A note on Mahalanobis and related distance measures in WinISI and Unscrambler

Journal:	*Journal of Near Infrared Spectroscopy*
Manuscript ID	Draft
Manuscript Type:	Original Research Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Garrido-Varo, Ana; University of Cordoba, Non-Destructive Spectral Sensors Unit, Faculty of Agriculture and Forestry Engineering Garcia-Olmo, Juan; University of Cordoba, NIR/MIR Spectroscopy Unit, Central Service for Research Support Fearn, Tom; University College London, Statistical Science
Keywords:	Mahalanobis distance, Leverage, Hotelling's T2, Principal components, Near infrared spectroscopy, Outliers
Abstract:	In identifying spectral outliers in near infrared calibration it is common to use a distance measure that is related to Mahalanobis distance. However, different software packages tend to use different variants, which leads to a translation problem if more than one package is used. Here the relationships between squared Mahalanobis distance D2, the GH distance of WinISI, and the T2 and leverage (L) statistics of Unscrambler are established as D2 = T2 ☐ L*n ☐ GH*k, where n and k are the numbers of samples and variables respectively in the set of spectral data used to establish the distance measure. The implications for setting thresholds for outlier detection are discussed. On the way to this result the principal component scores from WinISI and Unscrambler are compared. Both packages scale the scores for a component to have variances proportional to the contribution of that component to total variance, but the WinISI scores, unlike those from Unscrambler, do not have mean zero.

A note on Mahalanobis and related distance measures in WinISI and Unscrambler

A Garrido-Varo[1], J Garcia-Olmo[2] and T Fearn[3]

[1]Non-Destructive Spectral Sensors Unit, Faculty of Agriculture and Forestry Engineering, University of Cordoba, Spain
[2]NIR/ MIR Spectroscopy Unit, Central Service for Research Support, University of Cordoba, Spain
[3]Department of Statistical Science, University College London, UK.

Corresponding author: T Fearn, Department of Statistical Science, UCL, Gower Street, London WC1E 6BT, UK
Email: t.fearn@ucl.ac.uk

Keywords

Mahalanobis distance, Leverage; Hotelling's T^2, Principal components, Near infrared spectroscopy, Outliers

Abstract

In identifying spectral outliers in near infrared calibration it is common to use a distance measure that is related to Mahalanobis distance. However, different software packages tend to use different variants, which leads to a translation problem if more than one package is used. Here the relationships between squared Mahalanobis distance D^2, the GH distance of WinISI, and the T^2 and leverage (L) statistics of Unscrambler are established as $D^2 = T^2 \approx L*n \approx GH*k$, where n and k are the numbers of samples and variables respectively in the set of spectral data used to establish the distance measure. The implications for setting thresholds for outlier detection are discussed. On the way to this result the principal component scores from WinISI and Unscrambler are compared. Both packages scale the scores for a component to have variances proportional to the contribution of that component to total variance, but the WinISI scores, unlike those from Unscrambler, do not have mean zero.

Introduction

One of the necessary steps in developing or applying near infrared (NIR) calibrations is to check for spectral outliers. A common way to decide whether a given spectrum is an outlier is to calculate, using a distance measure that takes into account the pattern of spectral variability in the training set, its distance from the mean spectrum of that set. This distance can then be compared with some threshold that is either based on an assumption of some statistical distribution or is simply a rule of thumb based on experience.

So long as the user is faithful to one software package this approach is simple to apply. Problems can arise however if more than one package is used, because different packages tend to use different variants of the same underlying measure, Mahalanobis distance [1, 2]. Then the sort of question that can arise is "A threshold of 3 for GH in WinISI works well for my applications, what is the corresponding threshold for a leverage from Unscrambler?" The investigations reported here were carried out with the aim of answering some of these questions, by comparing the distance measures of WinISI and Unscrambler with Mahalanobis distances calculated from the same data set by Matlab code that implements the textbook formula.

An added complication is that the Mahalanobis formula involves the inversion of a variance matrix calculated from the spectra in the training set. This inversion is unstable in high dimensions and so the spectra need to be projected onto a lower dimensional space before the distances can be calculated. The obvious options are to use scores on either principal components (PCs) or partial least squares (PLS) factors. The use of PCs has the advantage that one can use them to screen for outliers before developing calibrations. PCs are also simpler to calculate and much more likely to match between different software packages, and so this is the approach adopted here. Given that PC scores needed to be calculated in each of the three packages, the opportunity was taken to compare the scores also.

