EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems

Research Article EAI.EU

# Controlling Sensitivity of Gaussian Bayes Predictions based on Eigenvalue Thresholding

Dongxu Han*, Hongbo Du and Sabah Jassim

School of Computing, the University of Buckingham, Buckingham, UK

## Abstract

Gaussian Bayes classifiers are widely used in machine learning for various purposes. Its special characteristic has provided a great capacity for estimating the likelihood and reliability of individual classification decision made, which has been used in many areas such as decision support assessments and risk analysis. However, Gaussian Bayes models tend to perform poorly when processing feature vectors of high dimensionality. This limitation is often resolved using dimension reduction techniques such as Principal Component Analysis. Conventional approaches on reducing dimensionalities usually rely on using a simple threshold based on accuracy measurements or sampling characteristics but rarely consider the sensitivity aspect of the prediction model created. In this paper, we have investigated the influence of eigenvalue selections on Gaussian Bayes classifiers in the context of sensitivity adjustment. Experiments based on real-life data have shown indicative and intriguing results.

**Keywords:** Gaussian Bayes classifier, decision support system, PCA, eigenvalue.

Received on 18 September 2018, accepted on 18 October 2018, published on 29 November 2018

doi: 10.4108/eai.29-11-2018.155885

## 1. Introduction

Gaussian Bayes classifier is a type of probability-based classifiers that has been applied to many different classification problems with promising performance. One of the major challenges in Gaussian Bayes classifiers is the curse of dimensionality. High dimensionality causes the probability models trained to become oversensitive, especially when feature dimensionality is higher than the number of training samples, which often happens when images of various kinds are involved. To solve this problem, most conventional solutions aim at either creating adaptive hierarchical models [1] or conducting statistical analysis to reduce dimensionalities in advance as a pre-processing task [2] [3].

As an effective method to reduce feature dimensionality, Principal Component Analysis (PCA) is widely used in many different scenarios prior to the training of the classifier. PCA is a well-known transformation method that intends to project a given set of possibly correlated observations into an independent space, where the variance of each dimension in the projected space, known as eigenvalues, will be recorded in descending order to form a vector. In principle, a larger variance on the dimension implies richer information contained within it, so that it is reasonable to remove the dimensions that have relatively small variance (eigenvalue) since they are expected not to influence the classification result severely.

Traditional ways of selecting the principal components from the projected space would rely on a simple threshold that filters out the dimensions with relatively small eigenvalues. The threshold is normally determined by visual or statistical analysis on the scree-plot [4], matrix properties [5], or information gain [6]. However, as one of the special properties of Bayes classifier, the estimation of the

[*] Corresponding author. Email: 1303092@buckingham.ac.uk

conditional likelihood, especially the sensitivity of the estimation, are not normally considered as a critical factor when deciding the threshold for selecting the principal components.

In the current era, machine learning techniques have been widely adopted in many different fields in assisting human activities, such as Decision Support System in the clinical environment [7] or Risk Analysis on critical infrastructures [8]. These newly adopted application scenarios often require more precise predictions, in a measurable way, to serve much serious decision making with a certain level of confidence. Therefore, estimating not only the class label of an unknown observation, but also the likelihood of the prediction, becomes to an essential need. Unfortunately, the conventional classification approaches have not put such a requirement into very serious concerns.

As a pilot study, this paper aims at emphasizing on the importance of evaluating the conventional PCA approach in the context of decision assessment with respect to decision sensitivity. The potential alternatives of adopting refined PCA for Gaussian Bayes classifiers adjustment will also be discussed, and their theoretical impacts are analysed in the context of accuracy/sensitivity evaluation. All the hypnoses are examined with a real-life dataset, followed by further analysis and discussion.

The rest of the paper is organised as follows. Section 2 introduces the essential background knowledge regarding Gaussian Bayes Classifiers with a discussion of their uses and potential limits in Decision Score estimation. Section 3 explains the approaches taken to resolve/refine the limitations/estimation mentioned by adopting the Principal Component Analysis. The possible impact to the classification accuracy/sensitivity is also discussed with justifications. Section 4 shows the experimental work in classifying calcium type breast cancer with Gaussian Bayes Classifiers under different PCA thresholds. Section 5 discusses a potential statistical method for selecting appropriate thresholds. Section 6 concludes the paper with a summary of the work reported and future work needed.

## 2. Background

The Bayesian probability model is a well-used conditional model that has been applied to many different fields in machine learning. According to the Bayes' theorem [9], a conditional probability $\mathrm{P}(\omega_i|\vec{x})$ of having the predicted class $\omega_i$ at a given feature vector $\vec{x}$ can be expressed as

$$\mathrm{P}(\omega_i|\vec{x}) = \frac{\mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)}{\mathrm{P}(\vec{x})} \tag{2.1}$$

where $\mathrm{P}(\omega_i)$ is the prior regarding the natural expectation of the classified class, $\mathrm{P}(\vec{x}|\omega_i)$ is the posterior regarding the probability of having the observed feature vector $\vec{x}$ given that the class has been classified as $\omega_i$, $\mathrm{P}(\vec{x})$ is the evidence regarding the natural expectation of the observed feature vector $\vec{x}$.

