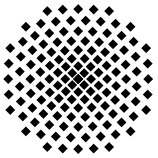


University of Stuttgart
Institute for Signal Processing and System Theory
Professor Dr.-Ing. B. Yang

LAB

Statistical Signal Processing

Pattern Recognition

Author: Yuxin Liu, Shanqi Yang, Qianqian Wei

Date of work begin: Date of work begin TBD
Date of submission: Date of submission TBD

Supervisor: Lukas Mauch

Keywords: Prostate Cancer Segmentation, Speaker Recognition

Abstract: Pattern recognition deals with a wide variety of problems. Methods of pattern recognition are prevalent in applications, such as the facial recognition and speech recognition systems. This pattern recognition lab includes two tasks, prostate cancer segmentation and speaker identification, which cope with medical imaging area and authentication field, respectively. The prostate cancer segmentation aims to detect the cancerous prostate area automatically, which saves cost and time for patients and medical workers. The speaker identification system can be used in several apps for authentication, like online banking, email services. As a consequence, the prostate cancer segmentation achieves an acceptable performance, and speaker identification system performs greatly.

Contents

1. Prostate Cancer Segmentation

1.1. Motivation

In the recently years, classification algorithms become more and more influence in the area of medical signal processing. Here we received different 3D image datasets from 14 patients and we implemented different classifier algorithms on the datasets to approach automatic prostate cancer segmentation(Separate cancer area and non-cancer area by algorithm selves). Before start, we can have a pre-view about the medical datasets.

Refer to Fig.1.1, it visualize 5 different image from medical datasets and what we are going to do in the first step is to extract the features before classifications .

1.2. Task Overview

Fig.1.2 reveals our process of Task.

1.3. Extract prostate region

We have seen the 5 different feature images in previous section. But the image we have seen contains feature vectors from regions with both prostate issue and non-prostate issue. In the Fig.1.3, the area with blue and red colors is the prostate area and blue color represents non-cancer area while red color represents cancer area respectively. We have to extract the prostate area because non-prostate area will not be considered in the classification.

The datasets of 14 patients actually can be divided into 2 parts, the labels of 1-11 stuck are made by 2 experts and the labels of 12-14 stuck are historical labels . So for the labels from last 3 datasets, we can directly choose the labels equal to 1 or 2 as prostate area because

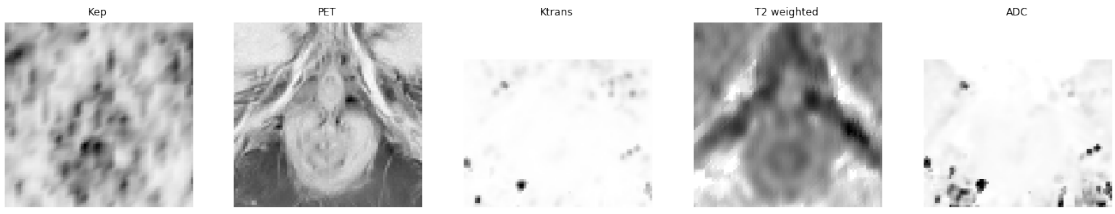


Figure 1.1.: Visualization of 5 different 3D images from left to right: K_{EP} , PET , K_{trans} , $T2weightedMR$, ADC

