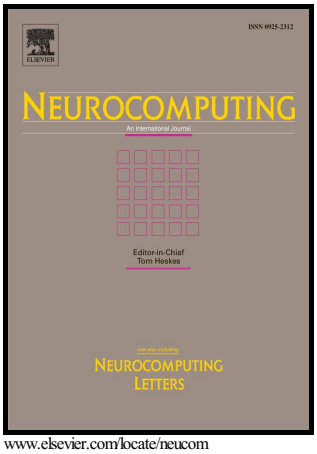


www.elsevier.com/locate/neucom

Properties of the Box-Cox Transformation for Pattern Classification

Manuele Bicego, Sisto Baldo

PII: S0925-2312(16)30977-8
DOI: http://dx.doi.org/10.1016/j.neucom.2016.08.081
Reference: NEUCOM17515

To appear in: *Neurocomputing*

Received date: 11 June 2015
Revised date: 5 July 2016
Accepted date: 27 August 2016

Cite this article as: Manuele Bicego and Sisto Baldo, Properties of the Box-Co Transformation for Pattern Classification, *Neurocomputing* http://dx.doi.org/10.1016/j.neucom.2016.08.081

Properties of the Box-Cox Transformation for Pattern Classification

Manuele Bicego and Sisto Baldo

Dipartimento di Informatica, University of Verona, Strada le grazie 15, 37134 Verona (ITALY)

Abstract

The Box-Cox transformation [1, 2] has been regarded as a parametric preprocessing technique aimed at making the distribution of a set of points approximately Gaussian. Since normality represents an assumption underlying many statistical data analysis tools, such technique has been widely applied in different fields of Computer Science. In this paper we will provide evidence that this technique can be useful also in the case of Pattern Classification, where Gaussianity of datasets is not so critical. By letting the Box-Cox transform work in operational ranges which do not necessarily correspond to an increase in Gaussianity, we will show that class separability can be improved: this is likely due to the *non linear* nature of the Box-Cox transformation, which deforms the space in a nonuniform way. We will also provide some suggestions on criteria that can be used to automatically estimate the best parameter of the Box-Cox transformation in the Pattern Classification context.

Keywords: Box-Cox transformation, non linear mappings, classification, preprocessing

1. Introduction

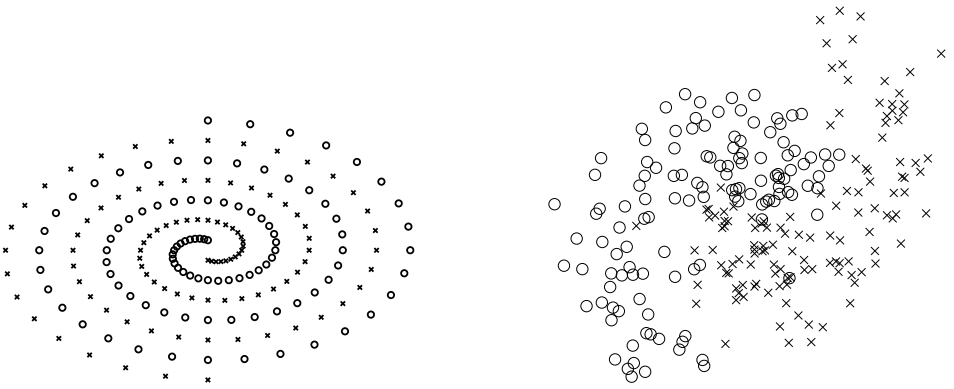
Many important results and techniques in statistical data analysis follow from the assumption that data is normally distributed. In situations where this condition does not hold, one of the possible options [3, 4] is to transform the data in such a way that the distribution is nearer to the normality assumption. The Box-Cox transformation, abstracted from the original context of the linear regression model where it was first introduced in the 60's [1, 2], belongs to this class of approaches and can be regarded as a parametric way to non linearly transform a set of points with the aim of making their distribution approximately Gaussian (see for instance [5]). Since its introduction, such transformation has been widely studied and applied to many different data analysis situations[1], mostly in economics, econometrics, statistics and medicine, but also – to be closer to the pattern recognition area – in medical image segmentation [6], EEG signal analysis [7], geoscience [8], system dynamics modeling and prediction [9], time series forecasting [10, 11] and expression microarray [12]. It is important to notice that in many of the above-referenced applications the final goal was not the design of a classification system: in fact, the usefulness of the Box-Cox transformation as a pre-processing tool for pattern classification has received much less attention, with not so many papers published in the literature – see Section 2 for a detailed list. Clearly, in the specific Pattern Classification context, the increase of Gaussianity in a dataset is no longer the crucial aspect, since Gaussianity

[1] To give an idea, GoogleScholar reports more than 8000 citations to the original paper of Box and Cox [1].

of the dataset does not imply Gaussianity of the classes – see for example the datasets in figure 1 –, the latter characteristic being the assumption of different standard classifiers, like the nearest mean classifier.

In this paper, we provide some evidence that the Box-Cox transformation may be useful also in the Pattern Classification context, typically operating in parameter ranges which can be very far from those optimal for Gaussianity. This success is likely due to the non linear nature of the Box-Cox transformation, which deforms the problem space in a nonuniform way, allowing to highlight useful structures or making the data more suitable for a given classifier (e.g. by increasing the class separability) – see the example in fig. 3. To investigate these aspects, we start from a large scope analysis, involving a set of different datasets and classifiers, and we empirically link the behaviour of accuracies while varying the Box-Cox parameter λ with three criteria (Gaussianity, Gaussianity of every class, Class separability), showing that accuracies curves are linked more to class separability than to Gaussianity. In the second part of the paper, we also derive some practical and simple criteria to select good and effective values of the parameter λ, exploiting the characteristics of the specific scenario – i.e. the labels.

The remainder of the paper is organized as follows: in Section 2 we briefly review the basics of the Box-Cox Transformation; subsequently, in Section 3, we empirically investigate its properties in the Pattern Classification domain. The analysis on the automatic estimation of the best parameter is then reported in section 4. Finally, in Section 5 conclusions are drawn.



(a) (b)

Figure 1: Two "nearly Gaussian" datasets whose classes are highly not Gaussian: (a) the spirals dataset; (b) the bananas dataset

2. The Box-Cox Transformation

In our paper we focus on the basic formulation of the Box-Cox transform as given in the original Box-Cox paper [1], which transforms a given variable x into $x^{(\lambda)}$ via the following equation:

$$x^{(\lambda)} = \begin{cases} \dfrac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases} \tag{1}$$

the transformation being defined for $x > 0$. Many other slightly different versions or formulations have appeared over the years [2], which are nevertheless all minor variations of the same idea. In some versions the parameter was restricted to the range $(0, 1]$, in some others a shift was included (in order to deal also with negative values); further approaches also apply a linear scaling based on the geometric mean (which represents the inverse of the Jacobian of the transformation); finally some others simply consider a

power transformation x^λ, which makes impossible to give a useful definition of the transformation for $\lambda = 0$. In all the cases, nevertheless, the main characteristic is to realize a non linear mapping of the points through a power operation.

In order to explain the effect of the Box-Cox transformation to the situations considered in this paper, we need to specify how our data is preprocessed. As a starting point, we perform a classical *range* standardization [13] (also called, in other scenarios, domain standardization), reducing all features to a range of length 1 (in our case we reduce all features between 1 and 2). Data standardization is a classical preprocessing operation in pattern recognition, which allows to remove scale factors: in this sense, it can be very beneficial for those classifiers which rely on geometric concepts (such as nearest neighbors or kernel machines). Moreover, in our case this standardization allows to avoid the somewhat extreme behaviour of the Box-Cox transformation for x approaching 0. The same standardization is applied again after transforming the data with the Box-Cox transformation. This way, we ensure that the range does not depend on the parameter λ and we have a uniform way of comparing the original data with the transformed ones.

