

WHITENING AND COLORING TRANSFORMATIONS FOR MULTIVARIATE GAUSSIAN DATA

A Slecture for ECE 662
by Maliha Hossain

1 Introduction

This slecture discusses how to whiten data that is normally distributed. Data whitening is required prior to many data processing techniques such as principal component analysis, independent component analysis, etc. The slecture also covers how to transform white data so that it resembles data drawn from some Gaussian distribution with known mean and covariance. This will be useful material for when the reader would like to generate data points from non-white Gaussian distributions.

Throughout this slecture, we will denote the probability density function (pdf) of the random variable \mathbf{X} as $f_{\mathbf{X}} : \mathbb{R}^d \rightarrow \mathbb{R}$ where $\mathbf{X} \in \mathbb{R}^d$ is a d -dimensional Gaussian random vector with mean μ and covariance matrix Σ . This slecture assumes that the reader is familiar with topics from linear algebra including but not limited to vector spaces, eigenvalues, and eigenvectors. It is no surprise that eigendecomposition plays an important role in data whitening since we would ultimately like to produce a diagonal covariance matrix.

The pdf of \mathbf{X} is given by

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right] \quad (1)$$

2 Whitening Transform

Let \mathbf{X} be a multivariate Gaussian random vector with arbitrary covariance matrix Σ and mean vector μ . We would like to transform it to another random vector whose covariance is the identity matrix. This means that the components of our new random variable are uncorrelated and have variance 1. This process is called whitening because the output resembles a white noise vector.

There are two steps to the procedure. First, we must decorrelate the components of the vector. This produces a data point drawn from a distribution with a diagonal covariance matrix. Finally, we must scale the different components so that they have unit variance. In order to do this, we need the eigendecomposition of the original covariance matrix Σ . This is either known or it can be estimated from your data set. We assume that Σ is positive definite. We also assume wlog that \mathbf{X} has zero mean. If not, then subtract the mean (estimated or otherwise) from your samples such that

$$\mu = \mathbb{E}[\mathbf{X}] = \mathbf{0} \quad (2)$$

$$\Rightarrow \Sigma = \mathbb{E}[\mathbf{X}\mathbf{X}^T] \quad (3)$$

The covariance matrix can be expressed as follows

$$\Sigma = \Phi \Lambda \Phi^{-1} = \Phi \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} \Phi^{-1} \quad (4)$$

where Λ is a diagonal matrix with the eigenvalues, λ_i , of Σ on the diagonal and Φ is the matrix of corresponding eigenvectors, \mathbf{e}_i . In addition, the columns of Φ are orthonormal so that

$$\Phi^{-1} = \Phi^T \quad (5)$$

Let

$$\mathbf{Y} = \Phi^T \mathbf{X} \quad (6)$$

Then \mathbf{Y} is a random vector with a decorrelated multivariate Gaussian distribution with variances λ_i on the diagonal of its covariance matrix. The next step then is quite obvious. Let

$$\mathbf{W} = \Lambda^{-\frac{1}{2}} \mathbf{Y} = \Lambda^{-\frac{1}{2}} \Phi^T \mathbf{X} \quad (7)$$

\mathbf{W} is then random vector with the standard multivariate distribution. Note that $\mathbf{X} = \mathbf{x}$, $\mathbf{Y} = \mathbf{y}$, and $\mathbf{W} = \mathbf{w}$ are at the same Mahalanobis distance away from their respective means (in this case, the origin) and in the case of \mathbf{W} , the Mahalanobis distance and the Euclidean distance are the same.