A priori the PC scores might be expected to differ between packages, because there is more than one option for scaling them, for example a vector of scores can be scaled to have length 1 or a squared length reflecting the contribution of the PC to total variance, to list just two of the most common options. In addition the sign of the PC loadings, and hence of the scores, is arbitrary, because if v is an eigenvector of the matrix M then so is –v. Different packages will often produce scores with different signs, and even using the same package the removal of one spectrum from the calculation can result in the sign of the PC flipping. This is unimportant, but can be disconcerting when a plot appears to change completely after a very small change to the data. None of this should matter so far as the distance calculations are concerned, since Mahalanobis distance is scale invariant, but it is still of interest to compare the scores from the three packages.

Materials and Methods

Data

The data set used to compare the results on different software comprised 349 spectra of liquid samples of subcutaneous fat of Iberian pigs, measured on a Foss-NIRSystems 6500 monochromator. The wavelength range was 400 to 2498nm in steps of 2nm, and thus the data matrix X was of dimension 349 x 1050. Since the purpose of the current investigation was simply to compare the software, no pre-treatments were applied to the spectra. The data were exported to a .csv file for transfer to Matlab, and to JCAMP-DX format for transfer to Unscrambler.

Software

The Matlab code in the appendices was run using version R2016b (The MathWorks Inc., Natick, MA, USA). The WinISI software was version 4.8 (FOSS Analytical A/S, Hillerød, Denmark), and the Unscrambler software was Unscrambler X version 10.4.1 (CAMO Software AS, Oslo, Norway). Calculations with much earlier versions of Win ISI and Unscrambler (see the acknowledgement) gave equivalent results.

Mahalanobis distance, Hotelling's T^2, and leverage

Mahalanobis [3] invented the statistic that bears his name [1] as a way to measure the distance between two groups of observations in k-dimensional space while taking into account the fact that the k-variables may have differing scales and may be intercorrelated. In this case the formula for the squared distance between the groups would be

$$D^2 = (m_1 - m_2)^T S^{-1} (m_1 - m_2)$$

where m_1 and m_2 are the k x 1 vectors of means for the two groups and S is a k x k within-group variance matrix, all of these quantities being estimated from the data on the two groups. This statistic is closely related to the subsequently developed Hotelling's T^2, which is a multivariate version of the two-sample t-test. The relationship when there are n_1 observations in group 1 and n_2 in group 2 is

$$T^2 = (n_1 n_2/(n_1 + n_2))\, D^2$$

When the observations come from multivariate normal distributions, the distribution of T^2, and hence that of D^2, is known to be a multiple of an F distribution. Full details of all the above can be found in almost any multivariate statistics textbook, for example the one by Krzanowski [2].

In NIR calibration a version of D^2 is commonly used to measure the distance of a single spectrum, or more precisely the scores of this spectrum on a set of PCs or PLS factors, from the centre of the cloud of calibration set spectra. Then the formula becomes

$$D^2 = (x - m)^T V^{-1} (x - m) \qquad (1)$$

where x is the k x 1 vector of spectral data for the observation of interest, and the k x 1 mean vector m and k x k variance matrix $V = X^T X/(n-1)$ are both calculated from the n x k matrix X of spectral data for the calibration set.

With m and V both calculated from the full calibration set of n observations, most of the random variability in this version of D^2 comes from x. If we assume x to be randomly sampled from the same multivariate normal distribution as the calibration set, and ignore that fact that m and V are sample estimates of population parameters, then the distribution of D^2 will be approximately chi-squared on k degrees of freedom. This distribution has a mean value of k. This same approximate distribution applies regardless of whether x belongs to the calibration set or is a new observation.

Leverage L is a statistic developed for identifying influential observations in multiple linear regression [4]. It is also used to identify outliers in NIR calibration [5, 6]. Starting from the standard statistical definition its relationship with D^2 should be

$$L = 1/n + D^2/(n-1). \quad (2)$$

The factor of (n-1) arises because leverage uses $(X^TX)^{-1}$ in place of V^{-1} in a formula analogous to Equation 1, and the 1/n represents the influence of x on the mean m. Because of the context in which it was designed to be used, there is an assumption here that x is one of the n rows of X and so has contributed to the estimation of m. In cases where it is not, for example in comparing the spectra of prediction samples with those of a calibration set, it would make sense to omit the 1/n, though if this makes any practical difference the calibration set is too small.

