Mon. Not. R. Astron. Soc. **000**, 1–12 (2014) Printed 20 November 2014 (MN LATEX style file v2.2)

Weighted principal component analysis: a weighted covariance eigendecomposition approach

L. Delchambre[1]*

[1] *Institut d'Astrophysique et de Géophysique, Université de Liège, Allée du 6 Août 17, B-4000 Sart Tilman (Liège), Belgium*

Accepted 2014 October 12. Received 2014 October 7; in original form 2014 July 4

ABSTRACT

We present a new straightforward principal component analysis (PCA) method based on the diagonalization of the weighted variance–covariance matrix through two spectral decomposition methods: power iteration and Rayleigh quotient iteration. This method allows one to retrieve a given number of orthogonal principal components amongst the most meaningful ones for the case of problems with weighted and/or missing data. Principal coefficients are then retrieved by fitting principal components to the data while providing the final decomposition. Tests performed on real and simulated cases show that our method is optimal in the identification of the most significant patterns within data sets. We illustrate the usefulness of this method by assessing its quality on the extrapolation of Sloan Digital Sky Survey quasar spectra from measured wavelengths to shorter and longer wavelengths. Our new algorithm also benefits from a fast and flexible implementation.

Key words: methods: data analysis – quasars: general.

1 INTRODUCTION

Principal component analysis (PCA) is a well-known technique initially designed to reduce the dimensionality of a typically huge data set while keeping most of its variance (Pearson 1901; Hotelling 1933). PCA is intimately related to the singular value decomposition (SVD) since the principal components of a data set, whose arithmetic mean is zero, will be equal to the eigenvectors of the covariance matrix sorted by their corresponding eigenvalue; or equivalently by the variance they account for. The principal coefficients are the linear coefficients allowing us to reconstruct the initial data set based on the principal components. Further details about PCA will be given in Section 3 of this paper. Interested readers are also invited to read Schlens (2009) for an accessible tutorial on this technique or Jolliffe (2002) for a deeper analysis.

PCA has many applications in a wide variety of astronomical domains from the classification of the Sloan Digital Sky Survey (SDSS) quasar spectra and their redshift determination (Yip et al. 2004; Pâris et al. 2014) to the study of the point spread function variation in lensing surveys (Jarvis & Jain 2004). The method described hereafter was originally developed in the framework of the *Gaia* astrophysical parameters inference system (Bailer-Jones et al. 2013) where it is used to provide learning data sets of spectrophotometric data based on SDSS quasar catalog spectra (Pâris et al. 2014). The latter cover the observed wavelength range 4000–10 000Å and are extrapolated by our algorithm to the wavelength range 3000–11 000Å covered by *Gaia*. Even if developed for an astronomical purpose, it can be used in any problems requiring PCA decomposition of weighted data. The case of missing data being simply the limiting case of weights equal to zero.

Classical PCA is a mature tool whose performance in dimensionality reduction and pattern recognition has been assessed for a long time. Nevertheless, its main limitation comes from the fact that it is not adapted to the case of samples having weighted and/or missing data. The inherent consequence is that the classical PCA implementations made no difference between variance coming from a genuine underlying signal and variance coming from measurement noise.

Most of the previous works cope with these limitations mainly by focusing on bypasses to the problem of noisy and/or missing data; or deal explicitly with particular cases. These encompass, for example, the interpolation of missing data (Beale & Little 1975) or cases where the weight matrix can be factorized into per-observation and per-variable weight matrices (Greenacre 1984). Jolliffe (2002) in sections 13.6 and 14.2 makes the point about these proposed solutions.

At the present time, some methods are still able to deal with weight matrices having the same size as the corresponding data set (Gabriel & Zamir 1979; Wentzell et al. 1997; Tipping & Bishop 1999; Srebro & Jaakkola 2003). Neverthe-

* E-mail: ldelchambre@ulg.ac.be

less, none of these are able to provide the orthogonal principal components ordered by the data set variance it accounts for. Rather, they provide an unsorted set of not-necessary orthogonal vectors whose linear combination is optimized to describe the underlying variance but whose goal is not compatible with the explanation of the variance given a minimal number of vectors.

Based on the idea of Roweis (1997), Bailey (2012) and Tsalmantza & Hogg (2012) have recently proposed methods based upon the expectation-maximization (EM) algorithm (Dempster et al. 1977) in order for the PCA to include weights associated with each variable within each observation. The objective of these methods is globally similar to the one of this paper. Differences mainly come from the fact that Tsalmantza & Hogg (2012) aim at finding an orthogonal decomposition of the original matrix such that the whole data set variance is the best accounted for. Instead, our implementation focuses on finding the orthogonal vectors that are the best at individually describing the data set variance at the expense of a lower explained global variance. This trade-off comes from the fact that in a weighted case, the solution to the problem of finding the set of N components explaining most of the variance of a data set is not guaranteed to contain the eigenvectors that are the best at individually describing this variance. The implementation of Bailey (2012) takes benefits of the flexibility of the EM algorithm in order to interpolate between these two solutions.

In Section 2, we explain the notation used in this paper. We summarize the properties of the classical PCA in Section 3. We see in details two current alternative implementations of weighted PCAs in Section 4. In Section 5, we describe our new algorithm while in Section 6, we see its application on simulated data and real cases and compare it against other algorithms. Finally, some properties and extensions are discussed in Section 7 and we conclude in Section 8.

2 NOTATION

This paper uses the following notations: vectors are in bold italic, $\boldsymbol{x}$; x_i being the element i of the vector $\boldsymbol{x}$. Matrices are in uppercase boldface or are explicitly stated; i.e. $\mathbf{X}$ from which the ith row will be denoted $\mathbf{X}_i^{\mathrm{row}}$ and the jth column by $\mathbf{X}_j^{\mathrm{col}}$, element at row i, column j will then be $\mathbf{X}_{ij}$. Amongst matrix operators, $\mathbf{a} \circ \mathbf{b}$ denotes the element-wise product (Hadamard product) of $\mathbf{a}$ and $\mathbf{b}$ and $\|\mathbf{a}\|$ denotes the Euclidian matrix norm of $\mathbf{a}$.

Consider a problem where we have N_{obs} observations each containing N_{var} variables, from which we want to retrieve N_{comp} principal components. For reference, here are the often used matrices along with their corresponding sizes: $\mathbf{X}$ the data set matrix ($N_{\mathrm{var}} \times N_{\mathrm{obs}}$) from which we have subtracted the mean observation $\bar{\mathbf{y}}$ ($\mathbf{X}_i^{\mathrm{col}} = \mathbf{Y}_i^{\mathrm{col}} - \bar{\mathbf{y}}$); $\mathbf{W}$ the weight of each variable within each observation ($N_{\mathrm{var}} \times N_{\mathrm{obs}}$); $\mathbf{P}$ the orthogonal matrix of principal components ($N_{\mathrm{var}} \times N_{\mathrm{comp}}$); $\mathbf{P}_i^{\mathrm{col}}$ being the ith principal component; $\mathbf{C}$ the principal coefficient matrix ($N_{\mathrm{comp}} \times N_{\mathrm{obs}}$); $\boldsymbol{\sigma}^{\mathbf{2}}$ the symmetric matrix of variance–covariance ($N_{\mathrm{var}} \times N_{\mathrm{var}}$) associated with $\mathbf{X}$.

Finally, $\mathbf{A} \overset{\chi^2}{=} \mathbf{B}$ means that $\mathbf{A}$ is the nearest matrix from $\mathbf{B}$ in a – potentially weighted – least-squares sense. Mathematically, this is equivalent to have $\mathbf{A}$ and $\mathbf{B}$ such that

$$\chi^2 = \sum_{ij} \mathbf{W}_{ij}^2 \left(\mathbf{A}_{ij} - \mathbf{B}_{ij}\right)^2 \tag{1}$$

is minimized.

3 PRINCIPAL COMPONENT ANALYSIS

Regarding classical PCA and assuming – without any loss of generality – that we would like to retrieve as many principal components as the number of variables (ie. $N_{\mathrm{comp}} = N_{\mathrm{var}}$), then the goal of the PCA will be to find a decomposition

$$\mathbf{X} = \mathbf{PC}, \tag{2}$$

such that

$$\mathbf{D} = \mathbf{P}^{\mathrm{T}} \boldsymbol{\sigma}^2 \mathbf{P} = \mathbf{P}^{\mathrm{T}} \mathbf{X} \mathbf{X}^{\mathrm{T}} \mathbf{P} \tag{3}$$

is diagonal and for which

$$\mathbf{D}_{ii} \geqslant \mathbf{D}_{jj};\ \forall i < j. \tag{4}$$

Note that based on equation (3) and according to the spectral theorem[1], $\mathbf{P}$ will be orthogonal.

Intuitively, the matrix $\mathbf{P}$ can be seen as a change of basis allowing us to maximize the variance within $\mathbf{D}$ and thus minimizing the off-diagonal elements corresponding to the covariance. Differently stated, each $\mathbf{P}_i^{\mathrm{col}}$ defines a privileged direction along which the data set variance is the best explained. The fact that $\mathbf{D}$ is ordered implies that for $i < j$, the principal component $\mathbf{P}_i^{\mathrm{col}}$ accounts for more – or equal – variance than $\mathbf{P}_j^{\mathrm{col}}$. For the sake of clarity, a comprehensive PCA example is given in Fig. 1.

