Position Prediction in CT Volume Scans

Jizhou Huang
Yutong Sun
Chih Yun Pai

Abstract

Our project aims to building a prediction model to precisely predict the reference depth with given CT slice image features of spines. Firstly, we will show our analysis that indicating the reference depth spanning(RDS)is the key property dominates the prediction loss. Then, we will show how we simplified the dataset for reducing training time. Moreover, we will present the analysis and the corresponding performance of several prediction models as well as their combinings and transformations. At last, we will conclude the results and show the best model we have ever found.

1. Introduction

The data used to train our model is from *UCI* dataset *Relative location of CT slices on axial axis Data Set*[2]. The dataset consists of 384 features extracted from CT images. The class variable is numeric and denotes the relative location of the CT slice on the axial axis of the human body. Each CT slice is described by two histograms in polar space. The first histogram describes the location of bone structures in the image, the second the location of air inclusions inside of the body. Both histograms are concatenated to form the final *feature vector*[2]. Bins that are outside of the image are marked with the value -0.25. And before using this data, we reduce feature dimension by using PCA. So that when we use the Bayesian method, the bias can be reduced which caused by features are not independent to each other.

2. Data Analysis

ID Based Analysis

Initially, we believe that each patient has a unique spine so that the data of each *ID* is biased compared with that of the other *ID*. Based on this belief, we started at building individual model for each patient and we choose *Gaussian Process Regression* (*GPR*)model as our prediction model.

To verify our assumption, we made an *Cross Prediction Matrix* (*CP Matrix*). In particular, we divided the whole dataset into 97 subsets that each subset has consistent *ID*. Then, we trained 97 *GPR* models using this subsets. We iterated through these models so that, at each iteration, we use current model to predict all the 97 datasets and calculated the corresponding *Rooted Mean Squared Error* (*RMSE*). Finally, we got a 97-by-97 *CP Matrix*. If our assumption is right, all the *RMSEs* should be very high except for the *RMSEs* on the diagonal.

However, the matrix defaulted our surmise since there are a large amount of *RMSEs* are far below our expectation as shown in *Figure* 2.1. Moreover, the *CP Matrix* is approximately symmetric which means the prediction error caused by using the i^{th} *GPR* model to predict the j^{th} dataset is approximately the same as its counterpart. In conclusion, the *CP Matrix* indirectly

shows that the individual bias of each patient does not matter so much. Understandably, the spines of different people should be similar to each other, since we are of the same species.