These probabilities can be modelled by referring to a set of training samples. The form of the probability models depends on the nature of the data set, which can adopt various kinds of statistic models such as Bernoulli, Gaussian or Multinomial, etc. One of the commonly used models is the Multivariate Gaussian Model (MGM), which has an essential form of

$$\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi^d|\Sigma|}} e^{-\frac{(\vec{x}-\vec{\mu})\,\Sigma^{-1}\,(\vec{x}-\vec{\mu})^T}{2}} \tag{2.2}$$

where $\mathcal{N}$ represents a standard Gaussian probability density function for a given set of $d$ dimensional data with a mean vector $\vec{\mu}$ and a covariance matrix $\Sigma$. Furthermore, MGM often appears in a mixture form to model the data set closely, leading to a Multivariate Gaussian Mixture Model (MGMM). In other words, an MGMM is essentially a collection of sub-MGMs, each of which can be determined by a parameter set $\theta = \{W, \vec{\mu}, \Sigma\}$; $W$ represents the weight of the sub-models in the mixture and the summation of the weights of all the models is equal to 1. Therefore, given a sequence of $K$ parameter sets $\{\theta_{i=1\ldots K}\}$, i.e., $K$ Gaussian sub-models, the overall mixture model is characterized as:

$$\mathcal{N}(\vec{x} \mid \theta_{i=1\ldots K}) = \sum_{i=1}^{K} W_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) \tag{2.3}$$

Following these definitions, the feature-dependent probabilities $\mathrm{P}(\vec{x})$ and $\mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)$ in the original Bayesian form can then be further defined as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(\vec{x} \mid \theta_{\omega_{i=1\ldots k}}) \\ \mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(\vec{x} \mid \theta_{\omega_i}) \end{cases} \tag{2.4}$$

where the parameter set $\theta_{\omega_i}$ is derived from each relevant class set $\Omega_i = \{\omega_{i1}, \omega_{i2}, \ldots, \omega_{ik}\}$ and the weight of each set is considered as its proportion in the whole training set, i.e., $W_i = \frac{|\Omega_i|}{|\Omega|}$.

One of the benefits of MGMM Bayes predictions is that it provides an estimation of the likelihood of each class at different feature values, which allows the user to compare the classification strength under different conditions in a uniform manner. In our previous work [7], we have introduced a method to estimate the level of certainty regarding a specific classification decision made based on MGMM Bayes predictions, which refers to a Decision Score $S_D$ in a form as

$$S_D(\omega_i|\vec{x}) = 2\mathrm{P}(\omega_i|\vec{x}) - 1 \tag{2.5}$$

This definition is justified by the assumption that the level of classification certainty is directly proportional to the difference of the probabilities of the target class and the others without any transition bias. The sign of the decision score indicates the belonging of the class, which positive value indicates confirmation of the chosen class $\omega_i$ and a negative value indicates a preference of other classes. The absolute value of the decision score is the level of certainty regarding the decision made on the class belongings, which has a range [0,1).

## 2.1. Singularity in MGM

Although many researches have adopted Bayesian probability models under assumptions of independent events for simplification purpose, the Bayes' theorem does not require data independency in advance. However, the covariance matrix $\Sigma$ calculated from each class must be positive semi-definite in satisfying the modelling function required in Formula (2.2); otherwise the term $|\Sigma|$ and $\Sigma^{-1}$ in the formula are undefined.

The covariance matrix of an $n$ dimensional data set is a $n \times n$ matrix, where the diagonal terms $\{\Sigma_{i,i}\}$ are the $i^{\text{th}}$ variances ${\sigma_i}^2$ among the $n$ dimensions. The rest of the terms $\{\Sigma_{j,k} \middle| j \neq k\}$ in the matrix can be defined as a linear transformation of the product of the $j^{\text{th}}$ and $k^{\text{th}}$ standard deviations $\sigma_j\sigma_k$ among the $n$ dimensions with a gradient of the pair wised Pearson correlation coefficient $r_{j,k}$. In this representation, it can be easily noticed that the covariance matrix $\Sigma$ would be singular if $|r_{j,k}| = 1$ exists in any part of the matrix. In addition to this, the singularity would also occur if it exists a linear relationship of any kind across all possible dimension combinations in the data matrix. This can be an unavoidable risk in practice since significant correlations may naturally exist in raw training data, especially with very high dimensionalities. Therefore, the singularity threat must be addressed to avoid potential errors in computing decision models.

