Robustness of Raw Images Classifiers Against the Class Imbalance – A Case Study

Ewaryst Rafajłowicz(✉)

Faculty of Electronics, Wrocław University of Science and Technology, Wrocław, Poland
ewaryst.rafajlowicz@pwr.edu.pl

Abstract. Our aim is to investigate the robustness of classifiers against the class imbalance. From this point of view, we compare several most widely used classifiers as well as the one recently proposed, which is based on the assumption that the probability densities in classes have the matrix normal distribution. As the base for comparison we take a sequence of images from that laser based additive manufacturing process. It is important that the classifiers are fed by raw images. The classifiers are compared according to several criterions and the methodology of all pair-wise comparisons is used to rank them.

Keywords: Matrix normal distribution · Bayesian classifier
Robustness of classifiers · Class imbalance

1 Introduction

The class imbalance phenomenon, i.e., a largely different fractions of examples from different classes in the learning and in the testing sequences, is known to cause troubles when learning and assessing the quality of classifiers. The reason is in that most of the known classifiers tend to give the priority to the largest class in the learning sequence. This, in turn, leads to a poor generalization properties. On the other hand, the class imbalance is unavoidable when classifiers are used for detecting rare events (e.g., faults in production processes or diagnosis of rare diseases).

Many attempts were proposed in order to circumvent this difficulty. They can be, roughly, clustered as follows.

1. Data editing strategies that attempt to artificially increase the fraction of the minority class (classes) examples in the learning and in the testing sequences. Typically, it is achieved by either the re-sampling from the minority class or by the under-sampling from the majority class or by combining them. These ways, although useful in many cases, have one common drawback, namely, they distort a priori class probabilities, which – in turn – may lead to undesirable preference voting for the minority class.

K. Saeed and W. Homenda (Eds.): CISIM 2018, LNCS 11127, pp. 154–165, 2018.
https://doi.org/10.1007/978-3-319-99954-8_14

2. Attaching a high cost for a minority class misclassification, in particular, by using a dedicated metrics.
3. Designing classifiers dedicated to cope with the class imbalance phenomenon.

Our approach differs from the above. Namely, we take several popular classifiers and we propose to rank them from the view point of their robustness against the class imbalance in the data. In addition to the popular classifiers, we consider also the classifier for matrix normal distributions (see [5,8,9]).

The second challenge in comparing the robustness of classifiers against the class imbalance is the choice of criterions for their comparisons. Again, a number of criterions is advocated in the literature. For this reason, we propose to use a pair-wise comparisons of classifiers, for which several criterions are calculated. This approach was originated by Slowinski [11] and its applicability is still growing (see [4]).

As an empirical material for case studies we take raw images of the laser additive manufacturing process (see [8] for more detailed description why this process is important).

An important issue in our case study is that we put raw images as the inputs of classifiers. This approach seems to be of importance at least for two reasons, namely,

– it demonstrates that easily available PC computers can be successful in a cheap way of classifying images, since the process of features extraction is time-consuming (expensive)
– the results of comparisons of classifiers are not biased by a human-dependent way of feature extraction.

The paper is organized as follows.

– In Sect. 2 we provide the description of a modified classifier for matrix normal distributions.
– The well known classifiers that are selected for comparisons are listed and briefly commented in Sect. 3.
– In Sect. 4 we describe the methodology of testing and comparisons as well as their results.
– Section 5 contains conclusions, while in the Appendix we summarize the known properties of matrix normal distributions.

2 A Modified Classifier for Matrix Normal Distributions

2.1 MND as Class Densities and Their Estimation

We assume that probability distributions of gray-level images from class $j = 1, 2, \ldots, J$ have MND with probability density functions (p.d.f.'s) $f_j(\mathbf{X})$ defined in Appendix.

MND densities have special covariance structure in comparison to a general multivariate Gaussian densities. Namely, their covariance matrices do not have inter rows-columns covariances, which makes them much easier to estimate (see Appendix).

Further on, we assume that we have J learning sequences of the following form: $\mathbf{X}_i^{(j)}$, $i = 1, 2, \ldots N_j$, $j = 1, 2, \ldots, J$.