A common solution to such a classical PCA is based on the SVD of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^{\mathrm{T}}, \tag{5}$$

where $\mathbf{U}$, $\mathbf{V}$ are orthogonals, $\boldsymbol{\Sigma}$ is diagonal and for which $|\boldsymbol{\Sigma}_{ii}| \geqslant |\boldsymbol{\Sigma}_{jj}|;\ \forall i < j$. By setting $\mathbf{P} = \mathbf{U}$ and $\mathbf{C} = \boldsymbol{\Sigma} \mathbf{V}^{\mathrm{T}}$, we find that equation (3) becomes

$$\mathbf{P}^{\mathrm{T}} \mathbf{X} \mathbf{X}^{\mathrm{T}} \mathbf{P} = \mathbf{C} \mathbf{C}^{\mathrm{T}} = \boldsymbol{\Sigma}^2, \tag{6}$$

that fulfils the conditions of equations (3) and (4). Note that in equation (3) the exact variance–covariance matrix should be normalized by N_{obs} but since we are solely interested in the diagonalization of $\boldsymbol{\sigma}^{\mathbf{2}}$, we drop it.

4 WEIGHTED EXPECTED MAXIMIZATION PCA

As already mentioned in Section 1, the current methods efficiently dealing with the problem of weighting PCA (Bailey 2012; Tsalmantza & Hogg 2012) aim at best explaining the whole data set variance according to a given number of principal components, which is equivalent to minimize

$$\chi^2 = \sum_{ij} \mathbf{W}_{ij}^2 \left(\mathbf{X}_{ij} - [\mathbf{PC}]_{ij}\right)^2, \tag{7}$$

[1] Any real symmetric matrix is diagonalized by a matrix of its eigenvectors.

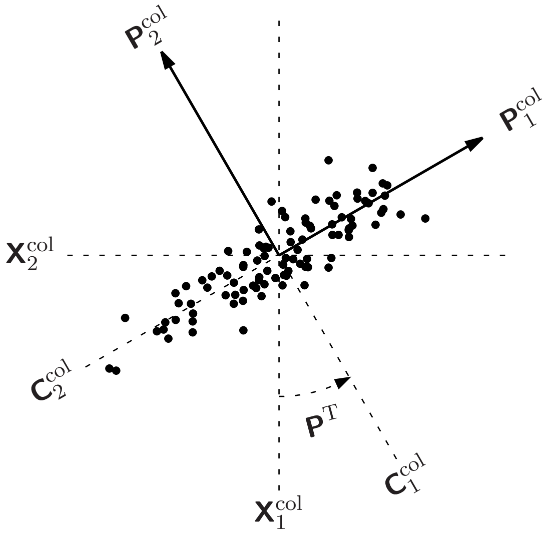
$\mathbf{P}_2^{\mathrm{col}}$
$\mathbf{P}_1^{\mathrm{col}}$
$\mathbf{X}_2^{\mathrm{col}}$
$\mathbf{C}_2^{\mathrm{col}}$
$\mathbf{P}^{\mathrm{T}}$
$\mathbf{C}_1^{\mathrm{col}}$
$\mathbf{X}_1^{\mathrm{col}}$

Figure 1. A two-dimensional PCA example: $\mathbf{P}^{\mathrm{T}}$ can be seen as an orthogonal coordinate transformation from $\left(\mathbf{X}_1^{\mathrm{col}};\mathbf{X}_2^{\mathrm{col}}\right)$ to $\left(\mathbf{C}_1^{\mathrm{col}};\mathbf{C}_2^{\mathrm{col}}\right)$ such that the data set variance is maximized along the vector $\mathbf{P}_1^{\mathrm{col}}$ and $\mathbf{P}_2^{\mathrm{col}}$. Note that, for a didactical purpose, we chose $\mathbf{P}$ to be a rotation matrix but practically it can be any orthogonal matrix.

where the weighted mean observation we subtracted (cf. Section 2) is given by

$$\bar{\mathbf{y}} = \sum_i \mathbf{W}_i^{\mathrm{col}} \mathbf{Y}_i^{\mathrm{col}} \Big/ \sum_i \mathbf{W}_i^{\mathrm{col}}. \tag{8}$$

We notice that equation (7) has latent variables such that it has to rely on an iterative procedure to be solved.

The EM algorithm is a statistical tool specifically designed to optimize a likelihood function for models having latent – or unknown/hidden – variables (Dempster et al. 1977). This iterative procedure is composed of two steps.

(i) E-step: find the expectation value of the latent variables given the parameters of the current model.

(ii) M-step: find the parameters of the model such that the likelihood function is optimized.

Based on the latter, Roweis (1997) has developed a fast and straightforward classical PCA algorithm for which the conditions in equations (2), (3) and (4) are all fulfilled. Regarding weighted PCA and more specifically the χ^2 described by equation (7), we will have the following weighted expected maximization PCA (WEMPCA) algorithm:

```
$\mathbf{P} \longleftarrow$ Random orthogonal matrix
While $\mathbf{P}$ and $\mathbf{C}$ have not converged
    (E-step) Find $\mathbf{C}$ that minimizes $\chi^2$ given $\mathbf{P}$.
    (M-step) Find $\mathbf{P}$ that minimizes $\chi^2$ given $\mathbf{C}$.
```

Note that the convergence criterion is still relative. It can be based on the χ^2 – or the change in the principal components $\Delta\mathbf{P}$ – falling under a given threshold, the fact that the algorithm has reached a given number of iterations or whatever criterion we consider as relevant.

4.1 Tsalmantza's implementation

Tsalmantza & Hogg (2012) designed a general approach to the modelling and dimensionality reduction of the SDSS spectra called 'Heteroskedastic Matrix Factorization'. More specifically, it attempts to minimize

$$\chi^2_\epsilon = \chi^2 + \epsilon \sum_{i>1} \sum_j \left[\mathbf{P}_{ij} - \mathbf{P}_{(i-1)j}\right]^2, \tag{9}$$

subject to

$$\left.\begin{array}{l} \mathbf{P}_{ij} \geqslant 0 \\ \mathbf{C}_{jk} \geqslant 0 \end{array}\right\} \forall i, j, k. \tag{10}$$

We recognize the first part of equation (9) as being equation (7) while the second part is a smoothing regularization term whose scalar ϵ defines the strength. Non-negativity constraints reflect a particular need to have a meaningful physical interpretation of the resulting spectra.

Regarding the fact that we would like to model the widest variety of data sets, we will drop the non-negativity constraints that otherwise would have restricted our search space. Concerning the smoothing regularization factor, we have to note that it will be highly problem-dependent and that it can be tricky to optimize, this will result in a potential unfair comparison with other methods. We will then consider the case $\epsilon = 0$. Moreover, as we will see in Section 7.3, our method can deal with principal components smoothing as well, consequently ignoring it will not constitute a major drawback to our implementation.

The resulting function to optimize will then be reduced to the sole equation (7). Nevertheless, and before going further, we have to note that minimizing equation (7) will provide us a lower-rank matrix approximation of $\mathbf{X}$ but it is not a sufficient condition for the resulting matrices $\mathbf{P}$ and $\mathbf{C}$ to be considered as a PCA decomposition. According to Tsalmantza & Hogg (2012), the solution to this problem can be solved in two steps.

First consider a lower-rank matrix decomposition of $\mathbf{X}$, similar to the one produced by the solution of equation (7),

$$\mathbf{X} \overset{\chi^2}{=} \mathbf{AB}, \tag{11}$$

where for clarity, the sizes of these matrices are $\mathbf{A}(N_{\mathrm{var}} \times N_{\mathrm{comp}})$ and $\mathbf{B}(N_{\mathrm{comp}} \times N_{\mathrm{obs}})$. Now suppose an orthogonal basis $\mathbf{P_0}$ of $\mathbf{A}$; such an orthogonal basis always exists for full-rank matrices and can be retrieved through a straightforward Gram–Schmidt process for example. The associated coefficients matrix $\mathbf{C_0}$ is then directly retrieved by

$$\mathbf{C_0} = \mathbf{P_0}^{\mathrm{T}} \mathbf{AB}. \tag{12}$$

Secondly, in a way similar to equation (6), we will take the classical PCA decomposition of $\mathbf{C_0}$,

$$\mathbf{C_0} = \mathbf{P_c} \mathbf{C}, \tag{13}$$

such that $\mathbf{CC}^{\mathrm{T}}$ is diagonal. The resulting principal coefficient matrix will then be given by

$$\mathbf{P} = \mathbf{P_0} \mathbf{P_c}, \tag{14}$$

that will be orthogonal and that will provide us with the final decomposition

$$\mathbf{X} \overset{\chi^2}{=} \mathbf{PC}. \tag{15}$$

The above-mentioned steps have thus to be performed after the EM algorithm minimizing equation (7) in order for $\mathbf{P}$ to be orthogonal and for the covariance matrix $\mathbf{C}\mathbf{C}^{\mathrm{T}}$ to be diagonal. Sections 4.1.1 and 4.1.2 will now focus on details of the EM algorithm.

