PUBLICATIONS OF THE ASTRONOMICAL SOCIETY OF THE PACIFIC, **124**:1015–1023, 2012 September

Principal Component Analysis with Noisy and/or Missing Data

STEPHEN BAILEY

Physics Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA, 94720

Received 2012 July 10; accepted 2012 August 17; published 2012 September 19

ABSTRACT. We present a method for performing principal component analysis (PCA) on noisy datasets with missing values. Estimates of the measurement error are used to weight the input data such that the resulting eigenvectors, when compared to classic PCA, are more sensitive to the true underlying signal variations rather than being pulled by heteroskedastic measurement noise. Missing data are simply limiting cases of weight $= 0$. The underlying algorithm is a noise weighted expectation maximization (EM) PCA, which has additional benefits of implementation speed and flexibility for smoothing eigenvectors to reduce the noise contribution. We present applications of this method on simulated data and QSO spectra from the Sloan Digital Sky Survey (SDSS).

Online material: color figures

1. INTRODUCTION

Principal component analysis (PCA) is a powerful and widely used technique to analyze data by forming a custom set of "principal component" eigenvectors that are optimized to describe the most data variance with the fewest number of components (Pearson 1901; Hotelling 1933; Jolliffe 2002). With the full set of eigenvectors, the data may be reproduced exactly; i.e., PCA is a transformation that can lend insight by identifying which variations in a complex dataset are most significant and how they are correlated. Alternately, since the eigenvectors are optimized and sorted by their ability to describe variance in the data, PCA may be used to simplify a complex dataset into a few eigenvectors plus coefficients, under the approximation that higher-order eigenvectors are predominantly describing fine-tuned noise or otherwise less important features of the data. Example applications within astronomy include classifying spectra by fitting them to PCA templates (Pâris et al. 2011; Connolly & Szalay 1999), describing *Hubble Space Telescope* point spread function variations (Jee et al. 2007), and reducing the dimensionality of cosmic microwave background map data prior to analysis (Bond 1995).

A limitation of classic PCA is that it does not distinguish between variance due to measurement noise and variance due to genuine underlying signal variations. Even when an estimate of the measurement variance is available, this information is not used when constructing the eigenvectors, e.g., by deweighting noisy data.

A second limitation of classic PCA is the case of missing data. In some applications, certain observations may be missing some variables, and the standard formulas for constructing the eigenvectors do not apply. For example, within astronomy, observed spectra do not cover the same rest-frame wavelengths of objects at different redshifts, and some wavelength bins may be masked due to bright sky lines or cosmic ray contamination. Missing data are an extreme case of noisy data, where missing data are equivalent to data with infinite measurement variance.

This work describes a PCA framework which incorporates estimates of measurement variance while solving for the principal components. This optimizes the eigenvectors to describe the true underlying signal variations without being unduly affected by known measurement noise. Code which implements this algorithm is available at https://github.com/sbailey/empca.

Jolliffe (2002) § 13.6 and § 14.2 review prior work on PCA with missing data and incorporating weights into PCA. Most prior work focuses on the identification and removal of outlier data, interpolation over missing data, or special cases such as when the weights can be factorized into independent per-observation and per-variable weights. Gabriel & Zamir (1979), Wentzell et al. (1997), Tipping & Bishop (1999), and Srebro & Jaakkola (2003) present iterative solutions for the case of general weights, though none of these find the true PCA solution with orthogonal eigenvectors optimally ranked by their ability to describe the data variance. Instead, they find an unsorted set of non-orthogonal vectors, which as a set are optimized to describe the data variance, but individually are not the optimal linear combinations to describe the most variance with the fewest vectors. Their methods are sufficient for weighted lower-rank matrix approximation, but they lack the potential data insight from optimally combining and sorting the eigenvectors to see which components contribute the most variation.

Within the astronomy literature, Connolly & Szalay (1999) discuss how to interpolate over missing data and use PCA eigenspectra to *fit* noisy and/or missing data, but they do not address the case of how to *generate* eigenspectra from noisy but nonmissing data. Blanton & Roweis (2007) generate template

spectra from noisy and missing data using non-negative matrix factorization (NMF). This method is similar to PCA with the constraint that the template spectra are strictly positive, while not requiring the templates to be orthonormal. Tsalmantza & Hogg (2012) present a more general "heteroskedastic matrix factorization" approach to study Sloan Digital Sky Survey (SDSS) spectra while properly accounting for measurement noise. Their underlying goal is similar to this work, though with an algorithmically different implementation.

The methods presented here directly solve for the PCA eigenvectors with an iterative solution based upon expectation maximization (EM) PCA (EMPCA). Roweis (1997) describes an unweighted version of EMPCA, including a method for interpolating missing data, but he does not address the issue of deweighting noisy data. We also take advantage of the iterative nature of the solution to bring unique extensions to PCA, such as noise-filtering the eigenvectors during the solution.

The approach taken here is fundamentally pragmatic. For example, if one is interested in generating eigenvectors to describe 99% of the signal variance, it likely does not matter if an iterative algorithm has "only" converged at the level of 10^{-5} even if the machine precision is formally 10^{-15}. We discuss some of the limitations of weighted EMPCA in § 8, but ultimately we find that these issues are not limiting factors for practical applications.

This work was originally developed for PCA of astronomical spectra, and examples are given in that context. It should be noted, however, that these methods are generally applicable to any PCA application with noisy and/or missing data—nothing in the underlying methodology is specific to astronomical spectra.

2. NOTATION

This paper uses the following notation: Vectors use boldface italic, $\boldsymbol{x}$, while x_i represents the scalar element i of vector $\boldsymbol{x}$. For sets of vectors, $\boldsymbol{x}_j$ represent vector number j (not element j of vector $\boldsymbol{x}$). Matrices are in non-italic upper-case boldface, $\mathbf{X}$. To denote vectors formed by selected columns or rows of a matrix, we use $\boldsymbol{X}_j^{\rm col}$ for the vector formed from column j of matrix $\mathbf{X}$ and $\boldsymbol{X}_i^{\rm row}$ for the vector formed from row i of matrix $\mathbf{X}$. The scalar element at row i column j of matrix $\mathbf{X}$ is $\mathbf{X}_{ij}$.