PCA calculations

PCA scores were calculated in Matlab using the code in Appendix 1 with the scaling parameter 'scal' set to 1. This uses the Matlab SVD function to decompose X and scales each vector t of scores so that its squared length t^Tt is equal to the corresponding eigenvalue of X^TX. In other words, the variance of the scores for a component is proportional to the contribution of that component to the total variance in X. The columns of X are centered but not rescaled in the computation. This would correspond to the 'covariance' option in most standard statistical packages. In WinISI there are no options; in Unscrambler mean centering and the SVD algorithm were selected.

Distance calculations

Squared Mahalanobis distances D^2 were calculated in Matlab using the code in Appendix 2. This implements the textbook formula in Equation 1. In WinISI and Unscrambler the desired statistics were selected from the appropriate menus, choosing to base the calculations on 10 PCs in each case.

Results and discussion

The aim of this investigation was to relate the formulas used by the packages compared, not to establish the accuracies of the computations. Thus statements like 'the scores matched' should be interpreted as meaning only that the correspondence was good enough to establish equivalence beyond reasonable doubt, not as a claim of identity to the level of machine precision. In any case the rounding errors involved in the transfers of data probably dominate the errors in any internal computations.

Comparison of PC scores

As expected, there were differences in signs between some of the scores returned by the three programs. With these differences resolved the scores produced by Unscrambler matched those from Matlab. The scores from WinISI had the same scaling as those from the other two programs but, unlike the other two sets, were not centred on zero. Figure 1 shows the Matlab and WinISI scores on the first two PCs after the directions of the Matlab scores have been reversed so that the plots match.

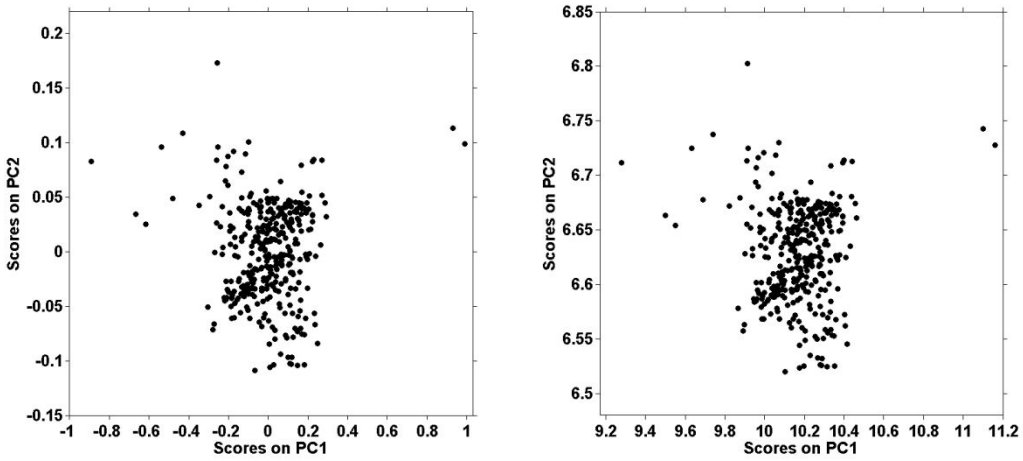

Figure 1. Scatter plots of first 2 PC scores from Matlab (left) and WinISI (right).

Further comparisons revealed that although WinISI centres the columns of X before carrying out the PCA, the scores it produces correspond to applying the loadings to an uncentred X. Running the Matlab code in Appendix 1 and then calculating X*L reproduces the WinISI scores. This is presumably done because it simplifies the calculation of scores for future samples by eliminating the need to subtract the mean spectrum of the set used to carry out the PCA.

Comparison of Mahalanobis and other distances

The T^2 results from Unscrambler correspond to the squared Mahalanobis distances D^2 from the Matlab program. The leverages L correspond to $D^2/(n-1) + 1/n$ as in Equation 2. The Unscrambler reference manual [7], which is generally quite precise, clearly defines leverage as the standard statistic, but is uncharacteristically vague about T^2.