4.1.1 E-step

As stated at the beginning of this section, the expectation step regarding WEMPCA will be given by the retrieval of the coefficient matrix $\mathbf{C}$ that minimizes equation (7) – or equivalently that optimizes equation (15) – while considering the principal component matrix $\mathbf{P}$ being held fixed. Since each observation – column of $\mathbf{X}$ – is a linear combination of the principal components, finding the solution of equation (15) is equivalent to solving

$$\mathbf{X}_i^{\mathrm{col}} \overset{\chi^2}{=} \mathbf{P}\mathbf{C}_i^{\mathrm{col}};\ \forall i, \tag{16}$$

whose solutions are given by the 'Normal Equations':[2]

$$\mathbf{C}_i^{\mathrm{col}} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{w}^2\mathbf{P}\right)^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{w}^2\mathbf{X}_i^{\mathrm{col}}, \tag{17}$$

with $\mathbf{w} = \mathrm{diag}\left(\mathbf{W}_i^{\mathrm{col}}\right)$.

4.1.2 M-step

Similarly to the E-step, solution to the M-step – that is the retrieval of $\mathbf{P}$ that optimizes equation (15) given $\mathbf{C}$ – can be decomposed by noting that within each observation, a given variable is the linear combination of the corresponding principal components variables. That is,

$$\mathbf{X}_i^{\mathrm{row}} \overset{\chi^2}{=} \mathbf{P}_i^{\mathrm{row}}\mathbf{C};\ \forall i, \tag{18}$$

whose solutions are

$$\mathbf{P}_i^{\mathrm{row}} = \mathbf{X}_i^{\mathrm{row}}\mathbf{w}^2\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\mathbf{w}^2\mathbf{C}^{\mathrm{T}}\right)^{-1}, \tag{19}$$

with $\mathbf{w} = \mathrm{diag}\left(\mathbf{W}_i^{\mathrm{row}}\right)$.

4.2 Bailey's implementation

As we have seen in Section 4.1, the implementation of Tsalmantza & Hogg (2012) focuses on the solution of equation (7) while the PCA decomposition can be seen as a supplemental step that is external to the EM algorithm. Moreover, a single iteration of the algorithm requires the solution of $N_{\mathrm{obs}} + N_{\mathrm{var}}$ systems of linear equations, each of size $(N_{\mathrm{comp}} \times N_{\mathrm{comp}})$, at each iteration of the EM algorithm. This becomes quickly unmanageable regarding huge data sets.

Bailey (2012) takes the pragmatic approach that the sole solution of equation (7) can lead to good insights about the principal components if the latter were fitted individually. This hypothesis is reasonable since this will allow each individual principal component to maximize the variance it accounts for.

The resulting implementation will be similar to the one of Tsalmantza & Hogg (2012) apart from the optimization function of the M-step. Indeed, in order for the principal components to be fitted separately, we have to consider the cross-product decomposition of equation (15), that is

$$\mathbf{X} \overset{\chi^2}{=} \sum_j \mathbf{P}_j^{\mathrm{col}}\mathbf{C}_j^{\mathrm{row}}, \tag{20}$$

from which each $\mathbf{P}_j^{\mathrm{col}}$ has to be individually fitted.

Suppose that we already retrieved the $(j-1)$ first principal components. Let us also assume that the data projection along these $(j-1)$ principal components was already subtracted from the data set, that is

$$\mathbf{X}' = \mathbf{X} - \sum_i^{j-1} \mathbf{P}_i^{\mathrm{col}}\mathbf{C}_i^{\mathrm{row}}. \tag{21}$$

Then the retrieval of $\mathbf{P}_j^{\mathrm{col}}$ based on equation (20) can be decomposed in a way similar to equation (18) as

$$\mathbf{X}_i'^{\mathrm{row}} \overset{\chi^2}{=} \mathbf{P}_{ij}\mathbf{C}_j^{\mathrm{row}}, \tag{22}$$

whose solution is straightly given by

$$\mathbf{P}_{ij} = \frac{\mathbf{X}_i'^{\mathrm{row}}\mathbf{w}^2\mathbf{C}_j^{\mathrm{row}^{\mathrm{T}}}}{\mathbf{C}_j^{\mathrm{row}}\mathbf{w}^2\mathbf{C}_j^{\mathrm{row}^{\mathrm{T}}}}, \tag{23}$$

with $\mathbf{w} = \mathrm{diag}\left(\mathbf{W}_i^{\mathrm{row}}\right)$.

Equation (21) theoretically ensures that the last-retrieved component, $\mathbf{P}_j^{\mathrm{col}}$ will be orthogonal to any previous one. Nevertheless, due to machine round-off errors, this has to be manually checked.

Finally, we have to note that solving equation (20) will not minimize the global χ^2 – as defined by equation (7) – such that the algorithm has to rely on a last E-step at the end of the main EM algorithm.

5 NEW IMPLEMENTATION

Though both mentioned algorithms (Bailey 2012; Tsalmantza & Hogg 2012) correctly find lower-rank orthogonal decompositions that are suitable to explain the whole data set variance at best, none of them assures us that the retrieved principal components will be those that maximize the individual variance described by each of them. These principal components are then efficient at reconstructing the initial data set but are not the best at individually describing the underlying data structure.

The basic idea of this new algorithm is to focus on the maximization of the weighted variance explained by each principal component through the diagonalization of the associated weighted covariance matrix. The resulting principal components will then be those that are the most significant – under the assumption that the definition of the used weighted variance is relevant – in identifying pattern within the data set even if their linear combination is not necessarily the best at explaining the total data set variance as described by equation (7).

In the following, we will consider that the weighted variance of a given discrete variable $\boldsymbol{x}$ having weights $\boldsymbol{w}$ is given by

$$\sigma_{\boldsymbol{x}}^2 = \frac{\sum_i w_i^2\left(x_i - \bar{x}\right)^2}{\sum_i w_i^2}, \tag{24}$$

[2] This method is known to suffer from numerical instabilities (Press et al. 2002) and is provided for a didactical purpose only. Methods such as SVD for linear least squares must be preferred in order to solve equation (16).

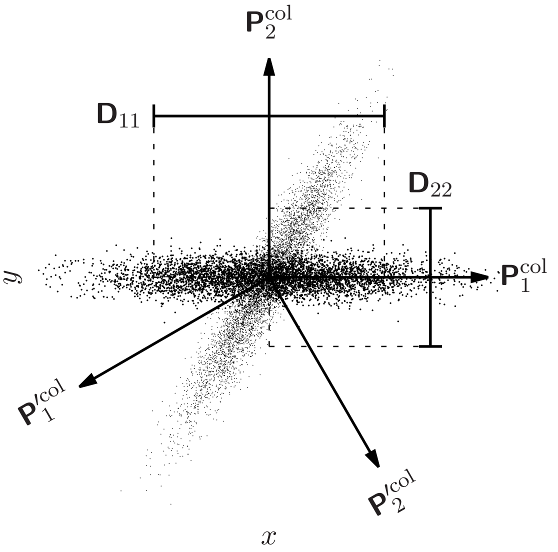

Figure 2. Comparison between weighted principal components $\mathbf{P}$ and classical principal components $\mathbf{P'}$ in presence of under-weighted observations corresponding to the brighter points. $|\mathbf{D}_{11}|$ and $|\mathbf{D}_{22}|$ correspond to the variances respectively explained by $\mathbf{P}_1^{\mathrm{col}}$ and $\mathbf{P}_2^{\mathrm{col}}$.

where $\bar{x} = \sum_i w_i x_i / \sum_i w_i$ and with the convention that $0/0 = 0$. The latter can be straightforwardly extended to the definition of the weighted covariance between two discrete variables, $\boldsymbol{x}$ and $\boldsymbol{y}$, for which the weights are given by $\boldsymbol{w}^{\boldsymbol{x}}$ and $\boldsymbol{w}^{\boldsymbol{y}}$ respectively, that is

$$\sigma^2_{\boldsymbol{x},\boldsymbol{y}} = \frac{\sum_i (x_i - \bar{x})\, w_i^x w_i^y\, (y_i - \bar{y})}{\sum_i w_i^x w_i^y}. \tag{25}$$

Based on these definitions, we can write the weighted covariance matrix of a data set $\mathbf{X}$ with associated weights $\mathbf{W}$ as

$$\boldsymbol{\sigma}^2 = \frac{(\mathbf{X} \circ \mathbf{W})\,(\mathbf{X} \circ \mathbf{W})^{\mathrm{T}}}{\mathbf{W}\mathbf{W}^{\mathrm{T}}}. \tag{26}$$

We know, from the spectral theorem, that there exists an orthogonal matrix $\mathbf{P}$ such that $\boldsymbol{\sigma}^2$ is diagonalized, and consequently that equation (3) is fulfilled (as well as equation 4 if $\mathbf{P}$ is ordered accordingly). This matrix $\mathbf{P}$ will then constitute the principal components of our implementation.

