TUNING PARAMETER SELECTION FOR NONNEGATIVE MATRIX FACTORIZATION

M.O. Ulfarsson† *and V. Solo*‡

†University of Iceland, Dept. Electrical Eng., Reykjavik, ICELAND
‡University of New South Wales, School of Electrical Eng., Sydney, AUSTRALIA

ABSTRACT

Finding low rank nonnegative decomposition of multivariate data has many important applications in signal processing. A standard method is the nonnegative matrix factorization (NMF). In recent years, many algorithm have been proposed for NMF. However, an important problem that has not received as much attention is the selection of the rank of NMF. In this paper we develop a method for selecting the rank of NMF based on the Stein's unbiased risk estimator (SURE). In simulations we compare the method against crossvalidation. In addition we apply the method for selecting the rank of NMF for high dimensional hyperspectral data.

Index Terms— Nonnegative matrix factorization, SURE, Hyperspectral data, Crossvalidation.

1. INTRODUCTION

Finding a low rank decomposition of multivariate data is of much interest in the signal processing community [2]. Examples include principal component analysis (PCA) [3] and independent component analysis (ICA) [4].

Often the components of low rank decomposition are only meaningful if they are nonnegative. Examples include hyperspectral data unmixing [5] and dimension reduction for face recognition [6] .

Nonnegative matrix factorization (NMF) is a popular method for finding hidden components in nonnegative data. A large number of algorithms have been developed to perform NMF. One of the most commonly used algorithm is the multiplicative update algorithm [7]. For other NMF algorithms and extensions thereof see [8].

An important problem for NMF is the rank selection. To the best of our knowledge there is not much research on this issue. In [9] a Bi-Crossvalidation (BCV) approach was developed. It is based on dividing the rows of the data matrix into h groups and the columns into l groups. During one round of BCV one row and column group (submatrix) is held out while the rest of the data is used to construct an estimate of it. The BCV estimate is based on repeating this for each row and column group and averaging the resulting hold out error estimates. An alternative crossvalidation approach, based on using the weighted NMF, was proposed in [10]. The paper [11] developed a method using automatic relevance determination for selecting the rank of NMF.

In this paper we consider rank selection for NMF using the Stein's unbiased risk estimator (SURE) [12, 1]. The idea behind SURE is to develop an unbiased computable estimator of the mean squared error (MSE) which can then be used to select tuning parameters. SURE has proven to be useful for variety of signal processing models but has never been used for NMF before. The advantage of SURE relative to crossvalidation methods is that SURE yields a closed form formula that gives greater insight and computational speed. We compare the method against BCV in simulation and apply it for selecting the rank of NMF for real high dimensional hyperspectral data.

The paper is organized as follows. In section 2 we review NMF. In section 3 we derive SURE for NMF. Section 4 presents simulations and an application of the proposed method on real hyperspectral data is given. Finally, in section 5, conclusions are presented.

1.1. Notation

We denote matrices and vector with boldface letters and scalars with lower case letters. The Frobenius norm is denoted as $\|\boldsymbol{W}\|_F^2 = \sum_i \sum_j w_{ij}^2$; $E[\cdot]$ denotes the expectation operator. The vectorizing operation vec$(\boldsymbol{A})$ stacks the columns of $\boldsymbol{A}$ on top of each other; $\otimes$ is the Kronecker product, and tr$(\boldsymbol{A})$ is the trace of a matrix $\boldsymbol{A}$.

2. NMF

There are two original NMF algorithms, one based on Kullback-Leibler diverence and the other based on least squares. In this paper we discuss the latter. The least squares NMF criterion is given by

$$J(\boldsymbol{W}, \boldsymbol{H}) = \|\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{H}\|_F^2 \tag{1}$$

where $\boldsymbol{Y} = [y_{ij}]$ is a nonnegative $m \times n$ matrix, $\boldsymbol{W}$ is a nonnegative $m \times r$ matrix, and $\boldsymbol{H}$ is a nonnegative $r \times n$ matrix. Without loss of generality we assume during the derivation of SURE that $m \geq n$. The Lee-Seung method [7] for minimiz-

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

ing (1) is a cyclic descent method

$$
\begin{aligned}
H_{1,ab} &= H_{0,ab}\frac{(\boldsymbol{W}_0^T\boldsymbol{Y})_{ab}}{(\boldsymbol{W}_0^T\boldsymbol{W}_0\boldsymbol{H}_0)_{ab}} \\
W_{1,ia} &= W_{0,ia}\frac{(\boldsymbol{Y}\boldsymbol{H}_1^T)_{ia}}{(\boldsymbol{W}_0\boldsymbol{H}_1\boldsymbol{H}_1^T)_{ia}}
\end{aligned}
$$

where 0 refers to current iterate and 1 refers to next iterate.