Denote by $p_j > 0$, $j = 1, 2, \ldots, J$, $\sum p_j = 1$, a priori class probabilities. It is well known that in a general case the MAP classifier assigns $\mathbf{X}$ to class j^* such that

$$j^* = \arg\max_j [p_j \, f_j(\mathbf{X})], \tag{1}$$

where $\arg\max_j [\,]$ stands for the argument for which the maximum is attained. It is also well known that this rule is the optimal one when the 0-1 loss function is used (see, e.g., [2]).

For symmetric and positive definite matrix A define the following function: $\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$, which indicates how large numerical errors can be committed when the inverse of A is calculated. Select $0 < \kappa_{max} < 100$.

A Modified Matrix Normal Distribution Classifier (MMNDCL)
I. The Learning Phase

Step (L1) Collect J learning sequences (for each class) of the following form: $\mathbf{X}_i^{(j)}$, $i = 1, 2, \ldots N_j$, $j = 1, 2, \ldots, J$.

Step (L2) Estimate the class mean matrices and a priori class probabilities as follows

$$\hat{M}_j = N_j^{-1} \sum_{i=1}^{N_j} \mathbf{X}_i^{(j)}, \quad \hat{p}_j = N_j/N, \quad j = 1, 2, \ldots, J. \tag{2}$$

Step (L3) Calculate the maximum likelihood estimates (MLE) of the inter-row and inter-column covariance matrices by solving the following set of equations:

$$\hat{U}_j = \frac{1}{N_j \, m} \sum_{i=1}^{N_j} (\mathbf{X}_i - \hat{\mathbf{M}}_j) \, \hat{V}_j^{-1} \, (\mathbf{X}_i - \hat{\mathbf{M}}_j)^T, \tag{3}$$

$$\hat{V}_j = \frac{1}{N_j \, n} \sum_{i=1}^{N_j} (\mathbf{X}_i - \hat{\mathbf{M}}_j)^T \, \hat{U}_j^{-1} \, (\mathbf{X}_i - \hat{\mathbf{M}}_j) \tag{4}$$

for $j = 1, 2, \ldots, J$. Equations (3) and (4) can be solved by the flip-flop method.

Step (L4) Estimate the normalization constants of class densities as follows:

$$\hat{c}_j = (2\,\pi)^{0.5\,n\,m} \det[\hat{U}_j]^{0.5\,n} \det[\hat{V}_j]^{0.5\,m}. \tag{5}$$

II. The recognition Phase

Step 1 Acquire $\mathbf{X}$ to be classified.

Step 2 Check whether all the inequalities:

$$\kappa(\hat{U}_j) < \kappa_{max}, \; j = 1, 2, \ldots, J \tag{6}$$

as well as

$$\kappa(\hat{V}_j) < \kappa_{max}, \; j = 1, 2, \ldots, J \tag{7}$$

are fulfilled. If so, go to Step 3, otherwise, go to Step 4.

Step 3 Classify new image (matrix) $\mathbf{X}$ according to the following rule:

$$\hat{j} = \arg \min_{1 \le j \le J} \left[\frac{1}{2} \operatorname{tr}[\hat{U}_j^{-1} (\mathbf{X} - \hat{\mathbf{M}}_j) \, V_j^{-1} \, (\mathbf{X} - \hat{\mathbf{M}}_j)^T] \right] - \log(\hat{p}_j / \hat{c}_j), \tag{8}$$

where $\hat{j}$ is the predicted class for $\mathbf{X}$.
Acquire the next image (matrix) $\mathbf{X}$ for classification and repeat (8).

Step 4 Classify new image (matrix) $\mathbf{X}$ according to the nearest mean rule, i.e., classify it to the class

$$\tilde{j} = \arg \min_{j} ||\mathbf{X} - \hat{M}_j||^2, \tag{9}$$

where the squared distance $||\mathbf{X} - \hat{M}_j||^2$ is defined as follows:

$$||\mathbf{X} - \hat{M}_j||^2 \; = \; \operatorname{tr}[(\mathbf{X} - \hat{\mathbf{M}}_j) \, (\mathbf{X} - \hat{\mathbf{M}}_j)^T]. \tag{10}$$