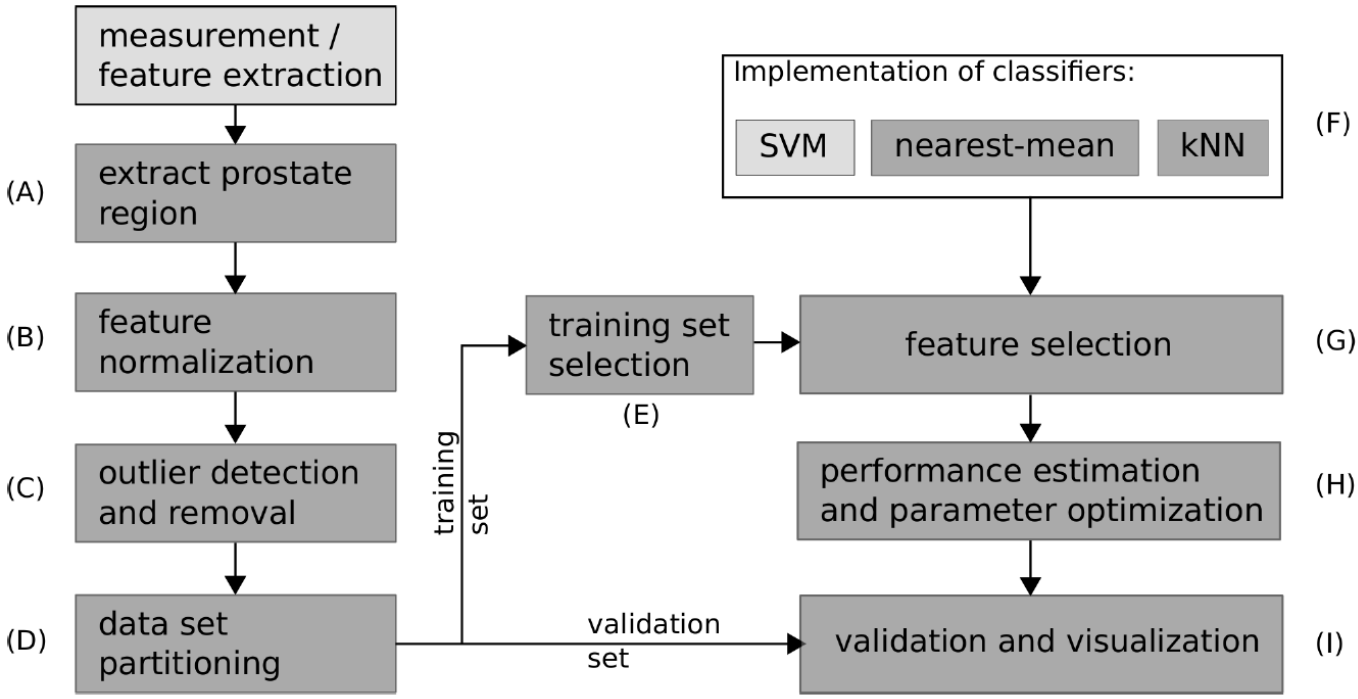


Figure 1.2.: Pipeline of Prostate Cancer Segmentation

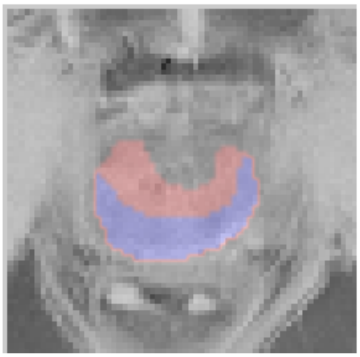


Figure 1.3.: Prostate area in the medical feature image

historical labels can be regarded as ground-truth. For the labels from first 11 datasets, we define the principle that :

$$m_c = \begin{cases} 0 & if\ m_1 \neq m_2 \\ m_1 & if\ m_1 = m_2 \end{cases} \tag{1.1}$$

The equation means that if expert A and expert B has the same conclusion on the voxel, we will take it into ground-truth dataset, otherwise abandon this voxel.

1.3.1. Feature Normalization

As we known , some classification algorithm are sensitive with the scales of features. Differences between features' value range will cause the inaccuracy to the result. To avoid that , we implement feature normalization and we select the standardization as the method:

$$\tilde{x}_k = \frac{x_k - \mu_k}{\delta_k} \tag{1.2}$$

Because scaling each dimension of the feature vector to zero mean and unit variance makes scaled features still fitting their original distribution.

1.3.2. Outlier detection and removal

However, even in the ground-truth datasets there is still a lot of anomaly samples in the dataset, these anomaly samples are so called outliers. Some of classifiers are also sensitive to the outliers. For instances, outliers will lead Nearst-Mean classifier to inaccuracy because the classifier will estimate wrong class centers $\hat{\underline{\mu}}_k$ when outliers within.

Outliers didn't follow the patterns of majority of samples, so it is hard to detect them in a simple way , the classical method to detect outliers is to compute the Mahalanobis distance of each sample:

$$MD_i = \sqrt{(x_i - \mu)C^{-1}(x_i - \mu)^T} \tag{1.3}$$

the mahalanobis distance illustrates the distance between class center and samples, also considering the distribution shape of datasets. After computing the mahalanobis distances, we then implement χ^2 distribution and compute the tolerance ellipse with cutoff value(setup w.r.t χ^2's quantile). There is no simple way to find an optimal number for cutoff value or χ^2's quantiles, only to visualize them. Fig.1.4 shows our detection result using mahalanobis distance with different χ^2's quantiles.

According to the Fig.1.4, we can clearly see the cutoff values' dash line, and those samples which are above the dash line are recognized as outliers. Fig.1.5 (a) also illustrate the locations of outliers, seems most of outliers locate in the center of prostate area. After detection of outliers, totally 23139 samples are sweep out from datasets.