3. SURE FOR NMF

Consider the model

$$y_{ij} = \mu_{ij} + \epsilon_{ij}, \quad i = 1, ..., m, \quad j = 1, ..., n$$

where $\epsilon_{ij} \sim N(0, \sigma^2)$. The MSE for $\hat{\mu}_{ij} = \hat{\mu}_{ij,r}(\boldsymbol{Y})$, an estimator of $\mu_{ij} = [\boldsymbol{W}\boldsymbol{H}]_{ij}$ where r is the rank, is given by

$$
\begin{aligned}
R_r &= \sum_{ij} E[(\mu_{ij} - \hat{\mu}_{ij})^2] \\
&= \sum_{ij} e_{ij}^2 - 2\sum_{ij} E[e_{ij}\epsilon_{ij}] + nm\sigma^2
\end{aligned}
$$

where $e_{ij} = y_{ij} - \hat{\mu}_{ij}$ is the residual and

$$
\begin{aligned}
E[e_{ij}\epsilon_{ij}] &= E[e_{ij}(y_{ij} - \hat{\mu}_{ij})] \\
&= \sigma^2 E[\frac{de_{ij}}{dy_{ij}}] \quad \text{Stein's Lemma [12]} \\
&= \sigma^2 - \sigma^2 E[\frac{d\hat{\mu}_{ij}}{dy_{ij}}].
\end{aligned}
$$

This yields

$$R_r = \sum_{ij} e_{ij}^2 + 2\sigma^2 \sum_{ij} E[\frac{d\hat{\mu}_{ij}}{dy_{ij}}] - nm\sigma^2.$$

By dropping expectation and irrelevant terms we get SURE

$$\hat{R}_r = \sum_{ij} e_{ij}^2 + 2\sigma^2 \sum_{ij} \frac{d\hat{\mu}_{ij}}{dy_{ij}}.$$

The rank r is chosen corresponding to the minimum of SURE.

3.1. Derivation of $\frac{d\hat{\mu}_{ij}}{dy_{ij}}$

To get the derivatives we need to differentiate through the fixed point equations of the Lee-Seung method

$$
\begin{aligned}
\frac{(\boldsymbol{W}^T\boldsymbol{Y})_{ab}}{(\boldsymbol{W}^T\hat{\boldsymbol{\mu}})_{ab}} &= 1 \\
\Rightarrow \boldsymbol{W}^T(\boldsymbol{Y} - \hat{\boldsymbol{\mu}}) &= \boldsymbol{0}
\end{aligned}
$$

and similarly $(\boldsymbol{Y} - \hat{\boldsymbol{\mu}})\boldsymbol{H}^T = \boldsymbol{0}$. Now change notation to $\boldsymbol{K} = \boldsymbol{H}^T$ so both $\boldsymbol{W}$ and $\boldsymbol{K}$ have r columns. The fixed point equations are

$$\boldsymbol{W}^T(\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{K}^T) = \boldsymbol{0} \text{ and } (\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{K}^T)\boldsymbol{K} = \boldsymbol{0}.$$

We need to compute

$$\frac{d\hat{\boldsymbol{\mu}}}{dy_{ij}} = \frac{d\boldsymbol{W}}{dy_{ij}}\boldsymbol{K}^T + \boldsymbol{W}\frac{d\boldsymbol{K}^T}{dy_{ij}}.$$

