Relative location of CT slices

Chih Yun Pai

Jizhou Huang

Yutong Sun

Contents

Data Description

- Data set: Size of 53500 x 386, 53500 CT images from 97 different patients.
- Attributes:
  - 1. Patent ID: Each ID identifies a different patient.
  - 2. – 241: Histogram describing bone structures.
  - 242. – 385: Histogram describing air inclusions.
  - 386: Reference: Relative location of the image on the axial axis. Values are in the range [0: 180] where 0 denotes the top of the head and 180 the soles of feet.
- Goal: After training, predict relative location given an unseen patient’s CT image.

Problems

- Each patient has different relative spine locations, hence each patient's slices has a bias. (The spines of different patients vary in different aspects, such as length and shape, hence, we believe each patient has a unique model for his/her spine.)

- Every patients' images begin and end in different location. (Different patients have their spine depth reference spanning over different intervals.)

- Not every patients' images have same number of slicing. (e.g. some fewer than 100 and others more than 1000 images)

- => Observing data behavior to find design intuition!

Contents

First Glance

- Running various models on the whole raw data (ignoring ID feature) with 10-fold cross-validation.
  - Models including Linear Regression, Interactions Linear Regression, Robust Linear Regression, Stepwise Linear Regression, Simple Tree, Medium Tree, Complex Tree, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Boosted Trees, Bagged Trees.

- Running over 48 hours, and GPRs didn't finish.
- The high cost computational complexity ($O(kn^3)$, where k is the number of function evaluations needed for maximization and n is the number of observations.) of GPRs

	Model	RMSE
top1	Bagged Trees	1.43
top2	Cubic SVM	1.74
top3	Medium Gaussian SVM	1.77

Table (1): Top three RMSE of all models expect GPRs running whole raw data with 10-fold cross-validation.

Individuals Analysis

- Taking first 5 patients data to learn individually and compare top 3 models, as the table below. In general, SE GPR (Squared Error GPR) performs stably well for given individual patient using raw data with 10-folds cross-validation.

	patient ID=0	patient ID=1	patient ID=2	patient ID=3	patient ID=4
top1	SE GPR/RMSE:0.18	Bagged Trees/RMSE:0.20	SE GPR/RMSE:0.18	SE GPR/RMSE:0.16	E GPR/RMSE:0.15
top2	M GPR/RMSE:0.18	M GPR/RMSE:0.19	E GPR/RMSE:0.18	RQ GPR/RMSE:0.16	SE GPR/RMSE:0.16
top3	RQ GPR/RMSE:0.18	SE GPR/RMSE:0.20	RQ GPR/RMSE:0.18	M GPR/RMSE:0.17	RQ/RMSE:0.16

Table (2): Top three RMSE of all models running raw data set for each individual patient (ID:0~4) with 10-fold cross-validation.

Individuals Analysis

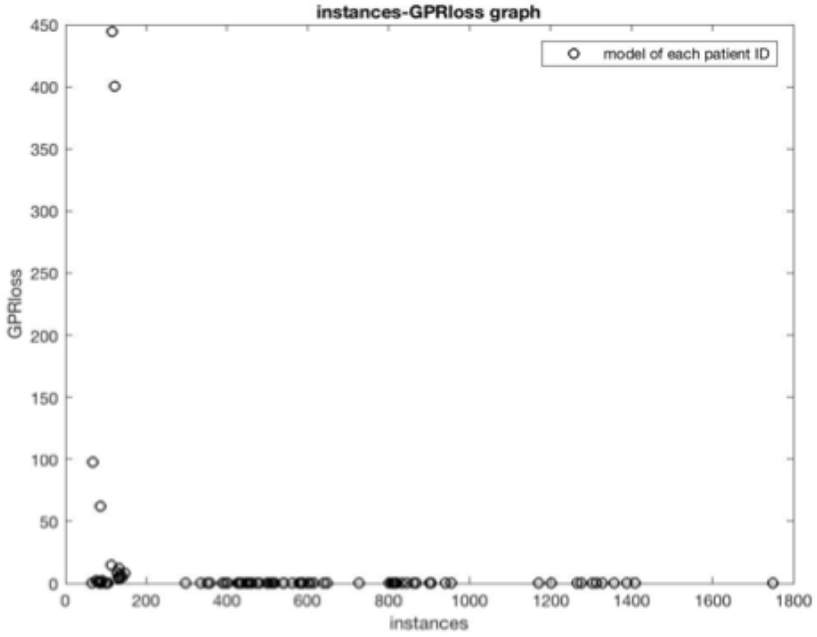
- Second observation from loss on each individual patient

