NAME

unmix − partial and full spectral unmixing

SYNOPSIS

unmix **−S** *end_member_spectra* | **−T** *train_image* [**−full**] [[**−saveS** *end_member_spectra* [**−a**]]
[**−saveC** *covmat_file*] [**−u**]] [[**−po** *n pow_1 ... pow_n*] [**−pr** [**−sqrt**]] [**−r** [**−sqr**]] [**−f**]]
[**−li** [**−c** | **−C**] | **−ls** [**−k**] [**−w**]] [**−i**] [**−1**] [**−L**] < inseq > outseq

DESCRIPTION

unmix performs partial and full linear unmixing. The abundance(s) of the end-member(s) can be constrained to be non-negative and not greater than one, and if several end-members are specified the abundances can be constrained to sum to one (full unmixing) or to a quantity not greater than one (partial unmixing). The term linear means linear in the coefficients (the abundances), not in the variables (the end-member spectra), see **−po**, **−pr** or **−r** below. For each pixel *unmix* calculates abundances for each of the input end-member spectra. Default action is partial unmixing with non-negative abundances the sum of which is not greater than one. If full unmixing is requested by specifying **−full** the abundances sum to one. Unconstrained linear regression can be requested by specifying **−li**.

The input end-member spectra are read from a double HIPS file stored row-wise if **−S** is specified. Alternatively, the end-member spectra can be calculated from the input if **−T** is specified. In this case **−saveS** causes the calculated end-member spectra to be written to a double HIPS file stored row-wise. If **−a** is specified with **−saveS**, the augmented end-member spectra are written, see **−po**, **−pr** or **−r** below. You must specify either **−S** or **−T**. **−saveC** writes the dispersion matrices for the variables in 1) each class, 2) all classes (also the background, class 0), 3) all classes pooled (excluding class 0) and 4) the entire input as individual frames to a double HIPS file.

This technique requires that the number of degrees of freedom in the least squares estimation is at least one. Depending on the options **−1** and **−i** this corresponds to requesting that the number of end-members is less than or equal to the number of input variables (approximately). If this is not the case or if for instance there are interactions between end-member spectra the original variable vector can be augmented by non-linear transformations of the original variables. The most obvious transformations are powers, products and ratios. The inclusion of these transformed variables is specified with **−po**, **−pr** and **−r**, respectively. For example, specifying **−po** *2 0.5 2.0* will augment the original variables vector with squareroots and squares of the original variables. If **−pr** is specified, **−sqrt** will replace all possible products with their squareroots to avoid very skewed distributions. If **−r** is specified, **−sqr** will replace all possible ratios with their squares. In case any of these options to augment the feature vector are specified (and often also if not) it is a good idea to check for correlations between the input variables. **−saveC** facilitates this check. If **−u** is specified no unmixing takes place. This is useful if end-member spectra or covariance matrices between input variables are needed only.

unmix reads byte, short, int, float and double and writes float. All calculations are performed in double. Input must be band-interleaved by line (BIL), output is BIL. Output consists of frames with abundances for each end-member spectrum, one frame with R-squared, and one frame with the root mean squared error. If **−L** is specified, frames with Lagrange multipliers for constraint on each abundance and for sum of abundances are also output.

OPTIONS

−S *end_member_spectra*

read end-member spectra stored row-wise from a double HIPS image

−T *train_image*

calculate end-member spectra from input, one spectrum for each class in the byte HIPS training

image (one class is defined by a non-zero number in the training image, 0 is background); the spectrum is the average of pixels for each training image value

−full perform full unmixing (abundances sum to one) as opposed to partial unmixing (abundances sum to something not greater than one)

−saveS *end_member_spectra*
write end-member spectra stored row-wise to a double HIPS image (with **−T** only)

−a write augmented end-member spectra stored row-wise to a double HIPS image (with **−saveS** only)