Introduce $\boldsymbol{E} = \boldsymbol{Y} - \hat{\boldsymbol{\mu}}$, let $\boldsymbol{\delta}_i$ be a vector of 0s but with a 1 in position i, and differentiate through the fixed point equations to get

$$
\begin{aligned}
\frac{d\boldsymbol{W}^T}{dy_{ij}}\boldsymbol{E} + \boldsymbol{W}^T(\boldsymbol{\delta}_i\boldsymbol{\delta}_j^T - \frac{d\boldsymbol{W}}{dy_{ij}}\boldsymbol{K}^T - \boldsymbol{W}\frac{d\boldsymbol{K}}{dy_{ij}}^T) &= \boldsymbol{0} \\
\boldsymbol{E}\frac{d\boldsymbol{K}}{dy_{ij}} + (\boldsymbol{\delta}_i\boldsymbol{\delta}_j^T - \frac{d\boldsymbol{W}}{dy_{ij}}\boldsymbol{K}^T - \boldsymbol{W}\frac{d\boldsymbol{K}^T}{dy_{ij}})\boldsymbol{K} &= \boldsymbol{0}.
\end{aligned}
$$

Rewriting gives

$$
\begin{aligned}
\frac{d\boldsymbol{W}^T}{dy_{ij}}\boldsymbol{E} - \boldsymbol{W}^T\frac{d\boldsymbol{W}}{dy_{ij}}\boldsymbol{K}^T - \boldsymbol{W}^T\boldsymbol{W}\frac{d\boldsymbol{K}^T}{dy_{ij}} &= -\boldsymbol{w}_i\boldsymbol{\delta}_j^T \\
\boldsymbol{E}\frac{d\boldsymbol{K}}{dy_{ij}} - \frac{d\boldsymbol{W}}{dy_{ij}}\boldsymbol{K}^T\boldsymbol{K} - \boldsymbol{W}\frac{d\boldsymbol{K}^T}{dy_{ij}}\boldsymbol{K} &= -\boldsymbol{\delta}_i\boldsymbol{k}_j^T
\end{aligned}
$$

where $\boldsymbol{w}_i^T$ and $\boldsymbol{k}_j^T$ are the i-th and j-th rows of $\boldsymbol{W}$ and $\boldsymbol{K}$, respectively. Now we vectorize and use the commutator matrix $\boldsymbol{L}_{rm}$ [13] that has the following properties $\boldsymbol{L}_{rm}\text{vec}\left(\frac{d\boldsymbol{W}^T}{dy_{ij}}\right) = \text{vec}\left(\frac{d\boldsymbol{W}}{dy_{ij}}\right)$ and $\boldsymbol{L}_{rn}\text{vec}\left(\frac{d\boldsymbol{K}^T}{dy_{ij}}\right) = \text{vec}\left(\frac{d\boldsymbol{K}}{dy_{ij}}\right)$ where $\boldsymbol{L}_{rm}$ is an $mr \times mr$ permutation matrix; similarly for $\boldsymbol{L}_{rn}$. We thus obtain

$$\boldsymbol{M}\begin{pmatrix} \text{vec}\left(\frac{d\boldsymbol{W}}{dy_{ij}}\right) \\ \text{vec}\left(\frac{d\boldsymbol{K}^T}{dy_{ij}}\right) \end{pmatrix} = -\begin{pmatrix} \text{vec}(\boldsymbol{w}_i\boldsymbol{\delta}_j^T) \\ \text{vec}(\boldsymbol{\delta}_i\boldsymbol{k}_j^T) \end{pmatrix}$$

where

$$
\begin{aligned}
\boldsymbol{M} &= \begin{bmatrix} (\boldsymbol{E}^T \otimes \boldsymbol{I}_r)\boldsymbol{L}_{mr} & \boldsymbol{0}_{nr\times nr} \\ \boldsymbol{0}_{mr\times mr} & [\boldsymbol{I}_r \otimes \boldsymbol{E}]\boldsymbol{L}_{rn} \end{bmatrix} \\
&- \begin{bmatrix} \boldsymbol{K}\otimes\boldsymbol{W}^T & \boldsymbol{I}_n \otimes \boldsymbol{W}^T\boldsymbol{W} \\ \boldsymbol{K}^T\boldsymbol{K}\otimes\boldsymbol{I}_m & \boldsymbol{K}^T\otimes\boldsymbol{W} \end{bmatrix} \\
&\equiv \boldsymbol{B} - \boldsymbol{A}.
\end{aligned}
$$