The effect of this operation is shown in the figure 2, where the graph of this re-normalized Box-Cox transformation is represented for several values of the parameter (axes are translated by putting the point $(1, 1)$ into the origin). For $\lambda = 1$ the transformation leaves the data unchanged. For $\lambda > 1$, the graph of the transformation is convex: its main effect is that of decreasing the relative distances of points near the minimum value 1, increasing at the

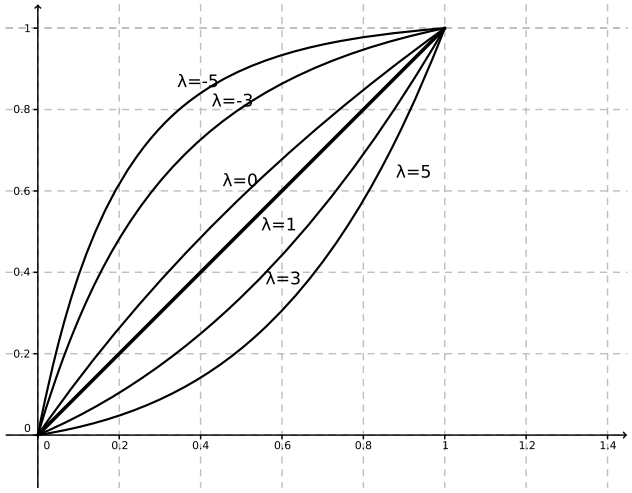


Figure 2: Graph of the 1-d Box-Cox transformation for different values of Lambda.

same time the distances of points near the maximum 2[2]. The behaviour of the transformation near the extrema of the range is more and more emphasized as λ grows away from 1. For $\lambda < 1$, the opposite behaviour is observed, the function being concave. It is important to stress that the function is increasing for every choice of λ: the Box-Cox transformation does not change data ordering.

Whereas the Box-Cox transform has been widely used in the literature, few approaches explore its usefulness as a preprocessing tool for pattern classification: such studies can be divided in systems related to specific applications – like radar signal classification [5, 14], sound indexing [15] or handwritten character recognition [16], just to cite a few – or studies related to the impact of the Box-Cox transformation on some specific classification tech-

[2]In the middle range relative distances are more or less preserved

niques – like knn in [5], dissimilarity-based classifiers in [17] or kernels for HMM-based generative embeddings in [18, 19]. In almost all of the above studies – except for [18, 19] – the main motivation to justify the use of the Box-Cox transformation in the pattern classification scenario is the general "Driving-to-Gaussianity" characteristic (i.e. making the dataset more Gaussian). As reported in the introduction, this motivation does not explain the success of the technique. We will show that the transform may increase dramatically the separability between classes, especially when operating in parameter ranges which can be very far from the range optimal for Gaussianity.

2.1. A measure of Gaussianity

The Box-Cox transformation is a parametric operation, whose effectiveness strongly depends on the chosen λ. Many approaches have been studied in the past to estimate such parameter, starting from historical works [1, 20, 21] up to more recent ones [9]. In their original paper [1], Box and Cox provided a criterion (called "Maximum Likelihood" – ML) employable to measure the Gaussianity of the transformed dataset; the λ maximizing such criterion represents the λ making the dataset as much Gaussian as possible. In this sense, we can interpret the ML criterion, evaluated for a given λ, as a measure of the "Gaussianity" of the dataset after applying the Box-Cox Transformation with parameter λ.

More precisely, given a set of points $\mathcal{X} = \{x_1, ..., x_N\}$, the ML criterion for a given λ can be computed as:

$$L_{max}(\lambda, \mathcal{X}) = -\frac{1}{2} N \log \hat{\sigma}^2(\lambda) + (\lambda - 1) \sum_{i=1}^{N} \log x_i \tag{2}$$

where $\hat{\sigma}^2(\lambda)$ is the standard estimate of the variance of points $\mathcal{X}^{(\lambda)} = \{x_1^{(\lambda)}, ..., x_N^{(\lambda)}\}$ after the application of the Box-Cox transform with parameter λ.

Let us call $\lambda_{MaxG,\mathcal{X}}$ the λ maximizing this criterion: $\lambda_{MaxG,\mathcal{X}}$ represents the parameter which makes the dataset as much Gaussian as possible. We will make use of $\lambda_{MaxG,\mathcal{X}}$ and the ML criterion in our experimental evaluation.

2.2. The multivariate Box-Cox transformation

The Box-Cox transformation is defined on one-dimensional data. The generalization to a d-dimensional set of points typically consists in performing d 1-dimensional transformations, one for each direction of the problem space. The transformation is then characterized by the vector parameter $\Lambda = [\lambda_1, ..., \lambda_d]$. Even if in principle such generalization may induce some complex aspects – especially for what concerns the inference of Λ, see [22] –, this way to extend 1-dimensional preprocessing tools is widely accepted and used in many Pattern Recognition cases [5, 17, 16, 19, 18].

Here we define two versions of the multivariate Box-Cox transformation, which we call – using the notation typically employed with multivariate Gaussian distributions – *Spherical* and *Diagonal* Box-Cox transformations. In the first of the two cases, λ is the same for every direction, namely $\lambda_j = \lambda$ $\forall j \in \{1..d\}$ (the transform is parametrized only by a single λ), while in the second we have a different λ for every direction, and the transformation is parametrized by the full vector $\Lambda = [\lambda_1, ..., \lambda_d]$. Even if the diagonal Box-Cox is in principle more flexible, allowing different transformations in different directions, there can be cases where a Spherical transform is more appropriate. Actually, especially for high dimensional data, it can be more difficult to properly estimate all the parameters of the diagonal transform (one param-

eter for every direction). This is particularly true when the transformation parameters are estimated on a hold-out training dataset, and applied to the testing set via a classifier. In case of high dimensional data, the risk of over-training becomes very high. In such cases, the simpler Spherical version (with just one parameter to be estimated) can be a viable and more robust solution – the empirical evaluation of section 4 confirms this aspect. Moreover, the spherical version represents the usual choice in most of the studies related to pattern classification and also allows a better evaluation of performances when varying this parameter (see for example [5, 18, 19]).

Also in these multivariate cases, it may be interesting to compute the parameters Λ which make the transformed $\mathcal{X}$ as much Gaussian as possible. By generalizing the definition of $\lambda_{MaxG,\mathcal{X}}$, we can provide a definition of $\lambda^{(S)}_{MaxG,\mathcal{X}}$ and $\Lambda^{(D)}_{MaxG,\mathcal{X}}$, for the Spherical and the Diagonal versions, respectively. In particular, the former is defined as the value which maximizes the sum of the criteria evaluated along the different directions:

$$\lambda^{(S)}_{MaxG,\mathcal{X}} = \arg\max_{\lambda} \sum_{k=1}^{d} L_{max}(\lambda, \mathcal{X}_{|_k}) \tag{3}$$

($\mathcal{X}_{|_k}$ represents the k-th direction of the dataset $\mathcal{X}$), where the latter is defined as the vector containing the set of λs maximizing the criterion $L_{max}(\lambda, \mathcal{X})$ along the different directions:

$$\Lambda^{(D)}_{MaxG,\mathcal{X}} = \begin{bmatrix} \arg\max_{\lambda_1} L_{max}(\lambda_1, \mathcal{X}_{|_1}) \\ \dots \\ \arg\max_{\lambda_k} L_{max}(\lambda_k, \mathcal{X}_{|_k}) \\ \dots \\ \arg\max_{\lambda_d} L_{max}(\lambda_d, \mathcal{X}_{|_d}) \end{bmatrix} \tag{4}$$

3. Experimental evaluation part I: understanding the Box-Cox transform

In this section, starting from an extensive analysis of different datasets, different classifiers, and different parameter configurations, we analyse the behaviour of the Box-Cox transform in the Pattern Classification scenario, trying to link accuracy improvements to different criteria relative to different aspects.