For reference, we summarize the names of the primary variables here: $\mathbf{X}$ is the data matrix with $n_{\rm var}$ rows of variables and $n_{\rm obs}$ columns of observations. $\mathbf{P}$ is the PCA eigenvector matrix with $n_{\rm var}$ rows of variables and $n_{\rm vec}$ columns of eigenvectors; $\boldsymbol{\phi}$ is a single eigenvector. These eigenvectors may fit the data using a matrix of coefficients $\mathbf{C}$, where $\mathbf{C}_{kj}$ is the contribution of eigenvector k to observation j. Indices i, j, and k index variables, observations, and eigenvectors, respectively. $\boldsymbol{X}$ is the vector formed by stacking the columns of matrix $\mathbf{X}$. The *measurement* covariance of dataset $\boldsymbol{X}$ is $\mathbf{V}$, while $\mathbf{W}$ is the weights matrix for dataset $\mathbf{X}$ for the case of independent measurement noise such that $\mathbf{W}$ has the same dimensions as $\mathbf{X}$.

3. CLASSIC PCA

For an accessible tutorial on classic PCA, see Schlens (2009). A much more complete treatment is found in Jolliffe (2002). Algorithmically, the steps are simple: The principal components $\{\boldsymbol{\phi}_k\}$ of a dataset are simply the eigenvectors of the covariance of that dataset, sorted by their descending eigenvalues. A new observation $\boldsymbol{y}$ may be approximated as

$$\boldsymbol{y} = \boldsymbol{\mu} + \sum_k c_k \boldsymbol{\phi}_k, \tag{1}$$

where $\boldsymbol{\mu}$ is the mean of the initial dataset and c_i is the reconstruction coefficient for eigenvector $\boldsymbol{\phi}_i$. For the rest of this article, we will assume that $\boldsymbol{\mu}$ has already been subtracted from the data, i.e., $\boldsymbol{y} \leftarrow (\boldsymbol{y} - \boldsymbol{\mu})$.

To find a particular coefficient $c_{k'}$, take the dot product of both sides with $\boldsymbol{\phi}_{k'}$, noting that because of the eigenvector orthogonality, $\boldsymbol{\phi}_k \cdot \boldsymbol{\phi}_{k'} = \delta_{kk'}$ (Kroeneker-delta),

$$\boldsymbol{y} \cdot \boldsymbol{\phi}_{k'} = \sum_k c_k \boldsymbol{\phi}_k \cdot \boldsymbol{\phi}_{k'} \tag{2}$$

$$= \sum_k c_k \delta_{kk'} \tag{3}$$

$$= c_{k'}. \tag{4}$$

Note that the neither the solution of $\{\boldsymbol{\phi}_k\}$ nor $\{c_k\}$ makes use of any noise estimates or weights for the data. As such, classic PCA solves the minimization problem

$$\chi^2 = \sum_{i,j} [\mathbf{X} - \mathbf{PC}]_{ij}^2, \tag{5}$$

where $\mathbf{X}$ is a dataset matrix whose columns are observations and rows are variables, $\mathbf{P}$ is a matrix whose columns are the principal components $\{\boldsymbol{\phi}_k\}$ to find, and $\mathbf{C}$ is a matrix of coefficients to fit $\mathbf{X}$ using $\mathbf{P}$. For clarity, the dimensions of these matrices are: $\mathbf{X}[n_{\rm var}, n_{\rm obs}]$, $\mathbf{P}[n_{\rm var}, n_{\rm vec}]$, and $\mathbf{C}[n_{\rm vec}, n_{\rm obs}]$, where $n_{\rm obs}$, $n_{\rm var}$, and $n_{\rm vec}$ are the number of observations, variables, and eigenvectors, respectively. For example, when performing PCA on spectra, $n_{\rm obs}$ is the number of spectra, $n_{\rm var}$ is the number of wavelength bins per spectrum, and $n_{\rm vec}$ is the number of eigenvectors used to describe the data and may be smaller than the total number of possible eigenvectors.

4. ADDING WEIGHTS TO PCA

The goal of this work is to solve for the eigenvectors $\mathbf{P}$ while incorporating a weights matrix $\mathbf{W}$ on the dataset $\mathbf{X}$:

$$\chi^2 = \sum_{i,j} \mathbf{W}_{ij} [\mathbf{X} - \mathbf{PC}]_{ij}^2. \tag{6}$$

We also describe the more general cases of per-observation covariances $\mathbf{V}_j$:

$$\chi^2 = \sum_{\text{obs}j} (\boldsymbol{X}_j^{\text{col}} - \mathbf{P}\boldsymbol{C}_j^{\text{col}})^T \mathbf{V}_j^{-1} (\boldsymbol{X}_j^{\text{col}} - \mathbf{P}\boldsymbol{C}_j^{\text{col}}), \quad (7)$$

where we have used the notation that $\boldsymbol{X}_j^{\text{col}}$ is the vector formed from the jth column of the matrix $\mathbf{X}$; $\boldsymbol{C}_j^{\text{col}}$ is described similarly. In the most general case, there is covariance $\mathbf{V}$ between all variables of all observations, i.e., we seek to minimize

$$\chi^2 = (\boldsymbol{X} - [\mathbf{P}]\boldsymbol{C})^T \mathbf{V}^{-1} (\boldsymbol{X} - [\mathbf{P}]\boldsymbol{C}), \quad (8)$$

where $\boldsymbol{X}$ and $\boldsymbol{C}$ are the vectors formed by concatenating all columns of $\mathbf{X}$ and $\mathbf{C}$, and $[\mathbf{P}]$ is the matrix formed by stacking $\mathbf{P}$ n_{obs} times.

This allows one to incorporate error estimates on heteroskedastic data such that particularly noisy data does not unduly influence the solution. We will solve this problem using an iterative method known within the statistics community as "expectation maximization."

5. WEIGHTED EXPECTATION MAXIMIZATION PCA

5.1. Expectation Maximization PCA

EM is an iterative technique for solving parameters to maximize a likelihood function for models with unknown hidden (or latent) variables (Dempster, Laird, & Rubin 1977). Each iteration involves two steps: finding the expectation value of the hidden variables given the current model (E-step), and then modifying the model parameters to maximize the fit likelihood given the estimates of the hidden variables (M-step).