Now we can write a formula for $\sum_{ij}\frac{d\hat{\mu}_{ij}}{dy_{ij}}$. We can write

$$
\begin{aligned}
\sum_{ij}\frac{d\hat{\mu}_{ij}}{dy_{ij}} &= \sum_{ij}\boldsymbol{\delta}_i^T\frac{d\hat{\boldsymbol{\mu}}}{dy_{ij}}\boldsymbol{\delta}_j \\
&= \sum_{ij}[(\boldsymbol{k}_j^T\otimes\boldsymbol{\delta}_i^T), (\boldsymbol{\delta}_j^T\otimes\boldsymbol{w}_i^T)]\begin{pmatrix} \text{vec}\left(\frac{d\boldsymbol{W}}{dy_{ij}}\right) \\ \text{vec}\left(\frac{d\boldsymbol{K}^T}{dy_{ij}}\right) \end{pmatrix} \\
&= -\text{tr}(\boldsymbol{A}(\boldsymbol{B}-\boldsymbol{A})^{-1}) \\
&= \text{rank}(\boldsymbol{M}) - \text{tr}(\boldsymbol{B}(\boldsymbol{B}-\boldsymbol{A})^{-1}) \\
&= (m+n)r - \text{tr}(\boldsymbol{B}(\boldsymbol{B}-\boldsymbol{A})^{-1}).
\end{aligned}
$$

It can be shown that (see the Appendix)

$$\text{tr}(\boldsymbol{B}(\boldsymbol{B}-\boldsymbol{A})^{-1}) = \sum_{i=1}^{r}\sum_{j=1}^{n-r}\frac{2\lambda_j^2}{\lambda_j^2 - \rho_i^2}$$

where ρ_i^2 is the i-th eigenvalue of $\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{W}^T\boldsymbol{W}$ and λ_j^2 is the j-th eigenvalue of $\boldsymbol{E}^T\boldsymbol{E}$. Putting everything together we get SURE for NMF

$$\hat{R}_r = \sum_{ij} e_{ij}^2 + 2\sigma^2 \left((m+n)r - \sum_{i=1}^{r} \sum_{j=1}^{n-r} \frac{2\lambda_j^2}{\lambda_j^2 - \rho_i^2} \right). \quad (2)$$

4. EXPERIMENTS

In these experiments we focus on remote sensing hyperspectral data. We assume that the hyperspectral data $\boldsymbol{Y}_{m\times n}$ follows the model

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{\epsilon}$$

where $\boldsymbol{Y}, \boldsymbol{W}$ and $\boldsymbol{H}$ are nonnegative matrices. Column p of $\boldsymbol{Y}$ contains spectral measurement at pixel p. The model assumes that the spectral measurement at pixel p is a linear combination of so called endmember signatures contained in the matrix $\boldsymbol{W}$ weighted by the abundances fractions contained in column p of the matrix $\boldsymbol{H}$. Ideally, there are few endmember signatures that correspond to known physical features of the remote sensing scene such as houses or trees.

4.1. Simulation

In the simulation $\boldsymbol{W}_{m\times r}$ contains $r = 4$ endmember signatures of dimensions $m = 162$. The endmembers are shown in Fig. 1 The $r \times (m = 1000)$ matrix $\boldsymbol{H} = [\boldsymbol{h}_{(p)}]$ contains

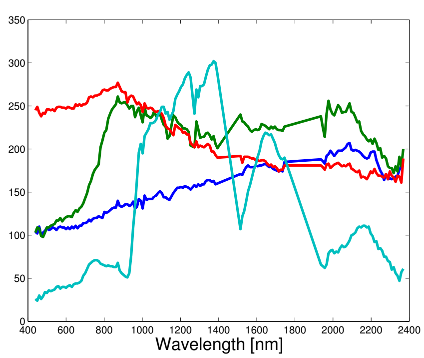

Fig. 1. The endmember signatures for the simulation.

the abundances fractions of each endmembers in its columns. The abundances are generated according to the Dirichlet distribution [14]

$$f(\boldsymbol{h}|\alpha_1, ..., \alpha_r) = \frac{\Gamma(\sum_k \alpha_k)}{\Pi_k \Gamma(\alpha_k)} \Pi_k h_k^{\alpha_k - 1}$$

where we set $\alpha_k = 1$. The noise matrix $\boldsymbol{\epsilon}$ contains i.i.d elements drawn from a Gaussian distribution with mean 0 and variance σ^2. The noise variance was set according to a specific signal to noise ratio (SNR)

$$\text{SNR} = 10\log_{10}\left(\frac{\|\boldsymbol{W}\boldsymbol{H}\|_F^2}{mn\sigma^2}\right).$$

If an element of $\boldsymbol{Y}$ is negative after generation, we set it equal to zero.