3.1. Experimental details

We employed 15 different standard datasets, whose details are reported in Table 1: we tried to select datasets with different characteristics (few or many objects, few or many features, few or many classes). All datasets have been preprocessed as explained in the previous section (namely with a range standardization between 1 and 2). In some specific cases, we also applied other standard pre-processings: in the microarray experiment (brain), gene selection has been applied (with a standard variance filter) in order to reduce the number of genes to 100; in the kimia images dataset (kimia), binary images have been re-sampled (64*64), vectorized and preprocessed with PCA (retaining 158 components, so to cover 99% of the variance). We also considered in the evaluation two real world challenging non standard problems: the classification of volcano seismic events (volcano – [23]) and a behavioural biometrics application using energy load profiles (energy – [24]). In the former case, the seismic signals, coming from the Nevado del Ruiz Dataset in Colombia, have been represented via averaged spectrograms (see [23]): the goal was to discriminate between different seismic events; in the second case,

the main goal was to identify a person by analysing the way he/she employs electrical energy in his/her house (the so-called load profiles). Profiles were extracted from internet[3]; we selected 50 users, using one month of recordings (one registration every hour), and the raw signal as signature (for more details, please refer to [24]).

The classification accuracy, to be computed before and after applying the Box-Cox transformation, has been assessed with different classifiers, described in Table 2, as implemented in the Matlab library PRTOOLS[4]. All the accuracies have been computed with the Averaged Hold Out Cross Validation protocol (30 repetitions), namely by repeating 30 random 50%-50% splitting of the dataset (half for training and half for testing) and averaging the obtained accuracies.

3.2. Classification Results

We start our evaluation by investigating the usefulness of the Box-Cox transformation by varying the parameter λ. We concentrate our analysis on the spherical version, which permits an exhaustive analysis with respect to the parameter. In particular, we compute, for every classifier/dataset/fold of the cross validation, the classification accuracy after applying the Box-Cox transformation, with λ varying in the range [-5, 5] (with step 0.1) – we denote it as $A^c_{\mathcal{X}}(\lambda, f)$ (accuracy of classifier c on the f-th fold of dataset $\mathcal{X}$ preprocessed with the Box-Cox transform with parameter λ). Then, for every $c, \mathcal{X}$ and f, we select the best result among the different λ (which we

[3] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.`

[4] A Matlab library for Pattern Recognition, see `prtools.org`.

Name	Description	Source	O	U	F
bananas	banana-shaped data	Synthetic	600	2	2
ecoli	Protein Localization sites	UCI-ML	272	3	7
diabetes	Diabetes dataset	UCI-ML	768	2	8
breast	Wisconsin breast cancer	UCI-ML	683	2	9
glass	Glass Identification	UCI-ML	214	4	9
wine	Wine Recognition	UCI-ML	178	3	13
brain	Brain Cancer Microarray	[25, 26]	90	5	100
biomed	Biomedical Data	StatLib	194	2	5
liver	Liver disorder data	UCI-ML	345	2	6
sonar	Sonar signals	UCI-ML	208	2	60
spirals	Spiral-shaped data	Synthetic	194	2	2
kimia	Kimia binary image datasets	[27]	216	18	158
fourier	Fourier features for digits	[28]	2000	10	76
volcano	Seismic signals	[23]	1233	5	65
energy	Energy Biometrics	[24]	1500	50	24

Table 1: Details of the employed datasets: **O** represents the number of objects, **U** the number of classes, **F** the number of features.

call $\lambda^c_{MaxA,\mathcal{X}}(f)$), computing the relative difference between this value and the accuracy obtained without applying the Box-Cox transform:

$$\text{Improvement } I^c_{\mathcal{X}}(f) = \frac{A^c_{\mathcal{X}}(\lambda^c_{MaxA,\mathcal{X}}(f), f) - A^c_{\mathcal{X}}(1, f)}{\min\{A^c_{\mathcal{X}}(\lambda^c_{MaxA,\mathcal{X}}(f), f), A^c_{\mathcal{X}}(1, f)\}} \tag{5}$$

where $A^c_{\mathcal{X}}(1, f)$ is the accuracy obtained when the Box-Cox transformation is not applied (i.e. with $\lambda = 1$). This represents the largest possible relative

Name	Description
nmc	Standard Nearest Mean Classifiers
ldc	Linear Bayes classifier assuming normal densities with equal covariance matrices (Covariance matrices are loosely regularized to avoid estimation errors)
qdc	Quadratic Bayes classifier assuming normal densities (Covariance matrices are loosely regularized to avoid estimation errors)
udc	Quadratic Bayes classifier assuming normal densities with uncorrelated features
1-nn	Standard Nearest Neighbor rule with euclidean distance
parzenc	Non parametric Bayes classifier where class densities are estimated with Parzen Windows (smoothing parameter set via cross validation on the training set)
loglc	Linear classifier by maximizing the likelihood criterion using the logistic function
linSVM	Support vector machine with linear kernel and regularization parameter set to 1

Table 2: Details of the employed classifiers: for all the details see [29] and the implementation description as reported in PRTOOLS.

improvement obtained with respect to the unprocessed case (i.e. when the Box-Cox transformation is not applied): clearly, a positive value indicates the potential usefulness of the Box-Cox transformation when using classifier c on $\mathcal{X}$. All these potential improvement values (averaged over the different cross validation folds), for all datasets and classifiers, are reported in Table 3: bold values indicate relative improvements which are statistically significant according to a standard t-test with a significance level of 5%.

From the table, we can observe that all relative improvements are positive, indicating that there is always the possibility of increasing the performances when applying the Box-Cox transformation: in some cases, the improvement

Dataset	nmc	ldc	udc	qdc	1-nn	parzenc	loglc	linSVM	*Avg*
bananas	**2.98%**	0.63%	**3.46%**	**0.95%**	0.11%	0.25%	0.55%	0.83%	*1.22%*
ecoli	**1.15%**	**0.99%**	**1.13%**	0.58%	**1.27%**	0.60%	0.90%	0.66%	*0.91%*
diabetes	**3.31%**	**1.10%**	**1.60%**	**2.26%**	**3.01%**	**3.13%**	**1.18%**	**1.28%**	*2.11%*
breast	**1.11%**	**1.58%**	**1.04%**	**2.00%**	**1.08%**	**1.52%**	**0.85%**	**0.82%**	*1.25%*
glass	**15.63%**	**5.77%**	4.60%	**6.52%**	**4.76%**	**5.84%**	**11.65%**	**6.95%**	*7.71%*
wine	**1.13%**	**1.38%**	**1.38%**	**2.55%**	**2.93%**	**2.19%**	**1.66%**	**1.93%**	*1.89%*
brain1	1.91%	0.33%	**6.06%**	0.00%	**3.14%**	**3.75%**	**7.07%**	**1.96%**	*3.03%*
biomed	**4.77%**	**2.78%**	0.97%	**1.32%**	**4.65%**	**3.41%**	1.35%	**2.88%**	*2.77%*
liver	**10.62%**	**10.86%**	**15.31%**	**14.21%**	**6.90%**	**11.42%**	**8.50%**	**10.76%**	*11.07%*
sonar	**7.20%**	**5.59%**	**10.02%**	**13.82%**	**3.69%**	**2.90%**	**8.25%**	**5.72%**	*7.15%*
spirals	**13.81%**	**12.91%**	**14.79%**	**16.56%**	2.16%	**15.58%**	**12.36%**	**15.54%**	*12.97%*
kimia	**1.90%**	3.25%	2.83%	7.10%	**15.00%**	**19.14%**	**5.64%**	**1.94%**	*7.10%*
fourier	**1.63%**	**1.26%**	**0.83%**	**1.28%**	**1.15%**	**1.21%**	**4.02%**	**1.63%**	*1.63%*
volcano	**4.92%**	0.87%	**2.45%**	**8.28%**	**1.52%**	0.87%	**1.81%**	**1.16%**	*2.74%*
energy	**0.75%**	**6.13%**	0.10%	1.84%	0.11%	**3.33%**	**2.14%**	**2.13%**	*2.07%*
Average	*4.85%*	*3.70%*	*4.44%*	*5.28%*	*3.43%*	*5.01%*	*4.53%*	*3.75%*	*4.37%*