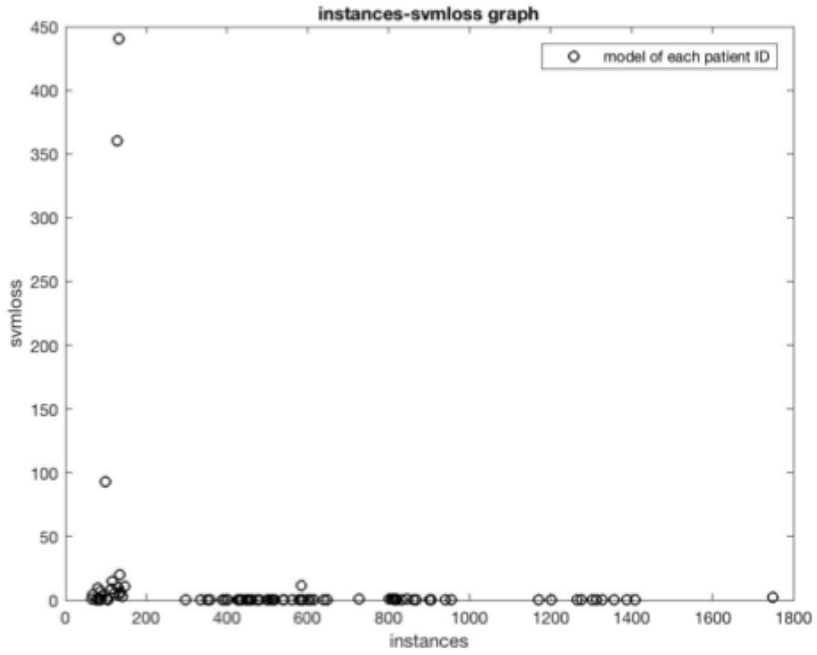
  Randomly selected 70% of images from each patient as training set.

  Other 30% as testing set.

  Calculate RMSE loss for each patient individually.

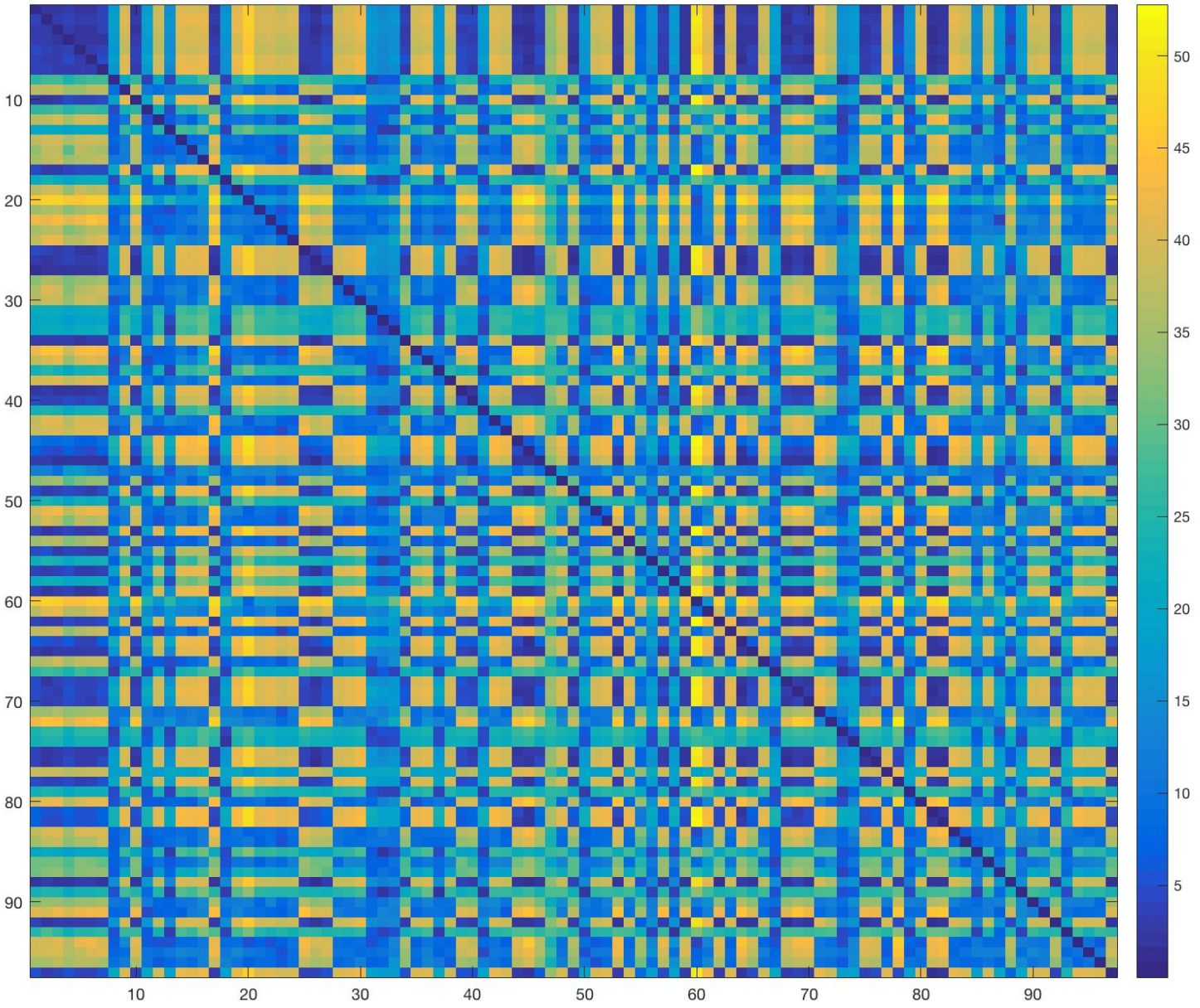
- From the plots, we discover
  - The prediction models built with the training data of patients who have less than 200 slice samples generally performs worse than the models of those who have more than 200 training examples.
  - There's no correlation between number of training points and loss for patients who have more than 200 training points as well as for those who have less than 200 training points.
  - Few "outliers" make enormously large loss which is probably caused by some random selection of training set doesn't cover the respective test reference.