As applied to PCA, the parameters to solve are the eigenvectors, the latent variables are the coefficients $\{c\}$ for fitting the data using those eigenvectors, and the likelihood is the ability of the eigenvectors to describe the data. To solve the single most significant eigenvector, start with a random vector $\boldsymbol{\phi}$ of length n_{var}. For each observation $\boldsymbol{x}_j$, solve for the coefficient $c_j = \boldsymbol{x}_j \cdot \boldsymbol{\phi}$ that best fits that observation using $\boldsymbol{\phi}$. Then using those coefficients, update $\boldsymbol{\phi}$ to find the vector that best fits the data given those coefficients: $\boldsymbol{\phi} \leftarrow \sum_j c_j \boldsymbol{x}_j / \sum_j c_j^2$. Then, normalize $\boldsymbol{\phi}$ to unit length and iterate the solutions to $\{c\}$ and $\boldsymbol{\phi}$ until converged. These steps are summarized below:

1. $\boldsymbol{\phi} \leftarrow$ random vector of length n_{var}
2. repeat the following until converged:
   (a) For each observation $\boldsymbol{x}_j$: $c_j \leftarrow \boldsymbol{x}_j \cdot \boldsymbol{\phi}$ (E-step)
   (b) $\boldsymbol{\phi} \leftarrow \sum_j c_j \boldsymbol{x}_j / \sum_j c_j^2$ (M-step)
   (c) $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} / |\boldsymbol{\phi}|$ (Renormalize)
3. return $\boldsymbol{\phi}$

This generates a vector $\boldsymbol{\phi}$ which is the dominant PCA eigenvector of the dataset $\mathbf{X}$, where the observations $\boldsymbol{x}_j$ are the columns of $\mathbf{X}$. The expectation step finds the coefficients $\{c_j\}$ which best fit $\mathbf{X}$ using $\boldsymbol{\phi}$ (see eqs. [2]–[4]). The likelihood maximization step then uses those coefficients to update $\boldsymbol{\phi}$ to minimize

$$\chi^2 = \sum_{\text{var}i,\text{obs}j} (\mathbf{X}_{ij} - c_j \phi_i)^2. \quad (9)$$

In practice, the normalization $\sum_j c_j^2$ in the M-step is unnecessary since $\boldsymbol{\phi}$ is renormalized to unit length after every iteration.

At first glance, it can be surprising that this algorithm works at all. Its primary enabling feature is that, at each iteration, the coefficients $\{c_j\}$ and vector $\boldsymbol{\phi}$ minimize the χ^2 better than the previous iteration. The χ^2 of equation (9) has a single minimum (Srebro & Jaakkola 2003); thus, when any minimum is found, it is the true global minimum. It is possible, however, to also have saddle points to which the EMPCA algorithm could converge from particular starting points. It is easy to test solutions for being at a saddle point and restart the iterations, as needed.

The specific convergence criteria are application specific. One pragmatic option is that the eigenvector itself is changing slowly, i.e., $|\Delta\boldsymbol{\phi}| < \epsilon$. Alternately, one could require that the change in likelihood (or $\Delta\chi^2$) from one iteration to the next is below some threshold. Convergence and uniqueness will be discussed in § 8.1 and § 8.2. For now, we simply note that many PCA applications are interested in describing 95% or 99% of the data variance, and the above algorithm typically converges very quickly for this level of precision, even for cases where the formal computational machine convergence may require many iterations.

To find additional eigenvectors, subtract the projection of $\boldsymbol{\phi}$ from $\mathbf{X}$ and repeat the algorithm. Continue this procedure until enough eigenvectors have been solved that the remaining variance is consistent with the expected noise of the data, or until enough eigenvectors exist to approximate the data with the desired fidelity. If only a few eigenvectors are needed for a large dataset, this algorithm can be much faster than classic PCA, which requires solving for all eigenvectors whether or not they are needed. Scaling performance will be discussed further in § 8.4.

5.2. EMPCA with per-Observation Weights

The above algorithm treats all data equally when solving for the eigenvectors and thus is equivalent to classic PCA. If all data are of approximately equal quality, then this is fine; but, if some data have considerably larger measurement noise, they can unduly influence the solution. In these cases, high signal-to-noise data should receive greater weight than low signal-to-noise data. This is conceptually equivalent to the difference between a weighted and unweighted mean.

In some applications, it is sufficient to have a single weight per observation so that all variables within an observation are equally weighted, but different observations are weighted more or less than others. In this case, EMPCA can be extended with per-observation weights w_i. The observations $\mathbf{X}$ should have their weighted mean subtracted, and the likelihood maximization step (M-step) is replaced with

$$\phi \leftarrow \sum_j w_j c_j \boldsymbol{x}_j. \tag{10}$$

The normalization denominator has been dropped because we re-normalize ϕ to unit length every iteration.

5.3. EMPCA with per-Variable Weights

If each variable for each observation has a different weight, the situation becomes more complicated since we cannot use simple dot products to derive the coefficients $\{c_j\}$. Instead, one must solve a set of linear equations for $\{c_j\}$. Similarly, the likelihood maximization step must solve a set of linear equations to update ϕ instead of just performing a simple sum. The weighted EMPCA algorithm now starts with a *set* of random orthonormal vectors $\{\phi_k\}$ and iterates through the following steps:

1. for each observation $\boldsymbol{x}_j$, solve coefficients c_{kj}:

$$\boldsymbol{x}_j = \sum_k c_{kj} \phi_k; \tag{11}$$

2. given $\{c_{kj}\}$, solve each ϕ_k one-by-one for k in $1..n_{\rm vec}$:

$$\boldsymbol{x}_j - \sum_{k'<k} c_{k'j} \phi_{k'} = c_{kj} \phi_k. \tag{12}$$

Both of the above steps can be solved using weights on $\boldsymbol{x}_j$, thus achieving the goals of properly weighting the data while solving for the coefficients $\{c_{kj}\}$ and eigenvectors ϕ_k. Implementation details will be described in the following two subsections, where we will return to using matrix notation.

5.3.1. *Notes on solving* $\{c_{kj}\} = \mathbf{C}$

In equation (11), the ϕ_k vectors are fixed and one solves the coefficients c_{kj} with a separate set of equations for each observation $\boldsymbol{x}_j$. Written in matrix form, $\mathbf{X} = \mathbf{PC}$ can be solved for each independent observation column j of $\mathbf{X}$ and $\mathbf{C}$:

$$\boldsymbol{X}_j^{\rm col} = \mathbf{P}\boldsymbol{C}_j^{\rm col} + \text{noise}. \tag{13}$$

Equation (13) is illustrated in Figure 1.