Table 3: Classification results. In bold differences which are statistically significant (based on a t-test, significance level 0.05).

is really significant (for example, a 20% of improvement is obtained in the kimia dataset for the Parzen classifier). Remarkably, in most of the cases (96 out of 120), the improvements are also statistically significant. The different classifiers benefit almost equally from the Box-Cox transformation, whereas for datasets the improvements are more variable, with some datasets (like glass, liver, sonar, spirals and kimia) gaining more than others; however, apparently, there is no obvious correlation between improvements and coarse grained characteristics of the datasets (number of features, number of classes,

number of objects).

3.3. Gaussianity is not the point

As shown in the previous Section, and as confirmed by many studies [5, 14, 15, 16, 17, 18, 19], the Box-Cox transform can be very useful in the Pattern Classification scenario. However, we are convinced that this usefulness is not strictly linked to the original spirit of such transform (i.e. the increase in the Gaussianity), and can be obtained when operating in parameter ranges far from those optimal for Gaussianity. As an example, let us consider the dataset shown in fig. 3(a), where a Italy-shaped class (crosses) should be separated from the French island Corsica (circles). It is clear that the problem is not linearly separable. In fig. 3(b) we show the same dataset transformed by applying the spherical Box-Cox transformation using $\lambda^{(S)}_{MaxG,\mathcal{X}}$, which represents the best parameter for Gaussianity of the transformed space (in this case, $\lambda^{(S)}_{MaxG,\mathcal{X}} = 2$). It is visually evident that the transform in this operational range does not help that much. In fig. 3(c) we show the same dataset after applying the spherical Box-Cox transformation with $\lambda = 5$. Now the problem is almost linearly separable, which is a better and more desirable situation from the Pattern Recognition perspective: such situation is obtained by using a value of λ which is very far from the one leading to the maximum Gaussianity.

From Figure 3(c) we can get the effect of a Box-Cox transformation: the problem space is deformed in a nonuniform way (higher value points were magnified more than lower value points), thus resulting in a non linear normalization of the feature space. Such non linear normalization can be evinced also by looking at figure 3(d), where we represent, for the Box-

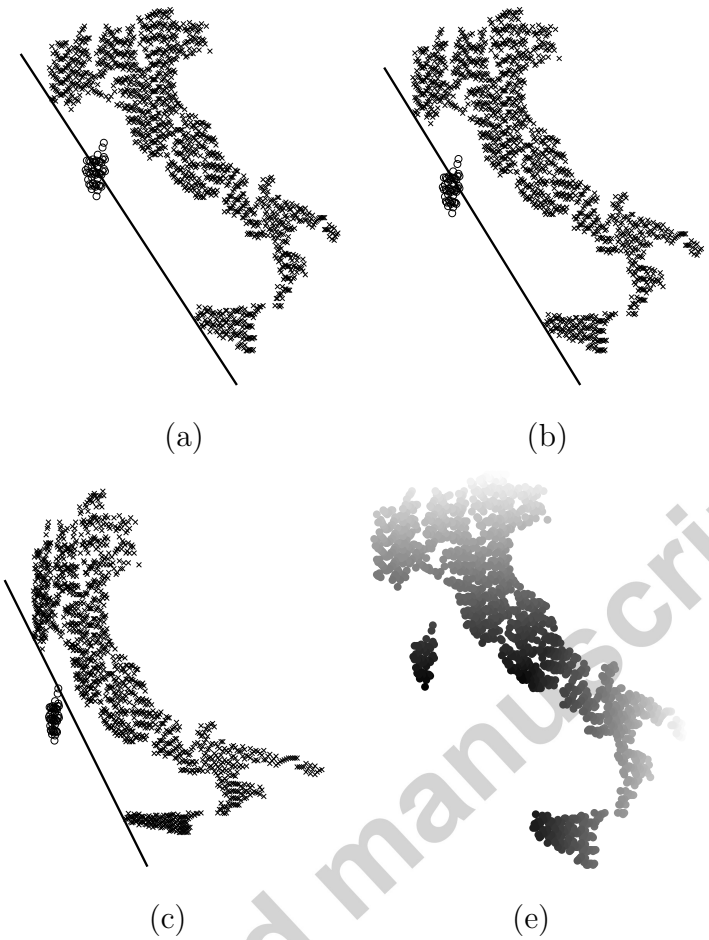


Figure 3: (a) Original data; (b) Data after applying a Spherical Box-Cox Transformation with $\lambda = 2$, which represents the best parameter for Gaussianity of the transformed space. (c) Data after applying a Spherical Box-Cox Transformation with $\lambda = 5$. Please note that Corsica becomes linearly separable from Italy after applying the Spherical Box-Cox transformation with a proper λ. (d) The deformation map for Spherical Box-Cox and $\lambda = 5$.

Cox Transformation with $\lambda = 5$, the deformation map for the dataset (the lighter the color the lower the deformation), obtained by determining the

local Lipschitz constant (the operator norm of the gradient) at every point: this constant indicates the maximum local deformation of the distance at a point after the transformation[5]. It is evident from the figure how different parts of the dataset are deformed in different ways. We are convinced that this non linear nature represents the most important characteristic of the Box-Cox transform in the pattern classification context: by deforming the problem space in a nonuniform way, it is possible to discover useful structures or to make the data more suitable for a given classifier.

As suggested above, this appealing behaviour can happen in operational ranges which are very far from those optimal for Gaussianity. In order to quantitatively investigate this aspect, we come back to our experiments, and, for every dataset $\mathcal{X}$, fold f and classifier c, we measure the absolute difference between the value $\lambda^{c}_{MaxA,\mathcal{X}}(f)$ (the λ giving the maximum accuracy for the classifier c) and the value $\lambda^{(S)}_{MaxG,\mathcal{X}}(f)$ (the best λ in terms of Gaussianity). Results, averaged over folds and classifiers, are reported in table 4, for the different datasets: in many cases the difference is very large, confirming that values of λ giving the best classification behaviour can be very far from those granting optimal dataset Gaussianity.

[5]The local Lipschitz constant is defined as the maximum modulus of the eigenvalues of the (diagonal) Jacobian matrix evaluated in that point:

$$\max\left\{\left|\frac{\partial x^{(\lambda)}}{\partial x}\right|, \left|\frac{\partial y^{(\lambda)}}{\partial y}\right|\right\}. \tag{6}$$

Dataset	Averaged Absolute Difference
bananas	1.27(1.54)
ecoli	2.04(1.16)
diabetes	2.32(1.50)
breast	1.55(1.96)
glass	3.79(2.59)
wine	3.21(1.94)
brain	2.66(2.93)
biomed	1.77(1.37)
liver	1.69(1.13)
sonar	1.68(1.13)
spirals	4.61(2.33)
kimia	3.36(2.93)
fourier	3.74(1.31)
volcano	1.84(2.17)
energy	2.27(3.40)

Table 4: Absolute differences between the λ leading to the Optimal Accuracy and the λ leading to maximum Gaussianity, averaged over classifiers and folds – between brackets the standard deviation.

