

Factorisation of Positive Valued Functions for Analysing Galaxy Spectra

Ata Kabán

School of Computer Science
University of Birmingham
Birmingham, UK
A.Kaban@cs.bham.ac.uk

Louisa Nolan

School of Physics and Astronomy
University of Birmingham
Birmingham, UK
lan@star.sr.bham.ac.uk

Somak Raychaudhury

School of Physics and Astronomy
University of Birmingham
Birmingham, UK
Somak@star.sr.bham.ac.uk

ABSTRACT

We develop a factorisation algorithm for a set of positive valued functions, each of which is specified over a common continuous domain, by a finite number of points. This is achieved by integrating both regression and non-negative factor analysis models in a unified probabilistic framework. Contrarily to existing work on clustering of functions, here non-parametric regression models may be employed without significantly increasing the computational cost over the analogous matrix factorisation algorithm. We apply this technique to decomposing de-redshifted galaxy spectra.

Keywords: functional data mining, exploratory analysis, non-negative data factorisation

1 INTRODUCTION

Matrix factorisations have been valuable general-purpose tools for multivariate data analysis, with applications in numerous fields of science and engineering. However often there is a long way until the real measurements get into a matrix form and this may be an unnatural process in some cases. Here we consider observations that are functions, specified by a finite number of points on a common continuous domain. The points where the function values are known are not necessarily the same for all functions. Therefore one would need to use some interpolation preprocessing to fit such data into a matrix and the natural topology of the continuous domain would be lost. In addition, any errors made at the preprocessing stage will be carried to all subsequent analyses, making the overall analysis suboptimal. It is therefore desirable to have data analysis methods that can directly operate on the set of functions.

Cluster analysis of a set of functions has recently been studied [1] in a fairly general setting. Earlier related

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

©2006 The University of Liverpool

work includes [7]. While clustering is an important data analysis technique, there are many reasons why factor analysis tools for functions is also desirable. For example, completing several related regression tasks may be more efficiently tackled if both their common and specific characteristics are captured in a consistent model. The exploratory analysis of a set of functions also calls for factor-type analysis – whereby a compact representation is sought in a particular form, primarily for understanding and interpreting the data.

Here we consider non-negative valued functions, such as spectra, and develop a non-negative factorisation approach suited for this scenario. Contrarily to the prohibitive computational demand noted with clustering real-valued functions [1], no matrix inversions are required in the proposed method. So we can employ non-parametric regression kernels without significantly increasing the computational cost over the analogous matrix factorisation algorithm. We illustrate the technique on both synthetic data and de-redshifted galaxy spectra.

2 METHOD

Consider N functions $y_n : \mathcal{R} \rightarrow \mathcal{R}^+$. The number of points where the n -th function is known will be denoted by T_n , and the associated function values will be denoted by $y_{nt} = y_{nt}(x_{nt})$. Each function is represented as a nonparametric regression model, in order to avoid any assumptions about the general relationship between the function value and the function argument. For the n -th function, $\mathbf{b}_t^{(n)}$ will denote the output vector of the regression basis functions evaluated at x_{nt} . The size of this vector (number of basis functions) is denoted by P . Naturally, for different $n \neq n'$, x_{nt} need not be equal to $x_{n't}$ and T_n may also be different from $T_{n'}$. However, P and the centres of the basis functions are global, e.g. they may consist of the union of all points where at least one of the function values is known, or they may be specified by the user.

Now, rather than having separate regression coefficients for each function, we model these in a factor form. The P regression coefficients of the n -th function will be sought as the linear combination $\mathbf{a}_n \mathbf{S}$, where \mathbf{a}_n is a $1 \times K$ vector of function-specific mixing coefficients, K is the number of factors and \mathbf{S} is a $K \times P$ matrix of regression coefficients. Thus, overall, the k -th factor in our model is a function of the form $s_k \mathbf{B}$, where \mathbf{B} is the design matrix

consisting of the values of the P basis functions evaluated at any desired points within an interval. These points may or may not coincide with those used with the N training function instances but should lie in the same interval.