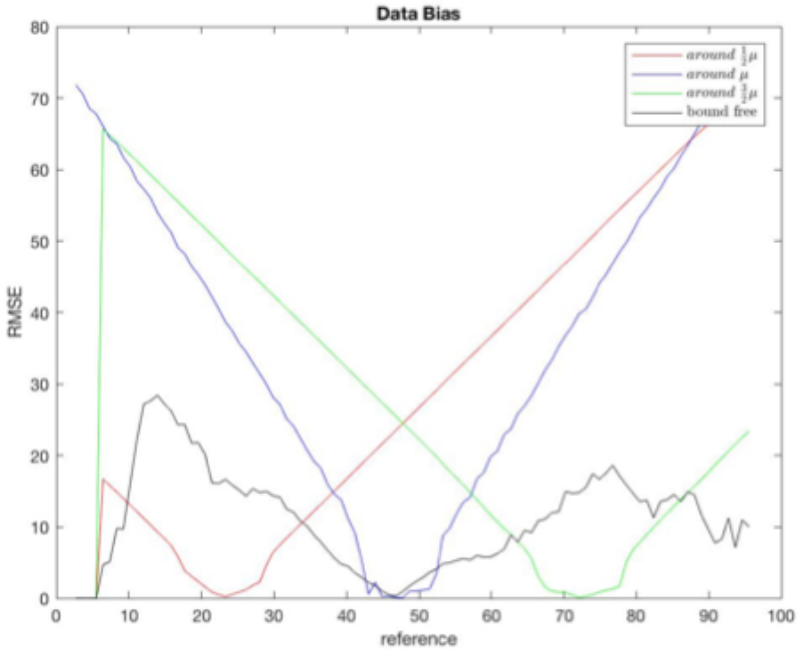


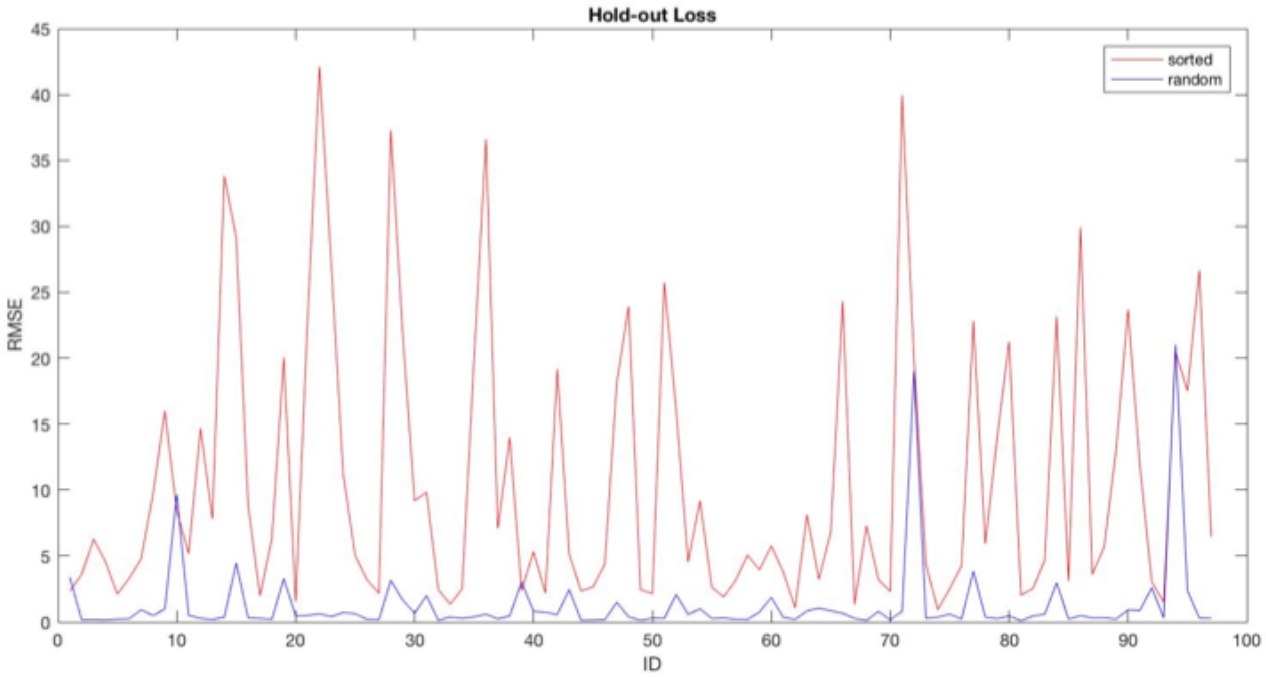
Cross Prediction

- We used each of the 97 individual models to predict the data comes from the other 96 patients so that obtained a 97-by-97 matrix which indicates the cross prediction loss.
- Not precisely but approximately symmetric!
- Generally, good predictions are caused by large reference overlap between training and testing data.
- Seems model bias is trivial, however, data bias dominates the loss…



Verification





- Train two models for each ID with the same training set but different data order.
