ECE 417, Lecture 7
PCA

Mark Hasegawa-Johnson

9/19/2017

Content

- Mahalanobis Distance review
- PCA = Eigenvectors of the covariance matrix
- Positive semi-definite matrices; pseudo-inverse
- PCA = Left singular vectors of the data matrix

Mahalanobis Distance Review

Mahalanobis form of the multivariate Gaussian, dependent dimensions

If the dimensions are dependent, and jointly Gaussian, then we can still write the multivariate Gaussian as

$$f_{\vec{X}}(\vec{x}) = \mathcal{N}(\vec{x}; \vec{\mu}, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})}$$

We call this the Mahalanobis form because the exponent is the squared Mahalanobis distance (with weight matrix Σ) between $\vec{x}$ and $\vec{\mu}$:

$$d_\Sigma^2(\vec{x}, \vec{\mu}) = (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})$$

Example

Suppose that x_1 and x_2 are linearly correlated Gaussians with means 1 and -1, respectively, and with variances 1 and 4, and covariance 1.

$$\vec{\mu} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Remember the definitions of variance and covariance:

$$\sigma_1{}^2 = E[(x_1 - \mu_1)^2] = 1$$
$$\sigma_2{}^2 = E[(x_2 - \mu_2)^2] = 4$$
$$\sigma_{12} = \sigma_{21} = E[(x_1 - \mu_1)(x_2 - \mu_2)] = 1$$

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$$

Example

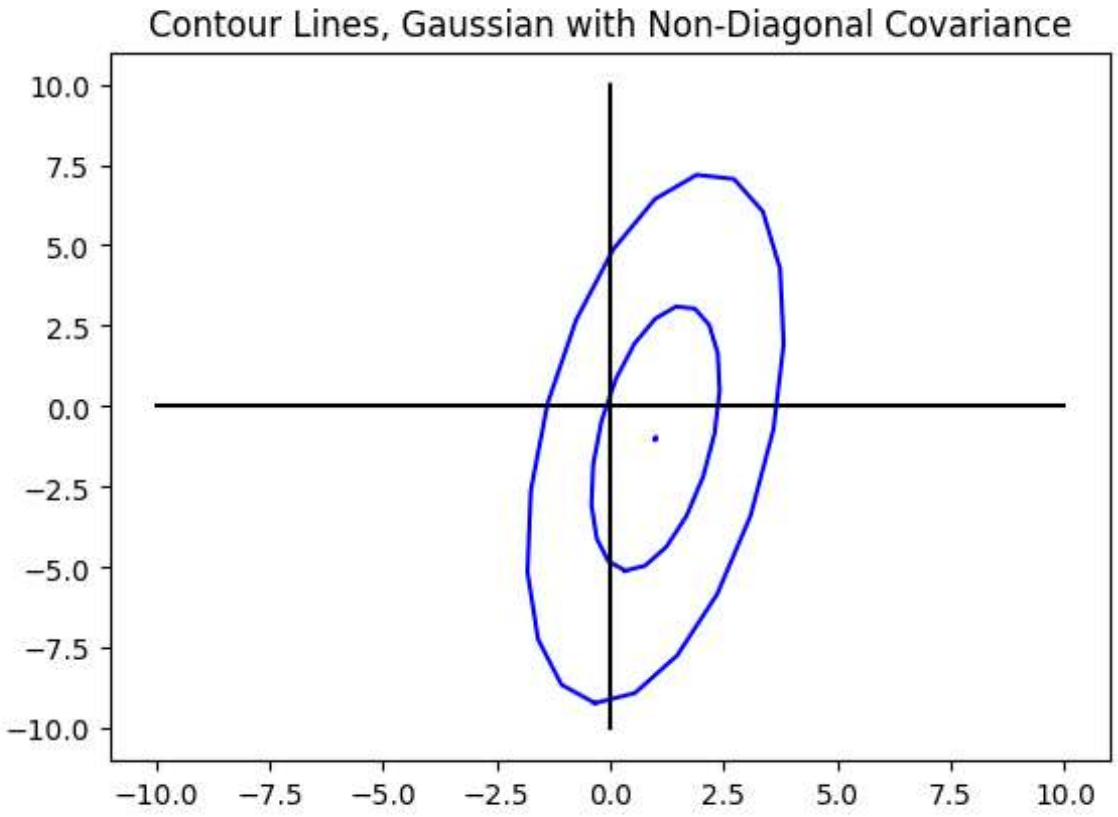
The contour lines of this Gaussian are the lines of constant Mahalanobis distance between $\vec{x}$ and $\vec{\mu}$. For example, to plot the $d_\Sigma(\vec{x},\vec{\mu}) = 1$ and $d_\Sigma(\vec{x},\vec{\mu}) = 2$ ellipses, we find the solutions of