For interpretability reasons, we require that the factors are also positive valued $\mathcal{R} \rightarrow \mathcal{R}^+$ functions. Since the design matrices will be non-negative, we require \mathbf{S} to be non-negative, and also require that all \mathbf{a}_n are non-negative. According to these model specifications, then up to constant terms, the complete data log likelihood, $\log L^C =$

$$\begin{aligned} & - \sum_{n=1}^N \sum_{t=1}^{T_n} \left\{ (y_{nt} - \mathbf{a}_n \mathbf{S} \mathbf{b}_t^{(n)})^2 e_{nt} - \log e_{nt} \right\} / 2 \\ & - \sum_{k,p} \left\{ (s_{kp} - m_k)^2 / 2 - \lambda_{kp} s_{kp} \right\} \\ & - \sum_{n,k} \left\{ a_{nk}^2 / 2 - \omega_{kn} a_{nk} \right\} \end{aligned} \quad (1)$$

where λ_{kp} and ω_{kn} are positive Lagrange multipliers in order to enforce the positivity constraints. Note we included a mean parameter for each regression coefficient component s_k — as we noted previously [2], unless there is reason to believe that the components have their mode at zero, this flexibility is beneficial. Further, we assume a heteroschedastic Gaussian noise, whose precisions will be denoted as e_{nt} and for the application considered here, this is partly determined by known measurement errors. The form of e_{nt} is therefore,

$$e_{nt} = 1 / (v^2 + \sigma_{nt}^2) \quad (2)$$

where σ_{nt} are known and v is the modelling error that may be estimated from the data. Considering different modelling error for each n is also straightforward.

2.1 Parameter estimation

To obtain maximum a posteriori (MAP) parameter estimates, we maximise the log complete data log likelihood (1). It should be noted that MAP estimates may be prone to overfitting when the sample size is small, and approximate Bayesian methods [2, 3] should be pursued instead. Here this is less of a concern, since our data consists of high-resolution detailed spectra.

From requiring $\frac{\partial \log L^C}{\partial \mathbf{S}} = 0$, we get

$$- \sum_{n=1}^N \mathbf{a}_n^T (\mathbf{y}_n - \mathbf{a}_n \mathbf{S} \mathbf{B}^{(n)})^T \mathbf{E}_n \mathbf{B}^{(n)T} + \mathbf{S} - \mathbf{m} \mathbf{1} = \mathbf{\Lambda} \quad (3)$$

where $\mathbf{1}$ is a $1 \times P$ vector of ones, \mathbf{E}_n is a diagonal matrix with elements e_{tn} where $t \in \{1, \dots, T_n\}$, $\mathbf{B}^{(n)}$ is the $P \times T_n$ design matrix of basis functions for the n -th instance and $\mathbf{\Lambda}$ is the $K \times P$ matrix with elements λ_{kp} .

Rearranging, and applying the KKT conditions, i.e. that $\mathbf{\Lambda} \odot \mathbf{S} = \mathbf{0}$, where \odot denotes element-wise product,

yields the following multiplicative fixed-point update.

$$\begin{aligned} \mathbf{S} & \leftarrow \mathbf{S} \odot \left\{ \sum_n \mathbf{a}_n^T \mathbf{y}_n \mathbf{E}_n \mathbf{B}^{(n)T} + \mathbf{m} \mathbf{1} \right\} \\ & \odot \left\{ \sum_n \mathbf{a}_n^T \mathbf{a}_n \mathbf{S} \mathbf{B}^{(n)} \mathbf{E}_n \mathbf{B}^{(n)T} + \mathbf{S} \right\} \end{aligned} \quad (4)$$

where \odot stands for element-wise division, and the remainder of notations is the same as before. Observe that no matrix inversion is required. Similarly, for each \mathbf{a}_n we obtain:

$$\begin{aligned} \mathbf{a}_n & \leftarrow \mathbf{a}_n \odot \left\{ \mathbf{y}_n \mathbf{E}_n \mathbf{B}^{(n)T} \mathbf{S}^T \right\} \\ & \odot \left\{ \mathbf{a}_n \mathbf{S} \mathbf{B}^{(n)} \mathbf{E}_n \mathbf{B}^{(n)T} \mathbf{S}^T + \mathbf{a}_n \right\} \end{aligned} \quad (5)$$

For \mathbf{m} , a closed form update is available:

$$\mathbf{m} \leftarrow \frac{1}{P} \sum_{p=1}^P s_p \quad (6)$$

Finally, due to the form of the variance in the likelihood term, unless we have no measurement errors ($\sigma_{nt} = 0$), there is no closed form solution for v^2 . Therefore numerical optimisation methods may be employed. However, noting that the variances are required to be always positive, we obtained a multiplicative fixed point update.

