City GDP and Population Prediction Using Machine Learning Algorithms

Wentao Li*
*George Washington University, unlimitediw@gwu.edu

***Abstract*—The current methods for measuring city's Gross domestic product (GDP) and Population are mainly city government collection and it is not easy to find a good estimate of a particular city's GDP[1]. In this project, I investigate the way using ground truth geographical features such as green area and satellite map to predict the city's GDP and Population. To evaluate city's GDP, I apply the geographic and administrative features to my Support Vector Regression (SVR) and Multilayer Percetron (MLP) based regression model. For predicting city's Population, I train several Convolutional Nerural Network (CNN) models with city's map images. Through extensive experiment, it is showed that the SVR based regression model with rbf kernel function outperforms other more sophisticated models such as MLP and Adaboost in terms of both accuracy and speed. At the same time, the CNN model with VGG16 architecture yields a very competitive result on population level classification problem.**

***Keywords*—Machine Learning, Support Vector Regression, Multilayer Perceptron and Convolution Neural Network.**

I. Introduction

In this projects, I focus on solving two problems. One is to predict the city's GDP value with geographic and administrative features of city. The other one is to classify the city's Population level with city's road map and satellite map images.

In the first problem, I predict the exact city's GDP which is a continuous numeric value. A wide range of machine learning methods such as SVR, MLP and Adaboost which can deal with this regression problem. Support vector machine is a machine learning method finding the maximum margin separating hyperplane. Chih-Chung Chang and Chih-Jen Lin provide a clear open source library[3] that using e-SVR[4] and v-SVR[5] to solve regression problem. Multi layer perceptron is a feedforward neural network and it is basically the same as the single layer perceptron but adding the hidden layers and activation function gives it nonlinearity and makes it a universal approximator. Meanwhile, Adaboost is a boosting method that convert a set of weak learner to a strong learner and it is normally based on decision tree weak learner while Harris shows a Improveing Regressor[2] to solve regression problem with boosting method. Normally it is appropriate to apply Adaboost in previous to make sure the problem is predictable.

I firstly choose SVR because support vector machine has a high performance with a small dataset (229 cities' data) and kernel functions such as rbf and polynomial make the model non-linear. SVM finds the hyperplane that best separates dataset and is more appropriate for small dataset because it will take too long to process and too many memory space for large dataset. The dual form of SVR is showing below.

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T Q(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + \boldsymbol{z}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) \\ \text{subject to} \quad & \boldsymbol{e}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) = 0,\ \boldsymbol{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) \le C\nu, \\ & 0 \le \alpha_i, \alpha_i^* \le C/l, \qquad i = 1,\dots,l. \end{aligned}$$

In this part, I implement my Sequential Minimal Optimal (SMO) method to solve this dual problem.

As a comparison, I also use the MLP model to predict GDP with self design hidden layers and different activation function. This MLP based model is non-linear as well and works better with large dataset. In this project, after eliminate the administrative features and several geographic features, the city dataset can enlarge to 3953 from 229 and only the coordinate, country name and population features are left. Apart from performance comparison on the same dataset, I specifically compare the performance of SVR model trained with small dataset but more features and MLP model trained with large dataset but less features. In this part, I implement the feedforward and backpropagation algorithm with friendly API for layer and activation function adujusting and optimizer usage.

In the second problem, I train several CNN models with city's road map and satellite map image dataset to classify the city population from 200,000 level to 2,000,000 level. Similar to MLP, CNN also needs the feedforward and backward steps. But only the last layers of it are fully connected and there are convolutional layer to adjust the features in small region, pooling layer to reduce the spatial size and control overfitting.

Many features of the previous problem such as green area and polycentricity are directly reflected on the city map and the satellite map images. The Convolutional layer is especially good at processing this kind of spatial features. For instance, the [[1,0,-1],[0,0,0],[-1,0,1]] filter can easily find the edge of an image. With different kind of convolutional layer, the model can find the low-level, mid-level feature and high-level feature of the map by itself especially to the satellite map.

In this project, I firstly use my self design 8 and 14 layers (the layer with weights) CNN and then apply two sophisticated CNN architectures, VGG16 and VGG19[6] to my model and finally get a 92% accuracy result in the 10 class classification problem. Moreover, I also use the CNN model to classify the GDP level in the first problem and get a high performance.

II. Problem Statement

A. City's GDP Prediction

Overview: In this part, I need to find a way to use city's geographic and administrative features to evaluate the city's GDP value.