If the class $\tilde{j}$ in (9) is selected in a sufficiently sure way, e.g., if the following condition holds for a pre-specified $\zeta > 0$

$$(1 + \zeta) \, ||\mathbf{X} - \hat{M}_{\tilde{j}}||^2 < ||\mathbf{X} - \hat{M}_j||^2, \quad j \neq \tilde{j}, \tag{11}$$

then update the estimates of $\hat{U}_{\tilde{j}}$ and $\hat{V}_{\tilde{j}}$ by adding current $\mathbf{X}$ to the learning sequence as $(\mathbf{X}, \tilde{j})$. Independently whether condition (11) is fulfilled or not, go to Step 1.

It was proved in [14] that it suffices to perform only one flip-flop operation in Step (L3) in order to obtain the efficient estimates of U_j and V_j.

2.2 The Methodology of Cross-Validation Testing

In order to test MMNDCL algorithm and to verify its robustness against the class imbalance, we used the cross-validation (CV) methodology in the following extensive version.

Step 1 Select from the set of images of the length 900 (at random with the same probabilities) a learning sequence of the length 450 and denote it as L_{450}. The rest of the sequence, denoted as T_{450}, use it for testing.

Step 2 Learn MMNDCL, using the L_{450} sequence.

Step 3 Test the classifier from Step 2, applying it to T_{450}, calculate and store the accuracy and other quality criterions (recall, precision, etc., see the next subsection).

Step 4 Repeat Steps 1–3 1000 times.

Step 5 Provide the averages of the quality indicators, obtained in Step 3, as the outputs.

Notice that this is an intensive testing procedure, because we have to estimate two matrices of the means and four covariance matrices, each of them 1000 times when learning MMNDCL.

3 Classifiers Selected for Verifying Their Robustness Against the Class Imbalance

The following classifiers are selected for comparisons.

(a) The MMND classifier in the version that was described in the previous section.
(b) The logistic regression classifier with L2 regularization coefficient equal 1. We refer the reader to [12] for a contemporary description of this classifier.
(c) The naive Bayes classifier. Despite of its simplicity, this classifier works quite well in many applications. It is of interest to check its robustness against the class imbalance.
(d) The feed forward, sigmoidal neural network classifier with the following parameters: two hidden layers, each containing 900 nodes with tanh activation functions. L2 regularization coefficient equal 0.1 was used (see [3]).
(e) The random forest (RF) classifier was proposed in the famous paper of Breiman [1] in which also the proof of consistency is provided. The popularity of the RF classifiers is still growing. In our experiments, the number of generated trees was 200.
(f) The support vector machine (SVM) classifier is currently considered as the classifier of the first choice in most of applications. In our experiments, the Gaussian radial basis functions were used. The soft margin parameter was selected to be 8.
(g) The nearest neighbors classifier is the golden classic. Its consistency and other properties are investigated in [2]. In the experiments reported in the next section, the version with 10 nearest neighbors (referred to as 10-NN) is reported.

We refer the reader to [2] for a wide and deep discussions concerning classifiers and their properties.

4 The Results of Testing Classifiers by Cross-Validation

Before providing the results of testing classifiers, we briefly discuss criterions that are selected for comparisons. We also provide a short description of the methodology of comparing classifiers when multiple criterions are used.

4.1 Criterions Selected for Comparisons

When testing a two class classifier on a large number of examples, we collect the following data:

– TP – the number true positive examples,
– TN – the number of true negative cases,
– FP – the number of false positive examples,
– FN – the number of of false negative cases.

Thus, the total number of test cases is $FP + FN + TP + TN$.

The following, widely used, measures of classifiers quality are selected for further comparisons.

Accuracy. The accuracy (Acc) is defined as the ratio of all properly classified patterns to all the patterns in the testing sequence:

$$Acc = \frac{TP + TN}{FP + FN + TP + TN}. \tag{12}$$

It is well known that Acc is not quite adequate, especially when we are faced with a large class imbalance, since it can provide a seemingly high accuracy just by classifying improperly all (or most) items from the minority class.

Recall. The recall (Rec), also known as the sensitivity, is defined as

$$Rec = TP/(TP + FN), \tag{13}$$

i.e., it is the proportion of positive patterns that are correctly classified. It does not take into account TN and FP cases.