## 2.2 Sensitivity of MGM

The sensitivity of a decision model computed refers to the rate of change in the decision score predicted in corresponding to the change in feature value. In this context, the sensitivity can be simply presented as a function regarding the first order derivate of the decision score function $S_D$ in Formula (2.5) as

$$\nabla S_D = S_D(\omega_i|\vec{x})\frac{\partial S_D}{\partial x} \tag{2.6}$$

The sensitivity measure is an essential element in decision score evaluation since it reflects the behaviour of the prediction model in online testing. On the one hand, high sensitivity would reflect rapid changes in decisions in corresponding to a minor change in feature values, which may imply potential over fittings in the trained decision model. On the other hand, low sensitivity would result in a marginal difference in the score computed, which makes the classification outcome being undistinguishable (over-generalised) and becoming useless. A desirable classifier must be able to distinguish different classes on the one hand, and predict decisions without being oversensitive on the other.

# 3. Theory and Methods

## 3.1 Data projections based on PCA

Conventional PCA transformation is performed by using either the EigenValue Decomposition (EVD) or the Singular Value Decomposition (SVD), where EVD can only be applied to a real symmetric matrix and SVD can be applied to any real rectangle matrix. The end computational result between EVD and SVD should not differ significantly. However, we have chosen to use the SVD in this research since it is more numerically reliable than computing EVD over a positive semi-definite matrix [10].

In SVD, if we define $A$ as the original row data matrix of size $m \times n$, it can be then uniquely represented as

$$A = USV^T \tag{3.1}$$

where $U$ and $V^T$ are $m \times m$ and $n \times n$ orthogonal matrix respectively; $U$ and $V$ contain the left and right singular vectors of A respectively (one singular vector per column, sorted in descending order); $S$ is a matrix of the same size as A that is zero except its main diagonal, which contains the corresponding singular values (with a one to one mapping to the descending singular vector computed).

In this definition, Formula (3.1) is believed to result in a full SVD if $m > n$, where $U$ is a large $m \times m$ matrix with the last m − n columns that are considered as unnecessary fields. Therefore, it is normally adapted into an economy-sized SVD [11], which is more memory efficient in real practice. In this form, the original $U$ and $S$ are pruned by only preserving the first n columns and first n rows, which eventually contribute to a $m \times n$ and $n \times n$ matrix respectively; $V^T$ remains the same.

In a special case, $U$ and $V$ are considered as identical when A is a positive semi-definite matrix, the covariance matrix for example. Under this condition, the singular vectors contained in $U$ and $V$ are essentially the eigenvectors of the covariance matrix; the singular values contained in $S$ are the corresponding eigenvalues of the covariance matrix. Therefore, the projected feature vector $\vec{x}_{\text{PCA}}$ of the original feature vector $\vec{x}$ could then being simply defined as

$$\vec{x}_{\text{PCA}} = U\vec{x} \tag{3.2}$$

Following this projection process, the original data feature will be de-correlated, producing an MGM that has the same reading as Formula (2.2) but in a different form as

$$\begin{cases} \mathcal{N}(\vec{x}|\vec{\mu},\Sigma) = \prod_{i=1}^{\dim \vec{x}} \mathcal{N}\left(\Lambda(\vec{x})_i \middle| \Lambda(\vec{\mu})_i, S_{i,i}\right) \\ \Lambda(\vec{x}) = U^{-1}\vec{x}U \end{cases} \tag{3.3}$$

This relieves the computed MGM from the potential singularity threat since correlations are diminished through the PCA projection.

Indeed, the singularity issue can also be solved by projecting the original data into other spaces as long as the dimensions in the projected space are not fully correlated. However, the independent projection is still preferred since it can ease the management and computation cost in the MGM, which will be discussed in detail in the next section.

Computing PCA over large matrices in practice can be expensive. Therefore, an iterative approach is normally taken to reduce the cost of computation. One of the most commonly used and well-established iterative solutions is the NIPALS-PCA algorithm [12]. However, one main issue regarding the algorithm is that the orthogonality may be lost eventually due to the errors accumulated from each iteration, especially when the data computed has high dimensionality. Therefore, we have further adopted the Gram-Schmidt reorthogonalization method to correct the non-orthogonal principal components computed by the NIPALS method [13].

## 3.2 Influence on sensitivity by PCA

The independent projection by SVD as described in the previous section allows us to present the diagonal of $S$ as a vector of variances in each independent dimension as $\langle {\sigma_1}^2, {\sigma_2}^2, \ldots, {\sigma_n}^2 \rangle$. Based on the special property of the diagonal matrix $S$, the original MGM in Formula (2.2) can be presented in a simplified form as a product of Univariate Gaussian Models (UGM) as

$$\begin{cases} \mathcal{N}\left(\vec{x}\middle|\vec{\mu}, diag(\lambda)\right) = \prod_{i=1}^{\dim \vec{x}} \frac{1}{\sqrt{2\pi{\sigma_i}^2}} e^{-\frac{(\Lambda(\vec{x})_i - \Lambda(\vec{\mu})_i)^2}{2{\sigma_i}^2}} \\ \Lambda(\vec{x}) = \mathrm{v}^{-1}\vec{x}\mathrm{v} \end{cases} \tag{3.4}$$