There are 4037 cities with more than 100,000 population around the world but only about 300 cities' GDP value is available on the Internet (previous 10 pages result with searching key word "world city GDP"). The city's government is not necessary to collect this data to make good policy decisions[1] or they just want to keep it as a secret. Furthermore, City's GDP datas collected from different website are sufficient with bias and incomplete. If an organization need to make some decision such as region investment based on GDP value, It is better to have their own city's GDP or other features prediction value to support their decision making.

To achieve this goal, I need to find a believable dataset to train my model by supervise learning with an appropriate algorithm.

B. City's Population Prediction

Overview: In this part, I need to find a way to use city's road map and satellite map image data to classify the city Population level.

Comparing to previous section. Image data such as satellite map image is much believable than numeric data from national or organization statistic because taking satellite image is basically just a camera work without too many bias if there is no malicious distort. Furthermore, map image data is much easier to access with Google or Baidu map API.

The label in this problem is Population but it can also be the GDP, Export volume or something else if you have enough label dataset. The pros of CNN is that you only need to consider the label set you can get and all features can be collect with the map image API works.

One of the potential usage of this population prediction model is finding the cities with large variance between the prediction value and the ground truth value. We can pick up this cities and analyze the reasons behind it. Why it is looks like a large city but not? Does it caused by aging of population? or depletion of energy resources?

To get a higher prediction accuracy. I need to preprocess the image in an appropriate way and test different CNN architecture. Moreover, combing image data with numeric data is also acceptable.

III. Related Work

Has this problem been solved before? What is different in your approach and why do you think it is the right time to focus on this problem. Please go over related work in a few paragraphs by comparing your results and findings to other approaches.

Previous studies on GDP prediction are mainly focus on future GDP prediction in Economic area based on Empirical forecasting. One of the famous paper is written by Patrick Pilstrm and Sebastian Pohl[7] which predict the international GDP growth. In 2016, Rickard Nyman and Paul Ormerod used random forests method to predict the economy recession[8] and do the GDP prediction at the same time.

However, their works can not be validated in a short time. My model is more focus on the GDP value prediction and can be tested at anytime. Moreover, unlike some future prediction base on the recursion model such as RNN. My model is more focus on the link between the features in a same time point. Therefore, it is acceptable to predict some features such as green area and population in the future and use it to predict the feature GDP value.

For the second population prediction part. There are many works for using satellite image to do classification work such as Transforming Satellite Imagery Classification with Deep Learning[9] and Remote Sensing Image Scene Classification: Benchmark and State of the Art[10]. The first work mainly focus on urban environment classification and clustering and the second one try to predict different object on a small scale satellite or airplane image. Both of this work is very helpful for understanding the inside of CNN and the second one also use the VGG16 architecture and achieve a good score.

IV. Dataset

There are two kinds of data: One is the geographic and administrative features numeric data and the other one is the map image data.

1. The first one, geographic and administrative features numeric data is collected from Organisation for Economic Cooperation and Devevelopment (OECD) Regions and Cities dataset, Fingolas's ergebnis dataset and BROOKINGS Global Metro Monitor. The first two dataset are both publiced at 2014, So I compare and combine this two dataset to eliminate bias and enlarge the feature size. Finally, I get two usable dataset. One is the 229 cities data with 10 features and the other one is 3953 cities data with 4 features.

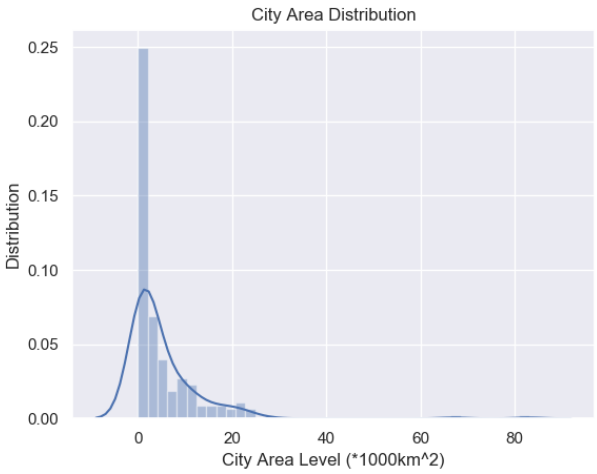
City Name	Tokyo	New York	Gothen
Polycentricity	1	2	1
Area	8592.12	9882.07	3850.19
GreenArea	4.56	39.39	292.17
PopInCore	93.66	97.47	56.15
NumOfGov	235	356	12
NumOfGovInCore	170	323	1
Latitude	35.7	40.7	57.7
Longitude	139.8	-73.9	12.0
Population	8,637,098	8,405,837	549,839
GDP	1,475,630	1,215,234	36,713

This table shows 3 cities' features and label extracted from the 229 cities' administrative and geographical dataset.