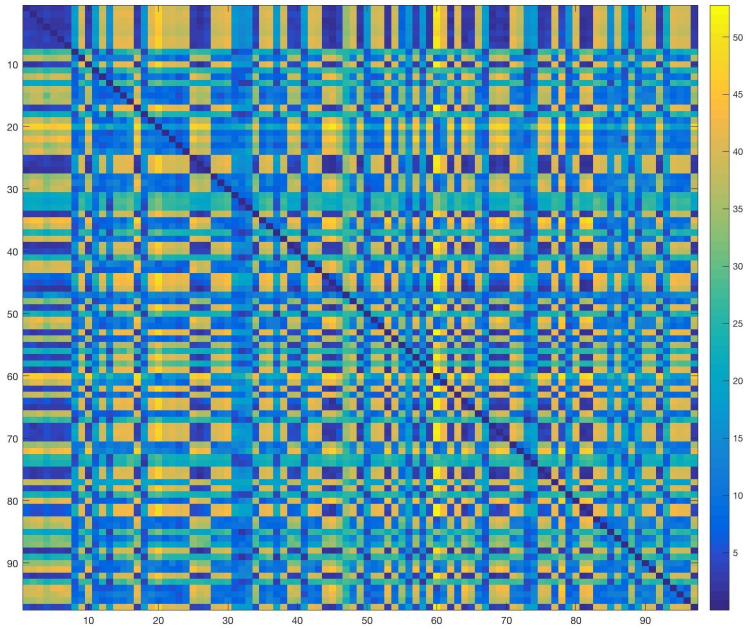


Figure 2.1

Reference based analysis

Even though the $CP\ Matrix$ is approximately symmetric, there is still small difference between each symmetrical element pair. To find out what leads to this phenomenon, we dug deeper into this matrix and found something very interesting. Generally, in each pair, if the $RMSE$ at (i, j) in the matrix is smaller than the $RMSE$ at (j, i), the i^{th} GPR is proved to be trained with dataset of larger $reference\ depth\ spanning\ (RDS)$. This is true even if the RDS of the two training sets do not overlap with each other. In addition, those $RMSEs$ indicating poor prediction performance are generally caused by none or small such overlapping between the training data of corresponding models. Since all the evidence pointed to the RDS, we made another two assumptions:

- $If\ the\ training\ set\ has\ larger\ RDS, the\ resulting\ loss\ should\ be\ smaller.$
- $The\ more\ the\ RDS\ of\ the\ testing\ set\ overlaps\ with\ that\ of\ the\ training\ set, the\ smaller\ the\ loss\ will\ be.$

Both of them indicates that the bias caused by different RDS of the training data may dominate the loss. To verify these two points, we designed another two experiments.

In the first experiment, we want to verify the first point, so we made two datasets for each patient. In particular, we divided the whole dataset into 97 subsets that each subset has consistent ID. Then, for each subset, we made the first dataset by choosing 70 percent of its data uniformly at random for training and the rest for testing. On the other hand, we sorted the data of each subset based on its reference, then, we made the second dataset by choosing the first 70 percent data for training and the rest for testing. Then, for each dataset, we trained 97 GPR models as well as calculated their loss. The results is shown on $Figure$ 2.2.