Solving equation (13) for $\boldsymbol{C}_j^{\rm col}$ with noise-weighting by measurement covariance $\mathbf{V}_j$ is a straight-forward linear least-squares problem, which may be solved with singular value decomposition (SVD), QR factorization, conjugate gradients, or other methods. For example, using the method of "normal equations"[1] and the shorthand $\boldsymbol{x} = \boldsymbol{X}_j^{\rm col}$ and $c = \boldsymbol{C}_j^{\rm col}$:

[1] Note that this method is mathematically correct but numerically unstable. It is included here for illustration, but the actual calculation should use one of the other methods (Press et al. 2002).

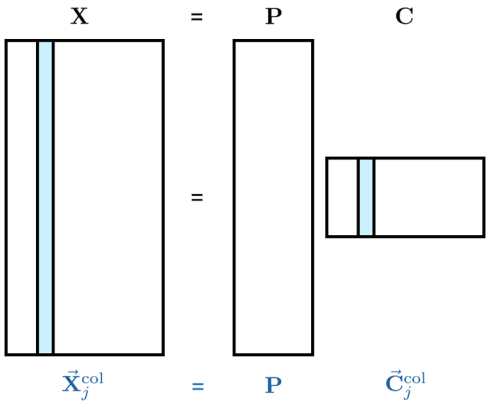

FIG. 1.—Illustration of solving **C** one column at a time, i.e. equation. (13). See the electronic edition of the *PASP* for a color version of this figure.

$$\boldsymbol{x} = \mathbf{P}\boldsymbol{c}; \tag{14}$$

$$\mathbf{V}^{-1}\boldsymbol{x} = \mathbf{V}^{-1}\mathbf{P}\boldsymbol{c}; \tag{15}$$

$$\mathbf{P}^T\mathbf{V}^{-1}\boldsymbol{x} = (\mathbf{P}^T\mathbf{V}^{-1}\mathbf{P})\boldsymbol{c}; \tag{16}$$

$$(\mathbf{P}^T\mathbf{V}^{-1}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{V}^{-1}\boldsymbol{x} = \boldsymbol{c}. \tag{17}$$

If the noise is independent between variables, the inverse covariance $\mathbf{V}^{-1}$ is just a diagonal matrix of weights $\boldsymbol{W}_j^{\rm col}$. Note that the covariance here is the estimated *measurement* covariance, not the total dataset variance—the goal is to weight the observations by the estimated *measurement* variance so that noisy observations do not unduly affect the solution, while allowing PCA to describe the remaining *signal* variance.

In the more general case of measurement covariance between different observations, one cannot solve equation (13) for each column of $\mathbf{X}$ independently. Instead, solve $\boldsymbol{X} = [\mathbf{P}]\boldsymbol{C}$ with the full covariance matrix $\mathbf{V}$ of $\boldsymbol{X}$, where $[\mathbf{P}]$ is the matrix formed by stacking $\mathbf{P}$ $n_{\rm obs}$ times, and $\boldsymbol{X}$ and $\boldsymbol{C}$ are the vectors formed by stacking all the columns of the matrices $\mathbf{X}$ and $\mathbf{C}$. This requires the solution of a single $(n_{\rm obs} \cdot n_{\rm vec}) \times (n_{\rm obs} \cdot n_{\rm vec})$ matrix rather than $n_{\rm obs}$ solutions of $n_{\rm vec} \times n_{\rm vec}$ matrices. If the individual observations are uncorrelated, it is computationally advantageous to use this non-correlation to solve multiple smaller matrices rather than one large one.

5.3.2. *Notes on solving* $\{\phi_k\} = \mathbf{P}$

In the second step of each iteration (eq. [12]), we use the fixed coefficients $\mathbf{C}$ (dimensions $n_{\rm vec} \times n_{\rm obs}$) and solve for the eigenvectors $\mathbf{P}$ (dimensions $n_{\rm var} \times n_{\rm vec}$). We solve the eigenvectors one-by-one to maximize the power in each eigenvector before solving the next. Selecting the kth eigenvector uses the kth column of $\mathbf{P}$ and the kth row of $\mathbf{C}$:

$$\mathbf{X} = \boldsymbol{P}_k^{\rm col} \otimes \boldsymbol{C}_k^{\rm row}, \tag{19}$$

where $\otimes$ signifies an outer product. If the variables (rows) of $\mathbf{X}$ are independent, then we can solve for a single element of $\boldsymbol{P}_k^{\rm col}$ at a time:

$$\boldsymbol{X}_i^{\rm row} = \mathbf{P}_{ik} \boldsymbol{C}_k^{\rm row}. \tag{20}$$

This is illustrated in Figure 2. With independent weights $\boldsymbol{W}_i^{\rm row}$ on the data $\boldsymbol{X}_i^{\rm row}$, we solve variable i of eigenvector k with:

$$\mathbf{P}_{ik} = \frac{\sum_j \mathbf{W}_{ij} \mathbf{X}_{ij} \mathbf{C}_{ik}}{\sum_j \mathbf{W}_{ij} \mathbf{C}_{ik} \mathbf{C}_{ik}}. \tag{21}$$

As with § 5.3.1, if there are measurement covariances between the data, equation (19) may be expanded to solve for all elements of $\boldsymbol{P}_k^{\rm col}$ simultaneously, using the full measurement covariance matrix of $\mathbf{X}$.

After solving for $\boldsymbol{P}_k^{\rm col}$, subtract its projection from the data:

$$\mathbf{X} \leftarrow \mathbf{X} - \boldsymbol{P}_k^{\rm col} \otimes \boldsymbol{C}_k^{\rm row}. \tag{22}$$

This removes any variation of the data in the direction of $\boldsymbol{P}_k^{\rm col}$ so that additional eigenvectors will be orthogonal to the prior ones.[2] Then, repeat the procedure to solve for the next eigenvector $k+1$.

6. EXTENSIONS OF WEIGHTED EMPCA

The flexibility of the iterative EMPCA solution allows for a number of powerful extensions to PCA, in addition to noise weighting. We describe a few of these here.

6.1. Smoothed EMPCA

If the length scale of the underlying signal eigenvectors is larger than that of the noise, it may be advantageous to smooth the eigenvectors to remove remaining noise effects. The iterative nature of EMPCA allows smoothing of the eigenvectors at each step to remove the high frequency noise. This generates the optimal smooth eigenvectors by construction, rather than by smoothing noisy eigenvectors afterward. This will be shown in the examples in § 7. Alternately, one can include a prior smoothing or a regularization term when solving for the principal components $\mathbf{P}$. That approach, however, requires solving equation (19) (plus a regularization term) for all elements of $\boldsymbol{P}_k^{\rm col}$ simultaneously instead of using the numerically much faster equation (21) for the case of diagonal measurement covariance.