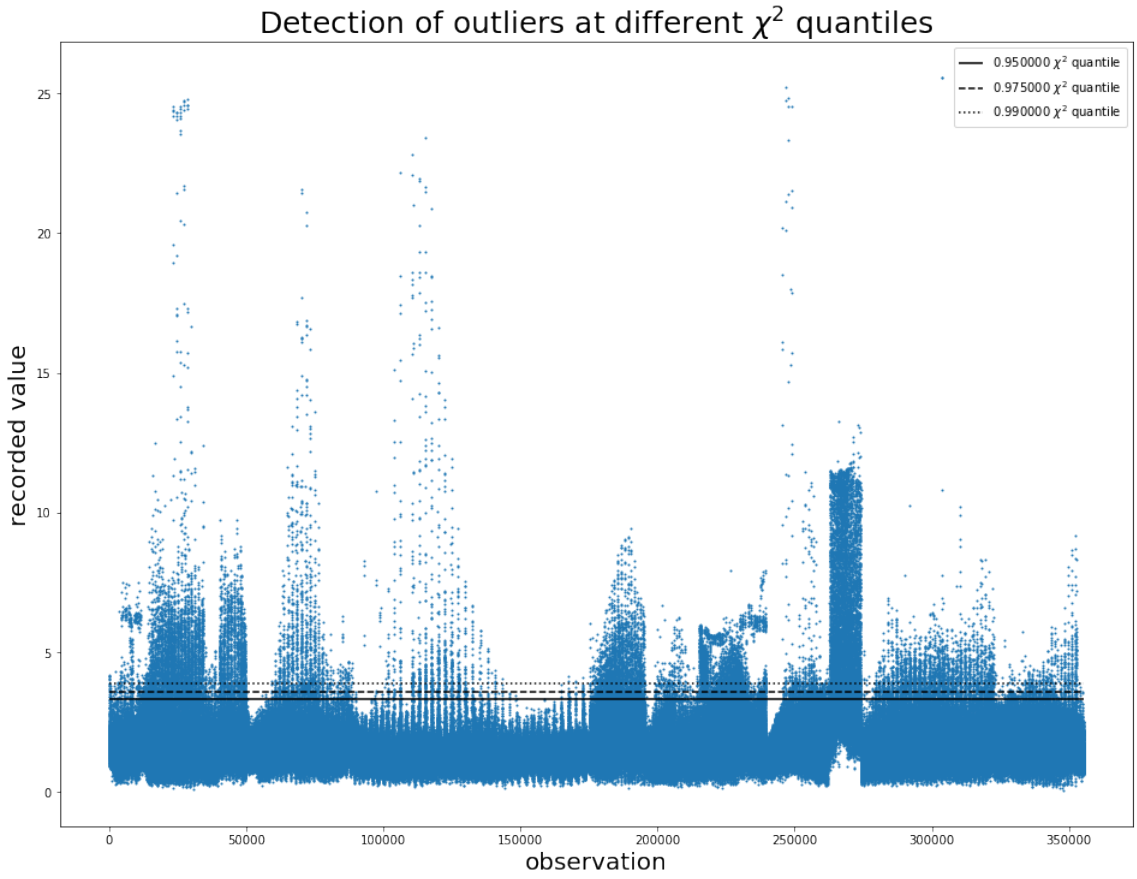
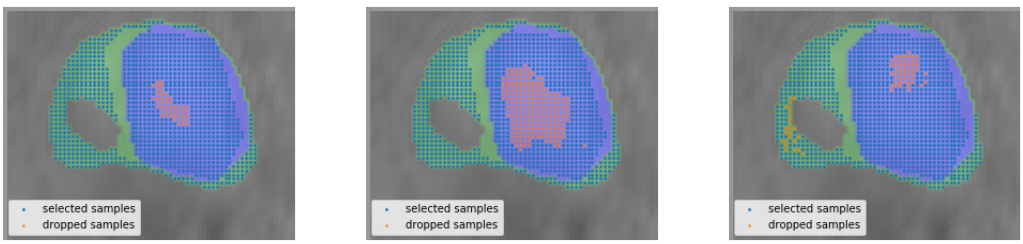


Figure 1.4.: Detection of Outliers at different χ^2 quantiles



(a) (b) (c)

Figure 1.5.: Visualization of outliers based on (a) Mahalanobis distance , (b) Robust distance with $m = 235$, (c) Robust distance with $m = 3000$

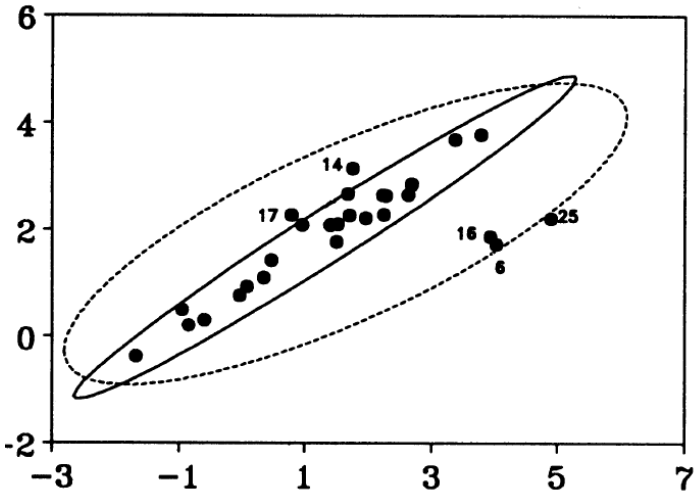


Figure 1.6.: Tolerance Ellipse based on Mahalanobis distance and Robust distance

As a further step, we think about estimators of multivariate location and covariance that have a high breakdown point. We will use minimum volume ellipsoid estimato (MVE), by inserting the MVE to estimate the mean and covariance then obtain the Robust distance RD_I. See Fig.1.6,

Compared with effects of obtaining Mahalanobis distance and Robust distance, the tolerance ellipse of applying Robust distance will be narrower as well as be more accurate. Here are how we define the Robust distance :

$$RD_i = \sqrt{(x_i - T(X))C^{-1}(x_i - T(X))^T} \tag{1.4}$$

The $T(X)$ and solution to minimize MVE is difficult to find, so we use resampling algorithm in our approach. Fig.1.7 shows the detection of outliers based on Robust distance.

The expectation of robust distances is obviously larger than mahalanobis distances'. Due to the same cutoff value, more samples will be recognize as outliers. After detection of outliers, totally 31936 samples are sweep out from datasets. In this approach, we set up the number of m subsample equal to 253, this number is determined by a probability augment:

$$1 - (1 - (1 - \epsilon)^{p+1})^m \geq p_0 \tag{1.5}$$

But the real value of subsample number m is far larger than the theoretical number. So we set up the subsample $m = 3000$, and made another detection. After detection, totally 43542 samples are recognized as outliers and the number is larger than the detection with $m = 235$. But when comparing with Fig.1.5 (b) and (c), we could find that the outliers are more distributed in the image applying $m = 3000$ and outliers are more concentrated in the image (b).It makes more sense that outliers are distributed widely instead of gathering in the one part of feature vectors.