In the simulation we set SNR $= 20$ and compare NMF-SURE to the Bi-Crossvalidation method in [9] which we call BCV. To compute SURE (2) we need to estimate the noise variance σ^2. We use the following median noise estimator

$$\hat{\sigma} = \frac{1}{n}\sum_{p=1}^{n} \text{median}(|z_{i,p}| : i = 1, ..., m)/0.6745$$

where $z_{i,p}, \quad i = 1, ..., n$ is obtained by highpass filtering $y_{i,p}, \quad i = 1, ..., m$. Similar noise estimator was used in [15]. We use (3×3)-fold BCV.

We run 50 simulations. For each simulation we picked the rank corresponding to the minimum of the SURE and BCV. Fig. 2 summarizes the result. We see that SURE almost always picks the correct rank. Fig. 3 (left) shows NMF-SURE vs the true MSE $\|\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{H}\|_F^2$. The NMF-SURE follows the MSE closely and minimum for both curves occur at the same rank. The discrepancy between MSE and NMF-SURE is mostly due to that a noise variance estimate is used instead of the true noise variance in the SURE formula. We note that the MSE is of course uncomputable in practice since it depends on the true signal. Fig. 3 (right) shows BCV as a function of rank. In this case the minimum of BCV occurs at $r = 6$.

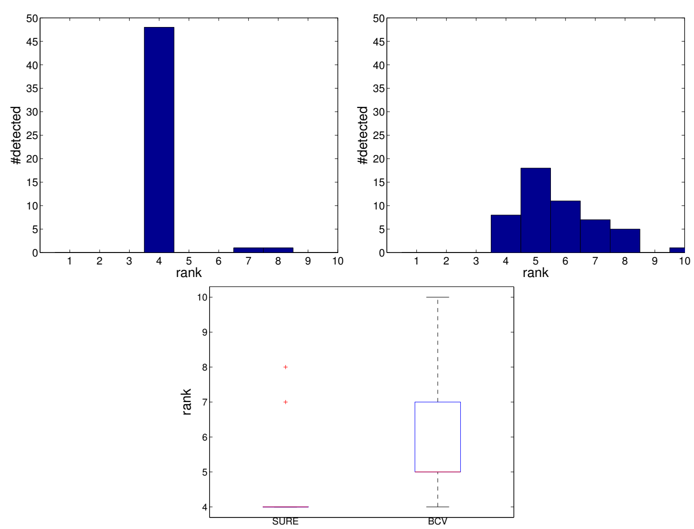

Fig. 2. A histogram summarizing the detection accuracy. $r = 4$ is the correct rank. Left: NMF-SURE. Right: BCV. Bottom: A box plot.

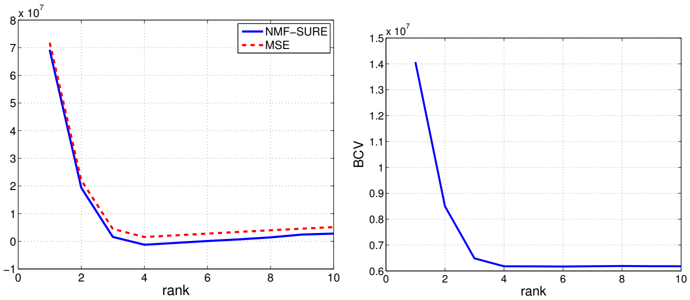

Fig. 3. Left: MSE vs NMF-SURE for one of the simulation runs. Right: BCV for one of the simulation runs.

4.2. Real data

In this section we apply NMF-SURE on the Indian Pine hyperspectral data set. This data set was collected by the AVIRIS sensor over the Indian Pines test site in northwestern Indiana in June 1992. The data consists of $128 \times 128 = 16384$ pixels and 220 spectral bands. Noisy bands and water absorptions bands were excluded leaving a 186×16384 data matrix $\boldsymbol{Y}$. Fig. 4 shows the NMF-SURE result. The method is picking $r = 18$ components. We note that the same data was analyzed in [16] where various methods from the remote sensing literature were used to identify the rank of the data. These methods picked the rank in the range from 16-25 so the SURE choice seems reasonable.