3.4. Understanding the success of the Box-Cox Transformation

In order to better understand why the Box-Cox transformation is useful, we tried to empirically link the behaviour of the accuracy of a given classifier after applying the Box-Cox transformation with a given parameter λ to some numerical measures or criteria which depend on the dataset and on

the parameter λ. In particular, we analysed three criteria, briefly summarized in the following list: the first criterion derives directly from eq. (2) and measures Gaussianity, the others two take into account the labels.

Given a dataset $\mathcal{X}$ involving a U class problem with N objects in a d dimensional space, the criteria are defined as follows:

- (**ML**) – Maximum Likelihood criterion: as explained before, for a fixed λ, the maximized likelihood defined in equation (2) may be intended as a measure of Gaussianity of the data after applying the Box-Cox transformation with parameter λ. This represents our first criterion, defined, for a d-dimensional dataset $\mathcal{X}$ as:

$$J_{ML}(\lambda, \mathcal{X}) = \frac{1}{d} \sum_{k=1}^{d} L_{max}(\lambda, \mathcal{X}_{|_k}) \tag{7}$$

  where $L_{max}(\lambda, \mathcal{X}_{|_k})$ represents the criterion in (2) computed along the k-th direction of $\mathcal{X}$, as defined in eq. (3).

- (**PCML**) – Per Class Maximum Likelihood criterion: this represents an extension of the ML criterion which takes into account the labels. In particular the idea is to measure the ML criterion class by class: Gaussianity of every class – which is not implied by the Gaussianity of the whole dataset – may be now a characteristic that can help those classifiers based on normality assumptions. More formally, the PCML criterion is defined as

$$J_{PCML}(\lambda, \mathcal{X}) = \sum_{\ell=1}^{U} J_{ML}(\lambda, \mathcal{X}_\ell) \tag{8}$$

  where $\mathcal{X}_\ell$ is the ℓ-th class of $\mathcal{X}$

$$\mathcal{X}_\ell = \{x \in \mathcal{X} | \text{ label}(x) = \ell\} \tag{9}$$

- (**Fisher**) – Fisher criterion: this is the classical Fisher Criterion [4] for measuring class separability in the transformed space; in particular we use the first of the versions introduced in chapter 10.2 of Fukunaga's book [4]; calling $M_\ell^{(\lambda)}$ and $\Sigma_\ell^{(\lambda)}$ the mean and the scatter matrix of the points belonging to the ℓ-th class after applying a Box-Cox transformation with parameter λ, P_ℓ the a priori probability of class ℓ, and $M_0^{(\lambda)}$ the total mean of the transformed points, we can define the within-class scatter matrix $S_w^{(\lambda)}$ and the between-class scatter matrix $S_b^{(\lambda)}$ as:

$$S_w^{(\lambda)} = \sum_\ell P_\ell \Sigma_\ell \tag{10}$$

$$S_b^{(\lambda)} = \sum_\ell P_\ell (M_\ell^{(\lambda)} - M_0^{(\lambda)})(M_\ell^{(\lambda)} - M_0^{(\lambda)})^T \tag{11}$$

which lead to the definition of the Fisher criterion

$$J_{FIS}(\lambda, \mathcal{X}) = tr((S_w^{(\lambda)})^{-1} S_b^{(\lambda)}) \tag{12}$$

where tr represents the trace operator.

To empirically link the accuracy to the criteria described above we adopted a visual approach as well as a quantitative approach. In general, given a dataset $\mathcal{X}$, a classifier c, and a cross validation fold f, the main goal is to compare, while varying λ, the accuracy $A_{\mathcal{X}}^c(\lambda, f)$ with the three criteria. To realize the comparison, for all folds/datasets, we computed the three criteria (7), (8) and (12) for λ in the range [-5,5] (step 0.1).

3.4.1. Visual Comparison

The main goal here is to perform a visual comparison between the four functions of λ (the accuracy $A_{\mathcal{X}}^c(\lambda, f)$ and the three criteria (7), (8) and

(12)). In order to have a proper visualization, we aggregate the accuracy values (among folds and classifiers) and the criteria values (among folds), obtaining for a given dataset $\mathcal{X}$ four functions of λ. The resulting curves are plotted in Fig. 4, 5 and 6, for different datasets; for visualization purposes they are all normalized between 0 and 1 (this operation does not affect the comparison, since we are interested in the behaviour and not in the values). A vertical line also denotes the value where the transformation has not been applied ($\lambda = 1$).

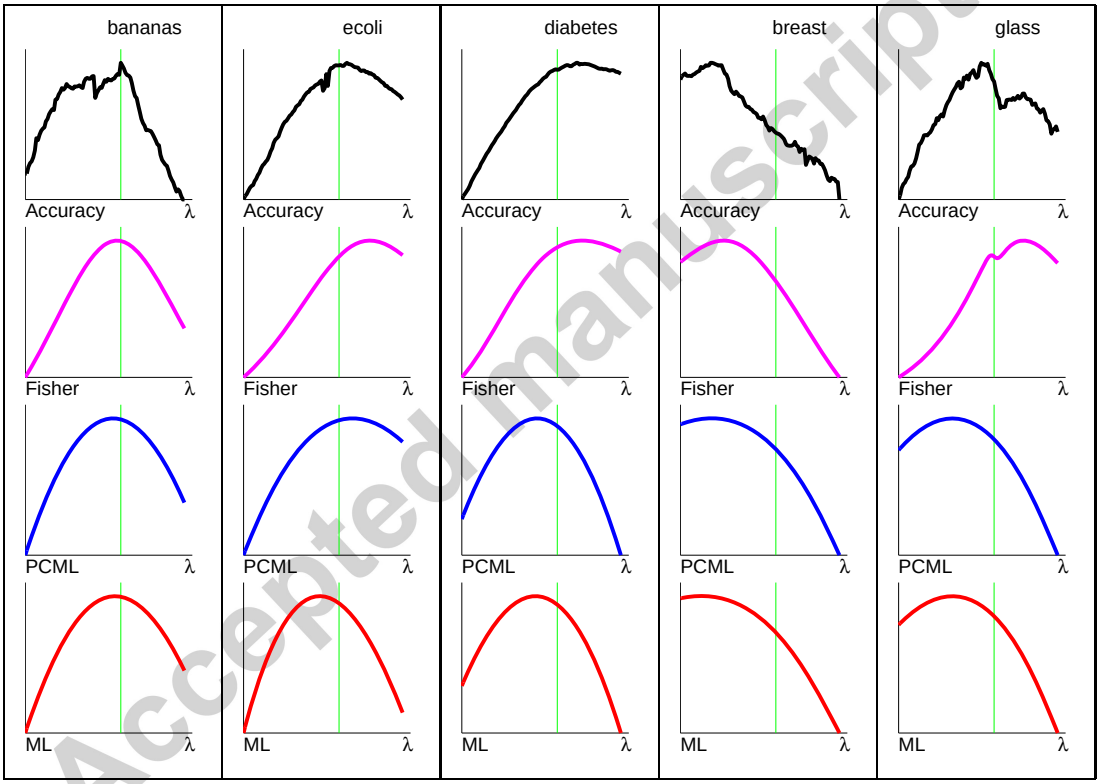


Figure 4: Comparison between criteria and accuracy (averaged over classifiers) – Spherical Box-Cox, part I.