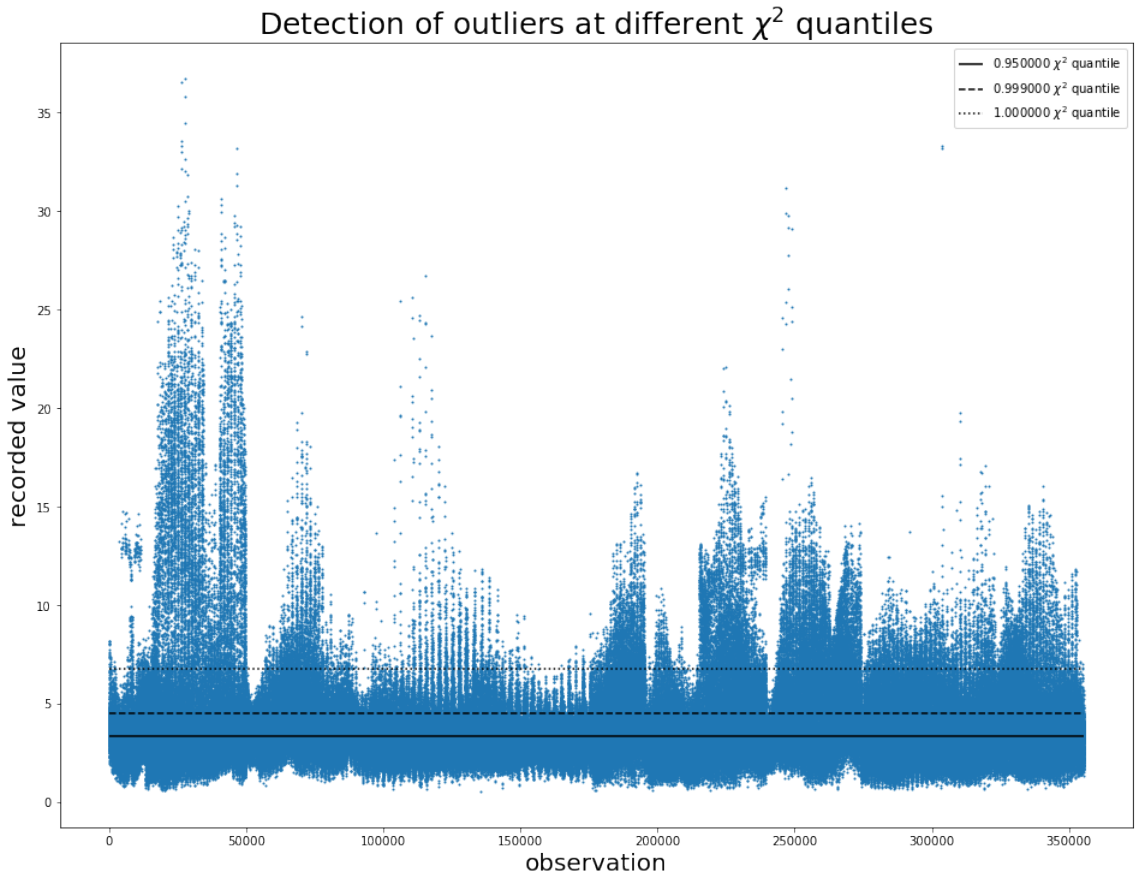


Figure 1.7.: Visualization of outliers based on Robust distance

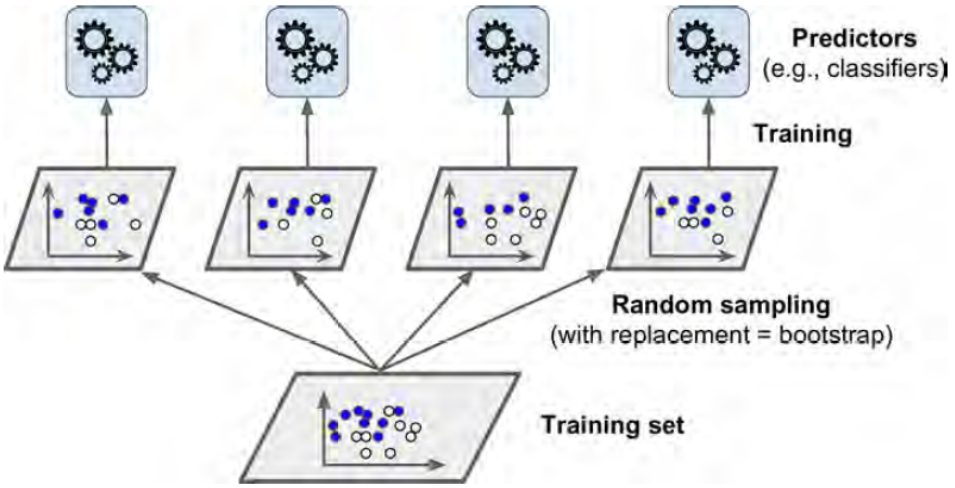


Figure 1.8.: The process of bagging classifier

1.4. Enhancement

We want to make further process to improve the performance of the algorithm so we implement one of the ensemble learning algorithms : Bagging classifier. Fig.1.8 illustrates the procedure of the bagging classifier. This algorithm can not reduce the bias of original model because

$$E[\frac{\sum X_i}{n}] = E[X_i] \tag{1.6}$$

But the variance change through the bagging shows below:

$$Var[\frac{\sum X_i}{n}] = Var[\frac{X_i}{n}] \tag{1.7}$$

According to the equation, the variance is significant reduced, this is because of the independences between child-models whose training data is random sampling from the whole dataset. Table.

