WVD, Singular Value Decomposition Extended to Three Dimensional Space and Beyond

Xishuo Wang*
Geo-X Processing, Divestco Inc., Caglary, AB, Canada
xishuo.wang@divestco.com

Introduction

A 2D data structure arranged as a matrix ***A*** of N_r rows by N_c columns has its *Singular Value Decomposition* (***SVD***):

$$A_{ij} = (U \Lambda V^t)_{ij} = \sum_l \lambda_l u_{il} v_{jl} \, , \tag{1}$$

in which λ_l is the l^{th} eigenvalue, u_{il} is the l^{th} element of l^{th} eigenvector spanning the row direction of matrix ***A***, and v_{jl} is the j^{th} element of l^{th} eigenvector spanning the column direction of matrix ***A***. Columns of matrix ***U***/***V*** are eigenvectors $\boldsymbol{u}_l$ /$\boldsymbol{v}_l$.

Wagon Value Decompostion (*WVD*)

In this work I try to extend this expansion into dimensions of three and higher. For 3D data ***A***, I try to find its *Wagon value decomposition* (***WVD***):

$$A_{ijk} = \sum_l \lambda_l u_{il} v_{jl} w_{kl} \, , \tag{2}$$

where λ_l is the l^{th} *wagonvalue* and vectors $\boldsymbol{u}_l$, $\boldsymbol{v}_l$, & $\boldsymbol{w}_l$ are the l^{th} *wagonvectors* spanning the 1st, 2nd & 3rd dimensions, respectively.

The choice of name, *Wagon*, is in admiration of the wagon as a basic carrier capable of carrying lots and any type of loads!

In both (1) & (2), eigen/wagonvalues give a measure of the magnitude of the l^{th} component, since for regularity, all eigen/wagonvectors always have L_2 norm of unity.

Expanding (2) into higher dimensions is straightforward:

$$A_{ijk...mn} = \sum_l \lambda_l u_{il} v_{jl} w_{kl} ... y_{ml} z_{nl} \tag{3}$$

Note that eqns. (1) through (3) have the same form: data in *N*-dimensional space is the sum of a finite number of eigen/wagoncomponents, each component being an outer-product of *N* 1D eigen/wagonvectors spanning each and every one of the *N* dimensions, and scaled by respective eigen/wagonvalues.

Solving Eigencomponents

For the 2D case, many textbooks (e.g., Feng Kang et al, 1978) describe a simple way of finding the most dominant eigencomponent. It takes advantage of the fact that eigenvectors are mutually orthogonal. From an initial guess of the eigenvector of the rows, $\boldsymbol{u}$, $\boldsymbol{Au}$ gives rise to a first approximation of column eigenvector $\boldsymbol{v}$. $\boldsymbol{A}^t\boldsymbol{v}$ gives an improved estimate for $\boldsymbol{u}$, etc. After a few such trial operations, the solution for the first eigencomponent stablizes. Subtract the 1^{st} eigencomponent from matrix $\boldsymbol{A}$, go through the above procedure again, and we get the 2^{nd} eigencomponent. After a few such subtractions, there is negligible energy remaining in matrix $\boldsymbol{A}$.

Solving Wagoncomponents

Inspired by the above 2-D approach, I use a similar procedure to solve for wagoncomponents in N dimensions, with N greater than 2. Starting with an initial guess of the dominant (i.e., first) wagonvector in any one dimension, say the 1st dimension, I have uguess (unit L2 norm and refer to eqn.3), and then I form:

$$A_{jk\ldots mn} = \sum_{i=1}^{n_i} A_{ijk\ldots mn} u_i^{guess} \quad ,$$

in which the summation upper limit n1 is the length of the data in dimension one. Notice that after such an operation the dimension count of the "data" (i.e. $A_{jk\ldots mn}$) drops by one. And in general, a J-dimensional subset of the original N dimensional data will undergo a dimension reduction of one (i.e. J$\Rightarrow$ J-1) whenever I take the "multi-dimensional dot product" of the J-dimensional data with a 1D vector spanning any one of the J dimensions. By chaining together N-2 of these multi-dimensional dot products, each time employing an initial guess for the wagonvector associated with the dimension at hand, the dimension of the resulting data structure drops to two, namely Amn—a 2D matrix for which computation of the most dominant eigencomponent is readily accomplished via SVD (say, using the technique described in the last section). This chaining process is then done in reverse order (i.e., form the first dot product using the most recently estimated wagonvector, form the second dot product using the second most recently estimated wagonvector, etc.) to improve the wagonvector estimates. After a few such forward/reverse dot product chaining operations, solutions to wagonvectors (and wagonvalues) in all dimensions stabilize. Now we have the first wagoncomponent (i.e., the l=1 term in eqn.3). Subtract this component from A and repeat the above process to find the second wagoncomponent. This procedure is repeated until there is little energy left in A.

Nature of *WVD*

1) It is easy to prove that a finite number of wagoncomponents can losslessly reproduce data $\boldsymbol{A}$.

2) Wagonvectors do *NOT* form an orthogonal set (except *N*=2 case).