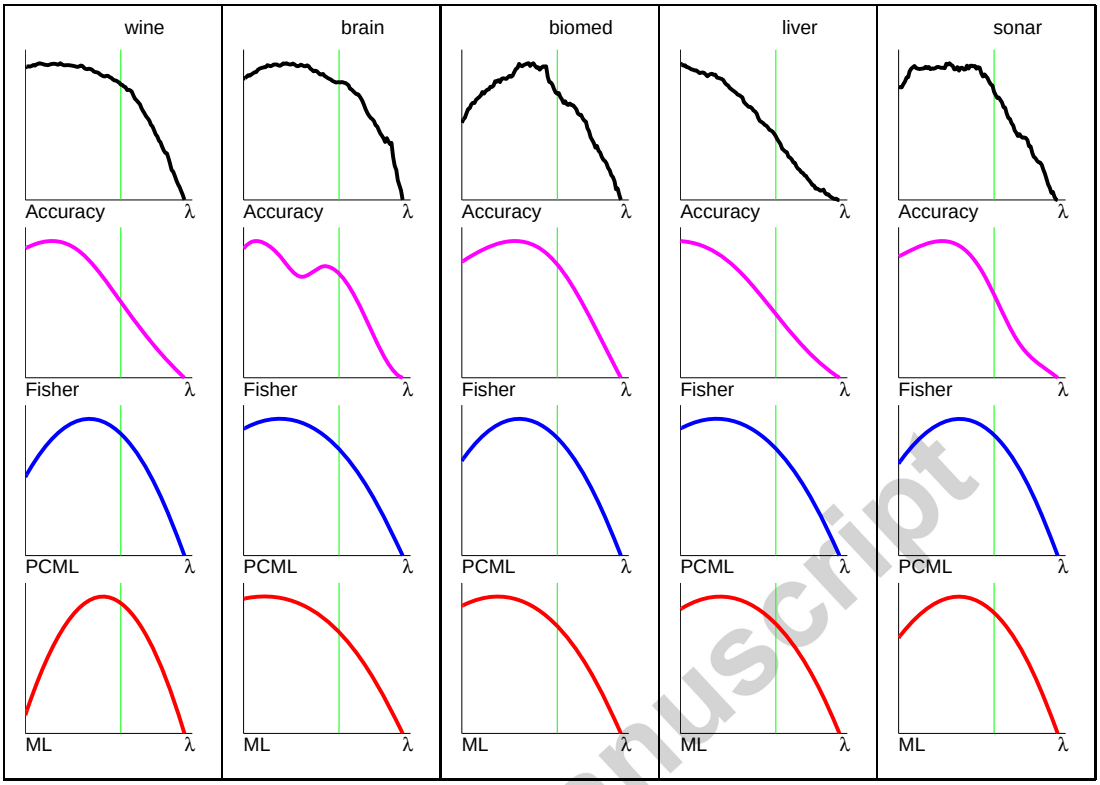


Figure 5: Comparison between criteria and accuracy (averaged over classifiers) – Spherical Box-Cox, part II.

Let us first consider the spirals dataset (Fig. 6, first column), which represents the most interesting case supporting our view: such dataset, as can be seen from Fig. 1(a), is already approximately Gaussian, with every class already approximately Gaussian: actually, both the ML and the PCML criteria, which measure the Gaussianity of the dataset and the classes, respectively, show a clear maximum around $\lambda = 1$ (i.e. no transform is needed, green vertical line). Of course, nevertheless, this dataset is not in a good position for being classified by all classifiers, and good improvements can

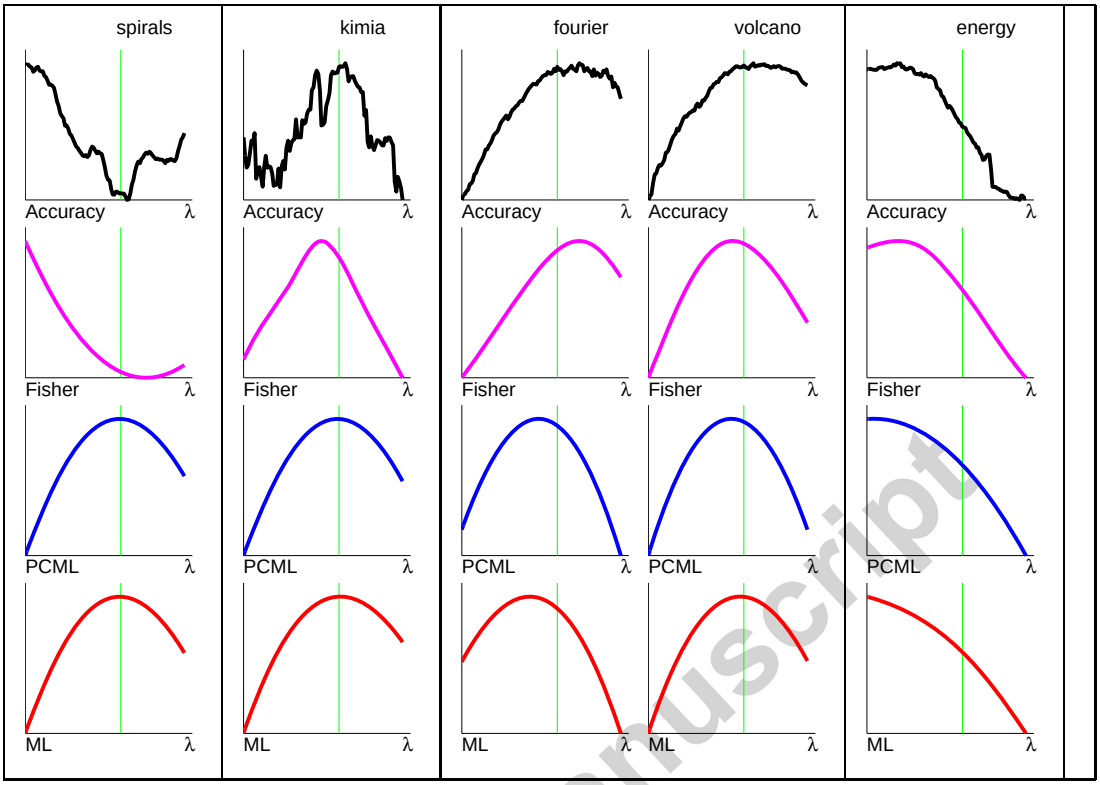


Figure 6: Comparison between criteria and accuracy (averaged over classifiers) – Spherical Box-Cox, part III.

be obtained when applying the Box-Cox Transformation with λ far from the Gaussianity: this is clearly shown by the accuracy curve, which has a maximum for $\lambda = -5$. Very interestingly, this curve is almost perfectly approximated by the Fisher criterion. Looking at this extreme dataset, we can note that i) the Box-Cox transformation improves the performances when working *as far as possible* from the values of λ corresponding to the maximum Gaussianity of the transformed dataset; and ii) it works because it increases the separability between classes, as measured by the Fisher Crite-

rion. In the other plots, such an extreme behaviour is no longer evident, but we notice a generalized coherence between the behaviour of the accuracy and those of the criteria: for most of the datasets there is *at least one of the criteria* (depending on the case) which is properly approximating the accuracy curve.

3.4.2. Quantitative Comparison

In order to quantitatively assess how much the accuracy is linked to every criterion, we compute the L_∞ (or uniform) distance between the accuracy and the three criteria (as functions of λ). Indeed, the L_∞ norm gives a sensitive measure of the difference between functions[6], in that it emphasizes even localized differences. Clearly, the lower this value the better the similarity between the criterion and the accuracy. Since our functions are normalized between zero and one, the maximum possible distance is 1. Such distances, for the different datasets, are displayed in table 5. From the table, it is evident that the best approximation of the accuracy curve is hardly given by the ML criterion (only one case), even if it is not so far for several datasets (thus explaining the success of the Box-Cox transformation even in its Gaussianity-improving flavour). The best criterion is Fisher, which measures class separability and can be very far from Gaussianity as confirmed in the extreme case of the Spirals dataset.