$$1 = d_\Sigma^2(\vec{x},\vec{\mu}) = (\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})$$

and

$$4 = d_\Sigma^2(\vec{x},\vec{\mu}) = (\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})$$

Example


Contour Lines, Gaussian with Non-Diagonal Covariance
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

PCA = Eigenvectors of the Covariance Matrix

Symmetric positive definite matrices

If Σ is symmetric and positive semi-definite we can write

$$\Sigma = U\Lambda U^T$$

and

$$U^T\Sigma U = \Lambda$$

Where Λ is a diagonal matrix of the eigenvalues, and U is an orthonormal matrix of the eigenvectors.

Inverse of a positive definite matrix

The inverse of a positive definite matrix is:

$$\Sigma^{-1} = U\Lambda^{-1}U^T$$

Proof:

$$\Sigma\,\Sigma^{-1} = U\Lambda U^T U\Lambda^{-1}U^T = U\Lambda\Lambda^{-1}U^T = UU^T = I$$

where

$$\Lambda^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & 0 \\ 0 & \frac{1}{\lambda_2} & \dots \\ 0 & \dots & \frac{1}{\lambda_D} \end{bmatrix}$$

Mahalanobis distance again

Remember that

$$d_{\Sigma}^{2}(\vec{x},\vec{\mu}) = (\vec{x}-\vec{\mu})^{T}\Sigma^{-1}(\vec{x}-\vec{\mu})$$

But we can write this as

$$d_{\Sigma}^{2}(\vec{x},\vec{\mu}) = (\vec{x}-\vec{\mu})^{T}U\Lambda^{-1}U^{T}(\vec{x}-\vec{\mu})$$
$$= \vec{y}^{T}\,\Lambda^{-1}\vec{y}$$

Where the vector $\vec{y}$ is defined to be the principal components of $\vec{x}$:

$$\vec{y} = U^{T}(\vec{x}-\vec{\mu}) = \begin{bmatrix} \vec{u}_1{}^{T}(\vec{x}-\vec{\mu}) \\ \dots \\ \vec{u}_D{}^{T}(\vec{x}-\vec{\mu}) \end{bmatrix}$$

Facts about ellipses

The formula

$$1 = \vec{y}^T \Lambda^{-1} \vec{y}$$

… or equivalently

$$1 = \frac{{y_1}^2}{\lambda_1} + \cdots + \frac{{y_D}^2}{\lambda_D}$$

… is the formula for an ellipsoid. If $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_D$ then the biggest main axis of the ellipse is the direction in which $y_1 \neq 0$ and all of the other principal components are $y_j = 0$. This happens when $(\vec{x} - \vec{\mu}) \propto \vec{u}_1$, because in that case:

$$\vec{u}_1{}^T (\vec{x} - \vec{\mu}) \neq 0$$
$$\vec{u}_j{}^T (\vec{x} - \vec{\mu}) = 0, \qquad j \neq 1$$

Example

Suppose that

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$$

We get the eigenvalues from the determinant equation: $|\Sigma - \lambda I| = (1-\lambda)(4-\lambda) - 1 = \lambda^2 - 5\lambda + 3$ which equals zero for $\lambda = \frac{5 \pm \sqrt{13}}{2}$.