which simplifies the original matrix multiplications into a linear form, and hence improves the time complexity from O($n^2$) to O($n$) while reducing the memory cost for storing the model computed. Moreover, it is important to note that Formula (3.4) always has a maximum reading when $\vec{x} = \vec{\mu}$. At this point, a density peak is formed and can be simply computed as

$$\prod_{i=1}^{\dim \vec{x}} \frac{1}{\sqrt{2\pi}\sigma_i} \tag{3.5}$$

Consequently, the range of the density function can then be defined as $(0,\ \prod_{i=1}^{\dim \vec{x}} \frac{1}{\sqrt{2\pi}\sigma_i}]$, which implies that the variation of the density value is directly proportional to $\prod_{i=1}^{\dim \vec{x}} \frac{1}{\sigma_i}$ with a coefficient of $\frac{1}{\sqrt{2\pi}}^{\dim \vec{x}}$. Therefore, the sensitivity of the prediction models can be justified by either controlling the dimensionality or the eigenvalues (variance in the independent space) in the projected model.

If we define ${\sigma_{\max}}^2$ as the maximum eigenvalue required to obtain a decision model that being sensitive/accurate enough to cover most of the feature observed while minimising the potential ambiguity between different classes, and define ${\sigma_{\min}}^2$ as the minimum eigenvalue required for computing decision scores without being over sensitive, the projection matrix $U$ can then be further pruned into a selective projection matrix $U'$ based on the independent covariance matrix $S$, also known as the singular value matrix in the economy SVD as

$$U' = \{U_i \mid {\sigma_{\min}}^2 < S_{i,i} < {\sigma_{\max}}^2\} \tag{3.6}$$

where any eigenvalues below the minimum threshold are considered as invalid since they cause the decision model being oversensitive; also, eigenvalues beyond the maximum threshold are again considered as invalid since they cause the decision model being over general. Finally, it is also worth noticing that

$$\frac{1}{\max\limits_{i=1\ldots\dim \vec{x}} \sigma_i} \le \frac{1}{\min\limits_{i=1\ldots\dim \vec{x}} \sigma_i} \tag{3.7}$$

which indicates that the Formula (3.5) can be dominated by $\min\limits_{i=1\ldots\dim \vec{x}} \sigma_i$. Therefore, although maximum threshold has more impact than the minimum threshold on the classification information in theory, controlling the minimum threshold is expectedly having a greater impact to the sensitivity, which is inversely proportional to the variation.

## 3.3 Sensitivity measurement

As we have defined in Formula (2.6), sensitivity is a measurement of the gradient of the decision score function at different feature values. However, finding the exact solution of the gradient of a decision score function can be very costly especially with high dimensionalities. Therefore, in our study, the gradient of the decision score function is calculated using the following method of approximation:

$$\nabla S_D = \frac{S_D(\omega_i|\vec{x}+h) - S_D(\omega_i|\vec{x})}{\|h\|} \tag{3.8}$$

where $h$ is considered to be an extremely small number and been set to $10^{-15}$ in our experiment. This form of calculation does not only decrease the computational cost, but also improves the adaptability of the sensitivity measurement; since the decision score function is treated as a black box and the result can be computed without knowing the detailed characteristics of the function in advance.

Following this definition, we are able to obtain a set of sensitivity measurements $\nabla S_{\mathcal{D}} = \{\nabla S_{D1}, \nabla S_{D2}, \ldots\ \nabla S_{Dn}\}$ for any given testing set $X' = \{\vec{x}'_1, \vec{x}'_2, \ldots, \vec{x}'_n\}$. Then, the variance of these sensitivity measurements $\nabla\sigma^2$ is calculated to reflect the change in sensitivities at different test readings as

$$\nabla\sigma^2 = \frac{\sum_{i=1}^{n}(\nabla S_{Di} - \overline{\nabla S_{\mathcal{D}}})^2}{n-1} \tag{3.9}$$

To uniform this measure, the computed $\nabla\sigma^2$ is passed through a transfer function and finally result into a coefficient $\nabla c$ regarding the sensitivity measurements of the test set provided as

$$\nabla c = \sqrt{\frac{\nabla\sigma^2}{1+\nabla\sigma^2}} \qquad (3.10)$$

where $\nabla c$ is a real number that has a range of [0,1). The decision score model is considered as sensitive to a set of testing samples if $\nabla c$ is close to 1 and be considered as stable if $\nabla c$ is close to 0.