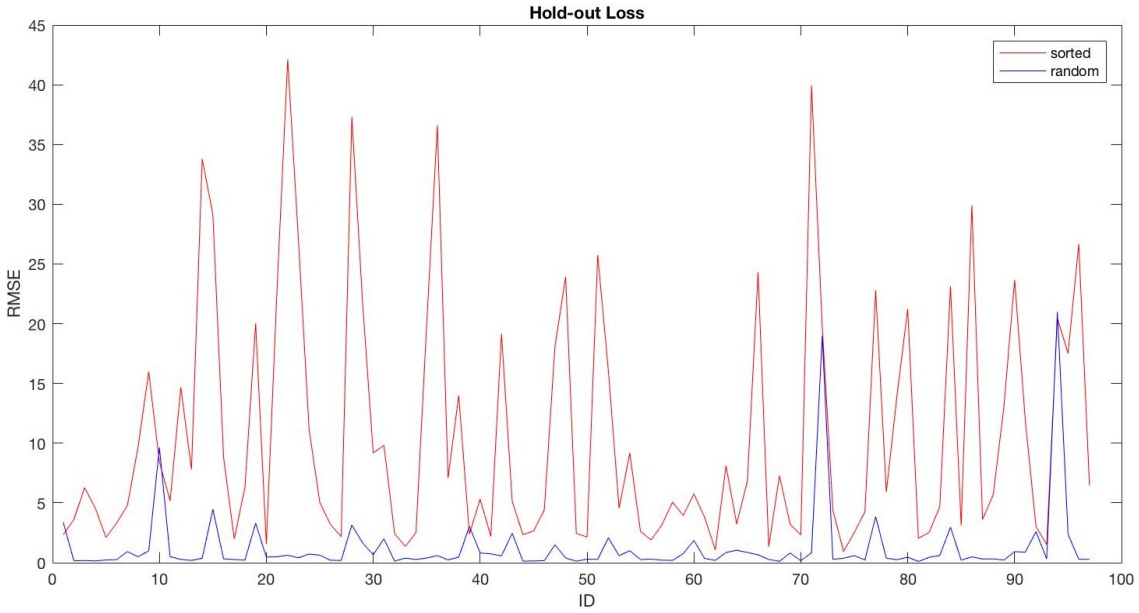


Figure 2.2

In the second trial, we built 4 GPR models. At this time, we completely ignored the ID feature, but separated the whole sorted (based on reference) dataset into three subsets with respect to the reference depth. Particularly, we created three training data sets of the same size, each of which is constrained to a small part of the whole RDS, specifically, they are $0.5\mu \pm 0.05RDS$, $\mu \pm 0.05RDS$, $1.5\mu \pm 0.05RDS$, where $\mu = mean(RDS)$. We also created another training set of the same size by picking data sample from the whole data set with equal space so that we can guarantee it has a very large RDS. Then, we exclude all the training sets from the data set and divided the remaining subset into another 100 subsets with equal RDS. From these 100 datasets, we evenly but randomly picked 100 subsets as our testing sets. Finally, we trained 4 GPR models and tested each of these models with the same 100 testing sets. The result is shown in *Figure* 2.3.

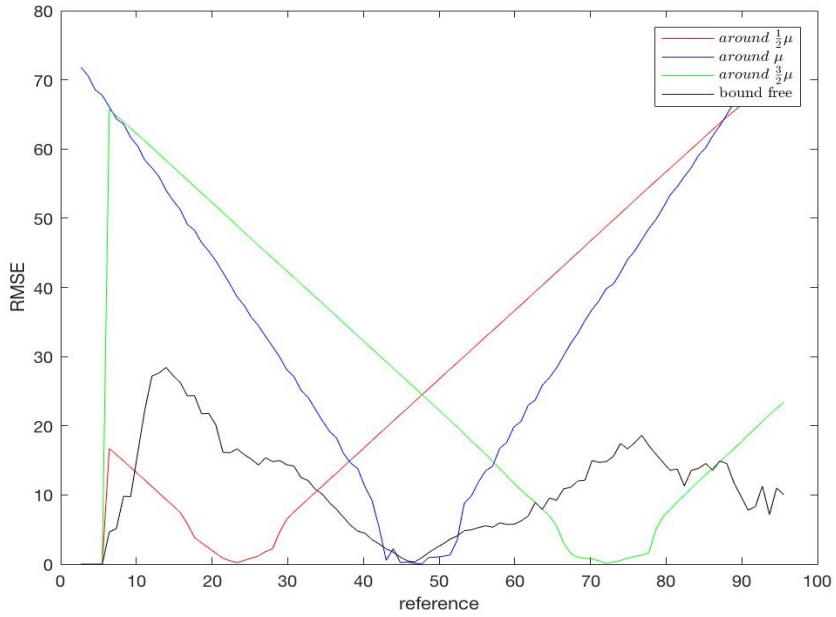


Figure 2.3

Conclusively, $Figure\ 2.2$ shows that, no matter which ID it is, the model trained by dataset of larger RDS outperforms its counterpart, which verifies our first assumption. $Figure\ 2.3$ shows that, no matter how the model is trained, the resulting loss is generally smaller if the RDS of the testing set overlaps more with the training set, which verifies our second assumption as well as enhances our first assumption.

3. Data Processing

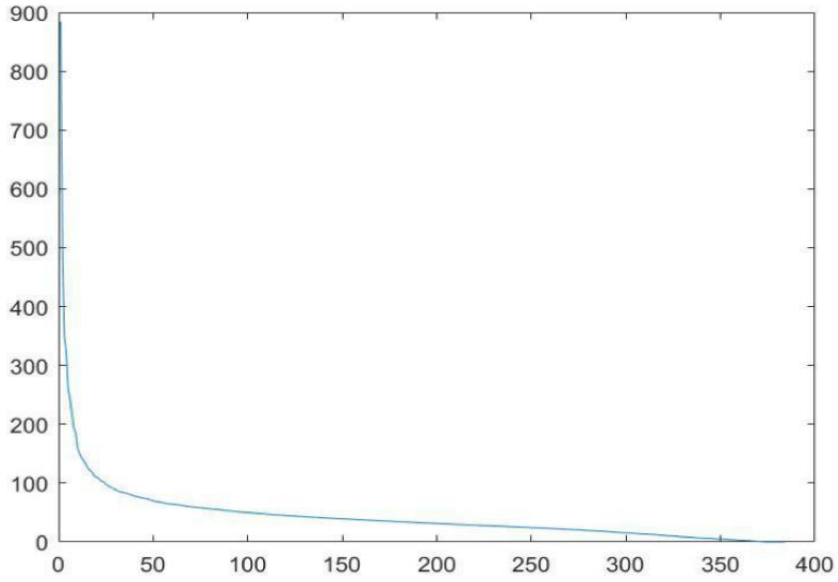
Dimension Reduction (PCA)

