SVD-based Techniques in Time Series Analysis, in Signal Processing, and in Image Processing

Anatoly Zhigljavsky

Cardiff University

Sophia Antipolis, 5pm on April 24, 2008

- Least squares
- Total least squares
- SVD
- Structured total least squares
- Singular Spectrum Analysis
- Applications in time series analysis, in signal processing, and even in image processing

Least squares (LS) and total least squares (TLS):
methods for solution of an overdetermined system of equations

$$A\theta \approx B$$

where $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times d}$ are known and $\theta \in \mathbb{R}^{n \times d}$ is unknown.

Least squares estimator:

$$\{\hat{\theta}_{ls}, \Delta B_{ls}\} = \arg\min_{\theta, \Delta B} ||\Delta B||_F \qquad \text{subject to } A\theta = B + \Delta B$$

Solution (if A has full rank):

$$\hat{\theta}_{ls} = (A^T A)^{-1} A^T B$$

This is the unique solution of the optimally corrected system of equations $A\theta = \hat{B}_{ls}$ where $\hat{B}_{ls} = B + \Delta B_{ls}$

Total least squares

TLS looks for the minimal (in the Frobenius norm sense) corrections ΔA and ΔB for A and B that make the corrected system of equation $\hat{A}\theta = \hat{B}$ solvable; here $\hat{A} = A + \Delta A$, $\hat{B} = B + \Delta B$:

$$\{\hat{\theta}_{tls}, \Delta A_{tls}, \Delta B_{tls}\} = \arg \min_{\theta, \Delta A, \Delta B} ||[\Delta A, \Delta B]||_F \;\; \text{subject to } (A + \Delta A)\theta = B + \Delta B$$

TLS estimator is MLE for the errors-in-variables model with i.i.d. normal errors.

Almost equivalent form of the TLS problem :

$$\hat{C}_{tls} = \arg \min_{\hat{C}} ||C - \hat{C}||_F \qquad \text{subject to } rank(\hat{C}) \leq n$$

This is low-rank approximation problem with $C = [A, B]$. The constraint $\hat{A}\theta = \hat{B}$ is the rank constraint $rank(\hat{C}) \leq n$.

Solution to the TLS problem: SVD of C

Let

$$C = [A, B] = U\Sigma V^T \text{ where } \Sigma = diag(\sigma_1, \ldots, \sigma_{n+d})$$

be an SVD of C, $\sigma_1 \geq \ldots \geq \sigma_{n+d}$ be singular values of C, and define the partitioning

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$

where $\Sigma_1 = diag(\sigma_1, \ldots, \sigma_n)$, $\Sigma_2 = diag(\sigma_{n+1}, \ldots, \sigma_{n+d})$.
TLS solution exists iff the matrix V_{22} (of size $d \times d$) is nonsingular. It is unique iff $\sigma_n \neq \sigma_{n+1}$ and is then given by $\hat{\theta}_{tls} = -V_{12}V_{22}^{-1}$,

$$\Delta C_{tls} = [\Delta A_{tls}, \Delta B_{tls}] = -U diag(0, \Sigma_2)V^T,$$

$$\hat{C}_{tls} = C + \Delta C_{tls} = -U diag(\Sigma_1, 0)V^T.$$

LS versus TLS estimator

Let $d = 1$, $B = b$, $\theta = \theta$. Then

$$\hat{\theta}_{ls} = (A^T A)^{-1} A^T b, \quad \hat{\theta}_{tls} = (A^T A - \sigma_{n+1} I)^{-1} A^T b$$

Sums of squares of residuals:

$$\text{LS: } ||A\theta - b||^2 \to \min_{\theta}; \qquad \text{TLS: } \frac{||A\theta - b||^2}{||\theta||^2 + 1} \to \min_{\theta}$$

Regularized TLS estimator

$$\hat{\theta}_{tls} = (A^T A - \sigma_{n+1} I + \lambda L^T L)^{-1} A^T b \quad \text{with some } \lambda > 0$$

the solution $\hat{\theta}_{tls}$ satisfies the additional condition $||L\theta||_2 \leq \gamma$.