The relation between WinISI's GH and D^2 was found to be

$$GH = (n/(n-1)).D^2/k$$

where k is the number of PCs used for the calculation of D^2 and GH. The factor n/(n-1) is presumably due to the use by WinISI of a divisor of n rather than the more usual n-1 in the calculation of the variance matrix in the Mahalanobis formula. The division by k scales GH to have typical values of around 1 whatever the value of k.

Ignoring subtleties like factors of n/(n-1), which is 1.003 for the data set used here for example, the relationships may be summarised as

$$D^2 = T^2 \approx L*n \approx GH*k. \quad (3)$$

Implications for thresholds

Using the approximate relationships in Equation 3, a threshold of T_M for squared Malahanobis distance corresponds to thresholds of T_M/k for GH, T_M for Unscrambler's T^2 statistic and T_M/n for Unscrambler's leverage statistic. So, for example, the GH rule of thumb of 3 would convert to 3k for squared Mahalanobis distance or for T^2, and to 3k/n for leverage. In fact 3k/n, along with 2k/n, is a commonly suggested rule of thumb for leverage [6].

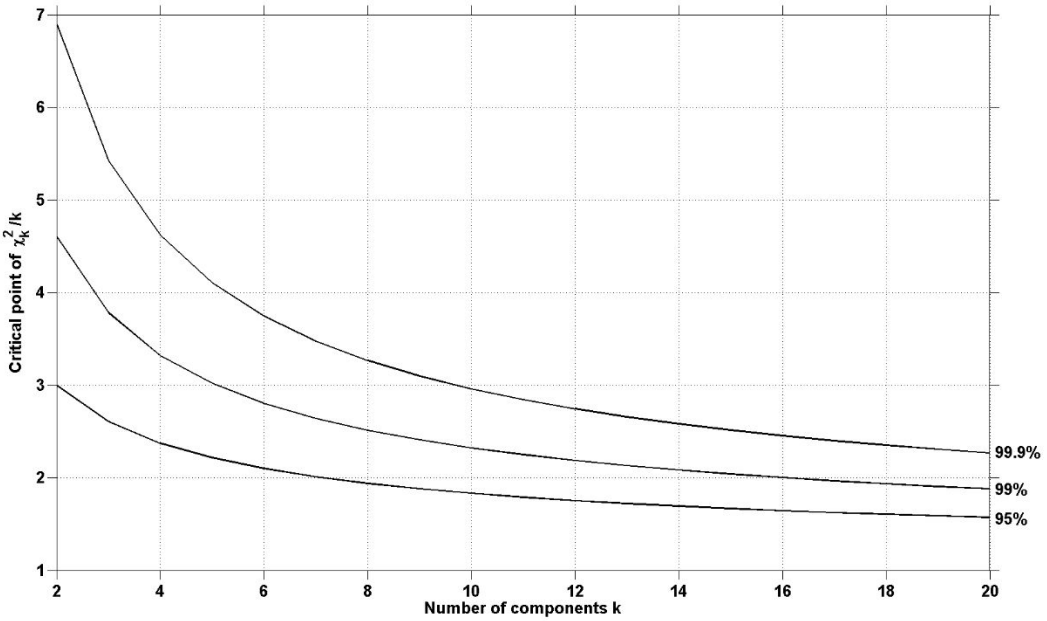

Figure 2. Thresholds for GH at three probability levels based on a chi-squared distribution with k degrees of freedom for D^2

The alternative to a rule of thumb is to base a threshold on a probability distribution. If we assume a multivariate normal distribution for the PC scores, the approximate distribution for D^2 is chi-squared on k degrees of freedom. Figure 2 shows thresholds at probability levels of 95, 99 and 99.9% for GH based on this distribution. While these probabilities should not be taken too seriously, since the normality assumption is unlikely ever to be correct, the figure does suggest that 3 is a sensible choice for a fixed threshold for GH, at least for modest k.

Acknowledgements

The authors would like to dedícate this article to the memory of Prof. Dr. Tomas Isaksson, who passed away on 12 July 2012. This work began as a collaboration between the first two authors and Tomas, paused upon his death, and has only now been completed.