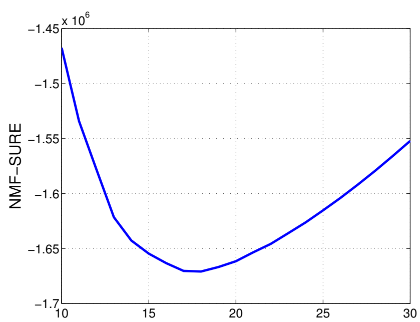

Fig. 4. NMF-SURE for the real data.

5. CONCLUSIONS

In this paper we have developed SURE for selecting the rank of NMF. We showed a simulation example where it outperformed a crossvalidation method specially designed for the NMF problem. We also applied the method on a real hyperspectral data example.

6. APPENDIX

Here we continue with the SURE derivation. Let $\boldsymbol{E} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^T$ be an SVD. Then we can write

$$\begin{aligned}
\boldsymbol{B} &= \begin{pmatrix} \boldsymbol{Q}\otimes\boldsymbol{I}_r & \boldsymbol{0}_{nr\times mr} \\ \boldsymbol{0}_{mr\times nr} & \boldsymbol{I}_r\otimes\boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{D}^T\otimes\boldsymbol{I}_r & \boldsymbol{0}_{nr\times nr} \\ \boldsymbol{0}_{mr\times mr} & \boldsymbol{I}_r\otimes\boldsymbol{D} \end{pmatrix} \\
&\cdot \begin{pmatrix} \boldsymbol{P}^T\otimes\boldsymbol{I}_r & \boldsymbol{0}_{mr\times nr} \\ \boldsymbol{0}_{nr\times mr} & \boldsymbol{I}_r\otimes\boldsymbol{Q}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{L}_{mr} & \boldsymbol{0}_{mr\times nr} \\ \boldsymbol{0}_{nr\times mr} & \boldsymbol{L}_{rn} \end{pmatrix} \\
&\equiv \boldsymbol{B}_1\boldsymbol{B}_2\boldsymbol{B}_3\boldsymbol{B}_4
\end{aligned}$$

we get

$$\begin{aligned}
\text{tr}(\boldsymbol{B}(\boldsymbol{B}-\boldsymbol{A})^{-1}) &= \text{tr}(\boldsymbol{B}_2(\boldsymbol{B}_2-\boldsymbol{B}_1^T\boldsymbol{A}\boldsymbol{B}_4^T\boldsymbol{B}_3^T)^{-1}) \\
&\equiv \text{tr}(\boldsymbol{B}\boldsymbol{C}^{-1})
\end{aligned}$$

where

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$$

and

$$\begin{aligned}
\boldsymbol{C}_{11} &= \boldsymbol{D}^T\otimes\boldsymbol{I}_r - \boldsymbol{L}_{nr}(\boldsymbol{W}^T\boldsymbol{P}\otimes\boldsymbol{Q}^T\boldsymbol{K}) \\
\boldsymbol{C}_{12} &= \boldsymbol{L}_{nr}(\boldsymbol{W}^T\boldsymbol{W}\otimes\boldsymbol{I}_n) \\
\boldsymbol{C}_{21} &= -\boldsymbol{L}(\boldsymbol{I}_m\otimes\boldsymbol{K}^T\boldsymbol{K}) \\
\boldsymbol{C}_{22} &= \boldsymbol{I}_r\otimes\boldsymbol{D} - \boldsymbol{L}_{rm}(\boldsymbol{P}^T\boldsymbol{W}\otimes\boldsymbol{K}^T\boldsymbol{Q}).
\end{aligned}$$

The inverse of $\boldsymbol{C}$ can be written as

$$\begin{aligned}
\boldsymbol{C}^{-1} &= \begin{pmatrix} \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \\ \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{0}_{rn\times rm} & \boldsymbol{I}_{rn\times rn} \\ \boldsymbol{I}_{rm\times rm} & \boldsymbol{0}_{rm\times rn} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{F}_{11i} & \boldsymbol{F}_{12i} \\ \boldsymbol{F}_{21i} & \boldsymbol{F}_{22i} \end{pmatrix} \begin{pmatrix} \boldsymbol{0}_{rn\times rm} & \boldsymbol{I}_{rn\times rn} \\ \boldsymbol{I}_{rm\times rm} & \boldsymbol{0}_{rm\times rn} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{F}_{12i} & * \\ * & \boldsymbol{F}_{21i} \end{pmatrix}
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{F}_{12i} &= -\boldsymbol{C}_{21}^{-1}\boldsymbol{C}_{22}\boldsymbol{J}^{-1} \\
\boldsymbol{F}_{21i} &= -\boldsymbol{J}^{-1}\boldsymbol{C}_{11}\boldsymbol{C}_{21}^{-1} \\
\boldsymbol{J} &= \boldsymbol{C}_{12} - \boldsymbol{C}_{11}\boldsymbol{C}_{21}^{-1}\boldsymbol{C}_{22}
\end{aligned}$$