To get rid of $curse\ of\ dimensionality\ (Bellman, 1957)$, preprocessing data to reduce it's dimension by using feature selection is necessary. Feature selection is the process of detecting relevant features and removing irrelevant, redundant, or noisy data. Careful selection of potential features can help achieve the higher accuracy at lower computational cost. In this paper, we use PCA to select features.

PCA is used to analyze data and extract variables with similar concepts (principal components). On the other hand, project the data onto a lower dimensional space. Principal components which explain a greater amount of the variance are considered to be more important.

$Figure\ 3.1$ shows eigenvalue of 384 features. From the $Figure\ 3.1$ we can see, the 82.64% information is contained by the Top200 features. Then we select Top200 features to continue building our predict model.

We calculate different PCA, 50 PCA, 100 PCA and 200 PCA whose eigenvalues contain 42.06%, 59.28% and 82.64% of total information respectively.



$Figure\ 3.1\ Eigenvalues\ of\ 384\ features$

4. Model Analysis

Kernel Selection

Due to limited computational power, we only compared 3 different kernels for our project, which are $Squared\ Exponential$, $Matern32$, and $Matern52$. We trained three models with the same training set but different kernels, and plot the result in the same way we did in the last experiment.

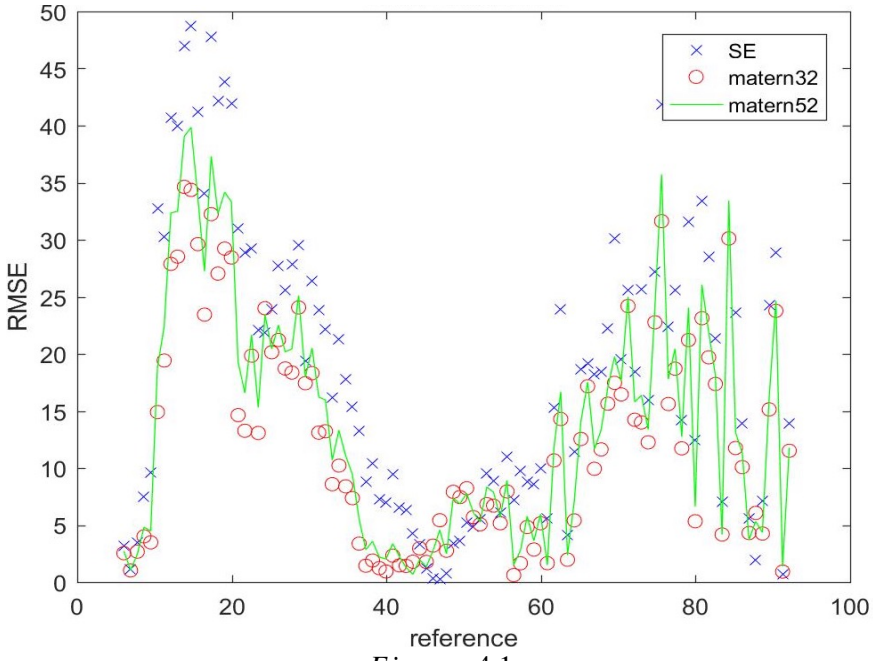


Figure 4.1

From $Figure\ 4.1$, we can see that the $Matern32$ performs the best, but there are no significant difference among these three kernels.