1.1. Feature Explanation and Normalization Policy - There are three types of policy in this section: Standard Normalization, divided by intuitive value and no normalization. The feature scaling is not used due to it's bad performance in experiment.

• Policentricity: number of centre of a city, no normalization. Normalization is not necessary because the domain of it is [1,2,3].

• Area: total land area(km2), normalized by standard normalization. The area feature of cities are basically normally distributed so it is good to use standard normalization. The following is the distribution of area for the 229 cities.



• GreenArea: Green area per million people(km2), standard normalization. Reason is same as Area.

• PopInCore: Concentration of population in the core, divided by 100. It is a percentage value so we can divieded it by 100 directly.

• NumOfGov: local governments(count), standard normalization. Reason is same as Area.

• NumOfGovInCore: local governments in the core(count), standard normalization. Reason is same as Area.

• Latitude, Longitude: World map coordinate, divided by 180. It is a intuitive normalization since Lat and Long are range from [-180,180] and coordinate relationship between cities is linear.

• Population: standard normalization.

• Country: country features is not implemented in the SVM model. However, in the population prediction with numeric value extensive experiment part, it is very important. We can use this data by preproccessing it by one hot encoding.

1.2. Nan Data Treatment Policy

There are plenty of nan data in the OECD dataset but I should not remove every city data containing nan data since our dataset is small size after merging several data sources. Instead, I choose use replacement policy to deal with nan data.

Replacement Policy: 1. Find the most important continuous numeric features without Nan data as weight. 2. Find the median value of the feature with missing value. 3. Replace all this Nan(missing) value with feature median value multiple corresponding weight value.

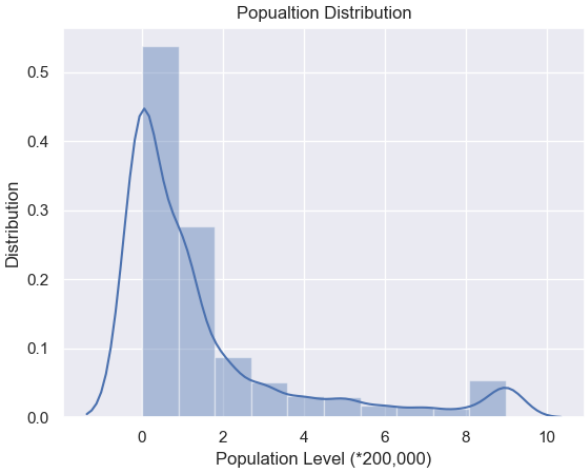
1.3. Reduce inaccurate national or organization data effect.

In this project, I collect data from two sources: OECD and BROOKINGS to do the data checking and Merging work. The main idea is find the weight of each data source and do the averaging work and finally test on both dataset.

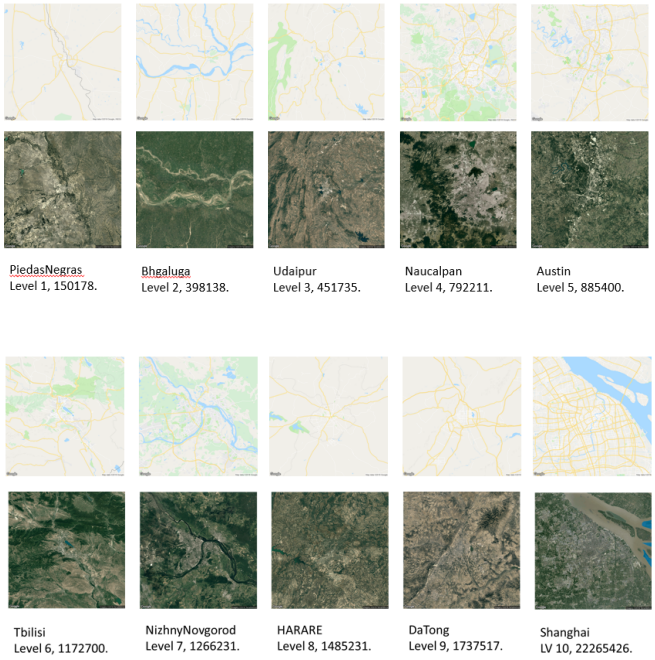
Merging Policy: 1. Apply data from different sources to the SVR model and evaluate their performance respectively. 2. Take this performance as a weight and do the feature combing base on this weight.