[2] Note that this approach is potentially susceptible to build up of machine rounding errors and should be checked explicitly when using EMPCA for solving a large number of eigenvectors.

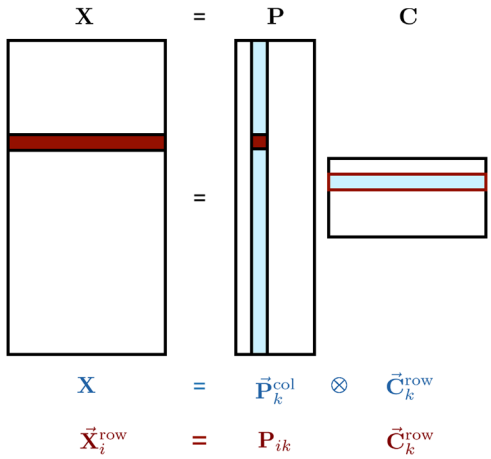

FIG. 2.—Illustration of solving $\mathbf{P}$ one element at a time, i.e., equation (20). See the electronic edition of the *PASP* for a color version of this figure.

6.2. A priori Eigenvectors

In some applications, one has a few *a priori* template vectors to include in the fit, e.g., from some physical model. The goal is to find additional template vectors which are to be combined with the *a priori* vectors for the best fit of the data. Due to noise weighting and the potential non-orthogonality of the *a priori* vectors, the best fit is a joint fit; one cannot simply fit the *a priori* vectors and remove their components before proceeding with finding the other unknown vectors.

This case can be incorporated into EMPCA by including the *a priori* vectors in the starting vectors $\mathbf{P}$ and simply keeping them fixed with each iteration rather than updating them. In each iteration, the *coefficients* for the *a priori* vectors are updated, but not the vectors themselves.

7. EXAMPLES

7.1. Toy Data

Figure 3 shows example noisy data used to test weighted EMPCA. One hundred data vectors were generated using three orthonormal sine functions as input, with random amplitudes drawn from Gaussian distributions. The lower frequency sine waves were given larger Gaussian sigmas such that they contribute more signal variance to the data. Gaussian random noise was added, with 10% of the data vectors receiving 25 times more noise from $[0, \pi/2]$ and 5 times more noise from $[\pi/2, 2\pi]$. For weighted EMPCA, weights were assigned as $1/\sigma^2$, where σ is the per-observation, per-variable Gaussian sigma of the added noise (not the sigma of the underlying signal). A final dataset was created where a contiguous 10% of each observation was set to have weight $= 0$ to create regions of missing data. As a crosscheck that this was applied correctly, the data in the regions with weight $= 0$ were set to a constant value of 1000—if these

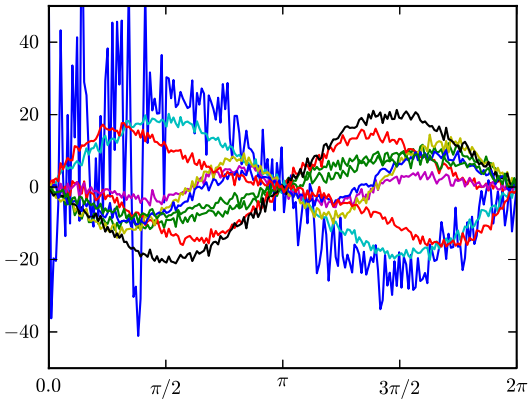

FIG. 3.—Example of noisy data used to test weighted EMPCA. See the electronic edition of the *PASP* for a color version of this figure.

data are not correctly ignored by the algorithm, they will have a large effect on the extracted eigenvectors.

Figure 4 shows the results of applying classic PCA and weighted EMPCA to these data. *Upper left*: Classic PCA applied to the noiseless data recovers the input eigenvectors, slightly rotated to form the best ranked eigenvectors for describing the data variance. *Upper right*: EMPCA applied to the same noiseless data recovers the same input eigenvectors. *Middle left*: When classic PCA is applied to the noisy data, the highest order eigenvector is dominated by the noise, and the effects of the non-uniform noise are clearly evident as increased noise from $[0, \pi/2]$. *Middle right*: Weighted EMPCA is much more robust to the noisy data, extracting results close to the original eigenvectors. The highest order eigenvector is still affected by the noise, which is a reflection that the noise does contribute power to the data variance. However, the extra-noisy region from $[0, \pi/2]$ is not affected more than the region from $[\pi/2, 2\pi]$

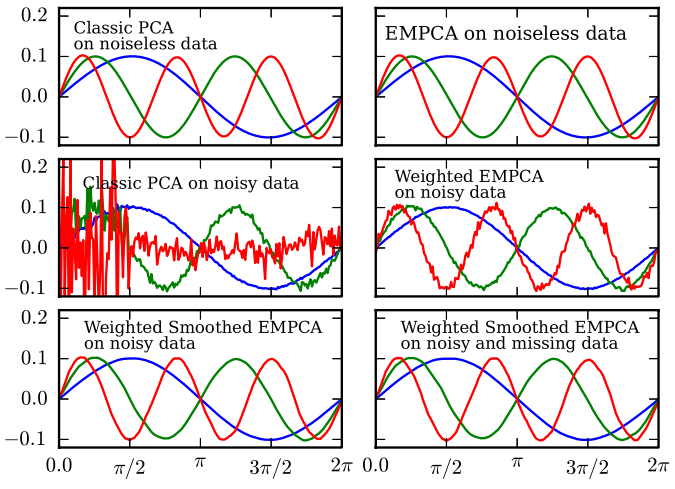

FIG. 4.—Examples of classic PCA and EMPCA applied to noiseless, noisy, and missing data. See the electronic edition of the *PASP* for a color version of this figure.

due to the proper deweighting of the noisy data. *Lower left*: Smoothed, weighted EMPCA is almost completely effective at extracting the original eigenvectors with minimal impact from the noise. *Lower right*: Even when 10% of every observation is missing, smoothed, weighted EMPCA is effective at extracting the underlying eigenvectors. All eigenvectors for all methods are orthogonal at the level of $\mathcal{O}(10^{-17})$.