- Train another two models, one of which is trained by data set whose reference depth is bounded in [42, 52], the other is trained by data set whose reference depth is uniformly distributed on [1, 98].
- Reference depth matters a lot!

Intuition

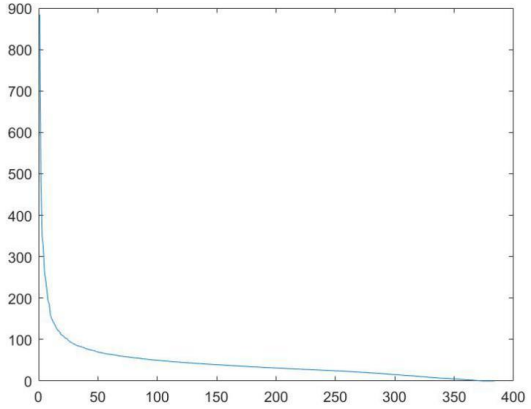
- Some Intuition after observing data…

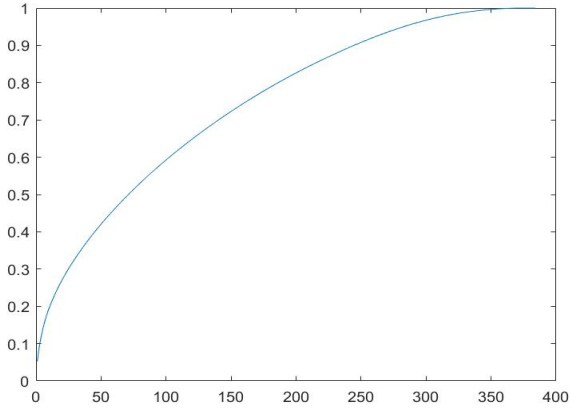
- 1. There’re reference bias in each patients’ spine and covering more different patients’ images in training set helps overcoming the bias.
- 2. GPR model perform well on each individual patient, yet not suitable for a large data set learning.
- 3. SVM is much faster in large data set.
- What if combining these features into a new method which is faster than pure GPR and more accurate than other methods?