2. In the second part, I collect a the map images from Google Cloud platform with using Google Map API. The image data is divided in road map, satellite map, road image and background map. The first two, road map and satellite map is used to predict the city GDP level and the other two is prepared for future map generation work.

The only numeric value used in this part is the label of population and I classified it into 10 level.



This is the population distribution in 10 class of population level. (10 levels: 0-199,999, 200,000-399,999, ... , 1,800,000-inf). This data will be used as my label set for 10 class classification.

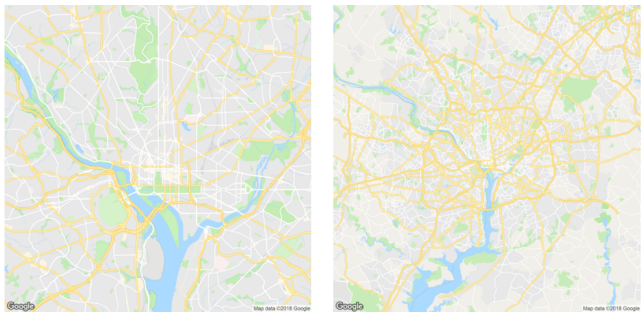


This is a brief visualization of the image features both in road map and satellite map and the correspondent label.

2.1. Image Size and Zooming.

In this part, I need to use CNN to train the map images to predict the population level. There are three things must be considered: Firstly, the map images should be in the same size, zooming level and same scaling level else the CNN model may just learns the prediction problem by the difference of size of same things such as road width. Secondly, the map images of the city must be in the same size and square to fit the CNN model. S Thirdly, the image can not be too large which will reduce the training speed and cause memory error for training machine.

In this project, the raw data size is set to 1280x1280 pixels and the zoom value is set to 10 which is 1:1155581 and the scale is 1 (2x640x2x640). In the following data preprocessing, I only adjust the scale parameter for the trade off of performance and training speed. The zoom value and size parameter is never changed.



Here is a comparison of DC maps with zoom value = 12 and zoom value = 10. In google API, the zoom value is used to adjust the area including in the image. The scale parameter is used to enlarge or shrink the map image size for the same area while the size parameter is used to choose image size of the map and it basically has the same function as zoom value.

2.2 Road map and Satellite map

There are two sources of images: road map image and satellite image. road map image is much more easy than satellite image with the same size(zoom and scale). Due to the limitation on memory I scale my image dataset to 256x256 pixels and it cause the information loss for some details such as city roadway. However the satellite has plenty of detail can not be seen directly. Thus, the experiment work of this CNN is mainly focused on both using road map image and satellite map image to predict GDP value.

V. Approach

1. Data Preprocessing.

For the GDP prediction part, as mention is previous section, I mainly use the standard normalization and intuitive division methods to deal with the numeric continuous feature value. To the classification features, I use one hot encoding to change this kind of feature into a list of boolean value.

For the map image data, I scale all the images to 256x256 pixels and keep the area size the same. After that, I simply divide the RGB values of image data by 255 for feature scaling.

I also tried to use PCA to scaling the image size. However, it will change the map image structure and is not appropriate for CNN training model. Moreover, CNN model will also do the features extracting job in non-linear way.

2. Support Vector Regression works for GDP prediction

For the first GDP prediction problem, I choose to use SVM model due to it's non-linearity with kernel function and high performance in small dataset. SVR is a popular method to predict the continuous value. In this project, I implement my own SMO program to solve v-SVR dual problem and it is a novel application to use SVM model to predict GDP value comparing other mature methods such as random forest.

Support Vector Machine is a linear classifier but it can be non-linear with a kernel trick and using the kernel functions such as rbf and polynomial. At the same time, SVM with soft constraints will allow some points out of the margin and enlarge the margin. The margin is high related to the C value where a higher C value will lead to a higher penalty for missing point.

Similar to Support Vector Classifier, SVR is used to predict the continuous value based on support vector and also need to find a line to maximize the margin (boundary lines). There is no different in the application of kernel function but the standard form of SVM is changed to:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*,\epsilon} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C(\nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)) \\ \text{subject to} \quad & (\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) - z_i \leq \epsilon + \xi_i, \\ & z_i - (\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l, \quad \epsilon \geq 0. \end{aligned}$$

This v-SVR is provided by Schlkopf in 2000[5] and has the same result with previous Vapnik's[4] e-SVR work. Meanwhile, the dual problem is showed below:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T Q(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \boldsymbol{z}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \\ \text{subject to} \quad & \boldsymbol{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0, \ \boldsymbol{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \leq C\nu, \\ & 0 \leq \alpha_i, \alpha_i^* \leq C/l, \qquad i = 1, \ldots, l. \end{aligned}$$

And the evaluation function is also showed below:

$$\sum_{i=1}^{l}(-\alpha_i + \alpha_i^*)K(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

To solve this dual problem I use a simplified SMO algorithm provided by Platt[11] which optimizes the two parameter respectively with the pre calculated upper and lower bound.