7.2. QSO Data

Figure 5 shows the results of applying classic PCA and weighted EMPCA to QSO spectra from the SDSS Data Release 7 (Abazajian et al. 2009), using the QSO redshift catalog of Hewett & Wild (2010). Five hundred spectra of QSOs with redshift $2.0 < z < 2.1$ were randomly selected and trimmed to $1340 < \lambda < 1620$ Å to show the Si IV and C IV emission features. Spectra with more than half of the pixels masked were discarded. Each spectrum was normalized to median $[\mathrm{flux}(1440 < \lambda < 1500\ \text{Å})] = 1$ and the weighted mean of all normalized spectra was subtracted. The left panel of Figure 5 plots examples of high, median, and low signal-to-noise spectra and a broad absorption line (BAL) QSO from this sample. Approximately 2% of the spectral bins have been flagged with a bad-data mask, e.g., due to cosmic rays, poor sky subtraction, or the presence of non-QSO narrow absorption features from the intergalactic medium. These are treated as missing data with weight $= 0$. The goal of weighted EMPCA is to deweight properly the noisy spectra such that the resulting eigenvectors are predominantly describing the underlying signal variations and not just measurement noise. Weights are $1/\sigma_{ij}^2$ where σ_{ij} is the SDSS pipeline estimated measurement noise for wavelength bin i of

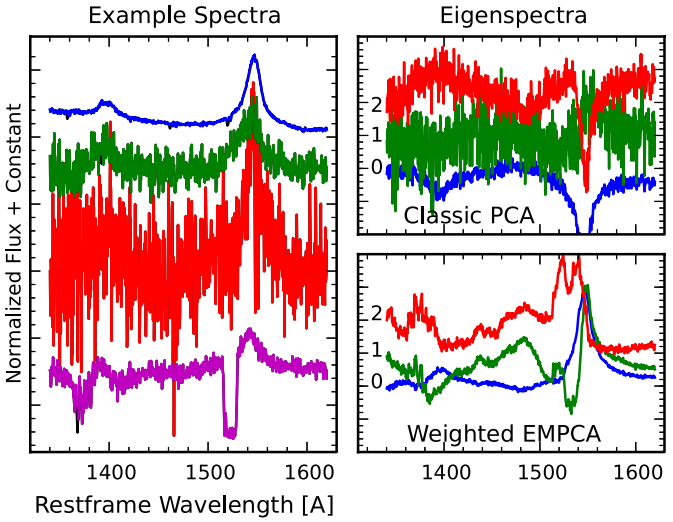

FIG. 5.—Example of high-, median-, and low signal-to-noise (S/N) input QSO spectra and a BAL QSO (*left*). The first three classic PCA eigenvectors are shown at top right; the first three weighted EMPCA eigenvectors are shown at bottom right. See the electronic edition of the *PASP* for a color version of this figure.

spectrum j. Weighted EMPCA also can properly ignore the masked data by using weight $= 0$ without having to discard the entire spectrum or artificially interpolate over the masked region.

The right panels of Figure 5 show the results for the first three eigenvectors of classic PCA (*top right*) and weighted EMPCA (*bottom right*). Eigenvectors 0, 1, and 2 are plotted in blue, green, and red, respectively. The mean spectrum was subtracted prior to performing PCA such that these eigenvectors represent the principal variations of the spectra with respect to that mean spectrum. Eigenvectors are orthogonal at the level of $\mathcal{O}(10^{-17})$.

The EMPCA eigenvectors are much less noisy than the classic PCA eigenvectors. As such, they are more sensitive to genuine signal variations in the data. For example, the sharp features between $1515 < \lambda < 1545$ Å in the EMPCA eigenspectra arise from BAL QSOs, an example of which is shown in the bottom of the left panel of Figure 5. These features are used to study QSO outflows, e.g. Turnshek (1988), yet they are lost amidst the noise of the classic PCA eigenspectra. Similarly, the EMPCA eigenspectra are more sensitive to the details of the variations in shape and location of the emission peaks used to study QSO metallicity (e.g. Juarez et al. 2009) and black hole mass (e.g. Vestergaard & Osmer 2009).

8. DISCUSSION

8.1. Convergence

McLachlan & Krishnan (1997) discuss the convergence properties of the EM algorithm in general. Each iteration, by construction, finds a set of parameters that are as good or better a fit to the data than the previous step, thus guaranteeing convergence. The caveat is that the "likelihood maximization step" is typically implemented as solving for a stationary point of the likelihood surface rather than strictly a maximum; e.g., $\partial\mathcal{L}/\partial\phi = 0$ is also true at saddle points and minima of the likelihood surface, thus it is possible that the EM algorithm will not converge to the true global maximum. Unweighted PCA has a likelihood surface with a single global maximum, but, in general, this is not the case for weighted PCA: the weights in equation (8) can result in local false χ^2 minima (Srebro & Jaakkola 2003). McLachlan & Krishnan (1997) § 3.6 also gives examples of this behavior taken (from Murray 1977 and Arslan, Constable, & Kent 1993) for the closely related problem of factor analysis. The example datasets are somewhat contrived and the minimum or saddle point convergence only happens with particular starting conditions.

We have encountered false minima with weighted EMPCA when certain observations have ~90% of their variables masked, while giving large weight to their remaining unmasked variables. In this case, the resultant eigenvectors can have artifacts tuned to the highly weighted, but mostly masked, input observations. When only a few (~10%) of the variables are masked per observation, we have not had a problem with false minima.

The algorithm outlined in § 5 solves for each eigenvector one at a time in order to maximize the power in the initial eigenvectors. This can result in a situation where a given iteration can improve the power described by the first few eigenvectors but degrade the total χ^2 using all eigenvectors. We have not found a case where this significantly degrades the global χ^2, however.

The speed of convergence is also not guaranteed. Roweis (1997) gives a toy example of fast convergence for Gaussian-distributed data (three iterations) and slow convergence for non-Gaussian-distributed data (23 iterations). In practice, we find that when EMPCA is slow to converge, it is exploring a shallow likelihood surface between two nearly degenerate eigenvectors. This situation pertains to the uniqueness of the solution, as described in the following section.

Weighted EMPCA may produce unstable solutions if it is used to solve for more eigenvectors than are actually present in the data or for eigenvectors that are nearly singular. Since EMPCA uses all eigenvectors while solving for the coefficients during each iteration, the singular eigenvectors can lead to delicately-balanced, meaningless values of the coefficients, which in turn degrades the solution of the updated eigenvectors in the next iteration. We recommend starting with solving for a small number of eigenvectors, and then increasing the number of eigenvectors if the resulting solution does not describe enough of the data variance.