Contents

Data Processing

- Dimensionality reduction:
  - PCA





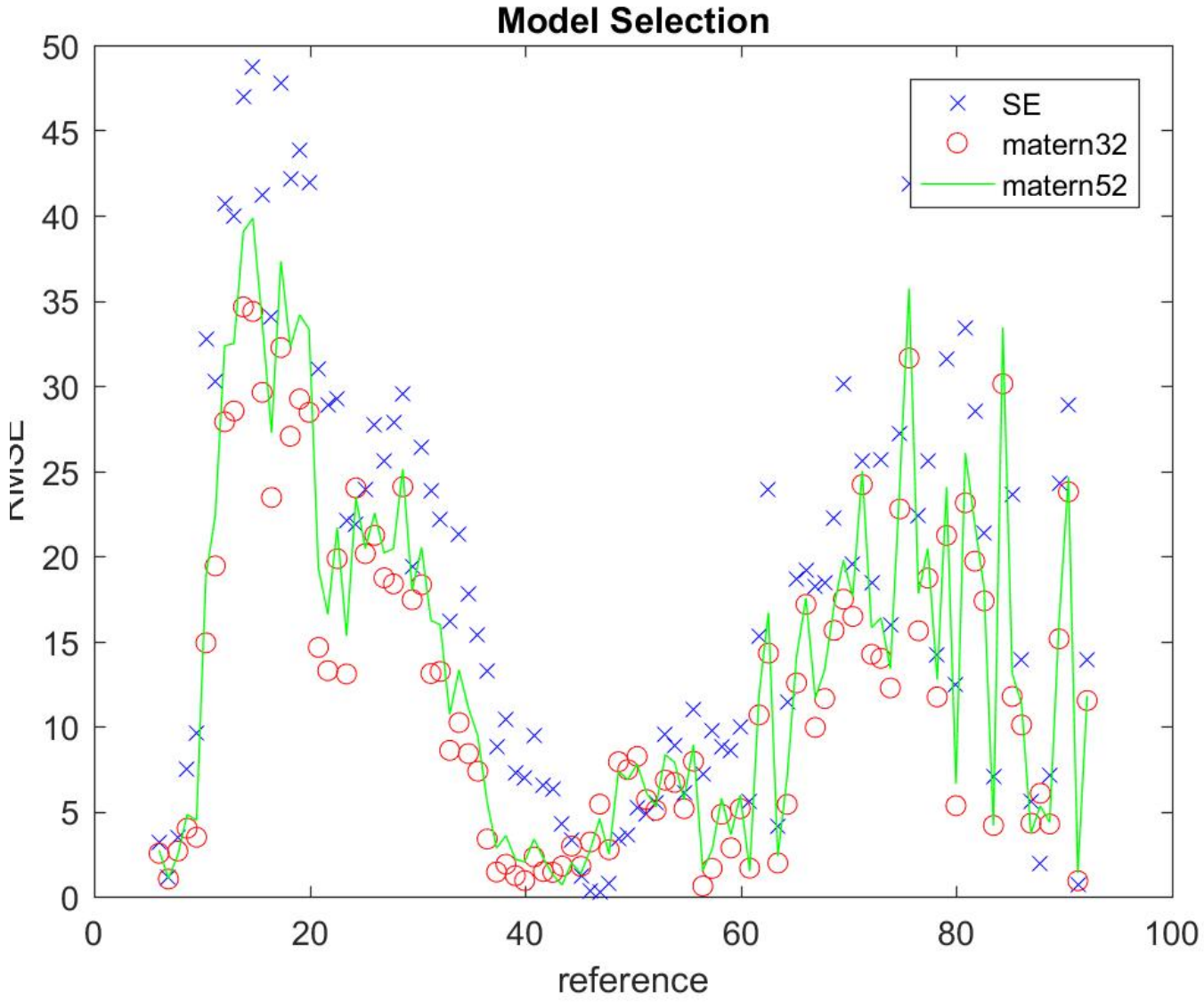
Left figure: Eigenvalues of 384 features(attributes 2~385)

Right figure: Information ratio of 384 features(attributes 2~385)

- Reducing data: Sampling 50 images randomly from each patient ID.