We also tried the using Stacking to combined NN,kNN and SVM classifiers but performance is not so well. The reason is that Stacking is sensitive with the diversity of models. In future work, we shall try some different types of model and apply Stacking on them to improve the performance.

2. Task2

2.1. Evaluation

In last section we built up our generation and identification model and in order to get best performance of the model, It is necessary to analysis the hyper-parameters. We applied grid-search method on the parameters including SNR (Signal Noise Ratio), Maximum iterations and covariance types. All of these parameters may have significant impact on the model performance.

2.1.1. Signal Noise Ratio Analysis

In the previous section , we have implemented a voice detector to separate voiced frames and unvoiced frames, its mathematic equation follows . Now the threshold of the voice detector is going to ensured according to cross-validation performance.

As it is seen in the Table.2.1, the model has best Detection Rate when SNR threshold $\gamma = 10$.

2.1.2. Convergence Analysis

In previous section, new model coefficients are generated using EM algorithm. However the model doesn't converge with only one iteration. Then we tuned the maximum iterations of Gaussian Mixture Model , aiming to figure out when convergence achieved and its impact on detection rate. See the Table.2.2, we take expected value among cross validation sets under different maximum iteration. When only one iteration, the GMM model seems to be underfitting because both detection rate and log-PDF are low. With maximum iteration increasing, both detection rate and log-PDF raise. When maximum iteration equals to 3, expected detection rate reaches the peak, and afterward GMM seems to be overfitting that detection rate goes down with log-PDF arising. Due to that GMM api from Sklearn will automatically report convergence, so we know that the model is converged until maximum iteration is close to 100, but it is definitely overfitting.

But only tunned maximum iteration is not sufficient for analyzing convergence and performance, we also put the covariance types into consideration. Table.2.2 implements with FULL

Table 2.1.: Detection Rate with different SNR threshold

	$\gamma = 1$	$\gamma = 5$	$\gamma = 10$	$\gamma = 50$	$\gamma = 100$
Expected Detection Rate	0.994117	0.994705	0.995294	0.993529	0.991764

Table 2.2.: Performance & Convergence Analysis with FULL covariance type

	Expected Detection Rate	Expected Log-PDF	Cross_Valid Time (minutes)
$Max\_iter = 1$	0.995294	-47.724	39.2
$Max\_iter = 2$	0.997058	-52.958	34.5
$Max\_iter = 3$	0.997647	-53.023	38.8
$Max\_iter = 4$	0.997058	-53.031	37.1
$Max\_iter = 5$	0.997058	-53.021	43.2
$Max\_iter = 6$	0.995294	-53.000	46.0
$Max\_iter = 7$	0.994117	-52.966	60.0
$Max\_iter = 8$	0.992352	-52.935	57.1
$Max\_iter = 9$	0.991176	-52.924	59.2
$Max\_iter = 10$	0.991764	-52.930	61.5

Table 2.3.: Performance & Convergence Analysis with DIAGONAL covariance type

	Expected Detection Rate	Expected Log-PDF	Cross_Valid Time (minutes)
$Max\_iter = 1$	0.864117	-36,822	11.4
$Max\_iter = 2$	0.93	-36.606	15.1
$Max\_iter = 3$	0.957058	-36.651	16.7
$Max\_iter = 4$	0.958235	-37.744	17
$Max\_iter = 5$	0.965882	-36.816	18.7
$Max\_iter = 6$	0.97	-36.870	21.2
$Max\_iter = 7$	0.971176	-36.921	21.3
$Max\_iter = 8$	0.968235	-36.969	21.1
$Max\_iter = 9$	0.972352	-37.006	19.8
$Max\_iter = 10$	0.972941	-37.031	20.9

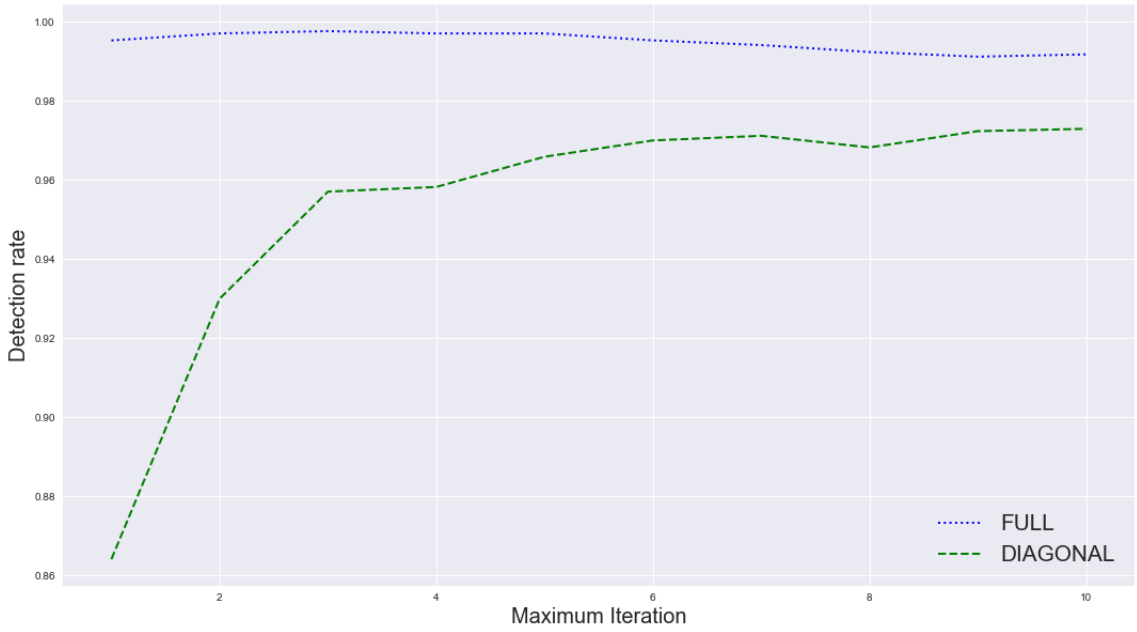


Figure 2.1.: Variation of Detection Rate with different covariance types in models