Precision. The precision (Prec), also called (specificity), defined as

$$Prec = TP/(TP + FP) \tag{14}$$

is – in fact – the true positive accuracy. It does not take into account TN and FN cases.

F1 Score. The F1 score (F1sc) attempts to reduce the drawbacks of Rec and Prec measures by calculating their harmonic mean:

$$F1sc = 2.0\, Prec\, Rec/(Prec + Rec). \tag{15}$$

Although F1sc is more informative than Prec and Rec separately, it still neglects TN cases, which are of importance in class imbalance cases.

Matthews Correlation Coefficient. A widely accepted alternative to F1sc is the Matthews Correlation Coefficient (MCC) that is defined as follows:

$$MCC = \frac{(TP\,TN - FP\,FN)}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}}. \tag{16}$$

MCC takes into account all the entries of a classifier confusion matrix. MCC is easy to interpret. Namely, if MCC is close to $+1$, then a classifier at hand provides a good prediction. Conversely, MCC being about -1 indicates that a classifier works properly, but it is advisable to exchange the roles of "true" and "false" classes. Finally, when MCC is near zero, then a classifier is not a good predictor at all, i.e., the tossing of the fair coin would provide comparable results.

4.2 Multiple Criteria Sorting for Assessing Classifiers Quality

The above discussion of the quality measures of classifiers indicates that all of them, although widely used, have also their drawbacks. For this reason, we propose to apply all of them in our case study. This leads to the need of selecting a method for multiple criteria sorting.

Problems of multiple criteria sorting (ranking) of objects have a long history that is documented in a large number of papers. For our purpose of sorting the classifiers according to the above criterions, we use a simplified version of the approach proposed in [4] (see also [11] for the discussion of the fundamental notion of the pair-wise comparisons).

Denote by $a_1, a_2, \ldots, a_7$ the set of algorithms (classifiers) to be compared. Let $g_1, g_2, \ldots, g_5$ stands for the set of criterions defined in the previous subsection. Then,

$$\bar{g}(a_i) \stackrel{def}{=} [g_1(a_i), g_2(a_i), \ldots, g_5(a_i)], \quad i = 1, 2, \ldots, 7 \tag{17}$$

is the vector of criterions that are evaluated for algorithm a_i.

Select $\epsilon_k > 0$ as the level of uncertainty of k-th criterion, i.e., if $|g_k(a_i) - g_k(a_j)| < \epsilon_k$, then a_i and a_j are considered to be equivalent with respect to k-th criterion, $k = 1, 2, \ldots, 5$. When algorithms (classifiers) a_i and a_j, $i \neq j$ are compared as one pair, then the following rules of adding scores to their total scores (denoted as S_i and S_j, respectively) are applied.

Scoring the comparison of a_i and a_j
For $k = 1, 2, \ldots, 5$ perform the following steps.

Step (C1) If $|g_k(a_i) - g_k(a_j)| < \epsilon_k$, do not change S_i and S_j and set k to $k+1$ (Step (C2) is not performed).

Step (C2) If $g_k(a_i) > g_k(a_j)$, then set $S_i := S_i + 1$ and $S_j := S_j - 1$. Set k to $k + 1$ and go to Step (C1), unless $k > 5$, otherwise, finish the comparison of a_i and a_j.

Overall Comparison. Initialize the all pairs comparison approach by setting $S_i = 0$, $i = 1, 2, \ldots, 7$. Perform Step (C1) and (C2) for all pairs of algorithms $i \neq j$, $i, j = 1, 2, \ldots, 7$. Sort S_i's as the output of the all pairs comparisons and consider the one with the largest S_i as the winner.
Remarks.

(1) In the next subsection $\epsilon_k = 0.01$ for $k = 1, 2, \ldots, 5$ was selected.
(2) An easy generalization of the above approach to multi-criteria comparisons is to attach nonnegative weights to criterions and to use them in Step (C2), instead of ± 1, but we skip this generalization in the next subsection.