(Clarification: Because of certain reason, all the experiments we will stated in Model Analysis section is based on $Squared\ Exponential$ kernel.)

Sparse GP

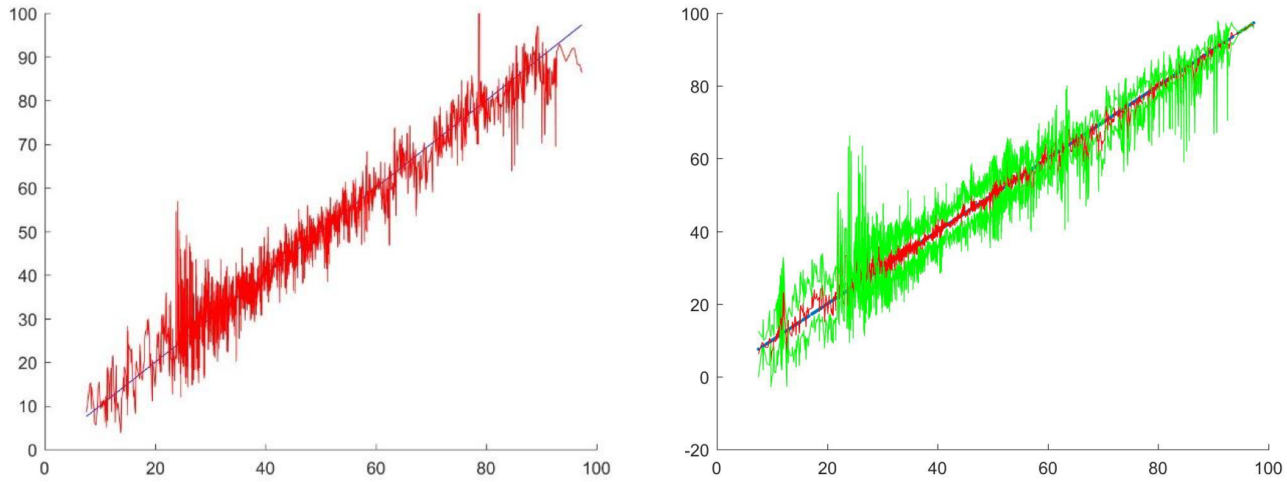


Figure 4.2 *Using* $ID = 0 - 12$ *train and test sparse GP model* : $x - axis\ is\ test\ reference, y - axis\ is\ prediction, red\ line\ is\ prediction\ mean, green\ line\ show\ the\ log - likelihood$

Using GPML toolbox to implement sparse GP, firstly, we randomly select 70% instances from $ID = 0 - 12$ as the training data and the rest as testing. Furthermore, we build a linear model as prior using the training data with square kernel to learn the hyper-parameters, and then make prediction. The more accurate the prediction were, the closer between test reference and prediction and hence approximate to a line which has slope 1, as the following graphs show.

From the $Figure\ 5.1$, the left graph show the relationship between test reference and prediction before sparse GP sampling, the right after. After sparse GP sampling (the right picture), the closeness between prediction (red line) and the slope-1 line (blue line).

The following graph show the result of training model with instances of ID0 to and testing with those in ID13 to. Not overlapping each other, the far distance between red and blue lines represent the bad performance.