−saveC *covmat_file*
write dispersion matrices for the variables in 1) each class, 2) all classes (also the background, class 0), 3) all classes pooled (excluding class 0) and 4) the entire input as individual frames to a double HIPS file to facilitate check for correlations between input variables; if **−po**, **−pr** or **−r** are specified **−saveC** calculates dispersions as the actual unmixing goes on; if not **−saveC** calculates dispersions before unmixing

−u do not unmix (with **−saveS** or **−saveC** only)

−po *n pow_1 ... pow_n*
use original variables as well as *n* powers of original variables

−pr use original variables as well as all possible products of original variables

−sqrt use original variables as well as squareroots of all possible products of original variables (with **−pr** only)

−r use original variables as well as all possible ratios of original variables

−sqr use original variables as well as squares of all possible ratios of original variables (with **−r** only)

−f augment feature vector only and not end-member spectra (with **−po**, **−pr** and **−r** only)

−li apply Linpack routines dsico/dsisl (if **−1** is specified) or Linpack routines dpoco/dposl (if not)

−c rescale abundances so that they lie in the interval [0;1] and add to one (with **−li** only)

−C set negative abundances to zero, rescale abundances so that they lie in the interval [0;1] and add to one (with **−li** only)

−ls apply LSSOL version 1.05-2 April 93; this implies non-negative abundances (default)

−k keep previous abundance estimates as starting guesses for current estimates (with **−ls**, default)

−w retain satisfied constraints from previous abundance estimates as starting constraints for current estimates (with **−ls**, "warm start", default)

−i estimate intercept as well as abundances; intercept is written after individual abundances just before R-squared and RMSE

−1 abundances must sum to one

−L output Lagrange multipliers for linear constraints, first for individual abundances (if any), last for sum of abundances

−d *level* debug; this produces (enormous amounts of) output to Fortran I/O file 9 (if **−ls** is specified) or to *stderr* (if not); *level* should be 0, 1, 5, 10, 20 or 30 (if *level* is not 0 with **−ls** *stdout* is contaminated)

BUGS

The training image must contain consecutively numbered classes. **−li** with **−1** is not implemented. **−saveS** *covmat_file* **−a** augments the values in the end-member spectra and does not calculate mean values

of the augmented features according to the **−T** specification. To avoid this use **−saveC** *covmat_file* **−f** and use *cov2spec* on *covmat_file*.

REFERENCES

ENVI User's Guide (1997). Research Systems, Inc.

J.J. Dongarra, C.B. Moler, J.R. Bunch and G.W. Stewart (1979). *LINPACK User's Guide.* SIAM, Philadelphia.

P.E. Gill, S.J. Hammarling, W. Murray, M.A. Saunders and M.H. Wright (1986). *User's Guide for LSSOL (version 1.0).* Department of Operations Research, Stanford University.

J.T. Kent and V. Mardia (1988). Spatial Classification Using Fuzzy Membership Models. *IEEE Transactions on Pattern Analysis and* **10**(5), 659-670.

A. A. Nielsen (2001). Spectral Mixture Analysis: Linear and Semi-parametric Full and Iterated Partial Unmixing in Multi- and Hyperspectral Image Data. *International Journal of Computer Vision*, **42**(1-2), 17-37, and *Journal of Mathematical Imaging and Vision*, **15**(1-2), 17-37.

J.J. Settle and N.A. Drake (1993). Linear Mixing and the Estimation of Ground Cover Proportions. *International Journal of Remote Sensing*, **14**(6), 1159-1177.

SEE ALSO

bil(1), discrim(1), disc(1), cov2spec(1)

ACKNOWLEDGEMENT

Dr. Rasmus Larsen, IMM, wrote C-code for provisional means calculation of means and covariance (dispersion) matrices with **−saveC**.

AUTHOR

Allan Aasbjerg Nielsen, M.Sc., Ph.D.
IMM, Informatics and Mathematical Modelling
Technical University of Denmark, Building 321
E-mail aa@imm.dtu.dk, Internet http://www.imm.dtu.dk/~aa