Now that you have the procedure, you may be asking why this transform works. Well, we can derive the covariance matrix of \mathbf{Y} . Recall that its mean is the zero vector so its covariance is

$$\begin{aligned} E[\mathbf{Y}\mathbf{Y}^T] &= E[\Phi^T \mathbf{X}\mathbf{X}^T \Phi] \\ &= \Phi^T E[\mathbf{X}\mathbf{X}^T] \Phi \\ &= \Phi^T \Sigma \Phi \\ &= \Phi^T \Phi \Lambda \Phi^T \Phi \\ &= \Lambda \end{aligned}$$

This proves what we previously stated; the components of \mathbf{Y} are uncorrelated since the covariance matrix is diagonal. In the Gaussian case, this is equivalent to independence since we can express the joint pdf of the components of \mathbf{Y} as a product of the marginals.

$$\begin{aligned} f_{\mathbf{Y}}(\mathbf{y}) &= \frac{1}{(2\pi)^{\frac{d}{2}} |\Lambda|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{y}^T \Lambda^{-1} \mathbf{y}\right) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\Lambda|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \sum_{i=1}^d \frac{y_i^2}{\lambda_i}\right) \\ &= \prod_{i=1}^d \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left(-\frac{y_i^2}{2\lambda_i}\right) \end{aligned}$$

Using the fact that diagonal matrix $\Lambda^{\frac{1}{2}}$ is symmetric, the covariance matrix of \mathbf{W} is

$$\begin{aligned} E[\mathbf{W}\mathbf{W}^T] &= E[\Lambda^{-\frac{1}{2}T} \mathbf{Y}\mathbf{Y}^T \Lambda^{-\frac{1}{2}}] \\ &= \Lambda^{-\frac{1}{2}T} E[\mathbf{Y}\mathbf{Y}^T] \Lambda^{-\frac{1}{2}} \\ &= \Lambda^{-\frac{1}{2}T} \Lambda \Lambda^{-\frac{1}{2}} \\ &= \Lambda^{-\frac{1}{2}} \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} \Lambda^{-\frac{1}{2}} \\ &= \mathbf{I} \\ &= \mathbf{I} \end{aligned}$$

This concludes the proof of how the linear transform given by matrix pre-multiplication by $\Lambda^{\frac{1}{2}} \Phi^T$ produces whitened data.

Let us now work through an example. For the purpose of illustration, I will use a bivariate Gaussian vector $X \sim N(\mathbf{0}, \Sigma)$ where

$$\Sigma = \begin{bmatrix} 10 & -6 \\ -6 & 5 \end{bmatrix} \quad (8)$$

Let \mathbf{S} be the collection of data points obtained by taking n samples from \mathbf{X} . The columns of \mathbf{S} are random vectors \mathbf{X}_i , $i = 1, 2, \dots, n$. \mathbf{S} can be generated in MATLAB using the following code.

```
% Generate n samples from colored bivariate Gaussian
n = 200;
CovX = [10 -6; -6 5];
mu = [0 0];
S = mvnrnd(mu, CovX, n);
S = S';
```

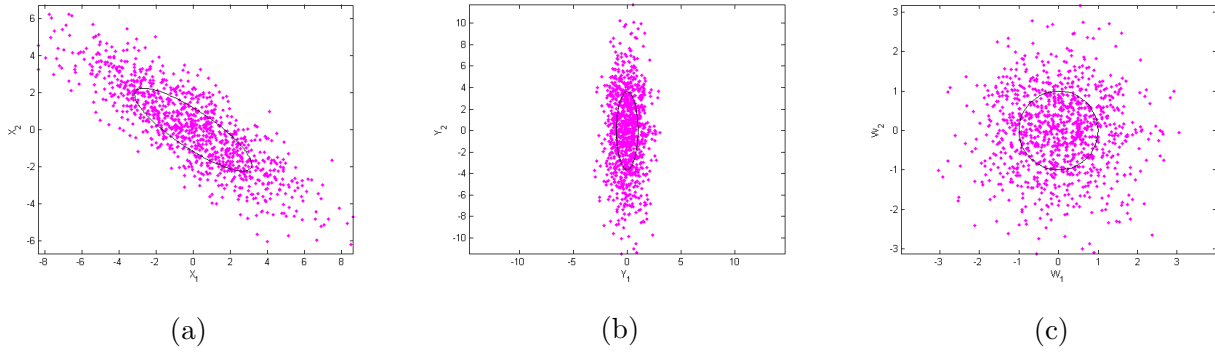


Figure 1: Scatter plots and Contours. All contours are at Mahalanobis distance 1. (a) Original colored density of \mathbf{X} , (b) decorrelated density of \mathbf{Y} , (c) whitenend density of \mathbf{W} .