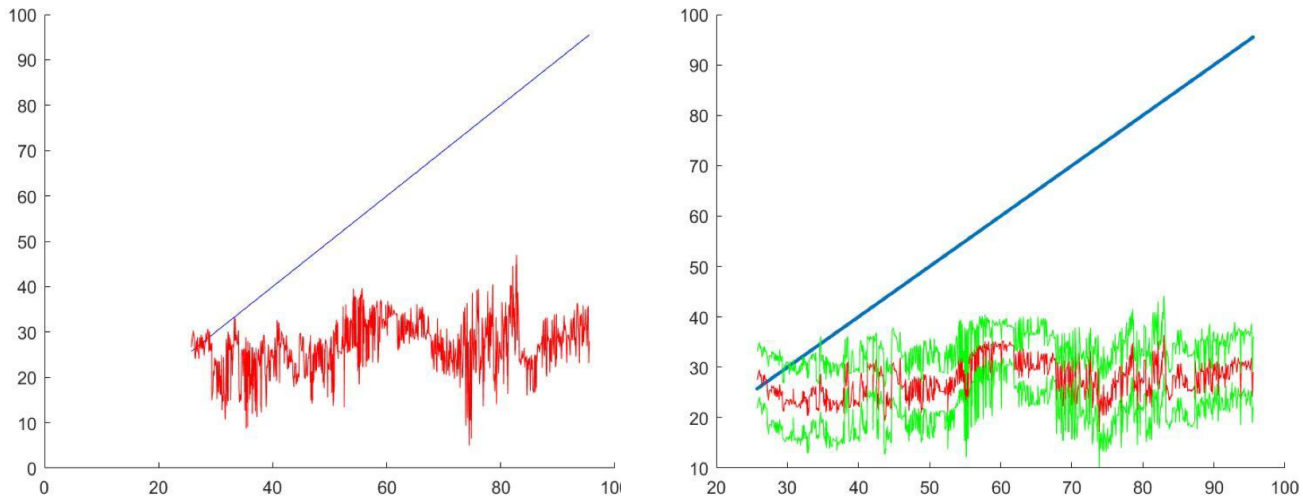


Figure 4.3 *Using* $ID = 0 - 12$ *train and* $ID = 13 - 15$ *test sparse GP model*

Combinatory Models

Now we introduce two combinatory models that we found to comparatively effective for our data, which are $GPR + SVM$ and $GPR + GPR$.

Where $x_1, x_2, \ldots, x_{70}$ are whole training points (whole raw data: 384 features, 200PCA: 200 features, 100PCA: 100 features) from each individual patient $ID = 0 - 69$ respectively.

$h_1, h_2, \ldots, h_{70}$ are hypothesis that are learnt from $x_1, x_2, \ldots, x_{70}$ respectively with GPR. is the final hypothesis that combines $h_1, h_2, \ldots, h_{70}$. In this layer, the H is formed by SVM in $GPR + SVM$ model and formed by GPR in $GPR + GPR$ model.

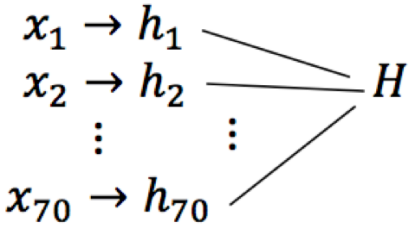


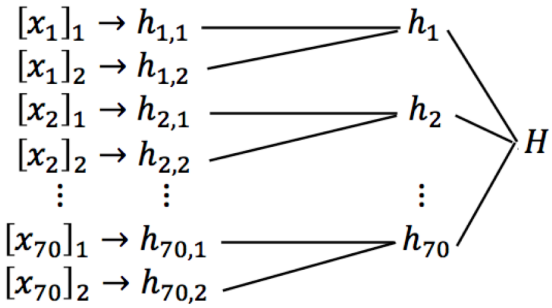
Figure 4.4

Transformed Models

Here, we introduce the most effective model we have ever built. The intuition is from $Neural\ Network$, while this model is trained in a much simpler way. The model is generally

consisted of three layers of GPR models, shown in $Figure\ 4.4$. More importantly, in this model, the two histograms in the original $feature\ vector$ are learned separately, which, we believe, will allow us to learning more information from the image slice.

$[x_i]_1$ represents the first histogram of the i^{th} patient, similarly, $[x_i]_2$ represents the second histogram of the i^{th} patient. $h_{i,j}$ represents the hypothesis learn from i^{th} patient's j^{th} histogram with GPR. h_i is the hypothesis learn from $h_{i,1}$ and $h_{i,2}$ with GPR, where $i = 1 - 70$ and $j = 1 - 2$.