$$v^2 \leftarrow v^2 \frac{\sum_{n=1}^N \sum_{t=1}^{T_n} (y_{nt} - \mathbf{a}_n \mathbf{S} \mathbf{b}_t^{(n)})^2 / (v^2 + \sigma_{nt}^2)^2}{\sum_{n=1}^N \sum_{t=1}^{T_n} 1 / (v^2 + \sigma_{nt}^2)} \quad (7)$$

This is convenient in that no optimisation-specific parameters (such as a learning rate) need to be set. Note, if there is no measurement error, i.e. $\sigma_{nt} = 0$, then v^2 cancels out from the r.h.s. and (7) reduces to the familiar closed form solution for the variance.

It should be noted that the convergence of the resulting alternating optimisation algorithm (4)-(7) may be studied as in [5]. In particular, the integration of the multiplicative variance update (7) requires that unless there is a guarantee that each iteration of (7) would not decrease (1), then (7) should be iterated until the maximisation objective is no lower than it was before the variance update. In our experiments, this has always been the case and so one iteration was alternated with the other parameter updates. A more rigorous study of convergence remains for further work.

2.1.1 Scaling.

Since no matrix inversion is required throughout of the algorithm, the scaling is multi-linear with the number of functions, the total number of function values and the number of components.

3 RESULTS

A first illustrative experiment is demonstrated on synthetic data. The true sources consisted of the following two

positive valued functions defined over the same interval, shown also on the left plot of Figure 1.

$$f_1(x) = 10\text{sinc}(x) + 3.16; x \in [-3, 3]$$

$$f_2(x) = 2\sin(4x) + 4\cos(x) + 6.73; x \in [-3, 3]$$

Of these, 20 noisy mixtures were created: For each mixture n , a set of T_n points was randomly sampled from the domain of definition $[-3, 3]$, and the function values at those points were mixed together with randomly generated mixing coefficients a_n . The mixing coefficients for all 20 instances can be seen on the right hand plot of Figure 1. Finally, a Gaussian noise with zero mean and unit variance was added and the resulting data set of curves is depicted on Figure 2. Indeed, it would not be easy to do reliable regression analysis on any of these curves, neither individually nor globally.

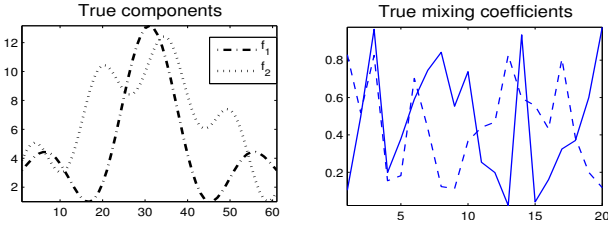


Figure 1: The true components and mixing coefficients in the synthetic data experiment.

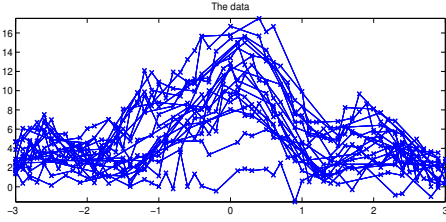


Figure 2: The noisy synthetic data set.

Radial basis functions were employed to construct the design matrices $B^{(n)}$ and P was set according to the number of equally distanced points with distance 0.1 on the domain of definition (which is 61). A global width parameter of 0.2 was used, and this value was set empirically¹. The measurement errors σ_{nt} have been set to 0.2, which represents an underestimation of the noise. Therefore we expect to find a nonzero modelling error v .

Figure 3 shows the evolution of the parameters as estimated through the iterations. The convergence is most apparent from the plot and indeed the estimated error is close to 1. Figure 4 shows the recovered components, as well as the recovered mixing coefficients. The recovery is up to scaling, permutation and translation.

Finally, figure 5 presents a subset of the reconstructed curves against the noisy data and the true functions

¹Cross-validation can be used to set an optimal width parameter, as well as the optimal P . Alternatively, local width parameters may be defined based on the distance from a point to its nearest points.

