_i = f(x_i) \tag{2}$$

where f is the encoding function, typically implemented using layers of convolutional or dense neural networks. The decoder then takes this latent representation z_i and reconstructs the original features and PM2.5 levels with similar decoding function. This process is repeated for batches of clouds and ensures that decoder's output closely matches the input features by minimizing the reconstruction loss, commonly used as

$$L_{recon} = \frac{1}{m} \sum_{i=1}^{m} ||x_i - \hat{x_i}||^2 \tag{3}$$

The stage 2 encoder-estimator again generates the latent representation z_i from features in cloud C_i but using the trained encoder in stage 1. At last the estimator takes this *LDF* to predict PM2.5 level $\hat{y}_i$ using some prediction function implemented using dense neural network and typical *MSE* loss function.

3 Reweighting : *ITL* holds a crucial role as the initial reweighting is performed to adjust the importance of samples from the source domain, so that they better represent the target domain and it improves the model's generalization to the target domain, particularly when there are significant differences between the two domains. Then reweighted source domain and target domain is combined and **LDF** values are integrated to each sample. Then *ITL* is performed using some weighted-loss function that leverages LDF to capture dependencies and improve generalization, leading to accurate predictions for target domain.

4 Important findings : This *TL* technique leveraging *LDF* is robust in scenarios with limited target domain data and it significantly improves PM2.5 prediction accuracy by effectively capturing spatial and semantic dependencies. The key part is making clouds of data-samples of sensors and train an auto-encoder model to extract such a feature from each cluster that identifies the dependencies and integrated that into combined set making a *rich feature set* for *ITL*. And most importantly, reweighting of source domain samples to align with target domain helped in better generalization to new, unseen data.

Limitations

The effectiveness of *TL* methods is highly dependent on the quality and relevance of the source domain data and if they're not well aligned even after reweighting, it might not yield significant improvements. Also since the model is designed for prediction for PM2.5 prediction, its applicability to other pollutants is not guaranteed because different pollutants may possess different dependencies to accurately predict their levels. The complexity of the model can impose limitation since the proposed method involves multiple steps starting from cloud formation to *LDF* calculation via auto-encoder training to reweighting for *ITL*, and having dynamic environmental factors, location with with numerous sensors and diverse features can pose scalability challenges too.

Future Scope

Incorporating hybrid approaches like using Reinforcement learning(*RL*) that can continuously adapt the weights based on feedback from the target domain, leading to more accurate and relevant weighting. The encoder training can be narrowed down to more than one dimension instead of only *LDF* and that may improve the interpretability of feature dependencies. And exploring this methodology to other domains like energy-consumption, disease outbreak monitoring etc. could demonstrate its broader impact.