4.3 The Empirical Material

As an empirical material for comparisons we selected 900 images of the laser based additive manufacturing process. In [8] it was explained in details why it

is important to distinguish cases when the laser head is in the middle of a wall to be constructed (class 1) versus the cases when it is near endpoints of the wall (class 2). Roughly speaking, when it is recognized that the laser head is near endpoints of the wall, it is desirable to reduce the laser power in order to prevent the endpoints to be too thick.

Clearly, one can expect that the empirical material contains a smaller number of examples of Class 2 than that of Class 1 since the laser head moves much longer along the middle of the wall than near its endpoints. Indeed, in the testing sequence of images we had 29 images from Class 2 out of all 450 images and a similar fraction in the learning sequence.

Typical examples of images from Class 1 and 2 are shown in Fig. 1 (upper row). These original images have the size of 111×241. Down-sampled images of the size 10×22 were supplied as inputs of the classifiers (see Fig. 1 – lower row). Notice that such images as in Fig. 1 (lower row) were inputs of the tested classifiers, without applying any features extraction.

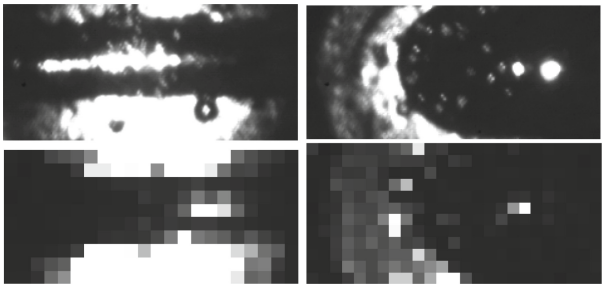

Fig. 1. Examples of images: Classes 1 and 2 (from the left). Original images – upper row and down-sampled images – lower row.

Table 1. The confusion matrix obtained when the MMNDCL (as described in Sect. 1) is applied to 450 long testing sequence.

	Pred. Cl. 1	Pred. Cl. 2	Sum
Act. Cl. 1	416	4	420
Act. Cl. 2	2	28	30
Sum	418	32	450

4.4 The Robustness Against the Class Imbalance – The Results of CV Testing

In this subsection we provide the comparisons of the classifiers that are important from the view-point of their robustness against the class imbalance. Conclusions

Table 2. The confusion matrix obtained when the logistic regression classifier (left panel) and the naive Bayes classifier (right panel) are applied to 450 long testing sequence.

	Pred. Cl. 1	Pred. Cl. 2	sum
Act. Cl. 1	415	6	421
Act. Cl. 2	0	29	29
sum	415	35	450

	Pred. Cl. 1	Pred. Cl. 2	sum
Act. Cl. 1	397	24	421
Act. Cl. 2	0	29	29
sum	397	53	450

Table 3. The confusion matrix obtained when the artificial neural network classifier (left panel) – with two hidden layers and 900 nodes, having tanh activation function – and the random forest classifier (right panel) are applied to 450 long testing sequence.

	Pred. Cl. 1	Pred. Cl. 2	sum
Act. Cl. 1	406	15	421
Act. Cl. 2	0	29	29
sum	406	44	450

	Pred. Cl. 1	Pred. Cl. 2	sum
Act. Cl. 1	420	1	421
Act. Cl. 2	0	29	29
sum	420	30	450

Table 4. The confusion matrix obtained when the SVM classifier (left panel) and 10-NN classifier (right panel) are applied to 450 long testing sequence.

	Pred. Cl. 1	Pred. Cl. 2	sum
Act. Cl. 1	418	3	421
Act. Cl. 2	0	29	29
sum	418	32	450

	Pred. Cl. 1	Pred. Cl. 2	sum
Act. Cl. 1	421	0	421
Act. Cl. 2	4	25	29
sum	425	25	450

Table 5. The summary of the tests of the classifiers for robustness against the class imbalance. In columns 3–5 the values of criterions for each classifier are displayed. Column 6 contains the scores collected by each classifier according to all the pairs comparisons (see Sect. 4.2). In column 7 the classifiers are ranked according to the scores gained in column 6.