covariance type and Table.2.3 implements with DIAGONAL covariance. We make a contrast with both covariance types, the calculation time of FUll covariance is 2 or 3 times of DIAGONAL covariance's time. It is easy to think over that FULL covariance has more coefficients than DIAGONAL covariance so it will takes more time when FULL covariance. Furthermore, we plot the variation of detection rates of two covariance types.

Figure.2.1 illustrates that FULL covariance has faster convergence speed and better performance than DIAGONAL covariance does, detection rate of DIAGONAL covariance has a gap 2% between detection rate of FULL covariance when both reach peak plain. To figure out why, Figure.2.2 illustrates covariance types have the different effects on the GMM model distribution.

From the Figure.2.2, FULL covariance type may lead to a sloped ellipse but DIAGONAL covariance lead to an upright ellipse distribution. So our case, few GMM model with FULL covariance type can fit our voice dataset but the same number of models is not sufficient for DIAGONAL covariance type. One sloped ellipse distribution can be made up of several regular and upright ellipses. In conclusion, FULL covariance type fit the irregularly shaped dataset better and lead to faster convergence but it will cost more calculation time than DIAGONAL covariance type dose.

2.1.3. Result Evaluation

Finally, we find out the parameters with best performance lets see how good it is. We plot the confusion matrix of 10-cross-validation of 170 speakers, see the Figure.2.3 , only 4 samples of 1700 samples are misidentified.

Due to the Table.2.4, we found no gender misidentified(one gender misidentified to the other one) and our custom voice (last 2 speakers) are both identified correctly.In addition, We observe from more cross validations with different parameters that most misidentified speakers

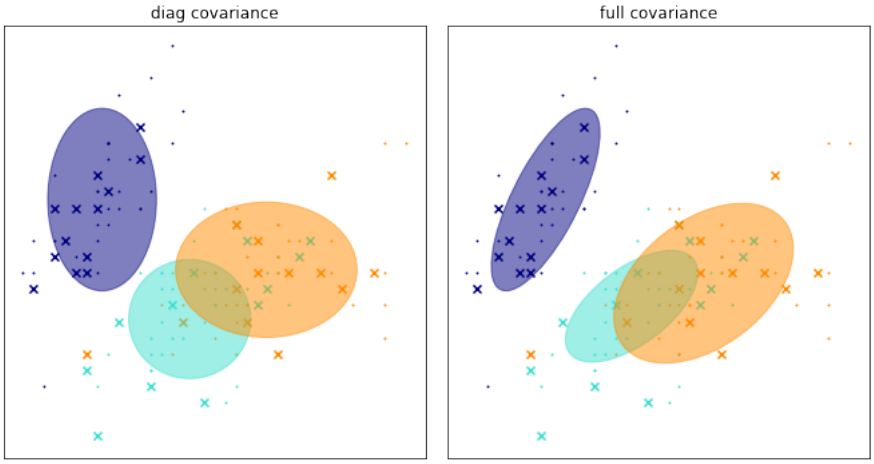


Figure 2.2.: GMM distribution Different Covariance Types

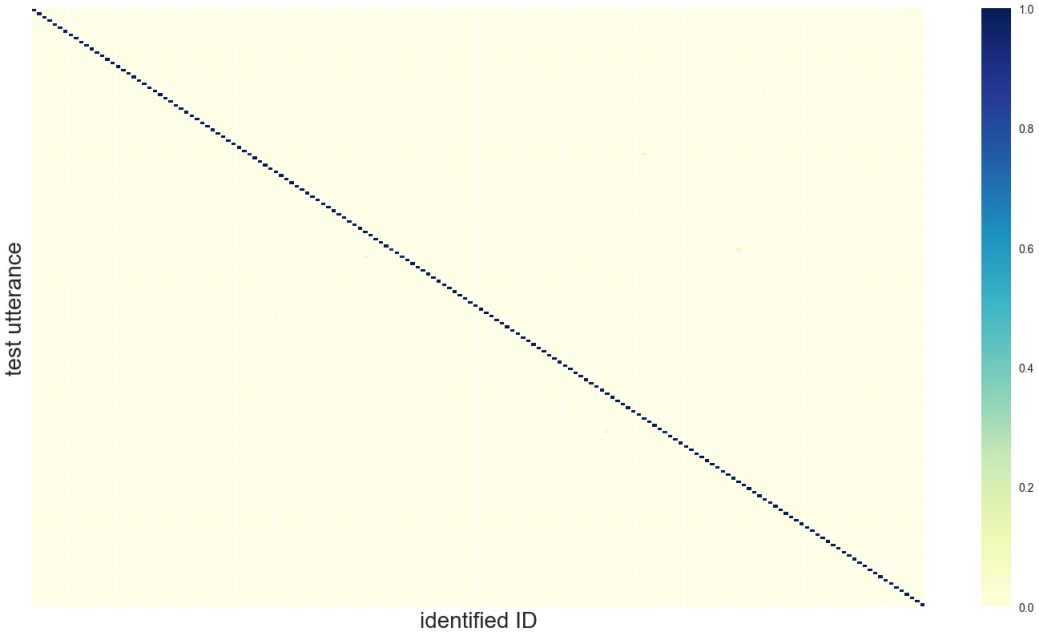


Figure 2.3.: Confusion Matrix Plot of 170 Speakers

Table 2.4.: Details of the misidentification

	True	False
Crossvalidation 4	mbwm0	mklt0
Crossvalidation 4	fjmg0	fadg0
Crossvalidation 5	mrrk0	mcrc0
Crossvalidation 9	fgjd0	fcau0