Figure 1(a) shows a scatter plot of the generated samples along with one contour at unit Mahalanobis distance. The contour is an ellipse that is centered at the origin and rotated through an angle. Say that we do not know the parameters of the distribution from which \mathbf{S} was obtained so we estimate them.

$$\begin{aligned} \hat{\mu} &= \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i = \begin{bmatrix} -0.0753 \\ 0.0568 \end{bmatrix} \\ \hat{\Sigma} &= \frac{1}{n-1} \sum_{i=1}^n (\mathbf{X}_i - \mu)(\mathbf{X}_i - \mu)^T \\ &= \frac{1}{n-1} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T \\ &= \begin{bmatrix} 9.9559 & -5.8718 \\ -5.8718 & 4.8099 \end{bmatrix} \end{aligned}$$

The code for the above estimation is given by

```
% estimate sample mean
mus = mean(S, 2);
% estimate covariance
Covs = bsxfun(@minus, S, mus) * bsxfun(@minus, S, mus)' / (n-1);
```

Next, we find the eigenvectors of the estimated covariance matrix by finding the orthonormal basis for the nullspace of matrix $\hat{\Sigma} - \lambda_i \mathbf{I}$. In MATLAB, we simply use the *eig* command to find both Φ and Λ .

```
[Phi,Lam] = eig(Covs);
```

$$\Lambda = \begin{bmatrix} 0.9881 & 0 \\ 0 & 13.8774 \end{bmatrix} \quad \Phi = \begin{bmatrix} -0.5576 & -0.8301 \\ -0.8301 & 0.5576 \end{bmatrix}$$

Now let

$$\mathbf{Y} = \Phi \mathbf{S} \tag{9}$$

```
Y = Phi.*S;
```

Then, \mathbf{Y} is a set of n data points drawn from an uncorrelated bivariate Gaussian distribution with diagonal covariance matrix Λ . The premultiplication by Φ essentially rotates the ellipse in figure 1(a) so that its principal axes are aligned with the $x_1 - x_2$ axes. The result of this transformation is shown in figure 1(b).

Finally, we scale the samples so the radii of the ellipse are of unit length.

$$\mathbf{W} = \Lambda^{-\frac{1}{2}} \mathbf{Y} \tag{10}$$

In MATLAB,

```
W = sqrt(Lam)\ Y;
```

The whitened data is shown in figure 1(c). Note that the normalization produces a contour that is now a circle. From MATLAB, the estimated covariance of the whitened samples is given by

$$\hat{\Sigma}_W = \begin{bmatrix} 1.0005 & -0.0001 \\ -0.0001 & 1.0000 \end{bmatrix}$$

Note that the whitening transform is not unique. If \mathbf{e}_1 is an eigenvector, then so is $-\mathbf{e}_1$. Also, Λ is not unique either. We could have taken the negative roots of the eigenvalues but since we think of them as standard deviations, it is easier to think of them as positive quantities. In our intermediary step, we rotated the ellipse so that the major axis is aligned vertically in figure 1(c) but we could have done it the other way around and arrived at the same conclusion.

3 Coloring Transform

The coloring transform is the inverse of the whitening transform. Here we see how to transform \mathbf{W} , a multivariate Gaussian variable whose components are i.i.d with unit variance to \mathbf{X} , a Gaussian random vector with some desired covariance matrix.