We get the eigenvectors by solving $\lambda\vec{u} = \Sigma\vec{u}$, which gives

$$\overrightarrow{u_1} \propto \begin{bmatrix} 1 \\ \frac{3+\sqrt{13}}{2} \end{bmatrix}, \qquad \overrightarrow{u_2} \propto \begin{bmatrix} 1 \\ \frac{3-\sqrt{13}}{2} \end{bmatrix}$$

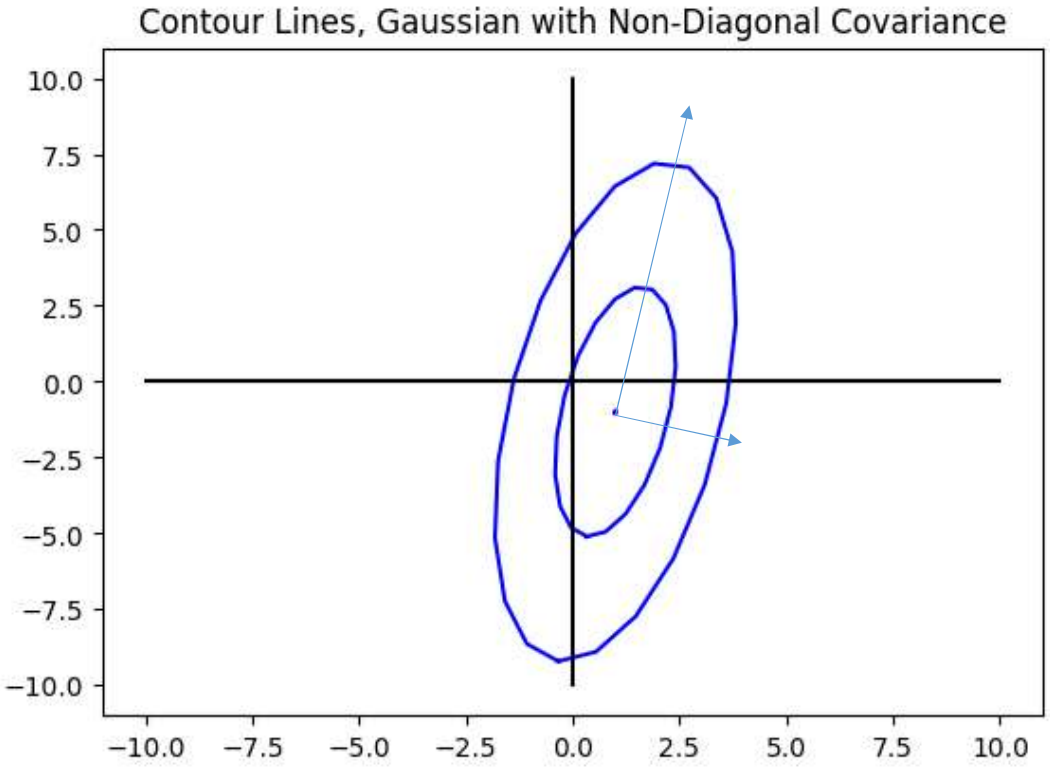
Where the constant of proportionality is whatever's necessary to make vectors unit-length; we don't really care what it is.

Example

So the principal axes of the ellipse are in the directions

$$\overrightarrow{u_1} \propto \begin{bmatrix} 1 \\ \dfrac{3+\sqrt{13}}{2} \end{bmatrix},$$