For these reasons, one should use caution when analyzing data with EMPCA, just as one should do with any problem which is susceptible to false minima or other convergence issues. In practice, we find that the benefits of proper noise-weighting outweigh the potential convergence problems.

8.2. Uniqueness

Given that EMPCA is an iterative algorithm with a random starting point, the solution is not unique. In particular, if two eigenvalues are very close in magnitude, EMPCA could return an admixture of the corresponding eigenvectors while still satisfying the convergence criteria. In practice, however, EMPCA is pragmatic: if two eigenvectors have the same eigenvalue, they are also equivalently good at describing the variance in the data and could be used interchangeably.

Science applications, however, generally require strict algorithmic reproducibility, and thus EMPCA should be used with a fixed random number generator seed or fixed orthonormal starting vectors, such as Legendre polynomials. The convergence criteria define when a given vector is "good enough" to move on to the next iteration, but they do not guarantee uniqueness of that vector.

Figure 6 shows the first three eigenvectors for five different EMPCA solutions of the QSO spectra from § 7.2, with different random starting vectors. After 20 iterations, the eigenvectors agree to $<10^{-5}$ on both large and small scales. Although this agreement is worse than the machine precision of the computation, it is much smaller than the scale of differences between the eigenvectors, and it represents a practical level of convergence for most PCA applications.

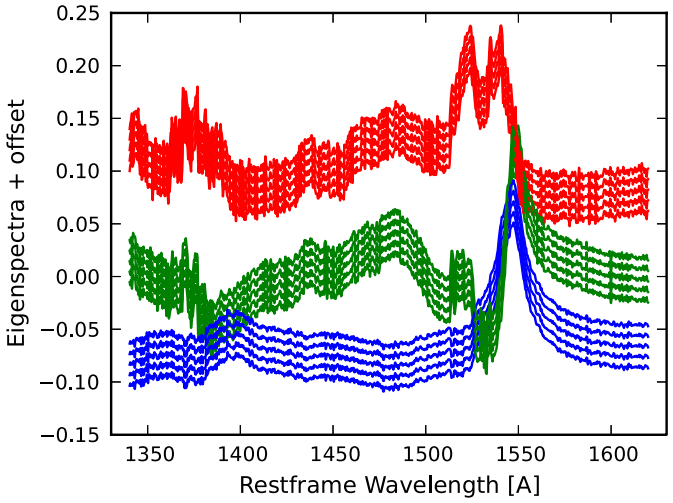

FIG. 6.—Example eigenspectra of the same dataset with different random starting values for the EMPCA algorithm. The eigenspectra are offset for clarity and have bin-for-bin agreement at the level of $\sim 10^{-5}$. See the electronic edition of the *PASP* for a color version of this figure.

8.3. Over-Weighted Data

Weighted EMPCA improves the PCA eigenvector solution by preventing noisy or missing data from unduly contributing noise instead of signal variation. However, the opposite case of high signal-to-noise data can also be problematic if just a few of the observations have significantly higher weight than the others. These will dominate the EMPCA solution just as they would dominate a weighted mean calculation. This may not be the desired effect since the initial eigenvectors will describe the differences between the highly weighted data, and subsequent eigenvectors will describe the lower weighted data. This may be prevented by purposefully down-weighting certain observations or by applying an upper limit to weights so that the weighted dataset is not dominated by just a few observations.

8.4. Scaling Performance

The primary advantage of EMPCA is the ability to incorporate weights on noisy data to improve the quality of the resulting eigenspectra. A secondary benefit over classic PCA is algorithmic speed for the common case of needing only the first few eigenvectors from a dataset with $n_{\rm obs} \lesssim n_{\rm var}$.

Classic PCA requires solving the eigenvectors of the data covariance matrix, an $\mathcal{O}(n_{\rm var}^3)$ operation. The weighted EMPCA algorithm described here involves iterating over multiple solutions of smaller matrices. Each iteration requires $n_{\rm obs}$ solutions of $\mathcal{O}(n_{\rm vec}^3)$ to solve the coefficients and $\mathcal{O}(n_{\rm obs} n_{\rm vec} n_{\rm var})$ operations to solve the eigenvectors. Thus, weighted EMPCA can be faster than classic PCA when $n_{\rm iter}(n_{\rm obs} n_{\rm vec}^3 + n_{\rm obs} n_{\rm vec} n_{\rm var}) < n_{\rm var}^3$, ignoring the constant prefactors. If one has a few hundred spectra ($n_{\rm obs}$) with a few thousand wavelengths each ($n_{\rm var}$) and wishes to solve for the first few eigenvectors ($n_{\rm vec}$), then EMPCA can be much faster than classic PCA. Conversely, if one wishes to perform PCA on all $\sim$1 million spectra from SDSS, then $n_{\rm obs} \gg n_{\rm var}$ and classic PCA is faster, albeit with the limitation of not being able to properly weight noisy or missing data. If the problem involves off-diagonal covariances, then weighted EMPCA involves a smaller number of larger matrix solutions for an overall slowdown, though it should be noted that classic PCA is unable to properly solve the problem at all.

As a performance example, we used EMPCA to study the variations in the simulated point spread function (PSF) of a new spectrograph design. The PSFs were simulated on a grid of 11 wavelengths and six slit locations, and were sampled over 200×200 μm spots on a 1 μm grid, for a total of 40,000 variables per spot. Classic PCA would require singular value decomposition of a $40,000 \times 40,000$ matrix. While this is possible, it is beyond the scope of a typical laptop computer. On the other hand, using EMPCA with constant weights, we were able to recover the first 30 eigenvectors covering 99.7% of the PSF variance in less than 6 min on a 2.13 GHz MacBook Air laptop.

For datasets where $n_{\rm var}$ is particularly large, the memory needed to store the $n_{\rm var} \times n_{\rm var}$ covariance matrix may be a limiting factor for classic PCA. The iterative nature of EMPCA allows one to scale to extremely large datasets since one never needs to keep the entire dataset (nor its covariance) in memory at one time. The multiple independent equations to solve in § 5.3.1 and § 5.3.2 are naturally computationally parallelizable.

9. PYTHON CODE

Python code implementing the weighted EMPCA algorithm described here is available at https://github.com/sbailey/empca. The current version implements the case of independent weights but not the more generalized case of off-diagonal covariances. It also implements the smoothed eigenvectors described in § 6.1, but not *a priori* eigenvectors (§ 6.2) nor distributed calculations (§ 8.4). For comparison, the `empca` module also includes implementations of classic PCA and weighted lower-rank matrix approximation.