nr	ind./meth.	Acc.	MCC	Rec.	Prec.	F1sc.	Comp.	Rank
a	MMNDCL	0.987	0.896	0.990	0.995	0.993	4	**1**
a	Log.-Reg.	0.987	0.904	0.986	1.000	0.993	2	**2≡ 3**
c	n.-Bayes.	0.947	0.718	0.943	1.000	0.971	−16	**7**
d	Neural n.	0.967	0.797	0.964	1.000	0.982	−12	**6**
e	Rand-for.	0.998	0.982	0.997	1.000	0.999	2	**2≡3**
f	SVM	0.993	0.948	0.993	1.000	0.996	1	**4**
g	10-NN	0.991	0.924	1.000	0.991	0.995	0	**5**

are based on the values of criterions and on the multiple criteria sorting of classifiers that were discussed in the previous subsections. Firstly, we display the confusion matrix for each classifier. Then, in Table 5 we provide the comparison of the classifiers and their sorting, according to the methodology of all the pairs comparisons that is described in Subsect. 4.2.

As expected in the class imbalance case, the confusion matrices in Tables 1, 2, 3 and 4 and the Acc. column in Table 5 display very high accuracies of all the classifiers. However, according to the rest of the criterions (columns 4–7 in Table 5), these classifiers are essentially different. In particular, when the methodology of all the pairs comparisons (see Subsect. 4.2) is applied, then they collect largely different scores (see column 8 in Table 5). In the analysis of this column, notice that a classifiers which is essentially the dominant over all the other classifiers, with respect of all the criterions, would be able to collect at most plus 30 scores. Conversely, a classifier that is dominated by all the other six classifiers would gain minus 30 scores.

From this point of view, the classifiers: MMNDCL, Logistic Regression, Random forests, SVM and 10-NN collected non-negative scores, which means that they are – to some extent – robust against the class imbalance. On the other hand, and somewhat unexpectedly, the naive Bayes the neural networks classifiers gained high negative scores for the comparisons.

In column 9 of Table 5 the ranking of the classifiers is presented, which is based on the scores that are shown in column 8. Formally, the winner is the MMND classifier, when the methodology of Subsect. 4.2 is used. Its success can be explained by the fact that it is essentially based on the rule of the nearest mean (in the Mahalonobis or the Euclidean distances). Notice however, that the winner MMNDCL is only slightly better than the last non-negatively tested 10-NN classifier. Notice also that both the MMNDCL and the 10-NN classifiers require only 10–30 images for a proper functioning. This is in contrast to all the others competing methods. On the other hand, the losers, i.e., the naive Bayes and the neural networks classifiers, are more global and they require relatively more longer learning sequences than they are usually in our disposal.

On the other hand, when only the MCC criterion is considered, the winner is the Random Forest method, while the MMNDCL is ranked at the 4-5 position.

5 Conclusions

A modified MAP classifier for images (matrices) having matrix normal distribution was extensively tested on down-sampled images of the laser additive manufacturing process. In parallel, the well known classifiers are tested using the same sequence of images. The main aim of the tests was to check the robustness of all these classifiers against the class imbalance troubles.

The conclusions are the following:

(I) There is the group of classifiers with positive scores in column 8 of Table 5. They can be considered as more robust against the class imbalance than others classifiers, i.e., the neural network and the naive Bayes one.

(2) The highest overall scores in column 8 of Table 5 was collected by the MMNDCL method.
(3) The above conclusions are based on extensive comparisons, but they are restricted to only one learning-testing sequence of images. These conclusions are – to some extend – confirmed by tests for another sequence of real-life images, namely, by the attempts to classify images of an industrial gas burner (see [10,13]).

Summarizing, although our attempts of selecting a group of classifiers that are robust against the class imbalance seems to be promising, it is highly desirable to verify these findings on other sets of real-life data.

A Appendix

The densities of the matrix normal distribution are defined as follows:

$$f_j(\mathbf{X}) = \frac{1}{c_j} \exp\left[-\frac{1}{2}\,\mathrm{tr}[U_j^{-1}(\mathbf{X}-\mathbf{M}_j)\,V_j^{-1}\,(\mathbf{X}-\mathbf{M}_j)^T]\right], \tag{18}$$

where the normalization constants are given by:

$$c_j \stackrel{def}{=} (2\,\pi)^{0.5\,n\,m}\,\det[U_j]^{0.5\,n}\,\det[V_j]^{0.5\,m}, \tag{19}$$

where $n \times m$ matrices M_j's denote the class means matrices. The covariance structure of MND class densities is as follows

1. $n \times n$ matrix U_j denotes the covariance matrix between rows of an image from j-th class,
2. $m \times m$ matrix V_j stands for the covariance matrix between columns of an image from j-th class.