After the model construction, I adjust the hyperparameter C to get larger margin and set the tolerance for the end of iteration.

3. Multi Layer Peceptron works for GDP prediction

As a comparison with previous SVR model, MLP is not suitable to this GDP prediction problem with only 229 data

points but it still works due to it's non-linearty. Moreover, in the experiment for 4 features (country, latitude, longitude and population) dataset with 3953 data points, Neural Network has a higher performance comparing to SVR model.

In this project, I write a MLP model and it's hidden layers and activation function can be easily adjusted with my self design API. There are mainly two part for the MLP model. The first one is feedforward which calculates the input value with hidden layers weights and activation function and outputs the prediction result. In the regression problem, there is no activation function after the last hidden layer and the output layer size is one.

Another part is the backpropagation which is used to train the weights for hidden layers by differentiating the error from next layer. With different activation function, we also need to adjust the backpropagation model's Delta function.

Meanwhile, different kinds of cost function such as squared error or l1-error have different performance. In my MLP model, only squared error works to predict the GDP value. The other function such as l1-error can not reduce the error rate with iterations.

Apart the model design, I also do many experiments with the hidden layer design and activation function selection. It is found that with deeper layers and larger hidden layer size such as (9,15,25,35,24,18,12,6,1) the result of the MLP model is easier to converge to certain value and after checking with the hidden layer output, I found the output for some points are drop in the local optimal point and may need to use momentum to get out of it. At the same time, a very simple layers such as (9,6,1) can not get a good performance in GDP prediction as well.

4. Convolutional Neural Network works for population prediction

In this project, I use Tensorflow to implement my CNN models. The self design model is showed below:

8 weight layers CNN
input (256 x 256 RGB image)
conv3-64
conv3-64
MaxPooling
conv3-128
conv3-128
MaxPooling
conv3-64
conv3-64
MaxPooling
FC 1024
FC 10
softmax

This model works well with two class classification such as cat-dog but can not predict the population accurately. Therefore, I select two more effective models VGG16 and VGG19[6] for population prediction. The VGG19 model is deeper than VGG16 with one more convolutional layer before each max pooling layer. However, it is not more powerful than VGG16. In my experiment, I will introduce the overfitting problem of it.

VI. Experiments

A. SVR GDP Prediction Model Experiment and Evaluation

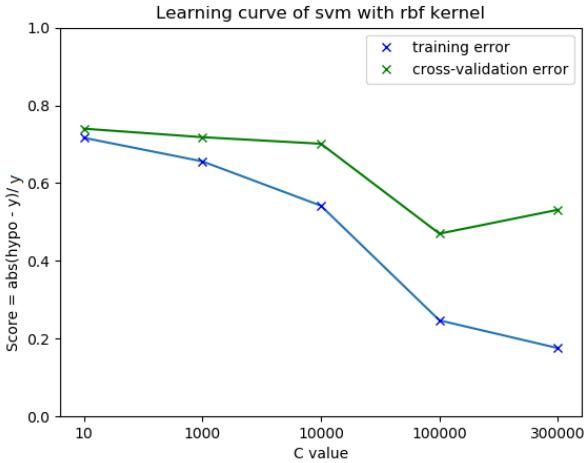
1. Kernel function selection: Linear, rbf or polynomial.

1.0 The score function of this experiment is $score = (hypoValue - trueValue)/trueValue$. In this project, a lower score means a higher performance and there is large penalty for exceeding large prediction result. At the same time, all validation means 10 class cross validation in this section.

1.1 In this step I will test the performance in different kernel functions, the C penalty parameters are optimized for each function.