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

References

[1] Mahalanobis PC. On tests and measures of group divergence. *Journal of the Asiatic Society of Bengal* 1930; 26; 541-588.

[2] Krzanowski WJ. *Principles of Multivariate Analysis: A User's Perspective.* 2nd ed. New York: OUP, 2000.

[3] Ghosh JK and Majumdar PP. Mahalanobis, Prasantra Chandra. In: Armitage P and Colton T (eds) *Encyclopaedia of Biostatistics.* New York: Wiley, 1998, pp. 2372-2375.

[4] Weisberg S. *Applied Linear Regression.* 3rd ed. New York: Wiley, 2005.

[5] Martens H and Næs T. *Multivariate Calibration*. Chichester UK: Wiley, 2001.

[6] Næs T, Isaksson T, Fearn T and Davies T. *Multivariate Calibration and Classification.* Chichester UK: NIR Publications, 2002.

[7] https://www.camo.com/downloads/U9.6%20pdf%20manual/The%20Unscrambler%20Method%20References.pdf

Appendix 1. Matlab code for PCA

This function uses Matlab's singular value decomposition on the n x p matrix X, thus avoiding the need to compute the matrix product X^TX. The 'econ' option stops the decomposition when the number of eigenvalues extracted corresponds to the smaller of n and p, the maximum number of nonzero eigenvalues for a matrix of this size. The PCA scores and loadings are easily computed from this decomposition.

```
function [S,L,v,m] = pcomp(X,k,scal)
% PCA
% Usage [S,L,v,m] = pcomp(X,k,scal)
% Inputs
%   X ..... n x p matrix of spectra (in rows)
%   k ..... number of components to return
%   scal .. scalar, options for scaling the scores
%           0 - orthonormal scores
%           1 - scores scaled to have squared length equal to
%                  the corresponding eigenvalue of X'X
% Outputs
%   S .... n x k matrix of scores
%   L .... p x k matrix of loadings
%   v .... k x 1 vector of eigenvalues of X'X
%   m .... 1 x p vector, column means of X
%
% Note: to calculate scores for a new data matrix M use
%       Scor = (M-ones(size(M,1),1)*m)*L
```

```
%
% Tom Fearn, February 2019
%
m = mean(X,1);                  % column means of X
Xc = X - ones(size(X,1),1)*m;   % centre columns of X
[U,E,V] = svd(Xc,'econ');       % decompose Xc as U*E*V'
e = diag(E);                    % all the eigenvalues of X
e = e(1:k);                     % first k eigenvalues of X
v = e.^2;                       % first k eigenvalues of X'X
if scal==0
    S = U(:,1:k);               % k scores, orthonormal
    L = V(:,1:k)*diag(1./e);    % k loadings, scaled to give
                                %    orthonormal scores
else
    S = U(:,1:k)*diag(e);       % k scores, scaled by eigenvalues
    L = V(:,1:k);               % k loadings, orthonormal
end
end
```

Appendix 2. Matlab code for Mahalanobis distance

Two separate functions were used. The first, MVinv, calculates the mean and inverse variance matrix from a set of data, the second takes these as inputs and calculates Mahalanobis distances. To use the PCA function above and the two functions below to calculate Mahalanobis distances using k PCs from a data matrix X, the code would be

```
[S,L,v,mx] = pcomp(X,k,1);
[ms,Vi] = MVinv(S);
D = MD2(S,ms,Vi);
```

Function MVinv

```
function [m,Vi] = MVinv(X)
% Calculates mean vector and inverse variance matrix of data set X
% Usage[m,Vi] = MVinv(X)
% Input
%   X ... n x p matrix of data, cases in rows
%
% Outputs
%   m ... 1 x p vector, column means of X
%   Vi .. p x p symmetric matrix, inverse of covariance matrix of X
%
% Notes
% 1. If p is very large, and in particular if p>n-1, computing the
%    inverse of the covariance matrix will give unstable results
% 2. The divisor in the computation of the covariance matrix is n-1
%
% Tom Fearn, February 2019
%
m = mean(X);
V = cov(X);
Vi = inv(V);
```

```
end
```