[6]It is defined, given two functions $g_1(\lambda)$ and $g_2(\lambda)$, by

$$L_\infty(g_1, g_2) = \max_\lambda |g_1(\lambda) - g_2(\lambda)| \tag{13}$$

Dataset	L_∞ ML	L_∞ PCML	L_∞ Fisher
bananas	0.511	0.452	0.406
ecoli	0.583	0.139	0.221
diabetes	0.921	0.921	0.076
breast	0.277	0.316	0.263
glass	0.790	0.768	0.402
wine	0.836	0.394	0.349
brain	0.253	0.243	0.350
biomed	0.367	0.128	0.283
liver	0.377	0.356	0.054
sonar	0.187	0.184	0.264
spirals	1.000	1.000	0.395
kimia	0.668	0.564	0.485
fourier	0.755	0.749	0.156
volcano	0.310	0.650	0.436
energy	0.247	0.295	0.232
Avg	*0.539*	*0.477*	*0.292*

Table 5: L_∞ norm between accuracy and different criteria, for the different datasets.

4. Experimental evaluation part II: automatic estimation of $\boldsymbol{\lambda}$

The main goal of this section is to provide some feasible and practical hints on how to set the λ parameter, a crucial problem in the actual application of the Box-Cox Transformation. Typically, in the literature, such a parameter is i) set by hand, or ii) found by an exhaustive search, or iii) automatically

estimated via numerical optimization of a criterion – many criteria have been studied, starting from the historical works [1, 20, 21] up to more recent ones [9]. In the peculiar pattern classification context, we can also resort to all the machinery developed in the community to solve the parameter selection issue – the most widespread approach being cross validation, a standard but computationally demanding procedure.

Here we analyse the behaviour of 4 automatic parameter estimation criteria, to be optimized to automatically find a reasonable parameter λ to be employed. Given the analyses shown in the previous section, it seems reasonable to start from the criteria presented there (ML, PCML and Fisher). Moreover, we also investigated the suitability of another criterion, called the *1-NN hypothesis-margin* [30, 31]. This criterion evaluates the goodness of a given space by measuring the (hypothesis) margin of the 1-Nearest Neighbor classifier. The margin represents another measure of separability of the classes, widely used in Pattern Recognition (e.g. in Support Vector Machines [32]); for the 1-NN case, it is possible to compute a simple and effective version of the margin called hypothesis margin [30, 31], which measures the distance between the decision boundary induced by the classifier and the closest decision boundary that assigns an alternative label to the given instance (on the contrary, the standard sample margin used in the SVM classifier evaluates the distance between the instances and the decision boundary induced by the classifier). For the 1-NN rule, given a set of points $\mathcal{X}$, the hypothesis margin can be computed as

$$\text{1NN-MM}(\mathcal{X}) = \sum_{x \in \mathcal{X}} \frac{1}{2} \left(\| \, x - \mathbf{nearmiss}(x) \, \| - \| \, x - \mathbf{nearhit}(x) \, \| \right) \quad (14)$$

where $\mathbf{nearhit}(x)$ and $\mathbf{nearmiss}(x)$ denote the nearest point in $\mathcal{X}$ with the

same and different label, respectively. For more details, interested readers can refer to [30], where such concept has been used for feature selection. We call this criterion *1NN-MM* (1NN-MaxMargin).

We already pointed out that the ML criterion has been widely used as parameter estimation criterion – also with some variants; on the contrary, the use of PCML, Fisher and 1NN-MM is novel in this context. We evaluated the four criteria by performing a set of experiments where, for every dataset $\mathcal{X}$ and fold f, the best λ is automatically estimated from the training set $\mathcal{X}_f^{train}$ through the optimization[7] of one of the criteria. We have different Box-Cox variants (spherical and diagonal) and different criteria (ML, PCML, Fisher, 1NN-MM), thus providing 8 different possibilities to be investigated. For each of them, after computing the best parameter, we applied the Box-Cox transformation on both training and testing set, computing the classification accuracies in the transformed space: as done in section 3, we employed different classifiers, and we computed again the relative improvement obtained with respect to the unprocessed case (i.e. when the Box-Cox transformation is not applied) – see eq. (5).

From the experiments, it was clear that there is not a single criterion which is appropriate for all situations. This is somehow reasonable, since the four criteria have different characteristics: some are completely unsupervised (such as the ML), some others heavily rely on labels (PCML, Fisher) or on labels and a classifier (1NN-MM); since optimal parameters are estimated on the training set, unsupervised approaches can be less prone to overtraining,

[7] Optimization has been carried out using the standard `fminsearch` function of Matlab, which implements the Nelder-Mead simplex (direct search) method.

but they can also be too simplistic to capture the complexity of the problem; on the contrary, supervised approaches can be more flexible, but can lead to overtraining in some cases; and this risk becomes more critical while considering the Diagonal version of the Box-Cox, which is more flexible since there is a different parameter for every direction. These aspects clearly depend on the classifier which is applied on top of the Box-Cox transform and on the problem at hand. Even if there is no universal criterion working well for *all* classifiers and *all* datasets, we can hope to find a criterion which is most suited for a *single* given classifier, or for a *single* given problem. To investigate this aspect, we present in Table 6, for every dataset, its "favourite" criterion (i.e. the criterion which leads to the best improvement for all classifiers in such dataset), together with the averaged relative improvement with respect to the unprocessed case (i.e. when the Box-Cox transformation is not applied). The corresponding analysis for classifiers is reported in Table 7: for every classifier, we report its "favourite" criterion (i.e. the criterion which leads to the best improvement for all datasets using such classifier), together with the average improvement obtained.

Looking at Table 6 and 7, we can observe that all results are positive, indicating that there is at least one criterion for dataset/classifier which permits to select a proper λ for a given dataset. For half of the datasets the improvement is interesting, while for the remaining half it is rather modest. For classifiers, however, the improvements are more consistent; actually it seems easier to find a proper criterion given the classifier rather than given the dataset. Concerning the favourite criteria, it is interesting to note that in the tables both spherical and diagonal versions are present. Even if, in principle,

Problem	Favourite Criterion	Averaged Improvement
bananas	ML Diag	0.081%
ecoli	PCML Spher	0.024%
diabetes	Fisher Spher	0.323%
breast	ML Diag	0.780%
glass	PCML Spher	0.225%
wine	Fisher Spher	0.667%
brain	ML Spher	1.225%
biomed	Fisher Diag	1.329%
liver	ML Diag	10.873%
sonar	ML Diag	3.311%
spirals	1NN-MM Spher	1.627%
kimia	Fisher Diag	9.987%
fourier	Fisher Spher	0.046%
volcano	1NN-MM Spher	0.253%
energy	ML Diag	3.171%

Table 6: Results obtained by automatic selection of λ: averaged improvements obtained by employing the best criterion for every dataset.

diagonal versions can be more flexible than spherical versions, it may be difficult to find a *single* diagonal criterion working properly in *all* cases (all classifiers for a given dataset or all datasets for a given classifiers). In order to confirm this, we present some other results, in which for every experiment (dataset/classifier/fold) we selected the best result obtained among the four different criteria (ML, PCML, Fisher and 1NN-MM), using both the spherical

Classifier	Favourite Criterion	Averaged Improvement
nmc	Fisher Spher	0.882%
ldc	Fisher Diag	2.435%
udc	PCML Diag	1.821%
qdc	PCML Diag	2.895%
1-nn	Fisher Diag	0.052%
parzenc	Fisher Diag	3.006%
loglc	ML Spher	1.529%
linSVM	ML Diag	1.581%

Table 7: Results obtained by automatic selection of λ: averaged improvements obtained by employing the best criterion for every classifier.

and the diagonal Box-Cox Transform: such results, for different datasets and different classifiers (averaged over classifiers and datasets, respectively), are reported in Table 8.