we do not use the $*$ terms. Now we continue with the derivative term in the SURE formula

$$\begin{aligned}
\sum_{ij}\frac{d\hat{\mu}_{ij}}{dy_{ij}} &= (m+n)r - \text{tr}\left[(\boldsymbol{D}^T\otimes\boldsymbol{I}_r)\boldsymbol{F}_{12i}\right] \\
&- \text{tr}\left[(\boldsymbol{I}_r\otimes\boldsymbol{D})\boldsymbol{F}_{21i}\right] \\
&= (m+n)r - 2\text{tr}\left[(\boldsymbol{D}^T\otimes\boldsymbol{I}_r)\boldsymbol{F}_{12i}\right].
\end{aligned}$$

If we let ρ_i^2 be the i-th eigenvalue of $\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{W}^T\boldsymbol{W}$ and λ_j^2 be the j-th eigenvalue of $\boldsymbol{E}^T\boldsymbol{E}$ then it can be shown that

$$\text{tr}\left[(\boldsymbol{D}^T\otimes\boldsymbol{I}_r)\boldsymbol{F}_{12i}\right] = \sum_{i=1}^{r}\sum_{j=1}^{n-r}\frac{\lambda_j^2}{\lambda_j^2-\rho_i^2}.$$

7. REFERENCES

[1] V. Solo, “A sure-fired way to choose smoothing parameters in ill-conditioned inverse problems,” in *Proc. IEEE International Conference on Image Processing (ICIP'96)*, Lausanne, Switzerland, 1996, vol. 3, pp. 89–92.

[2] L. Scharf, “The SVD and reduced rank signal processing,” *Signal Processing*, vol. 25, no. 2, 1991.

[3] I.T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, second edition, 2002.

[4] A. Hyvarien, J. Karhunen, and E Oja, *Independent Component Analysis*, John Wiley and Sons, New York, 2001.

[5] J.M. Bioucas-Dias, “An overview on hyperspectral unmixing: Geometrical, statistical, and sparse regression based approaches,” in *Proc. IEEE International Conference on Geoscience and Remote Sensing (IGARSS'11)*, Vancouver, Canada, 2011.

[6] D. Lee and S. Seung, “Learning the parts of objects by nonnegative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.

[7] D. Lee and S. Seung, “Algorithms for nonnegative matrix factorization,” in *Advances in Neural Information Processing Systems (NIPS'01)*, 2001, vol. 13, pp. 556–562.

[8] A. Cichocki, R. Zdunek, A.Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorization: Applications to Exploratory Multiway Data Analysis and Blind Source Seperation*, Wiley, 2009.

[9] A. Owen and P. Perry, “Bi-cross-validation of the SVD and the nonnegative matrix factorization,” *The annals of applied statistics*, vol. 3, no. 2, pp. 564–594, 2009.

[10] B. Kanagal and V. Sindhwani, “Rank selection in low-rank matrix approximations: a study of cross-validation for NMFs,” in *Advances in Neural Information Processing Systems (NIPS'10)*, 2010.

[11] V.Tan and C. Fevotta, “Automatic relevance determination in nonnegative matrix factorization,” in *Signal Processing with Adaptive Sparse Structured Representations (SPARSS'09)*, 2009.

[12] C Stein, “Estimation of the mean of a multivariate normal distribution,” *Ann. Stat.*, vol. 9, no. 6, pp. 1135–1151, 1981.

[13] J.R. Magnus and H. Neudecker, *Matrix Differential Calculus*, Wiley, New York, NY, 1999.

[14] T. Minka, “Estimating a Dirichlet distribution,” Tech Report, MIT, 2000.

[15] D.L. Donoho, “De-noising by soft-thresholding,” *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[16] J.M. Bioucas-Dias and J.M.P. Nascimento, “Hyperspectral subspace identification,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 46, no. 8, 2008.