Fig. 2 shows a two-dimensional example of classical principal components $\mathbf{P'}$ that are unable to individually describe the underlying data variance. In this example, we have supposed – for didactical purpose – that the observations corresponding to the bright points have far lower weights associated with the x variable. Weighted principal components, $\mathbf{P}$, diagonalizing $\boldsymbol{\sigma}^2$ as described by equation (26) are given along with the variance it explains (that is the diagonal elements of $\mathbf{D} = \mathbf{P}^{\mathrm{T}}\boldsymbol{\sigma}\mathbf{P}$). Note that such principal components maximize the variance explained by each variable as described by equation (24) and consequently set to zero the associated covariance as described by equation (25).

The goal of the algorithm is then to retrieve the dominant eigenvector $\boldsymbol{p}$ from the covariance matrix $\boldsymbol{\sigma}^2$ along with its associated eigenvalue λ, that is the dominant eigenpair $\langle \boldsymbol{p}, \lambda \rangle$. $\boldsymbol{p}$ will then be the principal component explaining most of the data set variance, λ. That is equivalent to find $\boldsymbol{p}$ in

$$\boldsymbol{\sigma}^2 \boldsymbol{p} = \lambda \boldsymbol{p}, \tag{27}$$

such that λ is maximized[3].

Equation (27), corresponding to the eigenvector definition, is a well-studied problem for which many methods already exist. The reference in the domain is Golub & Van Loan (1996) where the interested reader may find a rather exhaustive list of such methods as well as proofs of the algorithms described hereafter.

Unsurprisingly, in the context or our implementation, we choose the fastest and simplest algorithm called the power iteration method. The idea behind this algorithm is to recognize that given a diagonalizable square matrix $\mathbf{A}$ and a vector $\boldsymbol{u}^{(0)}$ having a nonzero component in the direction of $\boldsymbol{p}$, the iterative relation

$$\boldsymbol{u}^{(k)} = \mathbf{A}\boldsymbol{u}^{(k-1)} = \mathbf{A}^k \boldsymbol{u}^{(0)} \tag{28}$$

will converge to a vector that is proportional to the dominant eigenvector $\boldsymbol{p}$ as $k \to \infty$. Note that in practice, each vector $\boldsymbol{u}^{(k)}$ is normalized to unity in order to avoid numerical round-off errors inherent to the computer representation of large numbers. The final eigenvector will then be given by $\boldsymbol{p} = \boldsymbol{u}^{(k)} / \left\| \boldsymbol{u}^{(k)} \right\|$ and the associated eigenvalue by the Rayleigh quotient:

$$\mathcal{R}(\mathbf{A}, \boldsymbol{p}) = \boldsymbol{p} \cdot \mathbf{A}\boldsymbol{p} = \boldsymbol{p} \cdot \lambda \boldsymbol{p} = \lambda. \tag{29}$$

Convergence and assumptions made about this algorithm will be discussed in Section 7.1. Further principal components can be retrieved by considering application of the above-mentioned algorithm to

$$\mathbf{A'} = \mathbf{A} - \lambda \boldsymbol{p} \otimes \boldsymbol{p}, \tag{30}$$

that is the matrix obtained by subtracting the data variance along the found principal components.

5.1 Refinement

As we will see in Section 7.1, the power iteration method may have a slow convergence rate under some particular conditions. Consequently it may be that some vectors did not effectively converge to an eigenvector that would have diagonalized the covariance matrix. Nevertheless, Parlett (1974) proposed an algorithm – called Rayleigh quotient iteration – designed to tackle this kind of problem.

Even if the proof of this algorithm is beyond the scope of this paper, we still mention two basic facts to enable the reader to have a minimal understanding of how it works. First, as we have seen in equation (29), the Rayleigh quotient of a matrix with one of its eigenvector is equal to its associated eigenvalue. Secondly, given a matrix $\mathbf{A}$ with eigenvalues $\lambda_1, ..., \lambda_n$, we find that the eigenvalues of the matrix $(\mathbf{A} - d\mathbf{I})^{-1}$ will be $(\lambda_1 - d)^{-1}, ..., (\lambda_n - d)^{-1}$. Based upon these facts, we will have that the sequence

$$\boldsymbol{u}^{(k)} \quad = \quad \left(\mathbf{A} - d^{(k-1)}\mathbf{I}\right)^{-1} \boldsymbol{u}^{(k-1)} \tag{31}$$

[3] In fact, $\langle -\boldsymbol{p}, -\lambda \rangle$ is also solution of equation (27) but for the sake of clarity, we will only consider the case of positive eigenvalues

$$d^{(k)} = \mathcal{R}\left(\mathbf{A}, \boldsymbol{u}^{(k)}\right), \tag{32}$$

where each $\boldsymbol{u}^{(k)}$ is normalized to the unit length, will converge cubically to the eigenpair $\langle \boldsymbol{p}, \lambda \rangle$ that is the nearest – regarding the absolute value of their Rayleigh quotient – from a starting point $\left\langle \boldsymbol{u}^{(0)}, d^{(0)} \right\rangle$ as $k \to \infty$. Note that equation (31) is a power iteration towards the eigenvector for which $\left|\lambda - d^{(k-1)}\right|$ is minimized.

Finally, the principal coefficients are retrieved by solving $\chi^2 \overset{\chi^2}{=} \mathbf{PC}$ whose 'Normal Equation' solution is given by equation (17).

5.2 Variance regularization

Real-world data often have sparse and unevenly distributed weights such that it may happen for some variables to have their corresponding variances to be based only on a small number of observations. Such situations may become problematic since these few variables will have a strong impact on the resulting first principal components in a way that is irrespective to their total weight.

Such 'overweighted' variables can be damped by using a regularization factor within the expression of the weighted covariance, $\sigma^2_{\boldsymbol{x},\boldsymbol{y}}$, as defined by equation (25). The resulting regularized weighted covariance will take the form

$$\sigma^2_{\boldsymbol{x},\boldsymbol{y}}(\xi) = \left[\sum_i w_i^x \sum_i w_i^y\right]^{\xi} \sigma^2_{\boldsymbol{x},\boldsymbol{y}}, \tag{33}$$

where the regularization parameter ξ allows us to control the damping strength.

The typical value of the regularization parameter, ξ, goes from zero, where we get back to the classical behaviour of the algorithm, to two for a strong damping of these rare variables. Conversely one might want to highlight such underrepresented variables by setting the regularization parameter to a negative value.

6 COMPARISON

The performance of the method described in this paper was assessed regarding the two previously described algorithms, namely the one of Bailey (2012) and the one of Tsalmantza & Hogg (2012). The choice of these algorithms comes from the fact that they are fairly competitive and have goals that are comparable to those of the new algorithm. All methods were tested on both simulated data as well as on real observational ones.

6.1 Simulated data

Simulated data consist in random linear combinations of 10 orthogonal basis functions. These bases are produced by taking 10 shifted sine functions having periods between 0.2π and 2π and by applying a Gram–Schmidt orthogonalization process to the latter. Resulting observations are then sampled over 100 evenly spaced points in the interval $[0, 2\pi]$. To each variable, x, within each observation, $\boldsymbol{x}$, we also add a Gaussian noise having a standard deviation given by

$$\sigma_x = \sigma_{\text{in}}\left(1 + \sigma_{\text{obs}}\right)\left(1 + \mathcal{U}_\sigma\right) \max|\boldsymbol{x}|, \tag{34}$$

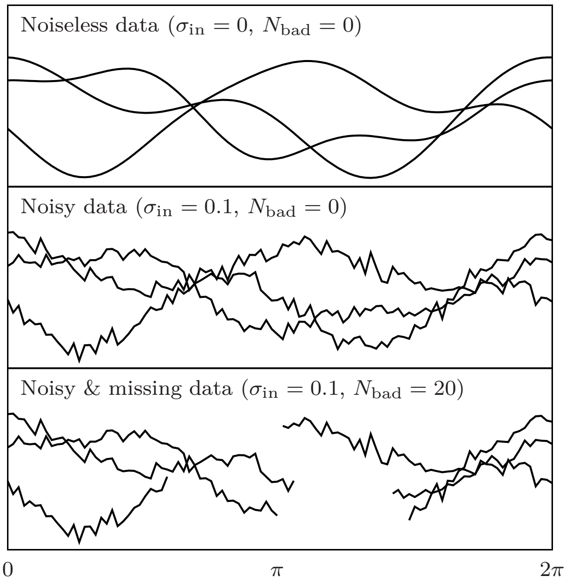

Figure 3. Examples of simulated data illustrating the effect of the simulation parameters σ_{in} and N_{bad}.

where σ_{in} is a user-provided parameter corresponding to the desired noise amplitude, σ_{obs} is an observation-specific noise ponderation uniformly drawn from $[-0.1, 0.1]$ and $\mathcal{U}_\sigma$ is a uniform random variable corresponding to the noise dispersion within observations and taking values in the range $[-0.1, 0.1]$. The weight associated with the variable x will then be set to $1/\sigma_x$.

Finally, we discard N_{bad} contiguous and randomly positioned variables from each observation. The latter will be used to assess performances of the various algorithms on data extrapolation while having their weights equal to zero during the PCA retrieval phase. Examples of such simulated data are illustrated in Fig. 3.