The above definitions are meaningful only when $\det[U_j] > 0$, $\det[V_j] > 0$.

The equivalent description of MND is the following:

$$\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}_{n\,m}(\mathrm{vec}(\mathbf{M}_j),\,\Sigma_j),\ \text{for } j = 1,\,2,\ldots,J, \tag{20}$$

where $\mathcal{N}_K$ stands for the classic (vector valued) normal distribution with K componentnts. In (20), $\mathrm{vec}(\mathbf{X})$ is the operation of stacking columns of matrix $\mathbf{X}$, while Σ_j is a $n\,m \times n\,m$ covariance matrix of j-th class, which is the Kronecker product (denoted as $\otimes$) of U_j and V_j, i.e.,

$$\Sigma_j \stackrel{def}{=} U_j \otimes V_j, \quad j = 1,\,2,\ldots,J. \tag{21}$$

Formulas (20) and (21) show clearly that MND's form a subclass of all normal distributions. Namely, MND's have the special structure of the covariance matrix given by (21) (see [7]). Thus, in practice, it suffices to estimate two much smaller matrices U_j and V_j instead of a general covariance matrix which is $n\,m \times n\,m$. As the consequence, it suffices to have:

$$N_j \geq \max\left\{\frac{n}{m}, \frac{m}{n}\right\} + 1, \quad j = 1,\,2,\ldots,\,J. \tag{22}$$

(see [6] for the proof).

References

1. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
2. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, Berlin (2013). https://doi.org/10.1007/978-1-4612-0711-5
3. Haykin, S.S.: Neural Networks and Learning Machines. Pearson, Upper Saddle River (2009)
4. Kadziński, M., Słowiński, R.: Parametric evaluation of research units with respect to reference profiles. Decis. Support. Syst. **72**, 33–43 (2015)
5. Krzysko, M., Skorzybut, M., Wolynski, W.: Classifiers for doubly multivariate data. Discuss. Math Probab. Stat., 31 (2011)
6. Manceur, A.M., Dutilleul, P.: Maximum likelihood estimation for the tensor normal distribution: algorithm, minimum sample size, and empirical bias and dispersion. J. Comput. Appl. Math. **239**, 37–49 (2013)
7. Ohlson, M., Ahmad, M.R., Von Rosen, D.: The multi-linear normal distribution: introduction and some basic properties. J. Multivar. Anal. **113**, 37–47 (2013)
8. Rafajłowicz, E.: Data structures for pattern and image recognition with application to quality control. Acta Polytechnica Hungarica, Informatics (accepted for publication)
9. Rafajłowicz, E.: Classifiers for matrix normal images: derivation and testing. In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.) ICAISC 2018. LNCS (LNAI), vol. 10841, pp. 668–679. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91253-0_62
10. Rafajłowicz, E., Rafajłowicz, W.: Image-driven decision making with application to control gas burners. In: Saeed, K., Homenda, W., Chaki, R. (eds.) CISIM 2017. LNCS, vol. 10244, pp. 436–446. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_37
11. Salvatore, G., Matarazzo, B., Slowinski, R.: Rough approximation of a preference relation by dominance relations. Eur. J. Oper. Res. **117**(1), 63–83 (1999)
12. Schein, A.I., Ungar, L.H.: Active learning for logistic regression: an evaluation. Mach. Learn. **68**(3), 235–265 (2007)
13. Skubalska-Rafajłowicz, E.: Sparse random projections of camera images for monitoring of a combustion process in a gas burner. In: Saeed, K., Homenda, W., Chaki, R. (eds.) CISIM 2017. LNCS, vol. 10244, pp. 447–456. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_38
14. Werner, K., Jansson, M., Stoica, P.: On estimation of covariance matrices with Kronecker product structure. IEEE Trans. Signal Process. **56**(2), 478–491 (2008)