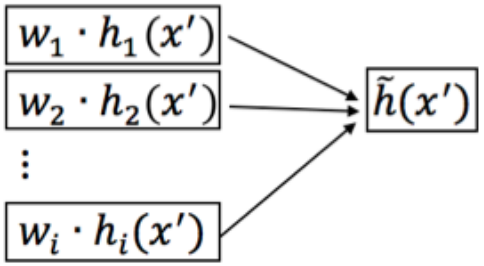
Kernel Selection

- Among Squared Exponential, Matern32, as well as Matern52, Matern32 performs the best.
- Squared Exponential is obviously worse than the other two kernels. However, there are not significant performance difference between Matern32 and Matern52.



Model 1 – GPR+SVM

- A combinational method which takes advantages of GPR's performance on each individual patient and SVM's speed for learning the second layer.



x': test point

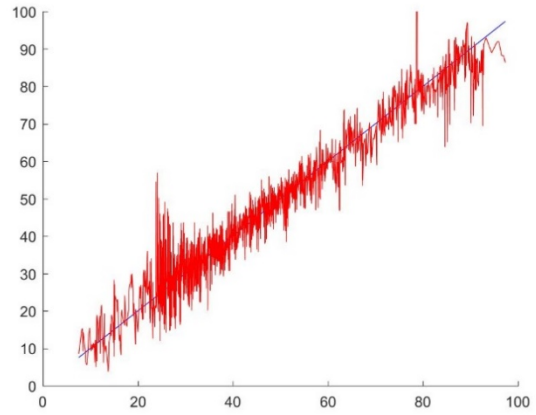
w_i:weight, $i = 1 \sim N$, where N: is training size

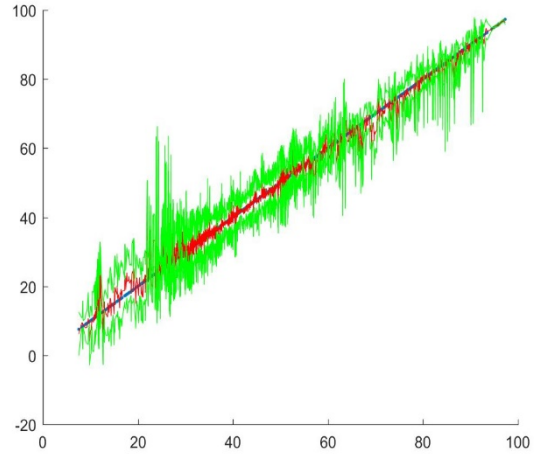
$h_i(\cdot)$: GPR model for each individual patient ID, $i = 1 \sim N$

$\tilde{h}(\cdot)$: final combinational model

Model 2 – Sparse GP

- Squared Exponential covariance
- Exact inference for a GP with Gaussian likelihood
- Linear mean function
- Gaussian likelihood function for regression





Using ID0-12 training and test sparse GP model

Model 3 – kernel for IDs

- We applied the following kernel function to kernelize the ID feature.
- The other part could be any kernels, such as Matern32 and Squared Exponential.

$$K(X, X') = K([id, x^c], [id, x^c]') = K(ID, ID')K(X^c, X^{c\prime})$$

$$K(id, id') = \begin{cases} 1 & if\ id = id' \\ c & if\ id \neq id' \end{cases}$$

Contents

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	Sparse GPR
reduced raw data	3500x384	1350x1	2.8205/84 sec	2.8875/2346 sec	8.286/18 sec
reduced PCA200	3500x200	1350x1	3.769/483 sec	3.8069/1693 sec	10.0659/499 sec
reduced PCA100	3500x100	1350x1	3.2344/559 sec	3.2453/1543 sec	9.2309/507 sec

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.

Future Work

- Data interpretation
- GP neural networks for accuracy prediction
- Advanced kernel function for ID feature, as following function[1]:

$$k_2(z_i, z_j) = \begin{cases} h(P_Z(z_i)) & \text{if } z_i = z_j \\ g(P_Z(z_i), P_Z(z_j)) & \text{if } z_i \neq z_j \end{cases}$$

[1] Villegas Garcia, M. A. (2013). An investigation into new kernels for categorical variables.

Q&A

- Thanks!
- Any Questions?