Table 2.5.: Configuration of the OSTI-SI Dataset

	Female	Male
Registered Speakers	64	106
Non-registered Speakers	66	124

are male, so we suppose that low frequency voice (the voice from most male has lower frequency than from most female) is a little bit difficult to be identified correctly in our model, it can be improved in future work.

2.2. Enhancement

2.2.1. Code Optimization

In comparison with using Matlab, Python has a big disadvantage that it is not good at handling large-size dataset. Although we transfered all the possible For-loops into Matrix computation, we still cost a lot of time on code implementation. To solve that the simplest way is to find a common-use api from good packages such as sklearn, scipy. We tried sklearn.gmm instead of mathematical implementation by ourselves and the speed rise to 50 times faster than before. However, api's framework is fixed and it can not fixed everything so our next approach is to write a Cython script to transfer *.py file into *.c and extend to .pyd. This method works well and the transfered code speed up 5 times faster than before.

2.2.2. Open-set,Text-independent Speaker Identification

From the content of previous sections, we have achieved high accuracy of Speaker identification among known speakers or we could see speakers registered in our model. However this model is not realistic because if there is an unknown speaker, the model will pair the unknown speaker to one of registered ID and it is obviously incorrect. So unregistered speaker identification is always tough challenge of general Speaker Identification. In order to solve that, we used OSTI-SI (Open-set,Text-independent Speaker Identification), evaluated the error rate and finally optimal the performance of Identification model.

Before start, we selected first 3 files of TIMIT's Training set $(dr1, dr2, dr3)$ as unknown speakers' voice set. Table.2.5 shows the people number, gender distribution of selected unknown speakers and in contrast with registered speakers.

The process of the open-set speaker Identification is shown in the Figure.2.4. The most important part is to set up threshold of comparison between registered and unregistered speaker. We introduced the ratio test between registered model and UBM's probability density function(PDF).

$$\frac{P(\underline{b}_{test}|\lambda)}{P(\underline{b}_{test}|\lambda_{UBM})} \underset{Unknown}{\overset{Known}{\gtrless}} \gamma \tag{2.1}$$

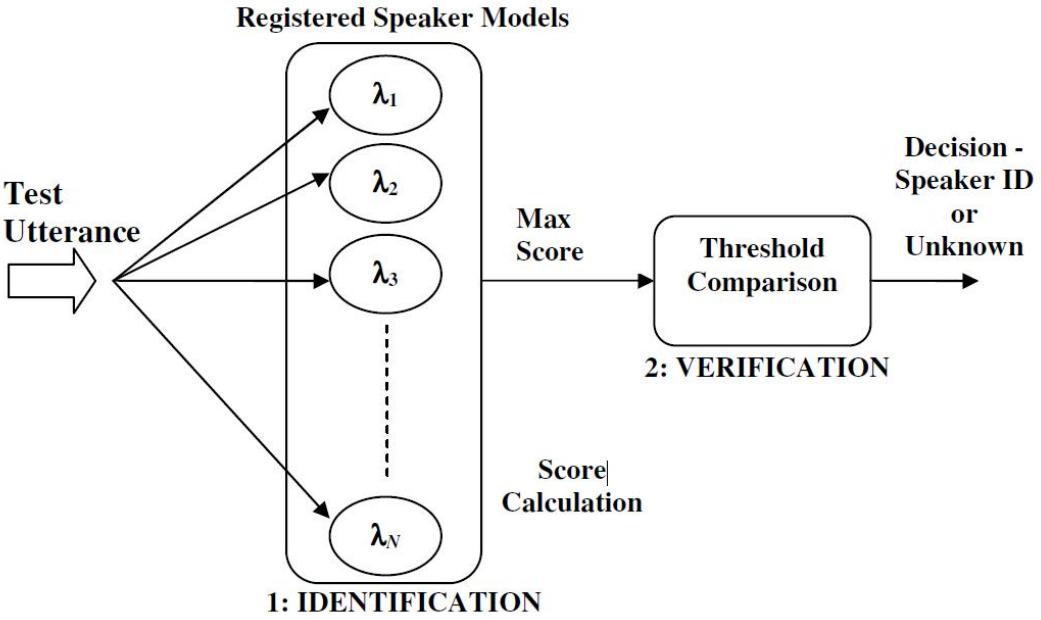


Figure 2.4.: Overview of the open-set, text-independent speaker identification process

As the Equation above, if the expected PDF from the registered model is larger than threshold ratio multiply PDF from UBM model, the speaker is one of known speakers, otherwise not. The reason why choosing UBM model for testing is that UBM model is trained from a large amount of people, so it has more universality than other registered models. Unknown speaker could have higher log-pdf in UBM model. Then we have to set up some indexes in order to evaluate the ratio test. In general, there are 3 types of error will happen in our model:

- a test utterance from one of registered speaker misidentified to another registered speaker, referred to Mislabelling (ML)
- a test utterance from one of registered speaker misidentified to unknown speaker, referred to False Rejection (FR)
- a test utterance from unknown speaker misidentified to one of registered speaker, referred to False Acceptance (FA)

So, the identification problem transfered into tradeoff problem between 3 types of error. In order to obtain the overall tradeoff performance, we set up Accumulative Error Rate (AER), it defines that

$$AER(\varsigma) = 100 * \frac{ML(\varsigma) + FR(\varsigma) + FA(\varsigma)}{T} \tag{2.2}$$

where ς is the threshold ratio and T is the total number of test voice set. Then we applied grid-search setting up several continuous threshold ratio and calculate the Expected ML rate, FR rate, FA rate and AER respectively through cross validation.

From the Figure.2.5, we got conclusion that with the threshold increasing, FA rate increases and FR rate drops. The best threshold is around 0.53 because it has minimum AER which is approximately 42%.

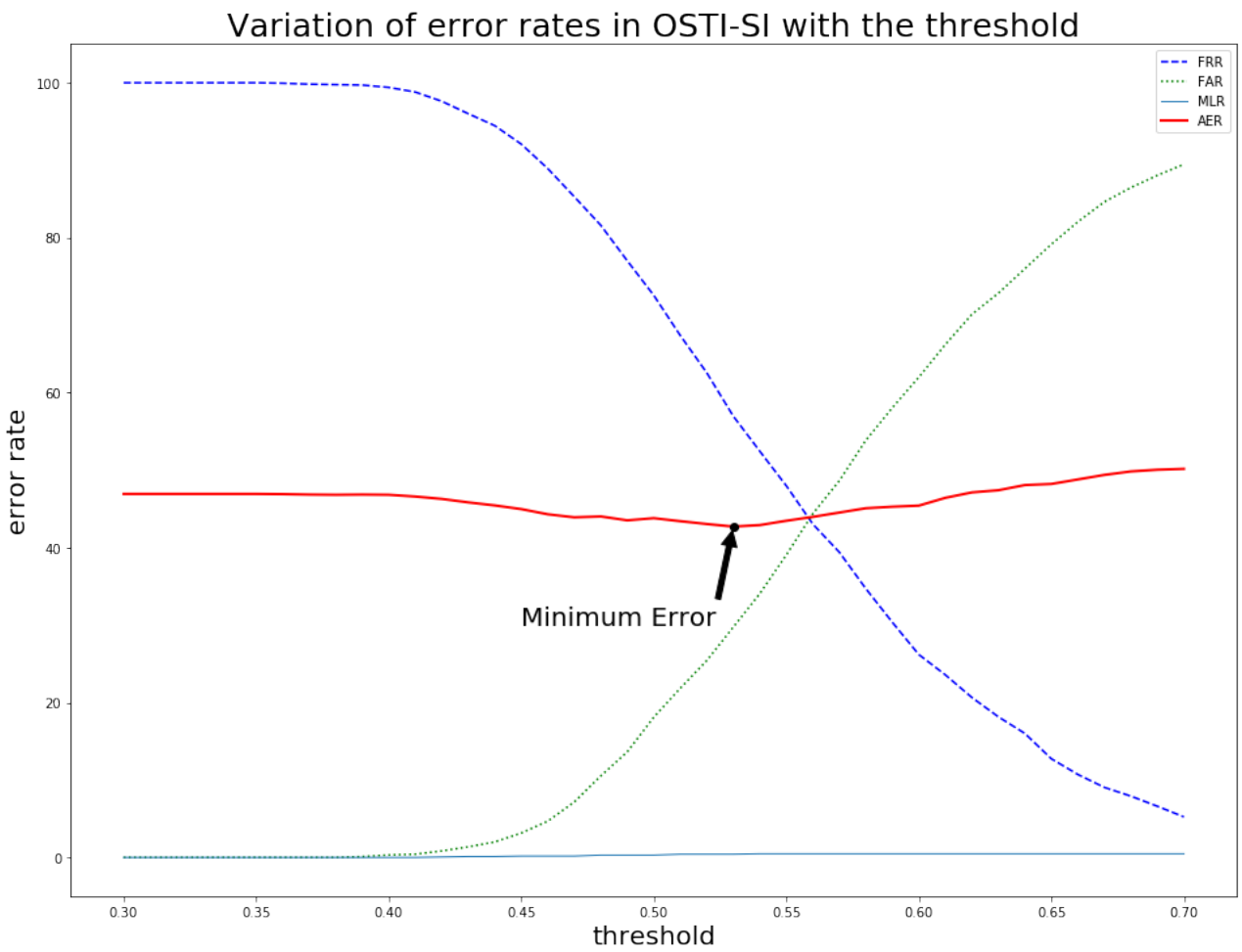


Figure 2.5.: Variation of error rates in OSTI-SI with threshold

In future work , we think it is promising to replace standard UBM with UBM with score normalization or other form, setting up more benchmarks and find out the best method with lowest error rate.

A. Additionally

You may do an appendix

List of Figures

List of Tables

Bibliography

Declaration

Herewith, I declare that I have developed and written the enclosed thesis entirely by myself and that I have not used sources or means except those declared.

This thesis has not been submitted to any other authority to achieve an academic grading and has not been published elsewhere.

Stuttgart, TBD Date of sign.

Yuxin Liu, Shanqi Yang, Qianqian Wei