Kernel function name	linear	rbf	polynomial
Train Score	0.61	0.265	0.60
Test Score	0.58	0.41	0.60
C value	90,000	90,000	1000

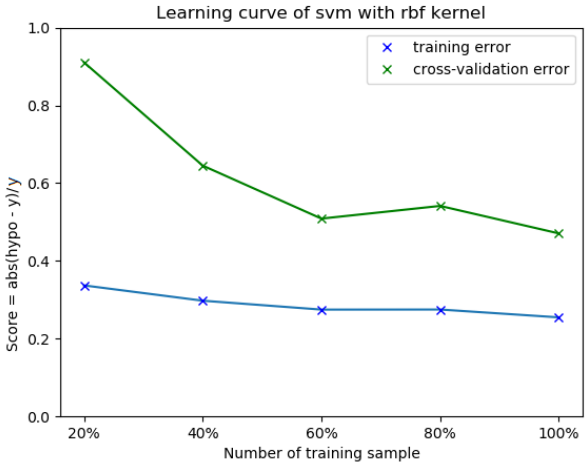
1.2 The hyperparameter C give a restriction on the misclassification on each sample. Generally speaking, with a larger C, the SVM will generate a small margin hyperplane but less misclassification at the same time, vice versa. A large C may cause the model be overfitting but a smaller C may also increase the bias of model so it is a trade off job. The C value selection should depend on the experiment.



Here is the learning curve of prediction score in relation to C value. In this learning sample, I use the best kernel function rbf from previous section.

As we can see, when C is smaller than 100,000 it will reduce the error(score) for both training prediction result and validation result, it is due to a larger C with more restrict on misclassification reduce the model's bias. However, after 100,000, the increasing C will only improve the performance on training data and make the validation error higher which is caused by overfitting.

1.3 In this section I test the relationship between the training data size and prediction performance. Thought SVM is good at dealing with small dataset but it is only due to it's high memory requirement and time consuming on large dataset.



As we can see, More training data will always reduce both the training error and validation error. Unlike the learning curve sample on CS6364 lecture, more data sample will also help reducing the training error in the regression model not obviously through. Moreover, more data points significantly reduce the validation error as showing in the learning curve. The little rise in the 80% degree may be a causal high error sample due to the small city dataset.

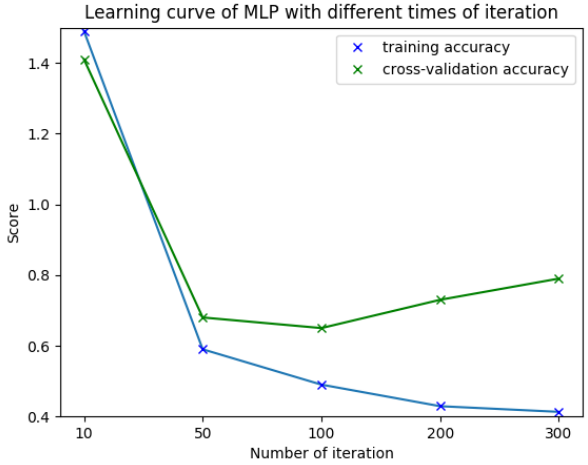
B. MLP GDP Prediction Model Experiment and Evaluation

2.0 The Score function in this part is the same as the SVM part. The cost function in my MLP models is squared error which performs better than the other cost function. Through the result of other cost function such as l1 loss may be the same, it works more stably and fast.

2.1 There are three kinds of MLP layers I design to use: the input layer size is 9, the output layer size is 1 and the hidden layer size is (6),(15,25,12,6) and (15,25,35,24,18,12,6) respectively. Here is the prediction result:

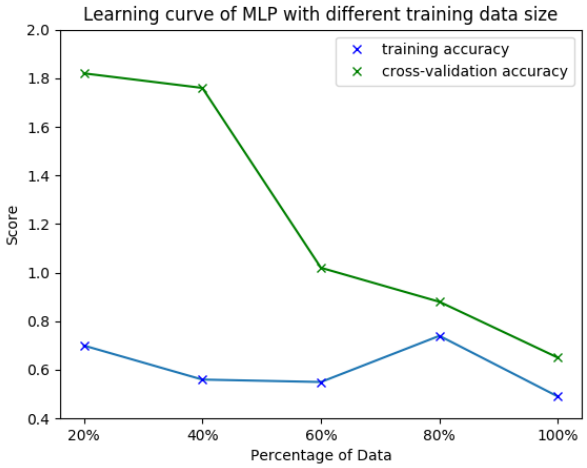
hidden layer size	(6)	rbf	polynomial
Train Score	0.61	0.265	0.60
Test Score	0.58	0.41	0.60
C value	90,000	90,000	1000

2.2 The relationship between number of iterations and the score performance.



As we can see, the increasing of iterations in the first 100 will improve the score performance for both training score and validation score. However, as iteration time grows larger, the training score will drop but the test score will increase which cause the model has a low bias but high variance.

2.3 The relationship between training datasize and prediction score.



As we can see, more training data makes the validation score perform better and it only has little effect on the training score.