Weighted TLS

Let W be a positive definite matrix of size $m(n+d) \times m(n+d)$. Define

$$||C||_W = \sqrt{vec^T(C^T) W vec(C)}$$

then

$$\hat{C}_{Wtls} = \arg \min_{\hat{C}:\, rank(\hat{C}) \leq n} ||C - \hat{C}||_W$$

In general there is no closed-form solution of the Weighted TLS problem in terms of SVD. Optimization problem is non-convex and hard to solve.

Structured TLS

Let $x_1, x_2, \ldots, x_N$ be a time series, $M \leq N/2$ be some integer called the window length and $K = N - M + 1$. Define the so-called trajectory matrix

$$\mathbf{X} = (x_{ij})_{i,j=1}^{M,K} = \begin{pmatrix} x_1 & x_2 & x_3 & \ldots & x_K \\ x_2 & x_3 & x_4 & \ldots & x_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_M & x_{M+1} & x_{M+2} & \ldots & x_N \end{pmatrix} \qquad (1)$$

The matrix $\mathbf{X}$ is Hankel: it has equal elements on the diagonals $i + j =$const.
If the time series $x_1, x_2, \ldots, x_N$ satisfies a linear recurrence equation

$$a_0 x_t + a_1 x_{t+1} + \ldots + a_n x_{t+n} = 0 \quad (t = 1, 2, \ldots, N - n) \qquad (2)$$

then $rank(\mathbf{X}) \leq n$ for all $M \geq n$.
Structured TLS (the matrix $\mathbf{X}$ plays the role of C above):

$$\hat{\mathbf{X}} = \arg\min_{\tilde{\mathbf{X}}} ||\mathbf{X} - \tilde{\mathbf{X}}|| \quad \text{where } \tilde{\mathbf{X}} \text{ is Hankel and } rank(\tilde{\mathbf{X}}) \leq n$$

There is no closed-form solution of Structured TLS problem.

The model standard in signal processing

If the time series $x_1, x_2, \ldots, x_N$ is 'the sum of damped exponentials'

$$x_t = \sum_{j=1}^{n} c_j e^{d_j t} e^{\mathbf{i}(\omega_j t + \phi_j)}$$

then it satisfies a linear recurrence equation (2).

Basic SSA

Step 1 (*Computing the trajectory matrix*) use (1) to construct $\mathbf{X}$.
Step 2 (*Constructing a matrix for applying SVD*) compute $\mathbf{X}\mathbf{X}^T$.
Step 3 (*SVD of the matrix* $\mathbf{X}\mathbf{X}^T$) compute the eigenvalues and eigen–vectors of the matrix $\mathbf{X}\mathbf{X}^T$ and represent it in the form $\mathbf{X}\mathbf{X}^T = U\Lambda U^T$. Here $\Lambda = diag(\lambda_1, \ldots, \lambda_M)$ is the diagonal matrix of eigenvalues of $\mathbf{X}\mathbf{X}^T$ ordered so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M \geq 0$ and $U = (U_1, U_2, \ldots, U_M)$ is the corresponding orthogonal matrix of eigen–vectors of $\mathbf{X}\mathbf{X}^T$.
Step 4 (*Selection of eigen–vectors*) select a group of l $(1 \leq l \leq M)$ eigen–vectors $U_{i_1}, U_{i_2}, \ldots, U_{i_l}$.
Step 5 (*Reconstruction*) compute the matrix $\tilde{\mathbf{X}} = ||\tilde{x}_{i,j}|| = \sum_{k=1}^{l} U_{i_k} U_{i_k}^T \mathbf{X}$ as an approximation to $\mathbf{X}$. Transition to the one–dimensional series can now be achieved by averaging over the diagonals of the matrix $\tilde{\mathbf{X}}$.

Multivariate SSA

Assume that we have an L-variate time series $x_j = \left(x_j^{(1)}, \ldots, x_j^{(L)}\right)$, where $j = 1, \ldots, N$ and let M be window length. Using (1), we define the trajectory matrices $\theta^{(i)}$ $(i = 1, \ldots, L)$ of the one-dimensional time series $\{x_j^{(i)}\}$ $(i = 1, \ldots, L)$. The trajectory matrix θ can then be defined as

$$\theta = \begin{pmatrix} \theta^{(1)} \\ \cdots \\ \theta^{(L)} \end{pmatrix} .$$

The other stages of the Basic Multivariate SSA procedure are identical to the Basic SSA as described above with an obvious modification that the diagonal averaging should be applied to each of the L components separately.