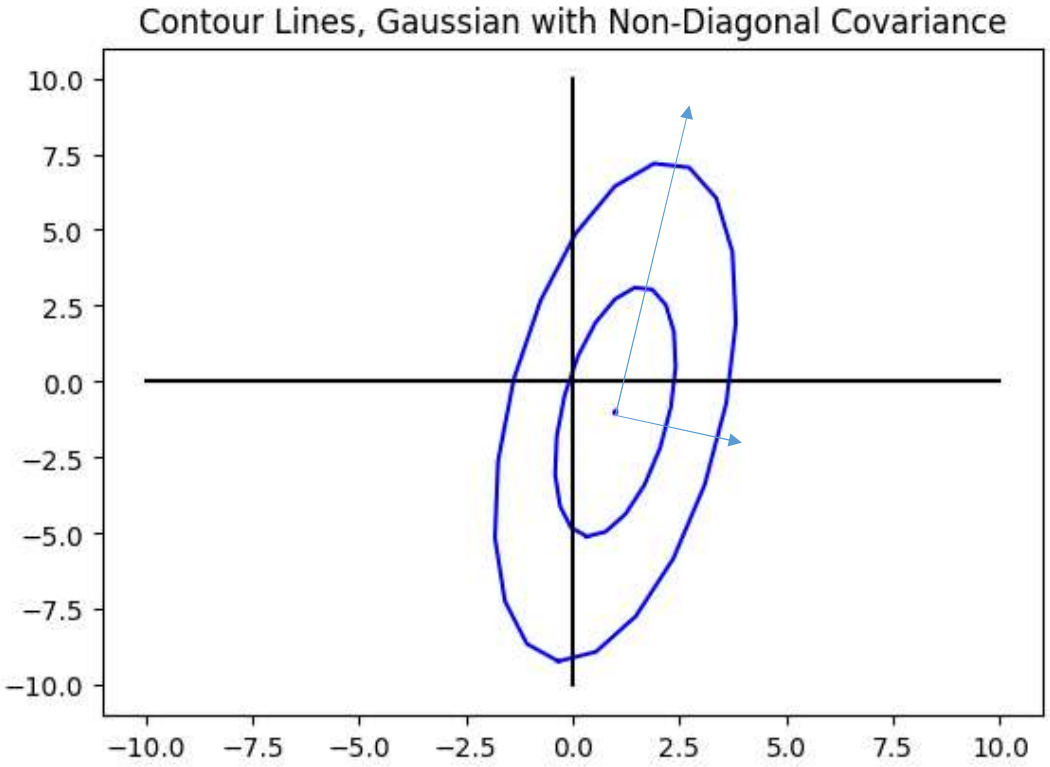
$$\overrightarrow{u_2} \propto \begin{bmatrix} 1 \\ \dfrac{3-\sqrt{13}}{2} \end{bmatrix}$$