C. CNN Population Prediction Model Experiment and Eva

3.0 My own model is tested with the accuracy around 50% accuracy on the validation set with 100 epochs. It is not good so I choose two more perfect model to test my thought.

3.1 Satellite map vs Road Map: With experiment I find that satellite map predicts much better than the road map in any way. When Applying both VGG19 and VGG16 to the road map it is easily drop into a overfitting condition. It is probably caused by the road map is too simple and the CNN just learn the exact details of it corresponding to each sample points. On contrast, the satellite map has a very high accuracy in the validation set. Using the VGG16 model with 1000 epochs training I get a 92% accuracy on the validation set.

validation accuracy with road map images.

Model Name	8-layers self design	VGG16	VGG19
50 epochs	0.59	0.66	0.55
100 epochs	0.60	0.55	0.51
500 epochs	nan	0.47	0.52
1000 epochs	nan	0.49	0.41

training accuracy with road map images.

Model Name	8-layers self design	VGG16	VGG19
50 epochs	nan	0.91	0.92
500 epochs	nan	0.9875	0.9962
1000 epochs	nan	nan	0.9997

In this part, I made a big mistake that reset the random seed after the the model is trained. This random seed is used to spilt my training data and testing data from the original one. The training accuracy will achieve 90% for both VGG16 and VGG19 on the training dataset, and finally achieve 99.97%

accuracy with 1000 times of epoch. However, the validation accuracy is just at a 50% level.

This is the first time I try to apply CNN on my model. However, due to the previous big mistake, I try many times to achieve the false result again. No matter of what, the result prove that CNN model has a good ability at doing the classification on the training set. No matter how big the dataset is, it can always find a non-linear solution to separate it.

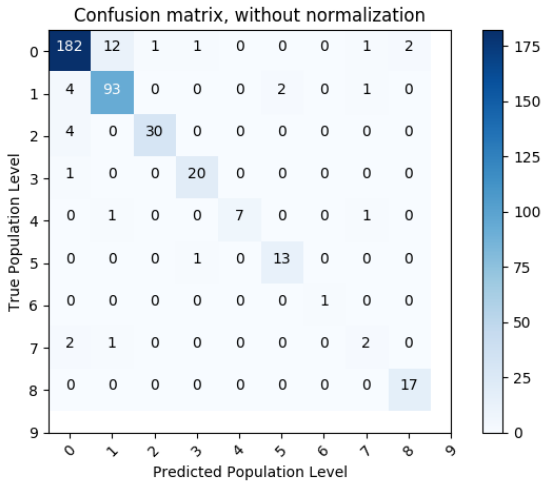
VII. Results

Here is a view of my prediction result for GDP Prediction Problem on the test set:

```
GDP Predict Result: 29535.4 GDP Real Label: 20561.8
GDP Predict Result: 58019.3 GDP Real Label: 55013.9
GDP Predict Result: 144271.1 GDP Real Label: 271449.3
GDP Predict Result: 33941.8 GDP Real Label: 44593.8
GDP Predict Result: 21332.2 GDP Real Label: 20212.8
GDP Predict Result: 32459.3 GDP Real Label: 25099.3
GDP Predict Result: 82204.5 GDP Real Label: 37762.5
GDP Predict Result: 27085.5 GDP Real Label: 31087.2
GDP Predict Result: 34220.3 GDP Real Label: 49311.7
GDP Predict Result: 26275.4 GDP Real Label: 21106.7
GDP Predict Result: 21919.5 GDP Real Label: 45865.8
GDP Predict Result: 60696.2 GDP Real Label: 91380.0
GDP Predict Result: 21104.2 GDP Real Label: 14733.1
GDP Predict Result: 23651.7 GDP Real Label: 15768.2
GDP Predict Result: 127969.8 GDP Real Label: 88521.2
GDP Predict Result: 18218.3 GDP Real Label: 17474.3
GDP Predict Result: 115143.1 GDP Real Label: 104677.9
GDP Predict Result: 22301.4 GDP Real Label: 15205.3
```

In the first problem, the best model I get is a soft SVR model with rbf kernel and c value equals to 90,000. This model get a score value of 0.41 finally at the test set (45 cities) validation. The score function is $score = (hypoValue - trueValue)/trueValue$.

Here is a view of my prediction reuslt on the Population Prediction Problem. This is the confusion matrix for VGG16 model trained by satellite map images works on the training dataset with 50 epoch:



In general, the SVM model used to predict the GDP value is more appliable. For the mistake in the second part, I will take more time to do a inside layer visulization job and find a more acceptable solution.