$Figure\ 4.5$

5. Results

*whole images from every patients	training data ID=0~69	testing data ID=70~96	SVM	GPR+SVM	Sparse GPR
whole raw data	39450x384	14050x1	5.9731/~2 weeks	4.7032/~56 hr	~
*randomly select 50 images from each patient	training data ID=0~69	testing data ID=70~96	SVM	GPR+SVM	Sparse GPR
reduced raw data	3500x384	1350x1	6.4797/330 sec	6.5980/1.234 sec	7.9149/460 sec
reduced PCA200	3500x200	1350x1	8.2640/386 sec	11.2633/1.2 sec	12.3590/485 sec
reduced PCA100	3500x100	1350x1	7.6416/343 sec	7.8010/0.98 sec	10.6548/ 463 sec
*sorted and randomly select	training data ID=0~69	testing data ID=70~96	SVM	GPR+GPR	Sparse GPR
reduced raw data	3500x384	1350x1	2.8205/84 sec	2.7842/516 sec	8.286/18 sec
reduced PCA200	3500x200	1350x1	3.769/483 sec	7.5595/1671 sec	10.0659/499 sec
reduced PCA100	3500x100	1350x1	3.2344/559 sec	4.9561/1802 sec	9.2309/507 sec
previous layer(s) with whole training data ID=0~69	the last layer with reduced training	testing data ID=70~96	GPR+GPR	GPR+GPR (separated histograms)	
39450x384	3500x384	1350x1	2.3953/6.3hr	2.1139/5.9hrs	
39450x200	3500x200	1350x1	2.8128/1.75hr	~	
39450x100	3500x100	1350x1	2.4686/1.73hr	~	

$Table\ 5.1$

All training points are obtained from patients' $ID = 0 - 69$ and all testing points are obtained from patients' $ID = 70 - 96$. "Whole raw data" contains all original features (384, without IDs) and all instances from each patient.

"Reduced raw data" randomly select 50 instances from each patient, hence create a 3500 training set and 1350 testing set. "Reduced PCA200" and "Reduced PCA100" are selected from PCA200 set and PCA100 set as the same method as "Reduced raw data".

"Reduced Sorted raw data" contains all original features, are randomly selected 3500 for training and 1350 for testing from sorted whole instances of training and testing set.

"Reduced Sorted PCA200" and "Reduced Sorted PCA100" are selected from PCA200 set and PCA100 set as the same method as "Reduced Sorted raw data".

"Whole raw data" contains patients' $ID = 0 - 69$ for training and $ID = 70 - 96$ for testing. "Reduced raw data", "Reduced PCA200" and "Reduced PCA100" all contain the same patient's ID as "whole raw data", but only randomly select 50 images for each patient in raw data, PCA200 data and PCA100 data respectively.

6. Conclusion

We started at a false assumption that the data itself has large bias. However, we proved that such bias is trivial for our prediction accuracy. On the contrary, the $reference\ depth\ spanning\ (RDS)$ turns out to be essential factor which affects our prediction loss the most. Then, we found, from data exploration phase, that large RDS in the training set has extremely positive influence on performance. Combining the need of large RDS and to avoid $GPR's$ high computational complexity, we built a model that learn $GPRs$ separately with smaller subset and combine them into next higher layer model. This method improves both in speed and performance comparing to just using single layer GPR.

Reference

[1] $https://archive.ics.uci.edu/ml/datasets/Relative+location+of+CT+slices+on+axial+axis$

[2] $F.\ Graf, H.-P.\ Kriegel, M.\ Schubert, S.\ Poelsterl, A.\ Cavallaro, 2D\ Image\ Registration\ in\ CT\ Images\ using\ Radial\ Image\ Descriptors, In\ Medical\ Image\ Computing\ and\ Computer-Assisted\ Intervention\ (MICCAI), Toronto, Canada, 2011.$