Function MD2

```
function D = MD2(X,x0,Vi);
% Calculates the squared Mahalanobis distance of each row of X from
%   each of the rows in x0 using the inverse covariance matrix Vi
% Usage D = MD2(X,x0,Vi)
% Inputs
%   X ... n x p, data matrix, observations in rows
%   x0 .. k x p, centres for calc of MD, in rows
%   Vi .. p x p, inverse variance matrix for calc of MD
%
% Output
%  D ... n x k matrix, squared Mahalanobis distances between each
%                      row in X and each row in x0
%
% Notes
% 1.  To get distances from a calibration set mean set x0=m where
%     m is the 1 x p mean vector of the cal set as given by MVinv
% 2.  To get distances from several individual observations
%     set these observations as the rows of x0
% 3.  This will obviously crash if the dimensions p of the inputs
%     do not match!
%
% Tom Fearn, February 2019
%
% make sure that if x0 is a vector it is a row vector
if size(x0,2)==1; x0=x0'; end;
%
% set up storage for results
n = size(X,1); k = size(x0,1);
D = zeros(n,k);
%
% loop over rows of x0
for i = 1:k
    m = x0(i,:);                  % set i'th row as as the centre
    Xc = X - ones(n,1)*m;         % center X
    D(:,i) = sum((Xc*Vi).*Xc,2); % calculate squared MDs
    % Note: The more obvious code would be diag(Xc*Vi*Xc') but
    % this would be less efficient
end
end
```

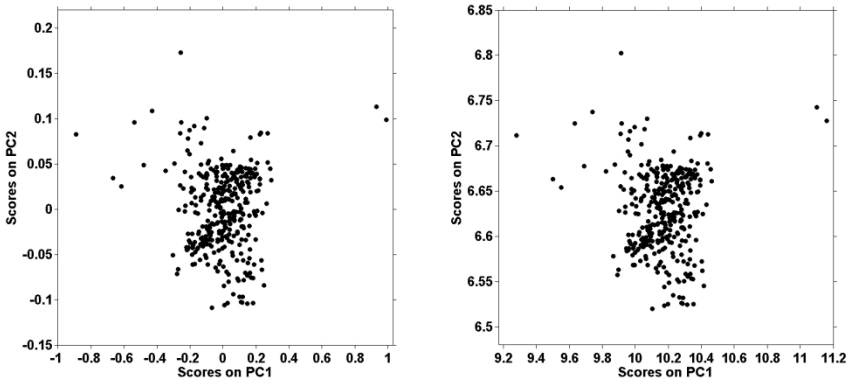

Figure 1. Scatter plots of first 2 PC scores from Matlab (left) and WinISI (right).

508x287mm (96 x 96 DPI)

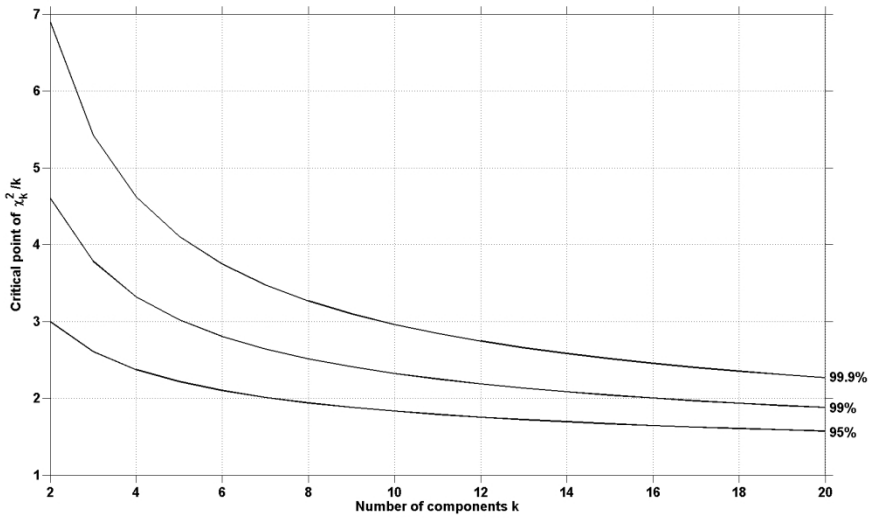

Figure 2. Thresholds for GH at three probability levels based on a chi-squared distribution with k degrees of freedom for D2

508x287mm (96 x 96 DPI)