In order to perform the comparison with other algorithms, we built a given number of data sets, N_{set}, each containing 1000 observations. Each of these sets was duplicated and altered through the realization of various values of the simulation parameters σ_{in} and N_{bad}. We then retrieved, for each of the mentioned algorithms, the five first principal components out of the resulting altered data sets and computed estimators based on the following χ^2 definition:

$$\chi^2 = \frac{\sum_{ij}\left[\mathbf{W} \circ (\mathbf{X} - \mathbf{PC})\right]^2_{ij}}{\sum_{ij} \mathbf{W}^2_{ij}}. \tag{35}$$

The following estimators were computed: χ^2_{fit}, the chi-square of the data set for which weights associated with the discarded variables are set to zero and χ^2_{test} where only rejected variables are considered and for which weights associated with the unrejected variables are set to zero. Let us note that χ^2_{fit} will typically account for the quality of the fit while χ^2_{test} will account for the quality of the extrapolation. If these estimators are to be computed based on the number of data set, N_{set}, we use a 3σ-clipped mean over all the χ^2_{fit} and χ^2_{test} associated with each data set, namely $\langle\chi^2_{\text{fit}}\rangle$ and $\langle\chi^2_{\text{test}}\rangle$.

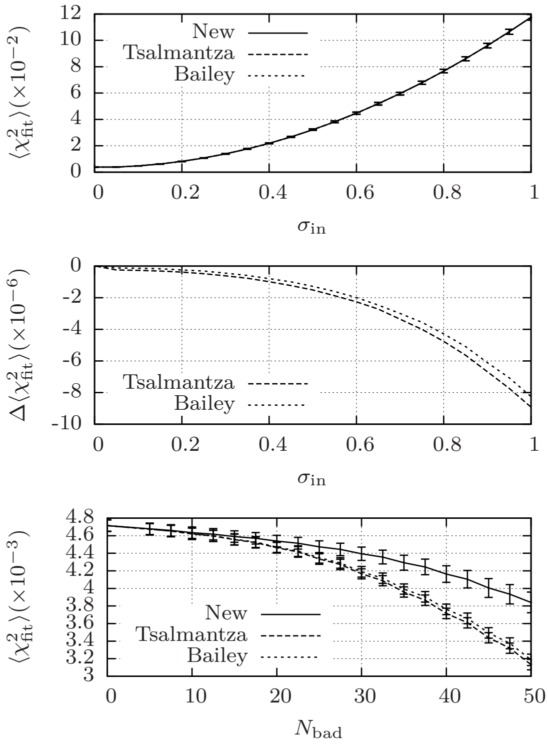

Figure 4. Averaged $\chi^2_{\rm fit}$ for increasing noise without missing data (top). Subtraction of Bailey and Tsalmantza $\chi^2_{\rm fit}$ from our algorithm $\chi^2_{\rm fit}$ (middle). Averaged $\chi^2_{\rm fit}$ with moderate noise and an increasing number of missing data (bottom).

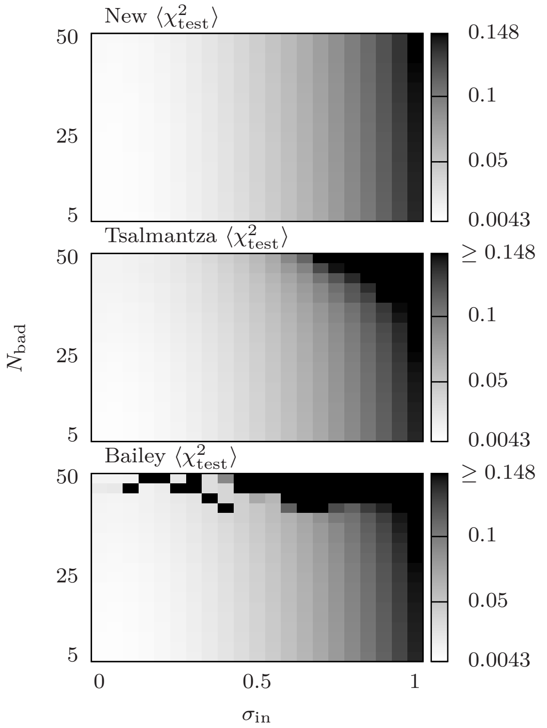

Figure 5. χ^2 maps of averaged $\chi^2_{\rm test}$ regarding $\sigma_{\rm in}$ and $N_{\rm bad}$ simulation parameters.

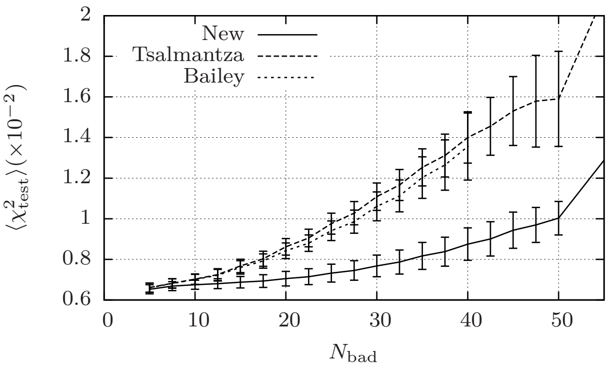

Figure 6. Averaged $\chi^2_{\rm test}$ for moderate noise with an increasing number of missing data.

For completeness, only data sets having $N_{\rm bad} \leqslant 50$ will be discussed here. This decision comes from the fact that efficiently estimating the principal components of such sparse data sets while having ‘only’ 1000 observations is a really tricky task strongly depending on the design of these data sets. Furthermore, as we will see in Section 7.4, the Bailey and Tsalmantza algorithms are big time consumers such that dealing with bigger data sets quickly becomes unmanageable.

Since all the studied algorithms are based on iterative procedures, we have to take into account the convergence criterion for each of them. To this aim, we performed a preliminary study whose goal is to determine the minimal number of iterations needed by each algorithm in order to reach convergence. This was assessed by running 100 times each algorithm on data sets similar to those previously described and by setting the initial eigenvectors estimates to random values within each run. In order to make sure we can model unseen and potentially more complex data sets, the number of iterations was set to twice the found number of iterations needed to converge, giving respectively 500, 500 and 10^4 iterations for Bailey, Tsalmantza and our algorithm (without refinement).

Regarding the quality of the fit, $\chi^2_{\rm fit}$, differences between the various algorithms are fairly low even if – as expected – our algorithm is proven to have somewhat larger $\chi^2_{\rm fit}$ with a higher dispersion over all the data sets. Fig. 4 shows the behaviour of the mean $\chi^2_{\rm fit}$ regarding two common cases, namely the case of increasing noise and no missing data and the case of moderate noise ($\sigma_{\rm in} = 0.1$) with increasing number of missing data. Each point on these graphs is averaged over $N_{\rm set} = 1000$ data sets. Practically, one cannot distinguish the various algorithms if all data are present since differences are in $\mathcal{O}\left(10^{-6}\right)$. In presence of missing data, differences start to be noticeable but still reasonable with differences in $\mathcal{O}\left(10^{-3}\right)$. For a didactical purpose, let us note that if only the first component – out of the five retrieved – was considered then our algorithm would have had a better or equal $\chi^2_{\rm fit}$ than Bailey and Tsalmantza implementations over all the values of the simulation parameters $\sigma_{\rm in}$ and $N_{\rm bad}$.

Regarding now the quality of the extrapolation, differences are more noticeable. Fig. 5 illustrates these discrepancies according to the averaged $\chi^2_{\rm test}$ over $N_{\rm set} = 100$ data sets having reasonable $\chi^2_{\rm fit}$ in each of the $\sigma_{\rm in}$ and $N_{\rm bad}$ simulation parameters. At first glance, our new algorithm shows a globally better $\langle\chi^2_{\text{test}}\rangle$ while suffering less from missing data: the latter being dominated by noise within the data. The Tsalmantza's algorithm shows good performances but has a stronger dependence on missing data, reaching a maximum $\langle\chi^2_{\text{test}}\rangle = 0.372$ while Bailey's algorithm shows strong numerical instabilities for $N_{\text{bad}} > 40$ that makes it unable to converge and makes it reach a maximum $\langle\chi^2_{\text{test}}\rangle \approx 3 \times 10^9$ for $\sigma_{\text{in}} = 0.9$ and $N_{\text{bad}} = 50$. Fig. 6 illustrates in more detail the behaviour of $\langle\chi^2_{\text{test}}\rangle$ averaged over $N_{\text{set}} = 1000$ data sets in the common case of moderate noise ($\sigma_{\text{in}} = 0.1$) with an increasing number of missing data. For clarity, the plot of Bailey having $N_{\text{bad}} > 40$ has been removed from the graph.

Finally, let us note that data sets having $\sigma_{\text{in}} = 0$ and $N_{\text{bad}} = 0$ can be solved using a classical PCA algorithm. Consequently, both the variance explained by each individual component and the total variance can be simultaneously optimized. We will thus find – in this particular case – that all algorithms will provide us with identical results as it was already suggested in Fig. 4. Similarly, if we choose to retrieve a single component, even with noisy and/or missing data, the algorithms of Bailey and Tsalmantza will maximize the variance explained by this component, the latter will thus match the first component that would have been retrieved by our algorithm.