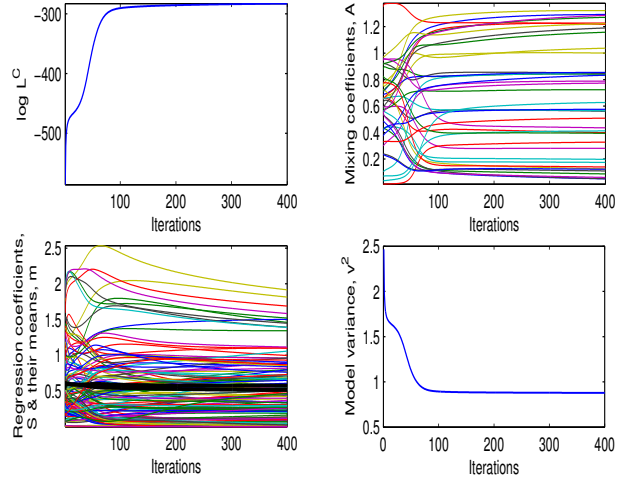


Figure 3: The evolution and convergence of of the model likelihood and the parameters during the iterations of the algorithm.

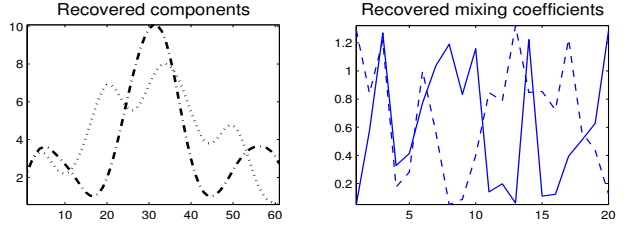


Figure 4: Recovered components and mixing coefficients.

$a_{n1}f_1(x) + a_{n2}f_2(x)$, together with the decomposition of each, in terms of the components identified. It is notable, e.g. on the middle left had plot that the hump at zero is correctly recovered despite there is no reliable data in that region. Of course, this is only possible as a result of 'borrowing strength' from the other 19 regressions through a unified model. Hence, the benefits of factorising learning machines is most apparent in principle.

3.1 Application to Galaxy Spectra

In previous work [4, 6] we have analysed spectra of nearby galaxies by various related factor analysis techniques. However those techniques rely on identical binning of all spectra. The approach presented here relaxes the equal binning requirement and allows us to decompose de-redshifted spectra, where without some interpolation preprocessing, the binning is different due to the shift. Missing data is also dealt with naturally, since now we have a set of regression models over a continuous domain and the input functions are not required to be specified in the same points. Using the known measurement errors (more details about the data can be found in [6]) we find a modelling variance v^2 of the order of 10^{-4} . This indicates the assumed model describes the data well and many of the details are genuine and not due to noise above of what

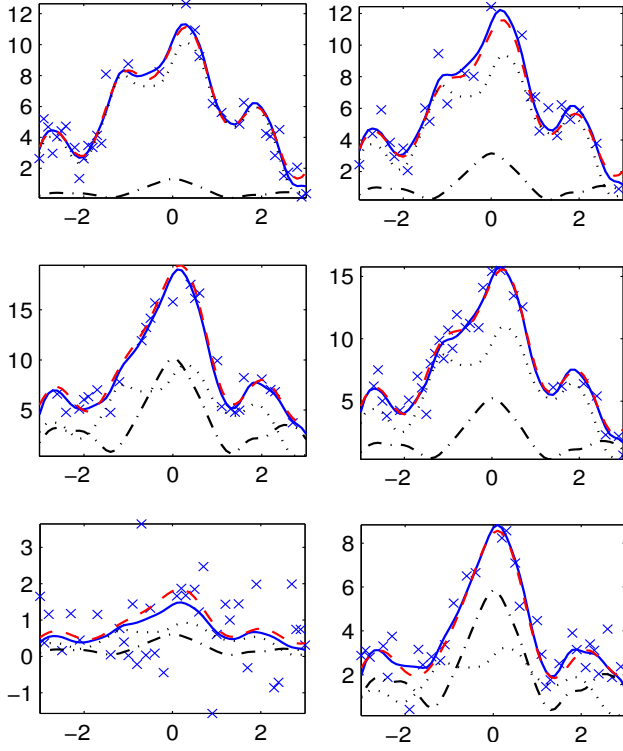


Figure 5: A selection of six regressions performed simultaneously by the proposed approach. Dashed line: the true function y_n ; 'x': the noisy data; Continuous line: the reconstructed function $a_n \mathbf{S} \mathbf{B}$; Dotted and dash-dotted lines: The two recovered component functions multiplied by their mixing proportion, i.e. $a_{nk} s_k \mathbf{B}$.