VIII. Discussion

In this project, I predict the city's GDP value which is acceptable but not perfect. In the second problem about population prediction, I make a big mistake in the random seed setting section and the final model only achieves 66% accuracy which has a little improve comparing to the ground truth distribution that 50% of cities have first level population size (0 - 200,000).

Base on these problems and previous works, I will discuss my project in two aspect: Limitation and Future work.

A. Limitation:

1. Feature Selection:

At the start of this project, I want to build a supervise learning model to predict the developing situation of city and achieve a high accuracy as mnist or pima dataset. However, the real problem is that it is very hard for me to collect large size data with many features and here is a tradeoff between feature size and data size. Therefore, the previous works for feature selection should be both in feature performance testing and accessibility investigation. In this project, to predict the GDP value, the population feature is one good selection due to it high accessibility and high related to the GDP value.

2. Data Bias:

When Professor Caliskan told about the problem of Nan data for national statistics I was only thinking about how to replace it by median value or something else and do the average work from different website. However, after deep investigate the population feature by randomly comparing with wikipedia, I found that not only the nan value, many existing data features in the same time point from different data source are different. We have no way to deal with this kind of "ground truth" failure but the only way I can think about is that apply some general machine learning model such as Adaboost to each data source and select the one with high confidence to do the new model training job after the data proving job.

3. Lack of Inside View of the CNN:

After doing many headless works with the CNN "black box", I found that it is necessary to understand what the CNN model is working for and learn more CNN architecture. In the related work part, MARK ALTAWEE[9] works for region definition is one of the example. Rather than directly apply CNN architecture to my machine learning problem, I should read more related work first and do the hidden layer visualization work after the model construction.

4. Training Timeline Management:

In this project, the deep hidden layers MLP and CNN model training is very time consuming. I waste a large amount of time to do the useless training job and therefore I summarize three points I believed that will be useful for training time management. Firstly, prove your dataset by some simple models such as Adaboost and SVM first and do the Neural Network training work if it is correct. In the Population prediction part, I once believed that I can achieve a very high accuracy with simply apply image set to CNN model. However, after comparing the population label with wikipedia and other data source. I found that there is a high bias with the label set so

the reuslt couldn't be achieved. Secondly, set a low epoch and go through all your neural network first then choose the one with high performance to be trained on your sleeping time. Thirdly, save your model with clear path and name for future usage.

B. Future work.

1. Semi Supervising GDP Prediction.

In this project, I found that the feature data is much more than the label data and it is good to apply the semi supervising model to predict the GDP value which allows us utilize more city feature data.

2. Continue Working on The Map to Population Work.

55% result is not satiable and I will do the hidden layer visualization work in the next step and read more related paper to get more background knowledge.

3. Map Generation Work.

Previously I went to use the background map, road line and road map to do the map generation work based on GAN technique. I had practiced several complete open source model to finish this job and I will implement it again with my own work in the future. However, after this project, I think that I should firstly learn more basic problems in CNN.

IX. Conclusion

In this project, I implement my own SVR , MLP and CNN model (Tensorflow based). The works of the first problem City GDP Prediction is acceptable but has limitation in the data source. In the second part, my CNN model only does a little improvement on population prediction and needs to solve the overfitting problem. Apart from the machine learning problem, I also learn many other useful techniques and tools such as multi-threading, Seaborn and AWS SageMaker. In the future, I will mainly focus on feature engineering and CNN visualization works which are believed can help me to improve this project.

References

[1] Don Sillers, Econo-geezer. Does a city have gdp? 2015.

[2] Harris Drucker. Improving Regressors using Boosting Techniques. ICML, 1997.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. March 4, 2013.

[4] Vapnik. Statistical Learning Theory. Wiley, New York, NY, 1998.

[5] Schlkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. Neural Computation, 12:12071245, 2000.

[6] Karen Simonyan, Andrew Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. arXiv:1409.1556v6 [cs.CV] 10 Apr 2015.

[7] Patrick Pilstrm, Sebastian. PohlForecasting GDP Growth. 2009.

[8] Rickard Nyman and Paul Ormerod, Predicting Economic Recessions Using Machine Learning Algorithms

[9] MARK ALTAWEEL. Transforming Satellite Imagery Classification with Deep Learning. Feb 14 2018.

[10] Gong Cheng, Junwei Han .Remote Sensing Image Scene Classification: Benchmark and State of the Art. 2017.

[11] Platt, John. Fast Training of Support Vector Machines using Sequential Minimal Optimization, in Advances in Kernel Methods Support Vector Learning, B. Scholkopf, C. Burges,A. Smola, eds. MIT Press 1998.