6.2 Observational data

Comparisons against a concrete case were performed using the SDSS DR10Q quasar catalog from Pâris et al. (2014). Out of the 166 583 QSO spectra present in the initial data release, 18 533 were rejected either due to spectra bad quality, strong uncertainties in redshift determination, presence of BAL or insufficient number of high-S/N points. The remaining spectra were set to the rest frame; the continuum was then subtracted and the spectra were normalized such as to have a zero mean and a variance of 1. Finally, visual inspection showed that in some cases the continuum was badly fitted such that the variance within these spectra can mainly be attributed to this error; these regions were removed using a k-sigma clipping algorithm for each variable among all observations. There remain 148 050 spectra having observed wavelengths between 4000 and 10 000Å and for which the variance within each spectrum is thought to be mainly caused by genuine signals. In the following tests, the 10 first components will be retrieved from each algorithm. We also suppose, as previously, that $\xi = 0$.

The number of iterations associated with each algorithm was assessed by using a subset of the above-described data set ($1000 \leqslant \lambda_{\text{rest}} \leqslant 2000$Å, $3 \leqslant z \leqslant 4$) and by running 10 times the various algorithms on it with random initial principal components. Convergence was assessed by minimizing the variance amongst the final principal components within these 10 runs. The fact that only a subset of the above-described data set was used can be explained by the large amount of time needed by each algorithm to run as well as by the fact that – as we will see – Tsalmantza and Bailey algorithms often fail to converge in presence of a large amount of missing data. The results show that the number of iterations chosen in Section 6.1 also match the complexity of this problem.

	New	Tsalmantza	Bailey
Data set χ^2_{fit}	0.107	0.088	0.094
Data set χ^2_{test}	1.064	2×10^5	8×10^{12}
Median χ^2_{test}	1.021	1.789	8×10^4
Ratio of observations having $\chi^2_{\text{test}} \geqslant 5$	0.014	0.33	0.81

Table 1. SDSS DR10Q data set fit and extrapolation chi-squares followed by per-observation median chi-squares of extrapolation and associated ratio of outliers.

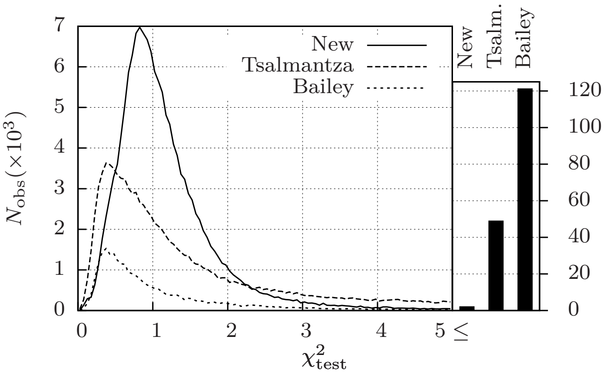

Figure 8. Distribution of the observation's χ^2_{test}. Right-hand part represents the number of QSOs ($\times 10^3$) having $\chi^2_{\text{test}} \geqslant 5$.

The initial test to be performed is similar to tests performed in Section 6.1 in the sense that PCA were retrieved for all algorithms using the region 5000–9000Å in observed wavelength while regions 4000–5000 and 9000–10 000Å are rejected and kept to assess the quality of the extrapolation. Fig. 7 shows an example of such a spectrum along with the fits by the various algorithms. The resulting SDSS DR10Q data set χ^2_{fit} and χ^2_{test}, as defined by equation (35), are given in Table 1.

We see that χ^2_{fit} is quite stable for all the algorithms while being a bit higher for our algorithm as expected. Now watching at the extreme differences of χ^2_{test} for the various algorithms and given the fact that we know it to be an estimator that is quite sensitive to outliers, we found relevant to see how these discrepancies are distributed among the observations. To this aim, we computed a χ^2_{test} per-observation whose distribution is summarized in Fig. 8.

Given the significant number of observations used in this test, we can already draw some general trends. Beside the fact that the other algorithms have a better χ^2_{fit}, they often fail to satisfactorily extrapolate the spectra. Indeed, for the Bailey algorithm, 81% of observations have $\chi^2_{\text{test}} \geqslant 5$ with a peak up to 9.59×10^{15} (median=8.24×10^4); the Tsalmantza algorithm has respectively 33% of observations having $\chi^2_{\text{test}} \geqslant 5$ and a peak up to 5.78×10^8 (median=1.789) while our new algorithm has only 1.4% of 'outliers' with a maximal peak of 172 (median=1.021). If the observations are individually compared, our algorithm has a lower χ^2_{test} in 90% of the time regarding Bailey algorithm and 68% of the time regarding Tsalmantza algorithm while in other cases differences remain quite moderate with a median $\Delta\chi^2_{\text{test}} \approx 0.3$. For completeness, we have to note that these huge discrepancies can be mainly attributed to the large amount of missing data (76%) within the resulting rest-frame DR10Q data set.

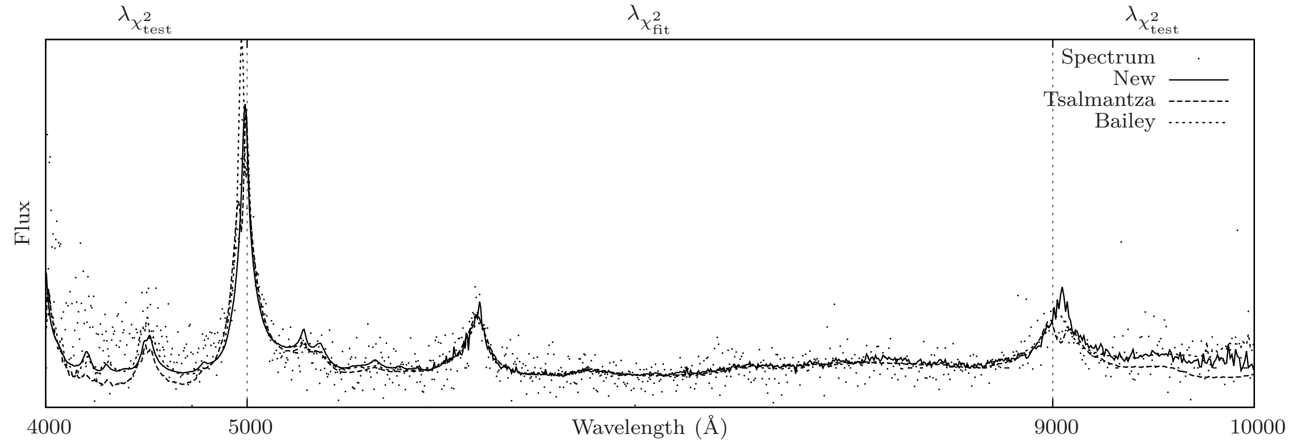

Figure 7. Example of SDSS DR10Q spectrum fits. $\lambda_{\chi^2_{\text{test}}}$ and $\lambda_{\chi^2_{\text{fit}}}$ denote regions used to compute, respectively, χ^2_{test} and χ^2_{fit} for this observation.

Another quality one would often desire is the ability to have the most general and representative set of principal components able to model unseen observations. In this optic, two tests were performed.

First, we split the initial DR10Q data set into two subsamples each of 74 025 spectra spanning from 4000 to 10 000Å (hereafter LS1 and LS2). From each of these subsamples, we extract ∼8600 spectra for which the rest-frame region from 1000 to 2000Å is entirely covered – and consequently 'without' missing data – and ∼62 000 spectra for which the mentioned region is only partially covered – and thus 'with' missing data. Note that according to the selected observed wavelengths and the previous definitions, spectra without missing data will correspond to those having $3 \leqslant z \leqslant 4$ and spectra with missing data to those having $1 \leqslant z \leqslant 9$.

The consistency and convergence of each algorithm were then tested by retrieving 10 principal components out of LS1 and LS2 data sets taken with and without missing data. Prior to discussing the results, we have to mention that the components may be swapped from data sets with missing data to the corresponding ones without missing data. This is easily explained by the fact that some patterns may be highlighted due to the uneven coverage of the wavelength range while being damped in the total variance if a full coverage is considered. Fig. 9 illustrates the results of this test regarding the first component of the data set without missing data and associated component with missing data. We see that in absence of missing data, all algorithms are consistent and succeed in converging towards the dominant eigenvector while in the case of missing data only our algorithm shows both a good consistency and a good convergence. More precisely, regarding our algorithm, the mean differences per-point between LS1 and LS2 are in $\mathcal{O}\left(10^{-6}\right)$ for both subsets with and without missing data. Differences noticed between subsets, that is mainly the general larger equivalent width and the stronger N Vλ1240Å emission line of the subset with missing data, are consistent with the underlying data set and make it able to model the larger variety of spectra coming from the larger redshift range. Let us mention, that no significant inconsistencies have been noticed up to the sixth component for LS1 and LS2 on both subsets for our algorithm. Concerning the Bailey's algorithm, we see, as already suggested in Section 6.1, that it fails to converge if a large amount of data is missing. Tsalmantza's implementation shows a better convergence but fails to correctly reproduce some emission lines (O VIλ1033, N Vλ1240, C IVλ1549 and C IIIλ1908Å) as well as the region ⩽1200Å in subsets with missing data. Similarly to Bailey's algorithm, analysis show that these are due to a convergence problem occurring on sparse data sets and arising from numerical instabilities.