is already known. In addition, the recovered components are also interpretable. The upper plot of Figure 6 shows an example where a galaxy spectrum is decomposed into two spectral factors using the proposed approach. Again, radial basis functions were employed and now the P centres were taken as the union of all wavelengths where at least one of the input spectrum is specified. The lower plot shows the decomposition of the same galaxy spectrum, as obtained by detailed fitting of physical models. The similarity is apparent and the two factors can readily be recognised and interpreted as typical young and mature factors of the galaxy.

References

- [1] S Gaffney, P Smyth. Joint Probabilistic Curve Clustering and Alignment. In Neural Information Processing Systems 2003.
- [2] M Harva and A Kabán, Variational Learning for Rectified Factor Analysis. Signal Processing, to appear.
- [3] P Hojen-Sorensen, O Winther and L.K Hansen. Mean Field Approaches to Independent Component Analysis. Neural Computation 14, 889-918, 2002.
- [4] A Kabán, L Nolan and S Raychaudhury. Find-

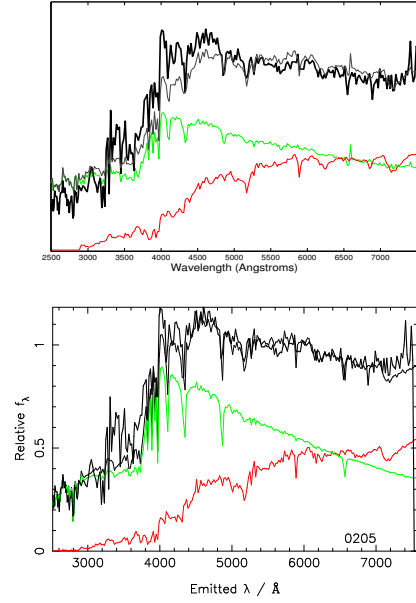


Figure 6: Upper plot: Decomposition of a spectrum as obtained from our method. Lower plot: Analysis based on astrophysical models and methods. The similarity is most apparent.

ing Young Stellar Populations in Elliptical Galaxies from Independent Components of Optical Spectra. Proc. SIAM Int'l Conf on Data Mining (SDM05), pp. 183–194.

- [5] D.D Lee and S. Seung. Algorithms for Non-Negative Matrix Factorisation. In Neural Information Processing Systems 2000.
- [6] L Nolan, M Harva, A Kabán and S Raychaudhury. A data-driven Bayesian approach to finding young stellar populations in early-type galaxies from their ultraviolet-optical spectra, Mon. Not. of the Royal Astron. Soc. 366(1), pp. 321-338.
- [7] J.O Ramsay and B.W Silverman. Functional Data Analysis. Springer-Verlag, New York, 1997.

Appendix: The estimation of v^2

Denoting $\mu_{nt} = a_n \mathbf{S} b_t^{(n)}$ and requiring that $\frac{\partial L^C + \nu v^2}{\partial v^2} = 0$, where ν is a positive Lagrange multiplier to ensure the positivity of the variance v^2 , the following is obtained:

$$0 = \frac{\partial}{\partial v^2} \sum_{nt} \left[\frac{\log(v^2 + \sigma_{nt}^2)}{2} + \frac{(y_{nt} - \mu_{nt})^2}{2(v^2 + \sigma_{nt}^2)} \right] - \nu v^2$$

$$\nu = \frac{1}{2} \sum_{nt} \left[\frac{1}{v^2 + \sigma_{nt}^2} - \frac{(y_{nt} - \mu_{nt})^2}{(v^2 + \sigma_{nt}^2)^2} \right]$$

From KKT, we have that $\nu v^2 = 0$, i.e.

$$v^2 \sum_{nt} \frac{1}{v^2 + \sigma_{nt}^2} = v^2 \sum_{nt} \frac{(y_{nt} - \mu_{nt})^2}{(v^2 + \sigma_{nt}^2)^2} \quad (8)$$

Rearranging, by isolating v^2 from the l.h.s, the fixed-point equation (7), given in the main text is obtained.