SSA for Image Processing

Let us have an image I of size $h \times w$ represented in the form of a matrix,

$$I = \|I_{i,j}\|_{\substack{i=1,\ldots,h \\ j=1,\ldots,w}}$$

where, for example, the values $I_{i,j}$ code the intensity of either a colour or a grey level, $0 \leq I_{i,j} \leq 255$.

The window will now have size $u \times v$, with $1 \leq u \leq h$, $1 \leq v \leq w$.

The window is placed at all possible positions in the image.

Construction of the trajectory matrix

$$\begin{pmatrix} \ulcorner I_{1,1} & I_{1,2} & I_{1,3}\urcorner & I_{1,4} & \dots \\ I_{2,1} & I_{2,2} & I_{2,3} & I_{2,4} & \dots \\ \llcorner I_{3,1} & I_{3,2} & I_{3,3}\lrcorner & I_{3,4} & \dots \\ I_{4,1} & I_{4,2} & I_{4,3} & I_{4,4} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \rightarrow \begin{pmatrix} I_{1,1} & \ulcorner I_{1,2} & I_{1,3} & I_{1,4}\urcorner & \dots \\ I_{2,1} & I_{2,2} & I_{2,3} & I_{2,4} & \dots \\ I_{3,1} & \llcorner I_{3,2} & I_{3,3} & I_{3,4}\lrcorner & \dots \\ I_{4,1} & I_{4,2} & I_{4,3} & I_{4,4} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \rightarrow$$

$$\dots \rightarrow \begin{pmatrix} \dots & I_{1,w-3} & \ulcorner I_{1,w-2} & I_{1,w-1} & I_{1,w}\urcorner \\ \dots & I_{2,w-3} & I_{2,w-2} & I_{2,w-1} & I_{2,w} \\ \dots & I_{3,w-3} & \llcorner I_{3,w-2} & I_{3,w-1} & I_{3,w}\lrcorner \\ \dots & I_{4,w-3} & I_{4,w-2} & I_{4,w-1} & I_{4,w} \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \rightarrow \dots \rightarrow \begin{pmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & I_{h-3,w-3} & I_{h-3,w-2} & I_{h-3,w-1} & I_{h-3,w} \\ \dots & I_{h-2,w-3} & \ulcorner I_{h-2,w-2} & I_{h-2,w-1} & I_{h-2,w}\urcorner \\ \dots & I_{h-1,w-3} & I_{h-1,w-2} & I_{h-1,w-1} & I_{h-1,w} \\ \dots & I_{h,w-3} & \llcorner\ I_{h,w-2} & I_{h,w-1} & I_{h,w}\ \lrcorner \end{pmatrix}$$

which gives

$$\mathbf{X} = \begin{pmatrix} I_{1,1} & I_{1,2} & \dots & I_{1,w-2} & \dots & I_{h-2,w-2} \\ I_{1,2} & I_{1,3} & \dots & I_{1,w-1} & \dots & I_{h-2,w-1} \\ I_{1,3} & I_{1,4} & \dots & I_{1,w} & \dots & I_{h-2,w} \\ I_{2,1} & I_{2,2} & \dots & I_{2,w-2} & \dots & I_{h-1,w-2} \\ I_{2,2} & I_{2,3} & \dots & I_{2,w-1} & \dots & I_{h-1,w-1} \\ I_{2,3} & I_{2,4} & \dots & I_{2,w} & \dots & I_{h-1,w} \\ I_{3,1} & I_{3,2} & \dots & I_{3,w-2} & \dots & I_{h,w-2} \\ I_{3,2} & I_{3,3} & \dots & I_{3,w-1} & \dots & I_{h,w-1} \\ I_{3,3} & I_{3,4} & \dots & I_{3,w} & \dots & I_{h,w} \end{pmatrix}$$