In the second test performed, 10 principal components were retrieved from a set of 98 700 spectra spanning from 4000 to 10 000Å (hereafter LS3). We tested the fact that the principal components are the best at individually describing the underlying data variance by fitting the remaining 49 350 spectra (hereafter VS) using a subset of five components retrieved from LS3. That is, if the variance explained by each individual component is maximized then the fit of a smaller number of components to a similar data set should be minimal. In this case, if we consider computing χ^2 of VS data set as defined by equation (35) for each of the algorithms, we have respectively $\chi^2 = 0.372$ for the Bailey algorithm, $\chi^2 = 0.316$ for the Tsalmantza algorithm and $\chi^2 = 0.309$ for our algorithm, thus supporting our last quality criterion. Note again that if all components were considered, Bailey and Tsalmantza algorithms would have had a better resulting χ^2.

7 DISCUSSION

7.1 Convergence and uniqueness

At first glance it is surprising that the power iteration algorithm works at all, but in practice, it is easily demonstrated. Consider a diagonalizable square matrix $\mathbf{A}$ of size $(n \times n)$ having eigenvectors $\boldsymbol{p_1}, ..., \boldsymbol{p_n}$ and associated eigenvalues $\lambda_1, ..., \lambda_n$ where $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$. Since the eigenvectors are orthogonal between each other, we can write the starting vector of the power iteration algorithm as

$$\boldsymbol{u^{(0)}} = c_1\boldsymbol{p_1} + ... + c_n\boldsymbol{p_n}. \tag{36}$$

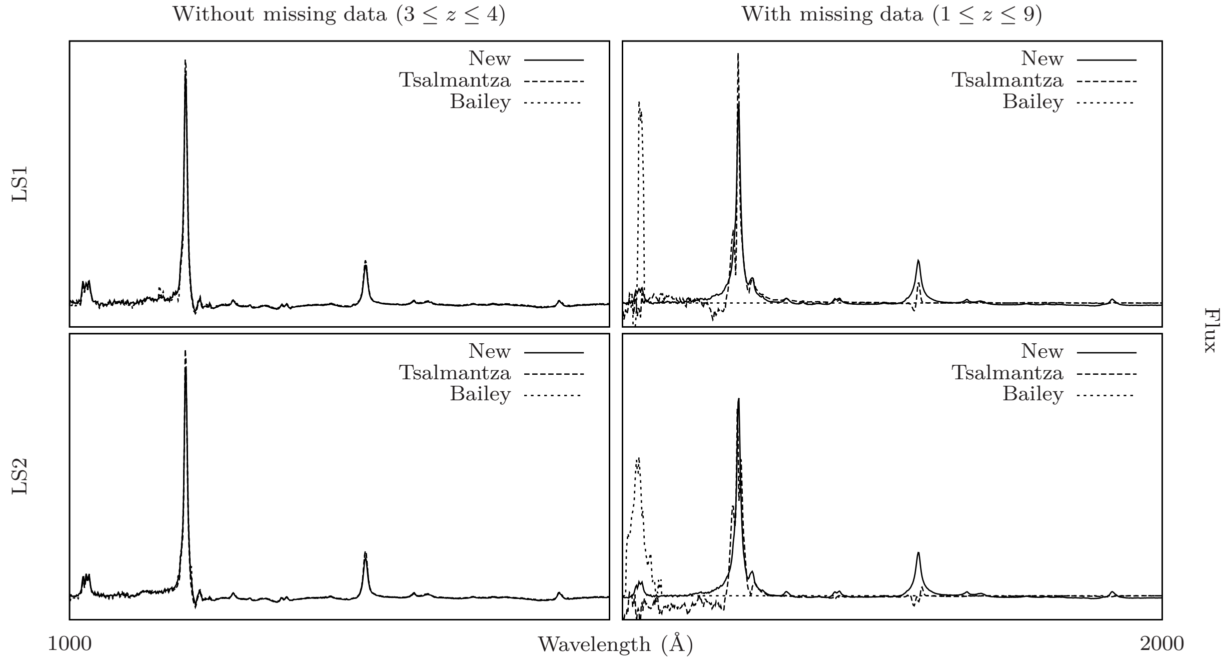

Figure 9. Principal components retrieved from two independent data sets (LS1 and LS2) coming from the DR10Q catalogue. The left-hand part corresponds to the dominant eigenvectors extracted from spectra having no missing data while the right-hand part are the associated components coming from spectra having missing fluxes.

Then we will have that the vector at iteration k given by

$$\begin{aligned} \boldsymbol{u^{(k)}} &= \mathbf{A}^k \sum_{i=1}^{n} c_i \boldsymbol{p_i} = \sum_{i=1}^{n} c_i \lambda_i^k \boldsymbol{p_i} \\ &= \lambda_1^k \left[c_1 \boldsymbol{p_1} + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1} \right)^k \boldsymbol{p_i} \right]. \end{aligned} \tag{37}$$

We see that $\boldsymbol{u^{(k)}}$ will converge to $\lambda_1^k c_1 \boldsymbol{p_1}$ as $k \to \infty$ under the following conditions: $c_1 \neq 0$, i.e. the starting vector has a nonzero component in the direction of the dominant eigenvector and $|\lambda_1| > |\lambda_2|$, i.e. the data set has only one dominant eigenvalue. Also note that, in this case, the rate of convergence to the dominant eigenvector will be principally given by λ_1/λ_2 and that in the case of $\lambda_1 = \lambda_2$, the uniqueness of the solution is not guaranteed as it depends on the starting vector $\boldsymbol{u^{(0)}}$. Refinement seen in Section 5.1 obeys to the same conditions. Additionally, the dominant eigenvalue of $(\mathbf{A} - d\mathbf{I})^{-1}$ will tend to ∞ as $d \to \lambda$, where $\mathbf{A}$ is the covariance matrix, d the current 'eigenvalue' (i.e. the Rayleigh quotient of $\mathbf{A}$ and $\boldsymbol{u}$, the current 'eigenvector') and λ a real eigenvalue of $\mathbf{A}$. This may thus lead to numerical instabilities that can strongly deteriorate the vector used in the next step of the Rayleigh quotient iteration.

That being said, failures against convergence can be easily checked. For example, a satisfactory solution should have $\mathbf{A}\boldsymbol{p} \approx \lambda\boldsymbol{p}$; and if not, the power iteration algorithm can be resumed with another starting vector $\boldsymbol{u^{(0)}}$. Under the condition that we have $\lambda_1 = \lambda_2$, we will have an infinite number of eigenvectors that are equally good at describing the data set variance, thus choosing one over another is irrelevant. Concretely, during the ~100 000 tests performed in the context of this paper, no such problems arose. Checks performed on eigenvectors from Section 6.1 and 6.2 show that the latter are orthogonal to a machine precision of $\mathcal{O}\left(10^{-16}\right)$ and that the weighted covariance matrix (as described by equation 26) is diagonalized with the same precision. Nevertheless, people wanting extreme reproducibility and/or secure convergence may still use the SVD in order to extract all eigenvectors from equation (26) at the expense of a lower flexibility (see Sections 7.2 and 7.3).

7.2 A priori eigenvectors

In some situations, one may already have an approximation of the wanted principal components corresponding to a given data set. Typical examples include principal components update according to new observations added to the data set or in a real case we encountered, the use of SDSS DR9Q eigenvectors to the DR10Q data set. These will constitute a priori eigenvectors that it would be regrettable not to use. The design of our algorithm allows us to easily take benefits of these a priori vectors. Instead of using a random vector, $\boldsymbol{u}^{(0)}$, in equation (28) one can straightforwardly substitute the known vectors to the random starting vector usually used. Doing so will typically decrease the number of iterations needed to converge towards the new vectors. Beware that vectors may be swapped between a priori eigenvectors and effective data set eigenvectors – as encountered in Section 6.2 – and consequently iterations needed to converge will be the same as if we had used a random starting vector. In this case, one can perform no power iteration and use only the refinement seen in Section 5.1 in order to converge to the nearest eigenvector.

7.3 Smoothing eigenvectors

It sometimes happens for some variables within eigenvectors to exhibit some sharp features that we know to be artefacts. These occur mainly in cases where we have noise that is comparable in amplitude to the data variance, data regions covered only by a few numbers of observations or data sets containing corrupted observations. Again the flexibility of our algorithm allows us to efficiently deal with these drawbacks. Suppose that we retrieved such a 'noisy' eigenpair $\langle \boldsymbol{p}, \lambda \rangle$ from a given data set whose covariance matrix is given by A, then, before removing the variance in the direction of the found component through equation (30), one can filter $\boldsymbol{p}$ thanks to any existing smoothing function. Assuming that the resulting vector, $\boldsymbol{p}'$, is near $\boldsymbol{p}$ regarding the norm of their difference, we can suppose that the variance accounted for by $\boldsymbol{p}'$ (i.e. its 'eigenvalue') is similar to the one of $\boldsymbol{p}$ and then subtract it along the direction of $\boldsymbol{p}'$ by

$$\mathsf{A}' = \mathsf{A} - \lambda \boldsymbol{p}' \otimes \boldsymbol{p}'. \tag{38}$$

Note that the kind of smoothing function to use is highly data-dependent and no rule of thumb exists. Nevertheless, a quite general and commonly used filter producing efficient results in the field of QSOs can be found in Savitzky & Golay (1964). Finally, let us note that, when applying a filter, the orthogonality of the components has to be manually checked and that they will obviously not diagonalize the covariance matrix anymore.