## 4. Experimental Analysis

### 4.1 Dataset Description

It makes good sense to put the theory into tests using data sets collected from practice. To test the idea, we need the data set to be reasonably big size and of high dimensionality. In our experiment, therefore, we have used the CBIS-DDSM dataset [14]. The dataset contains 2,620 mammography images for breast cancer studies with the relevant region of interests that have been verified by human experts. The dataset included two major types of tumours as the “mass” and “calcium” with relevant pathology results. For this study, the experiment has been conducted on calcium type tumours only based on the concern of feature visibility. Images have been cropped based on the region of interest provided. The final selected dataset contains 1,872 images that include 1,199 benign cases and 673 malignant cases.

These images have been further randomly sampled into 10 individual patches to implement a 10-fold cross-validation process. Nevertheless, this random sampling process was stratified to ensure the original prior of the dataset remaining undistorted. In the other words, each of the sampled patches is designed to contain benign classes and malignant classes in a ratio close to 1.78 : 1, which is the ratio in the whole data set before partitioning.

### 4.2 Feature Extraction

As a well-known texture-based feature, Grey Level Co-occurrence Matrix (GLCM) has been used in many studies that relate to mammography classifications [15] [16]. In GLCM, the matrix is computed based on pixel neighbours to reflect the frequency of the occurrence of certain patterns, where the pattern can be identified in different distance and angles. After the raw data matrix has been computed, statistical texture measurements are normally conducted to extract high-level descriptors with uniform dimensions.

In our experiment, following the well-used guidance that has been proposed by Haralick [17]. GLCM has been computed on three different distances (1, 2 and 3) with four different angles (0°, 45°, 90°, and 180°). Then, 13 statistical moment measurements (excluding the maximal correlation coefficient) are extracted from the raw matrix according to Haralick’s suggestion, which finally results in a feature of 3 × 4 ×13 = 156 dimensions.

### 4.3 Classification Modelling

Classification modelling in this research follows a Naïve Bayes scheme with MGMM embedded within it for simplicity. In addition, the PCA method presented in Section 3.1 projects the original data of the MGMM model into an orthogonal space, which does not only ensure the independency requirement of the Naïve Bayes classifier, but also simplified the original MGMM into a Univariate Gaussian Mixture Model (UGMM).

The dimensions in the orthogonal space are essentially the projections of multiple dimensions in the original space. Consequently, modelling each projected dimension with a UGM only is no longer sufficient, since the data points in the projected space are reflecting information from multiple dimensions, especially when the projected dimension has relatively large eigenvalues. Therefore, it is more reasonable to model the data points on each projected dimension with a UGMM, where eventually the class model on the projected multidimensional space should be presented as a mixture of UGMM. However, the creation of UGMM on each projected dimension can no longer follow the approach mentioned in Formula (2.3) due to the ambiguity of projection from different dimensions. As a solution, UGMM on each projected dimension was created by adopting the well-used Expectation Maximization (EM) method, where the threshold of the log-likelihood has been set to $10^{-5}$.

### 4.4 Experiment Results

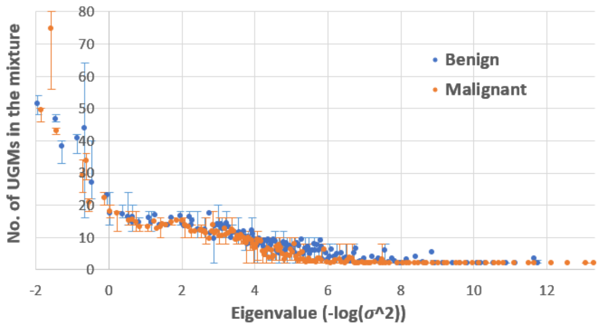

**Figure 1.** Number of UGMs in the mixture in each projected dimension with different eigenvalues

In our experiment, we have first observed the number of UGMs in the mixture on each projected dimension with different eigenvalues. The average of the 10 test patches is plotted in Figure 1, where the error bars are indicating the maximum and minimum readings among the 10 test patches.

The scatter plot has shown a clear positive relationship between the number of UGMs in the mixture and their eigenvalues on the dimension projected, where the dimensions with larger eigenvalues tend to require more UGs in the mixture to describe the behaviours of the class. This does match our expectations since the dimensions with larger eigenvalues tend to contain more information and therefore yields a more complex projection. We further discovered that the number of UGMs required on the projected dimensions falls significantly as the corresponding eigenvalues decrease; however, this trend started to be stabilized after the eigenvalue has fallen below 1. This fact can be seen as an experimental evidence of the "eigenvalue-one criterion", which states that the projected dimension with an eigenvalue that is less than 1 can be abandoned due to the relatively small information gain from them [18].

Following the analysis, the classification accuracy and sensitivity measurements under different thresholds on maximum and minimum eigenvalues are recorded and plotted in Figure 2 and 3 respectively. The scatter points in these plots represent the average of the cross-validation results and the error bars reflect the best and worst readings among them. The initial seed used for generating the 10-fold random samples is fixed. Therefore the testing environments under different eigenvalue thresholds are identical and comparable.