3) Data *super-slices* (for example, a time slice in stacked 3D volume; and in general a subset in $A_{jk\ldots mn}$ obtained by fixing k=k_0: $A_{jk_0\ldots mn}$) can be rearranged without altering the ***WVD*** (so long as elements of the relevant wagonvector are rearranged the same way).

4) It works the same way in both real and complex number systems.

Possible Applications of ***WVD***

1) Data compresion (e.g., Sacchi et al, 1998): for example, a 3D data volume of size N3, has N3 degrees of freedom, whereas L wagoncomponents are fully defined by only L∗(3N-2) degrees of freedom.

2) Noise reduction (e.g., Sacchi et al, 1998): reconstruct data by dropping insignificant wagoncomponents.

3) Edge detection (e.g., Gersztenkorn & Marfurt, 1996): similar to the SVD approach but our approach keeps the "stereo view" of the 3D data, as it avoids the compromise of rearranging 3D data into 2D.

Dimensionality of Seismic Data

For prestack data, I can think of up to eight dimensions: source *x*, source *y*, receiver *x*, receiver *y*, time, time-lapse, source type and receiver type. Four other "reluctant" dimensions are *x* & *y* of single source/sensor in source/receiver groups. For poststack data, the first four and the last "reluctant" four disappear and are replaced by CMP *x* & CMP *y*, but the time and time-lapse dimensions remain, giving a total of four dimensions (assuming we have just one source type & one receiver type).

In the source type dimension, the wagonvector length could be 4; for example, explosive plus 3 vibrator types: up/down, east/west, north/south.

In the receiver type dimension, the wagonvector length could also be 4; for example, pressure plus 3 components of particle velocity: vertical, radial & transverse.

Now, *that* is complete time-lapse 3D *land* seismic data and our every day "conventional" data is a *small* subset of it!

Three is much "*larger*" than Two

SVD is quite different from ***WVD***, mostly in the orthogonality (or not) of eigen (wagon) vectors. In other fields of science and/or mathematics, we are familiar with lots of such "harsh boundaries" from 2 to 3 (and beyond). One famous one is in geometry: using a compass and a straight edge a finite number of times to divide an arbitrary angle into 2/3 equal parts is easy/impossible. With this observation in mind I am not too embarrased at not being able to give a rigorous proof of the way I solve wagoncomponents.

Data Examples

A synthetic example is shown below. The 3D data are modeled using eqn.3 with two known wagoncomponents. In all three figures, the coloured one is a time slice, the upper right is an in-line slice and lower right a x-line slice. Thin red lines mark the “slicing” positions.

Fig.1m is the clean data, fig.1n is the same data with added noise, and fig.1w is synthesized after wagoncomponents are solved from noisy data of fig.1n. Cross correaltion between fig.1m and fig.1w is 98.3%. The performance of WVD is good: despite the high noise level, wagoncomponents are well resolved.

I will also give application examples in noise reduction & edge detection on seismic data.

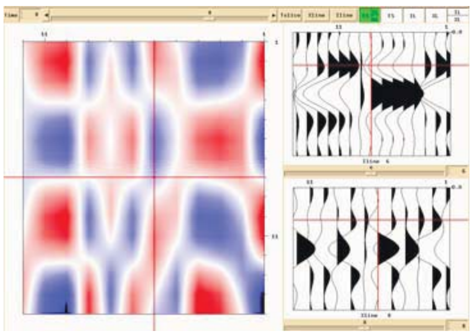

Figure 1m

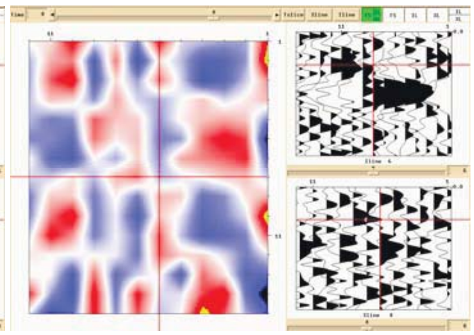

Fig ure 1n

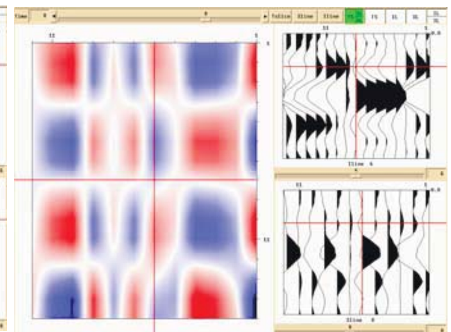

Figure 1w

References

Feng, Kang et al, 1978, Numerical methods, National Defense Industry Publication, China,15034.1653.

Gersztenkorn, A. & K.J. Marfurt, 1996, Eigenstructure based coherence computations, 66th Annual International Meeting, SEG Expanded abstract, 328-331.

Sacchi, M.D., Ulrych, T.J. & C. Magnuson 1998, Eigenimage analysis of common offset sections: pre-stack data compression and S/N ratio enhancement, CSEG National Convention, Calgary, Canada.