Say you have \mathbf{S} , a matrix whose n columns are n samples drawn from a whitened Gaussian distribution. So \mathbf{S} is of the form $[\mathbf{W}_1, \dots, \mathbf{W}_n]$ but what you want is n samples from a distribution that has covariance matrix Σ and mean μ . The first step is to perform the eigendecomposition of your desired covariance matrix Σ as in equation 4, which we reproduce here.

$$\Sigma = \Phi \Lambda \Phi^T = \Phi \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} \Phi^T \tag{11}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues, λ_i , of $\mathbf{\Sigma}$ on the diagonal and $\mathbf{\Phi}$ is the matrix of corresponding eigenvectors, \mathbf{e}_i . In addition, the columns of $\mathbf{\Phi}$ are orthonormal so that

$$\mathbf{\Phi}^{-1} = \mathbf{\Phi}^T \quad (12)$$

The first step is to scale the samples so their spread reflects your desired variances along the principal axes. This step changes the spherical contours to elliptical ones.

$$\mathbf{Y} = \mathbf{\Lambda}^{\frac{1}{2}} \mathbf{S} \quad (13)$$

The next step is to rotate the data points so that they are now correlated.

$$\mathbf{X} = \mathbf{\Phi} \mathbf{Y} = \mathbf{\Phi} \mathbf{\Lambda}^{\frac{1}{2}} \mathbf{S} \quad (14)$$

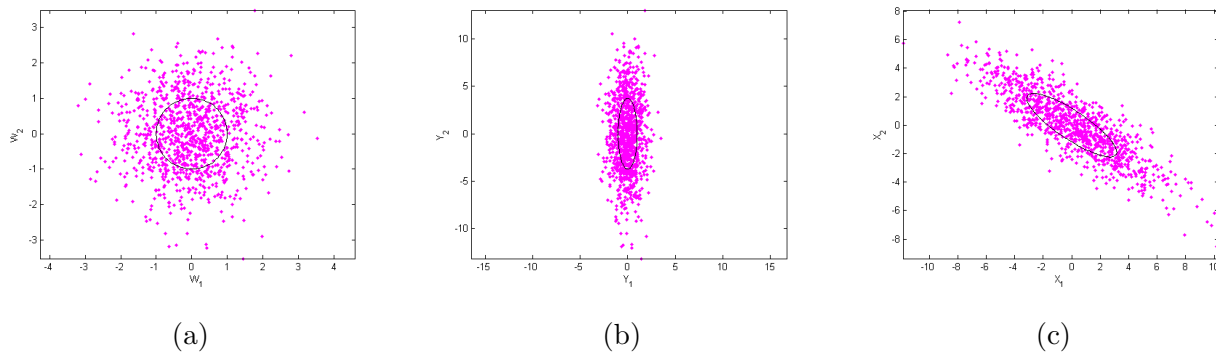


Figure 2: Scatter plots and Contours. All contours are at Mahalanobis distance 1. (a) Original white density of \mathbf{W} , (b) scaled density of \mathbf{Y} , (c) colored density of \mathbf{X} .

Figure 2 shows the mapping from \mathbf{W} to \mathbf{Y} to \mathbf{X} for a bivariate distribution with

$$\mathbf{\Sigma} = \begin{bmatrix} 10 & -6 \\ -6 & 5 \end{bmatrix}$$

It is just like figure 1 from right to left. Finally, add μ to your samples so that the mean is translated from the origin to your desired location. The transform can be implemented in MATLAB with the following code.

```
function[X] = colortran(mu, CovX)

% Find eigenvectors and eigenvalues
[Phi,Lam] = eig(CovX);
% Generate 1000 samples of white data
S = mvnrnd([0 0],eye(2),1000)';
% Scale to desired variances
Y = sqrt(Lam)*S;
% Rotate or correlate samples
X = Phi*Y;
% Add the mean vector to every column of X
X = bsxfun(@plus,X,mu);

end
```

4 Summary

So we see that data whitening can be broken down into two steps:

1. Decorrelation. In other words, a rotation that maps the data so that the axes along which the original data has the largest variance are aligned with your co-ordinate axes. The resulting covariance matrix is diagonal. This is really just a change of basis.
2. Scaling so that the principal axes are each of unit length. The data is squeezed or stretched along individual axes depending on whether the variance along said axis is greater or less than one. The resulting covariance matrix is identity.