As expected in Section 3.2, thresholding on eigenvalues indeed affects overall accuracy in the experiment. In general, it is expected that the accuracy decreases as the dimensionality reduces. However, thresholding minimum and maximum eigenvalues have shown more specific behaviours.

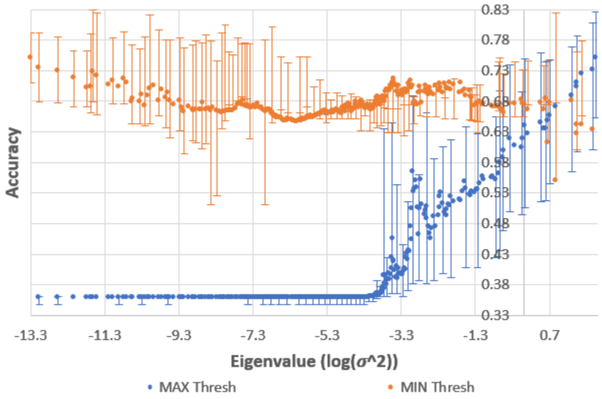

**Figure 2.** Accuracy measurements in relation to different eigenvalue thresholds

As shown in Figure 2, thresholding on maximum eigenvalues had a clear and significant linear impact to the overall accuracy, which reached a minimum and remained stable at $10^{-3.9}$. This linear impact was caused by the strong proportional relationship between the scale of eigenvalues and the amount of information gained from them. Abandoning dimensions with large eigenvalues directly reduces the information gained by the classifier and therefore impair the classification accuracy, which the impact appears to surpass the ambiguities contained within these dimensions. The stabilized minimum reading after the decay indicates that the remaining dimensions can no longer provide a sufficient amount of information to support further classifications. Thresholding on maximum eigenvalues appears to have a consistently large error margin. This could be caused by the eigenvalues on the minimum side as we have discussed in Formula (3.7), since small eigenvalues tend to cause the prediction model being extremely sensitive to the testing samples, which will be discussed in more detail in the next paragraph.

In contrast to the previous readings, thresholding on minimum eigenvalues had a moderate effect in general, where the accuracy was decaying in the beginning, then followed by a steady increase after $10^{-6}$ and finally ended with another significant decrease. The initial decrease in accuracy reading is very much understandable since the reduction in dimensionality causes more ambiguities in the lower dimensional space. Meanwhile, as we have mentioned in Formula (3.7), smaller eigenvalues should have more dominating effects compare to the large ones, which are more likely to cause the decision model being overfitted in the high dimensional space. This explained the reason for the increase between $10^{-6}$ to $10^{-1.7}$, which implies that the projected dimensions with eigenvalues that is less than $10^{-1.7}$ can be potential causes of the overfitting in the classification. Removing the overfitted dimensions essentially makes the classifier more robust and improving testing accuracy. An evidence that supports this argument is the error bars reflected on the scatter plots. The error bars remain consistently large at the beginning of the plot and then starts to decrease in size along with the increase in accuracy, which indicates that the initial classification results were very sensitive and unstable to different test sets but then becoming more and more robust along with pruning the dimensions with small eigenvalues. At the end of the plot, the continuous pruning of dimensions starts to have an escalating effect on the information lost and eventually

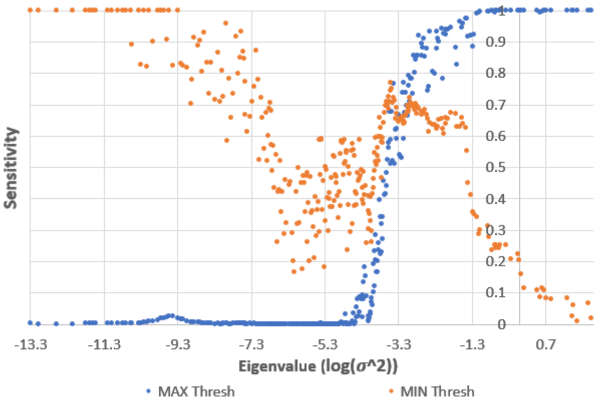

**Figure 3.** Sensitivity measurements in relation to different eigenvalue thresholds

causes the classifier predicting more errors, which is reflected by the drop on accuracy and by the increase in error margins.

Regarding the sensitivity measurements, the method introduced in Section 3.3 reflects the sensitivities of a group of samples, which makes the testing results being strongly dependent on the testing data. Consequently, it yields a large variation in the test readings. Nevertheless, the average reading among the testing patches will still be a good indication of the overall sensitivity under different thresholds.