Examples for this paper were prepared with tagged version v0.2 of the code. When using the code, note that the orientation of the data and weights vectors is the transpose of the notation used here, i.e., `data[j,i]` is variable `i` of observation `j` so that `data[j]` is a single observation.

10. SUMMARY

As a brief summary of the algorithm, a data matrix $\mathbf{X}$ can be approximated by a set of eigenvectors $\mathbf{P}$ with coefficients $\mathbf{C}$:

$$\mathbf{X} \approx \mathbf{PC} + \text{measurement noise}. \tag{23}$$

A covariance matrix $\mathbf{V}$ describes the estimated measurement noise.

The weighted EMPCA algorithm seeks to find the optimal **P** and **C**, given **X** and **V**. It starts with a random set of orthonormal vectors **P** and then iteratively alternates solutions for **C**, given $\{\mathbf{P}, \mathbf{X}, \mathbf{V}\}$, and **P**, given $\{\mathbf{C}, \mathbf{X}, \mathbf{V}\}$. The problem is additionally constrained by the requirement to maximize the power in the fewest number of eigenvectors (columns of **P**). To accomplish this, the algorithm solves for each eigenvector individually, before removing its projection from the data and solving for the next eigenvector. If the measurement errors are independent, the covariance can be described by a weights matrix **W** with the same dimensions as **X**, and the problem can be factorized into independent solutions of small matrices.

This algorithm produces a set of orthogonal principal component eigenvectors **P**, which are optimized to describe the most signal variance with the fewest vectors, while properly accounting for estimated measurement noise.

11. CONCLUSIONS

We have described a method for performing PCA on noisy data that properly incorporates measurement noise estimates when solving for the eigenvectors and coefficients. Missing data are simply limiting cases of weight $= 0$. The method uses an iterative solution based upon EM. The resulting eigenvectors are less sensitive to measurement noise and more sensitive to true underlying signal variations. The algorithm has been demonstrated on toy data and QSO spectra from SDSS. Code which implements this algorithm is available at https://github.com/sbailey/empca.

The author would like to thank Rollin Thomas and Sébastien Bongard for interesting and helpful conversations related to this work. The anonymous reviewer provided helpful comments and suggestions which improved this manuscript. The initial algorithm was developed during a workshop at the Institut de Fragny. This work was supported under the auspices of the Office of Science, U.S. DOE, under Contract No. DE-AC02-05CH1123. The example QSO spectra were provided by the SDSS. Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. The SDSS web site is http://www.sdss.org/. The SDSS is managed by the Astrophysical Research Consortium for the Participating Institutions. The Participating Institutions are the American Museum of Natural History, Astrophysical Institute Potsdam, University of Basel, University of Cambridge, Case Western Reserve University, University of Chicago, Drexel University, Fermilab, the Institute for Advanced Study, the Japan Participation Group, Johns Hopkins University, the Joint Institute for Nuclear Astrophysics, the Kavli Institute for Particle Astrophysics and Cosmology, the Korean Scientist Group, the Chinese Academy of Sciences (LAMOST), Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, Ohio State University, University of Pittsburgh, University of Portsmouth, Princeton University, the United States Naval Observatory, and the University of Washington.

REFERENCES

Abazajian, K. N., Adelman-McCarthy, J. K., Agüeros, M. A., et al. 2009, ApJS, 182, 543
Arslan, O., Constable, P. D. L., & Kent, J. T. 1993, Stat. Comput., 3, 103–108
Blanton, M. R., & Roweis, S. 2007, AJ, 133, 734
Bond, J. R. 1995, Phys Rev Lett, 74, 4369
Connolly, A. J., & Szalay, A. S. 1999, AJ, 117, 2052
Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977, J. R. Stat. Soc. B, 39, 1–38, http://www.jstor.org/stable/2984875
Gabriel, K. R., & Zamir, S. 1979, Technometrics, 21, 489–498, http://www.jstor.org/stable/1268288
Hewett, P. C., & Wild, V. 2010, MNRAS, 405, 2302
Hotelling, H. 1933, J. Educ. Psychol., 24, 417–441
Jee, M. J., Blakeslee, J. P., Sirianni, M., Martel, A. R., White, R. L., & Ford, H. C. 2007, PASP, 119, 1403
Jolliffe, I. T. 2002, Principal Component Analysis (2nd ed.; New York: Springer)
Juarez, Y., Maiolino, R., Mujica, R., Pedani, M., Marinoni, S., Nagao, T., Marconi, A., & Oliva, E. 2009, A&A, 494, L 25
McLachlan, G. J., & Krishnan, T. 1997, The EM Algorithm and Extension (New York: John Wiley & Sons)
Murray, G. D. 1997, contribution to the discussion section of Dempster, Laird, & Rubin (1977)
Pâris, I., Petitjean, P., & Rollinde, E., et al. 2011, A&A, 530, A 50
Pearson, K. 1901, Phil. Mag., 2, 559–572
Press, W. H.,Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2002, Numerical Recipes in C++: The Art of Scientific Computing (2nd ed.; New York: Cambridge University Press)
Roweis, S. 1997, CNS Technical Report, CNS-TR-97-02, http://cs.nyu.edu/~roweis/papers/empca.pdf
Srebro, N., & Jaakkola, T. 2003, Proc. of the twentieth Intl Conf. on Machine Learning, ed. T.Fawcett, & N.Mishra(Washington, DC: AAAI Press), 720, http://www.aaai.org/Press/Proceedings/icml03.php
Shlens, J. 2009, A Tutorial on Principal Component Analysis, http://www.snl.salk.edu/~shlens/pca.pdf
Tipping, M. E., & Bishop, C. M. 1999, J. R. Stat. Soc. B, 61, 611–622, http://www.jstor.org/stable/2680726
Tsalmantza, P., & Hogg, D. W. 2012, ApJ, 753, 122
Turnshek, D. A. 1988, Proceedings of the QSO Absorption Line Meeting (Baltimore), ed. J. C. Blades, D. A. Turnshek, & C. A. Norman (Cambridge: Cambridge University Press), 17
Vestergaard, M., & Osmer, P. S. 2009, ApJ, 699, 800
Wentzell, P. D., Andrews, D. T., Hamilton, D. C., Faber, K., & Kowalski, B. R. 1997, J. Chemometr., 11, 339–366