We can summarize coloring as the inverse of whitening. I also find it helpful to compare it with the simplest case, when we are working with scalars. Recall that multiplying a random variable with a constant produces another whose standard deviation is scaled by a factor equal to said constant. In other words If W is a random variable of unit variance and $X = c.W$, then W has standard deviation $c.1$ and variance c^2 . This multiplication is a linear transform. In the multivariate case, we are expressing our desired covariance matrix as a product

$$\Sigma = \mathbf{E}\mathbf{E}^T$$

where $\mathbf{E} = \Phi\Lambda^{\frac{1}{2}}$. This \mathbf{E} is our linear transform and is analogous to c in the scalar case.

I am aware that MATLAB comes with the *mvnrnd* command that can generate data points from any multivariate Gaussian distribution but hopefully, this material provides sufficient background to allow the reader to not only whiten data but also to transform data from one distribution to another in other programs that are not equipped with the same out of the box capabilities as MATLAB. A summary of this process is presented in figure 3.

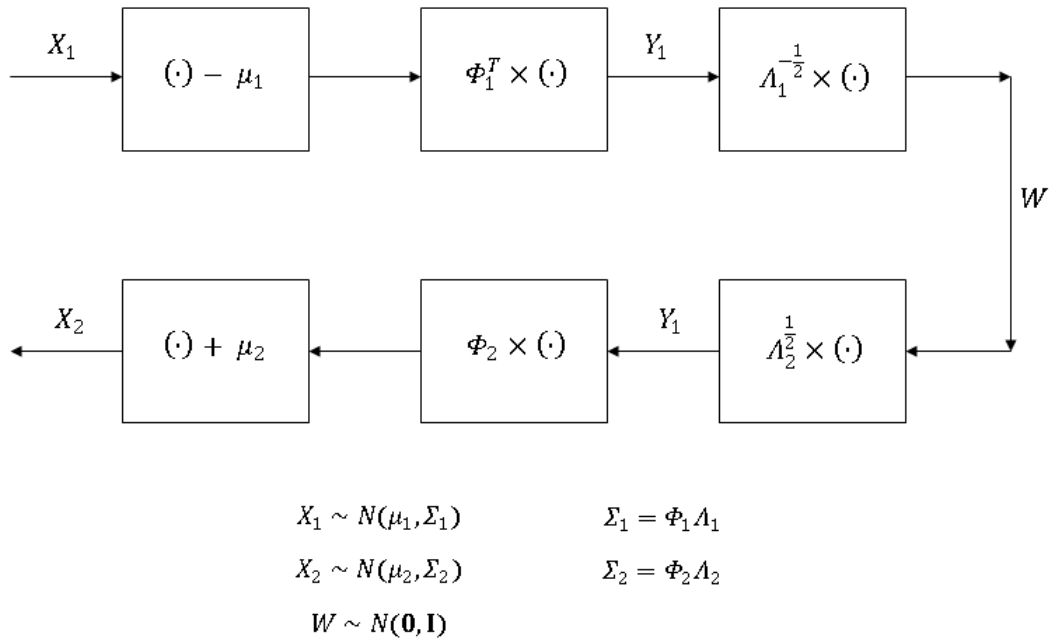


Figure 3: Summary diagram of whitening and coloring process

5 References

References

- [1] Charles A. Bouman, *Digital Image Processing Laboratory: Eigen-decomposition of Images*. <<https://engineering.purdue.edu/~bouman/grad-labs/Eigen-Image-Analysis/pdf/lab.pdf>>. February 22, 2013
- [2] Mirreille Boutin, "ECE662: Statistical Pattern Recognition and Decision Making Processes," Class Lecture. Purdue University, Spring 2014.
- [3] Gilbert Strang, *Linear Algebra and its Applications*. Thomson Brooks/Cole. 4th Edition, 2006