As shown in Figure 3, thresholding on minimum eigenvalues had a clear impact to the classification sensitivity. This again reflects our expectation since small eigenvalues tend to generate abrupt MGMM decision model that being very distinct from the others, which cause the measurements being susceptible. As the experiment result shows, cohering to the observations from Figure 2, sensitivity decreases significantly in the beginning and then being stabilized after $10^{-6}$ and finally ended with another significant drop after $10^{-1.9}$. The initial decrease on sensitivity essentially demonstrates the reducing on classification overfitting along with the reducing on dimensionality, which reached a floor eventually when the robustness of classification was established in testing. However, the classification accuracy will decrease consistently along with the reducing of dimensionality. As a result, accuracy eventually falls to a point where the predictions made by decision models become unstable. The consistent error made during the testing subsequently causes a rise in the sensitivity measured, shown as the increase around $10^{-1.9}$. This increase is immediately followed by a second decrease in sensitivity, which highlighted that the consistent errors made starts to cause the classifier bias towards one of the classes and therefore led to insensitive predictions.

In general, on the one hand, the clear drop in sensitivity at the beginning of the scatter plot was an indication on potential overfittings; on the other hand, the significant change on sensitivity at the end of the plot was reflecting potential underfittings of the prediction model. An ideal threshold should be a value that positions around the first trough of the plot, where the decision model with pruned dimensions will be neither underfitted nor overfitted.

Comparing to minimum eigenvalue thresholding, thresholding on maximum eigenvalues has shown a much consistent impact to the sensitivity. However, this does not imply that thresholding on maximum eigenvalues indeed affects sensitivities. The constant reading of extremely sensitive results at the beginning of the plot was essentially an observation of the dominating effect from minimum eigenvalues. The significant decrease in sensitivities after $10^{-1.1}$ was again caused by the decrease in classification accuracy, where the bias generated by classification error eventually yields insensitive predictions. Therefore, maintaining minimum eigenvalue unchanged eventually preserves the high sensitivity yield by the overfitted prediction models, causing the effect of thresholding on maximum eigenvalues being less obvious and noticeable.

## 5. Discussions

Our preliminary experiments were conducted on stringently controlled variables, where one of the two thresholds always remain constant at the maximum/minimum evaluations. However, it is still desirable to further test our hypnosis in a fully variable environment to reveal the relationship between the two thresholds.

Tuning the eigenvalue thresholds on both maximum side and minimum side coherently can be a challenging task due to the different magnitudes of information contained in each side. A practical solution in determining the appropriate thresholds could rely on the use of Confidence Interval (CI). In statistics, confidence interval is a type of estimation that defines the likelihood of observing a certain event at a given confidence level [19]. In a two-tails test, confidence level is always bounded with an upper limit and a lower limit, which can be adapted as the minimum and maximum thresholds of the given observation on eigenvalues computed. In practice, a probability distribution model can be first created from the eigenvalues observed. The CI analysis can then be applied to this distribution. As a result, we are able to obtain the maximum and minimum eigenvalue threshold as the upper and lower limits of the CI at any confidence level specified. In this form, the eigenvalue thresholds can be determined and tuned in the context of confidence levels as an imperial method depends on the environmental requirements.

Conventional CI analysis is assuming a normal distribution of the data sample, where the symmetric characteristic of the distribution should ease the modelling and computing of the analysis. Unfortunately, eigenvalues do not follow a normal distribution. As we observed in Figure 1, most of the eigenvalues computed are relatively small and the frequency of the observation decreases along with the increase of eigenvalues. Therefore, it would be better to define the eigenvalue distribution as a positively skewed distribution.

Currently, the essential form of the eigenvalue distribution has not been fully investigated. Most of the theories regarding the distribution of eigenvalues can only be supported by inductive approximations and massive computing simulations [20] [21]. As a result, validating the method proposed in this section can be extremely ambiguous and unpractical due to the consistent debating on the newly proposed hypnosis. In addition, the unsymmetrical property of the eigenvalue distribution causes the computation of the CI being very difficult. Determination of the appropriate CI can only be done through massive computing with Monte Carlo method [22] or approximation in controversial kinds [23]. Therefore, at the current stage, we would still like to recommend to threshold the eigenvalues with predefined and constant values. However, approaches based on CI can be further tested and validated in future along with the growing understanding on eigenvalue distributions.

## 6. Concluding Remarks

In this paper, we have first introduced MGMM based decision prediction models with relevant discussion regarding their potential challenges. Following these, we have investigated a PCA based solution and its properties in the context of classification sensitivities. Experiment results have shown a positive proof on the theory proposed. Thresholding on minimum eigenvalues has indeed shown an evident effect on reducing sensitivities, where reducing dimensionality continuously eventually increases modelling sensitivity in corresponding to the increase in classification error. The ideal threshold of the minimum eigenvalues would be recommended at the first trough of the sensitivity curve plotted, where it has shown to be around $\mathbf{10^{-6}}$ to $\mathbf{10^{-4}}$ in our experiment. However, further investigation of different data sets is still required to prove the ubiquitous of the values found.

Thresholding on maximum eigenvalues has also reflected our expectation to a certain extent. However, the effect on sensitivity controlling cannot be observed clearly due to the overwhelmed influence from the lower eigenvalues. Further study regarding the impact of thresholding on maximum eigenvalues is still required, especially in an environment where the minimum threshold is controlled in various magnitude.