7.4 Scaling performance

PCA is often used in cases where we have a lot of observations and a reasonable number of variables, typically we have $N_{\rm obs} \gg N_{\rm var}$. We know, for example, the classical algorithm to require $\mathcal{O}\left({N_{\rm var}}^3\right)$ basic operations in order to solve for the eigenvectors of the covariance matrix and $\mathcal{O}\left({N_{\rm var}}^2 N_{\rm obs}\right)$ operations to build this matrix. In the following, we will compute the algorithmic complexity of the various explored algorithms in a similar way.

The Tsalmantza algorithm requires for each iteration (hereafter $N_{\rm iter}$) the solution of linear systems of equations for each of the E and M-step, respectively, in each observation and in each variable. We will then have that its algorithmic complexity is given by $\mathcal{O}\left(N_{\rm iter} {N_{\rm comp}}^3 \left(N_{\rm obs} + N_{\rm var}\right)\right)$. The Bailey algorithm will be identical except for the M-step that will be in $\mathcal{O}\left(N_{\rm var} N_{\rm obs} N_{\rm comp}\right)$ and can thus be discarded, giving $\mathcal{O}\left(N_{\rm iter} {N_{\rm comp}}^3 N_{\rm obs}\right)$. Finally, our algorithm mainly requires the building of the covariance matrix, $N_{\rm iter}$ matrix multiplications, potentially $N_{\rm refine}$ matrix inversions in order to refine the eigenvectors and a final single E-step similar to the one for the previous algorithm (discarded here), thus giving $\mathcal{O}\left({N_{\rm var}}^2 N_{\rm obs} + N_{\rm iter} {N_{\rm var}}^2 N_{\rm comp} + N_{\rm refine} {N_{\rm var}}^3\right)$.

We see that in case $N_{\rm obs} \gg N_{\rm var}$, our algorithm is much faster than the other ones as the computing time is mainly spent in the covariance matrix building. As an illustration, if we take data similar to those described in Section 6.1 with $N_{\rm obs}$ = 10 000, PCA retrieval takes ∼ 140s for the Bailey and Tsalmantza algorithms (with $N_{\rm iter} = 100$) and ∼ 3s for the new one (with $N_{\rm iter}$ =10 000) on a 2.4Ghz CPU.

8 CONCLUSIONS

We presented a new method for computing principal components based on data sets having noisy and/or missing data. The underlying ideas are intuitive and lead to a fast and flexible algorithm that is a generalization of the classical PCA algorithm. Unlike existing methods, based on lower-rank matrix approximations, the resulting principal components are not those that aim to explain at best the whole variance of a given data set but rather those that are the most suitable in identifying the most significant patterns out of the data set while explaining most of its variance. The main benefits of the current implementation are a better behaviour in presence of missing data as well as faster run times on data set having a large amount of observations. Privileged problems encompass data set extrapolation, patterns analysis and principal component usage over similar data sets. We assessed the algorithm performance on simulated data as well as on QSO spectra to which many applications are already foreseen in the field of the *Gaia* mission.

9 ACKNOWLEDGEMENTS

The author acknowledges support from the ESA PRODEX Programme 'Gaia-DPAC QSOs' and from the Belgian Federal Science Policy Office.

Funding for SDSS-III has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation and the U.S. Department of Energy Office of Science. The SDSS-III web site is http://www.sdss3.org/.

SDSS-III is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS-III Collaboration including the University of Arizona, the Brazilian Participation Group, Brookhaven National Laboratory, Carnegie Mellon University, University of Florida, the French Participation Group, the German Participation Group, Harvard University, the Instituto de Astrofisica de Canarias, the Michigan State/Notre Dame/JINA Participation Group, Johns Hopkins University, Lawrence Berkeley National Laboratory, Max Planck Institute for Astrophysics, Max Planck Institute for Extraterrestrial Physics, New Mexico State University, New York University, Ohio State University, Pennsylvania State University, University of Portsmouth, Princeton University, the Spanish Participation Group, University of Tokyo, University of Utah, Vanderbilt University, University of Virginia, University of Washington and Yale University.

REFERENCES

Bailer-Jones C.A.L. et al., 2013, A&A, 559, A74

Bailey S., 2012, PASP, 124, 1015

Beale E. M. L., Little R. J. A., 1975, J. R. Stat. Soc. B, 37, 129

Dempster A. P., Laird, N. M., Rubin D. B., 1977, J. R. Statist. Soc. B, 39, 1

Gabriel K.R., Zamir S., 1979, Technometrics, 21, 489

Golub G.H., Van Loan C.F., 1996, Matrix Computations, 3rd edn. The Johns Hopkins Univ. Press, London

Greenacre M.J., 1984, Theory and Applications of Correspondence Analysis. Academic Press, London

Hotelling H., 1933, J. Educ. Psychol., 24, 417

Jarvis M., Jain B., 2004, preprint(astro-ph/0412234)

Jolliffe I.T., 2002, Principal Component Analysis, 2nd edn. Springer, New York

Pâris I. et al., 2014, A&A, 563, A54

Parlett B.N., 1974, Math. Comput., 28, 679

Pearson K., 1901, Phil. Mag., 2, 559

Press W.H., Tuekolsky S.A., Vetterling W.T., Flannery B.P., 2002, Numerical recipes in C++: The Art of Scientific Computing, 2nd edn. Cambridge Univ. Press, New York

Roweis S., 1997, CNS Technical Report CNS-TR-97-02, EM Algorithm for PCA and SPCA. Available at: http://cs.nyu.edu/~roweis/papers/empca.pdf

Savitzky A., Golay M.J.E., 1964, Anal. Chem., 36, 1627

Schlens J., 2009, A Tutorial on Principal Component Analysis. Available at: http://www.snl.salk.edu/~shlens/pca.pdf

Srebro N., Jaakkola T., 2003, Proc. 20th ICML, Weighted Low Rank Approximation, AAAI Press, Menlo Park, p. 720

Tipping M.E., Bishop C.M., 1999, J. R. Stat. Soc. B, 21, 611

Tsalmantza P., Hogg D.W., 2012, ApJ, 753, 122

Wentzell P.D., Andrews D.T., Hamilton D.C., Faber K., Kowalski B.R., 1997, J. Chemometr., 11, 339

Yip C.W. et al., 2004, AJ, 128, 2603

APPENDIX A: MATLAB/OCTAVE SOURCE CODE

```
%  Function designed to find the weighted PCA of a
% given data set such that the resulting principal
% components diagonalize the associated weighted
% covariance matrix and that the associated principal
% coefficients are the linear coefficients of these
% principal components that best match the data set
% in a least-squares sense.
%
% INPUT:
%  X       : data set matrix from which the weighted
%            mean was already subtracted. Rows
%            correspond to variables, columns to
%            observations.
%  W       : Weight matrix (same size as X).
%  ncomp   : Number of principal components to
%            retrieve.
%  niter   : Number of iterations (allows
%            convergence to the dominant principal
%            component)
%  nrefine : Number of refinments (allows refinement
%            of the dominant principal component)
%  xi      : Regularization factor controlling the
%            influence of rare variables
% OUTPUT:
%  P       : Principal components arranged in columns
%  C       : Principal coefficients such that X = P*C
%            in a least-squares sense
% AUTHOR:
%  Ludovic Delchambre
%  Extragalactic Astrophysics and Space Observations,
%  Institute of Astrophysics and Geophysics,
%  University of Liege, Belgium
function [P, C] = WPCA(X, W, ncomp, niter, nrefine, xi)
 nvar = size(X,1); nobs = size(X,2);
 P=zeros(nvar,ncomp);C=zeros(ncomp,nobs);
 ws = sum(W,2);
 covar = ((ws * ws’).^xi).*((X.*W)*(X.*W)’)./(W*W’);
 covar(isnan(covar)) = 0;
 for i=1:ncomp
  u = ones(nvar,1) / nvar;
  for j=1:niter
   u = covar*u;
   d = sqrt(u’*u);
   u = u / d;
  end
  d = u’*covar*u;
  for j=1:nrefine
   u = inv(covar - d * eye(nvar)) * u;
   u = u / sqrt(u’*u);
   d = u’*covar*u;
  end
  covar = covar - u*d*u’;
  P(:,i) = u;
 end
 for i=1:nobs
  w = diag(W(:,i)).^2;
  C(:,i) = inv(P’*w*P) * (P’*w*X(:,i));
 end
end
```