It can be noted that now the diagonal criteria are in most of the cases better than the spherical ones, with drastic improvements in two cases (the qdc classifier and the kimia dataset). These results suggest that diagonal criteria may allow for more flexibility, but the improvement is not consistent for all criteria, all datasets and all classifiers.

5. Conclusions

In this paper we provided an empirical analysis of the behaviour of the Box-Cox transformation for pattern classification, showing that it represents an useful preprocessing tool, also when used in operational ranges which are

Dataset	Spherical	Diagonal
bananas	0.183%	0.537%
ecoli	0.494%	0.643%
diabetes	0.903%	0.417%
breast	0.922%	1.000%
glass	3.308%	4.023%
wine	1.381%	1.686%
brain	2.080%	1.596%
biomed	1.689%	2.339%
liver	8.783%	12.095%
sonar	2.991%	6.457%
spirals	7.087%	6.917%
kimia	4.894%	41.515%
fourier	0.654%	0.350%
volcano	0.985%	1.230%
energy	3.417%	6.016%

(a)

Classifier	Spherical	Diagonal
nmc	2.379%	2.794%
ldc	2.406%	9.166%
udc	2.562%	3.276%
qdc	2.632%	14.485%
1-nn	2.022%	4.109%
parzenc	3.564%	5.731%
loglc	3.081%	3.324%
linSVM	2.566%	3.045%

(b)

Table 8: Averaged improvements using best Spherical and best Diagonal criteria: (a) for different datasets, (b) for different classifiers.

far from those leading to the maximum Gaussianity. These results open the door to the analysis of different non linear data pre-processing methods, which can be successfully exploited in the pattern recognition field.

Acknowledgements

MB would like to thank Bob Duin for pointing out similarities between the Box-Cox transformation and the power transformation, and P. Lovato for helpful discussions on the experimental evaluation. Authors would also like to thank the Observatorio Vulcanológico y Sismológico de Manizales, Colombia (in particular John Makario Londoño-Bonilla) for providing the volcano data set.

References

[1] G. Box, D. Cox, An analysis of transformations, Journal of the Royal Statistical Society: Series B (Methodological) 26 (2) (1964) 211–252.

[2] R. Sakia, The box-cox transformation technique: a review, The Statistician 41 (1992) 169–178.

[3] A. Graybill, The Theory and Applications of the Linear Model, London, Duxbury Press, 1976.

[4] K. Fukunaga, Statistical Pattern Recognition, 2nd Edition, Academic Press, 1990.

[5] R. V. D. Heiden, F. Groen, The box-cox metric for nearest neighbour classification improvement, Pattern Recognition 30 (2) (1997) 273–279.

[6] J.-D. Lee, H.-R. Su, P. Cheng, M. Liou, J. Aston, A. Tsai, C.-Y. Chen, Mr image segmentation using a power transformation approach, IEEE Trans. Med. Imaging 28 (6) (2009) 894–905.

[7] L. Li, W. Chen, X. Shao, Z. Wang, Analysis of amplitude-integrated eeg in the newborn based on approximate entropy, IEEE Transactions on Biomedical Engineering 57 (10) (2010) 2459–2466.

[8] A. Barb, C.-R. Shyu, Visual-semantic modeling in content-based geospatial information retrieval using associative mining techniques, IEEE Geoscience and Remote Sensing Letters 7 (1) (2010) 38–42.

[9] X. Hong, A fast identification algorithm for box-cox transformation based radial basis function neural network, IEEE Trans. on Neural Networks 17 (4) (2006) 1064–1069.

[10] A. da Costa, A. Crepaldi, The bias in reversing the boxcox transformation in time series forecasting: An empirical study based on neural networks, Neurocomputing 136 (2014) 281 – 288.

[11] X. Wang, K. Smith-Miles, R. Hyndman, Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series, Neurocomputing 72 (1012) (2009) 2581 – 2594.

[12] B. Durbin, D. Rocke, Estimation of transformation parameters for microarray data, Bioinformatics 19 (11) (2003) 1360–1367.

[13] C. Chen, L. Pau, P. Wang, Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Company, 1993.

[14] M. Pan, L. Du, P. Wang, Z. B. H. Liu, Multi-task hidden markov modeling of spectrogram feature from radar high-resolution range profiles, EURASIP Journal on Advances in Signal Processing 2012 (2012) 86.

[15] G. Peeters, E. Deruty, Sound indexing using morphological description, IEEE Trans. on Audio, Speech, and Language Processing 18 (3) (2010) 675–687.

[16] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37 (2) (2004) 265–279.

[17] E. Pekalska, R. Duin, Classifiers for dissimilarity-based pattern recognition, in: Proc. Int. Conf. on Pattern Recognition, Vol. 2, 2000, pp. 12–16.

[18] A. Carli, M. Bicego, S. Baldo, V. Murino, Non-linear generative embeddings for kernels on latent variable models, in: Proc. of ICCV09 Workshop on Subspace Methods, 2009, pp. 154–161.

[19] A. Carli, M. Bicego, S. Baldo, V. Murino, Nonlinear mappings for generative kernels on latent variable models, in: Proc. Int. Conf. on Pattern Recognition, 2010, pp. 2134–2137.

[20] L. Pericchi, A bayesian approach to transformations to normality, Biometrika 68 (1981) 35–43.

[21] R. Carroll, A robust method for testing transformations to achieve approximate normality, Journal of the Royal Statistical Society. Series B (Methodological) 42 (1) (1980) 71–78.

[22] S. Velilla, A note on the multivariate box-cox transformation to normality, Statistics and Probability Letters 17 (1993) 259–263.

[23] M. Orozco-Alzate, P. Castro-Cabrera, M. Bicego, J. Londoño-Bonilla, The dtw-based representation space for seismic pattern classification, Computers & Geosciences 85 (2015) 86–95.

[24] M. Bicego, F. Recchia, A. Farinelli, S. Ramchurn, E. Grosso, Behavioural biometrics using electricity load profiles, in: Proc. Int. Conf on Pattern Recognition (ICPR2014), 2014, pp. 1764–1769.

[25] S. Pomeroy, P. Tamayo, M. Gaasenbeek, L. Sturla, M. Angelo, M. McLaughlin, J. Kim, L. Goumnerova, P. Black, C. Lau, J. Allen, D. Zagzag, J. Olson, T. Curran, C. Wetmore, J. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. Louis, J. Mesirov, E. Lander, T. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (2002) 436–42.

[26] M. Bicego, A. Ulaş, U. Castellani, A. Perina, V. Murino, A. Martins, P. Aguiar, M. Figueiredo, Combining information theoretic kernels with generative embeddings for classification, Neurocomputing 101 (2013) 161–169.

[27] T. Sebastian, P. Klein, B. Kimia, Recognition of shapes by editing shock graphs, in: Proc. Int. Conf. on Computer Vision, 2001, pp. 755–762.

[28] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax, S. Verzakov, PR-Tools4.1, a matlab toolbox for pattern recognition, `http://prtools.org` (2007).

[29] P. H. D.G. Stork and, R. Duda, Pattern Classification, 2nd Edition, Wiley, 2000.

[30] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection - theory and algorithms, in: Proc. Int. Conf. on Machine Learning (ICML), 2004, p. 43.

[31] K. Crammer, R. Gilad-Bachrach, A. Navot, N. Tishby, Margin analysis of the lvq algorithm, in: Proc. Int. Conf. on Neural Information Processing Systems (NIPS), 2002, p. 479.

[32] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.