## References

[1] Palatucci, M. and Mitchell,T. (2007) “Classification in Very High Dimensional Problems with Handfuls of Examples” *Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD).*

[2] Zhang L.M., Peña M. J. and Robles V. (2009) “Feature selection for multi-label naive Bayes classification,” *Information Sciences*, vol. 179, no. 19, pp. 3218-3229.

[3] Liu J. and Chen S. D. (2009), “Fault Detection and Identification Using Modified Bayesian Classification on PCA Subspace,” *Industrial & Engineering Chemistry Research,* vol. 48, no. 6, pp. 3059-3077.

[4] Kanyongo Y. G. (2005) “Determining The Correct Number Of Components To Extract From A Principal Components Analysis: A Monte Carlo Study Of The Accuracy Of The Scree Plot,” *Modern Applied Statistical Methods,* vol. 4, no. 1, pp. 120-133.

[5] Choi Y., Taylor J. and Tibshirani R. (2017) “Selecting the number of principal components: Estimation of the true rank of a noisy matrix,” *The Annals of Statistics,* vol. 45, no. 6, pp. 2590-2617.

[6] Tamura M. and Tsujita S. (2007) “A study on the number of principal components and sensitivity of fault detection using PCA,” *Computers and Chemical Engineering,* vol. 31, no. 9, p. 1035–1046.

[7] Han D., Du H. and Jassim S. (2016) “Towards a Confidence-Centric Classification Based on Gaussian Models and Bayesian Principles,” *The Ninth York Doctoral Symposium on Computer Science and Electronics*, York,.

[8] Staalduinen A. M., Khan F., Gadag V. and Reniers G. (2017) “Functional quantitative security risk analysis (QSRA) to assist in protecting critical process infrastructure,” *Reliability Engineering & System Safety,* vol. 157, pp. 23-34.

[9] Carlin P. B. and Loui A. T. (2000) *Bayes and empirical Bayes methods for data analysis (2nd edn),* New York: CHAPMAN & HALL.

[10] Zhang W., Arvanitis A. and Al-Rasheed A. (2012) “Singular Value Decomposition and its numerical computations,” Michigan Technological University, Michigan.

[11] MathWorks, (2013), “*Eigenvalues and Singular Values*,”. [Online]. Available: https://www.mathworks.com/content/dam/mathworks/mathworks-dot-m/moler/eigs.pdf. [Accessed 25 Jun 2018].

[12] Risvik H. (2007), “*Principal Component Analysis (PCA) & NIPALS algorithm*,” University of Oslo, Oslo,.

[13] Andrecut M. (2008) “Parallel GPU Implementation of Iterative PCA Algorithms,” *Journal of computational biology: a journal of computational molecular cell biology,* vol. 16, no. 11, pp. 1593-1599.

[14] Lee S. R., Gimenez F., Hoogi A. and Rubin D. (2017) “A curated mammography data set for use in computer-aided detection and diagnosis research,” *SCIENTIFIC DATA,* vol. 4.

[15] Majeed F. T., Al-Jawad N. and Sellahewa. H. (2013) “Breast Border Extraction and Pectoral Muscle Removal in MLO Mammogram Images,” in *5th Computer Science and Electronic Engineering Conference (CEEC)* , Essex.

[16] Elshinawy M., Badawy H. A., Abdelmageed W. and Chouikha M. (2011) “Effect of breast density in selecting features for normal mammogram detection,” in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Chicago.

[17] Haralick M. R. (1979) “Statistical and structural approaches to texture,” *Proceedings of the IEEE,* vol. 67, no. 5, pp. 786-804.

[18] Cardoso S. J. and Cruz-Almeida Y. (2016) “Moving beyond the eigenvalue greater than one retention criteria in pain phenotyping research,” *Pain,* vol. 157, no. 6, pp. 1363-1364.

[19] J. Kragten (1994), “Tutorial review. Calculating standard deviations and confidence intervals with a universally applicable spreadsheet technique,” *Analyst,* vol. 119, no. 10, pp. 2161-2165.

[20] L. Pastur and M. Shcherbina (2011) “Eigenvalue Distribution of Large Random Matrices,” Providence, Rhode Island: American Mathematical Society.

[21] Y.-K. Liu (2001) “Statistical Behavior of the Eigenvalues of Random Matrices,” in *Mathematics Junior Seminar,* Princeton, New Jersey.

[22] R. Y. Rubinstein and D. P. Kroese (2016) Simulation and the Monte Carlo method, John Wiley & Sons.

[23] V. V. Patil and H. V. Kulkarni (2012) “COMPARISON OF CONFIDENCE INTERVALS FOR THE POISSON MEAN: SOME NEW ASPECTS,” REVSTAT – Statistical Journal, vol. X, no. 2, pp. 212 - 227.