Example

In fact, another way to write this ellipse is

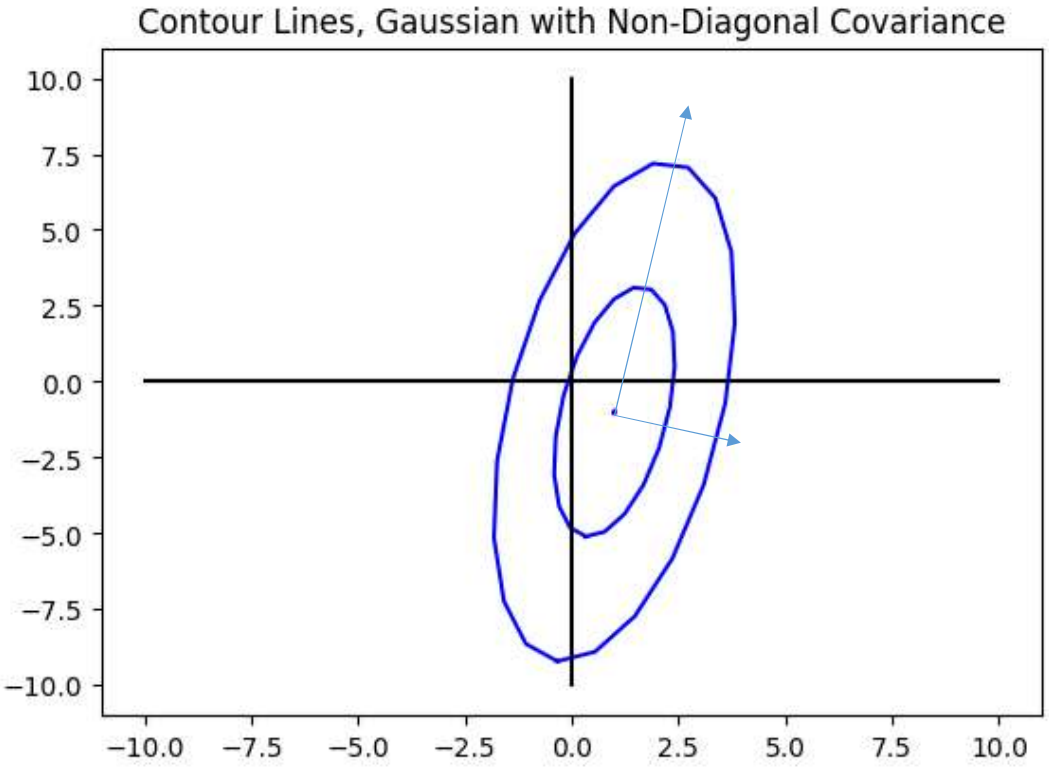
$$1 = \frac{\left(\vec{u}_1{}^T(\vec{x}-\vec{\mu})\right)^2}{\lambda_1} + \frac{\left(\vec{u}_2{}^T(\vec{x}-\vec{\mu})\right)^2}{\lambda_2}$$



Example

In fact, it's useful to talk about Σ in this way:

- The first principal component, y_1, is the part of $(\vec{x} - \vec{\mu})$ that's in the $\vec{u}_1$ direction. It has a variance of λ_1.
- The second principal component, y_2, is the part of $(\vec{x} - \vec{\mu})$ that's in the $\vec{u}_2$ direction. It has a variance of λ_2.
- The principal components are uncorrelated with each other.
- If $\vec{x}$ is Gaussian, then y_1 and y_2 are independent Gaussian random variables.



Positive semi-definite matrices

Symmetric positive semi-definite matrices

Positive semi-definite ($\Sigma \succcurlyeq 0$) means that for any vector $\vec{x}$, $\vec{x}^T \Sigma \vec{x} \geq 0$. This is equivalent to saying that all of the eigenvalues of Σ are non-negative ($\lambda_i \geq 0$).

- This kind of thing often happens if $D > N$ (the vector dimension is larger than the number of training tokens, as in MP2.
- Now it will turn out that some of the eigenvalues are zero. In fact, only M of the eigenvalues will be nonzero, for some number $M \leq \min(N, D)$.
- The number of zero eigenvalues depends on how many different values of $\vec{x}$ cause $\vec{x}^T \Sigma \vec{x} = 0$. Suppose that there is a $(D - M)$-dimensional subspace of vectors $\vec{x}$ such that any $\vec{x}$ from that subspace causes $\vec{x}^T \Sigma \vec{x} = 0$. Then there are $(D - M)$ zero-valued eigenvalues.
- If we've sorted the eigenvalues so that $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_D$, then $\lambda_i = 0$ for $i > M$, and

$$\Sigma = \sum_{i=1}^{D} \lambda_i \vec{u}_i \vec{u}_i^T = \sum_{i=1}^{M} \lambda_i \vec{u}_i \vec{u}_i^T$$

Symmetric positive semi-definite matrices

It's useful now to define the eigenvalue matrix to be only $M \times M$, and to define the eigenvector matrix to be $D \times M$, where $D > M$. That way we can keep the idea that Λ is all zeros off the diagonal, and all positive elements on the main diagonal:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & \dots \\ 0 & \dots & \lambda_M \end{bmatrix}, \qquad U = [\vec{u}_1, \dots, \vec{u}_M]$$

With these definitions, we can still write

$$\Sigma = U\Lambda U^T \text{ (a } D \times D \text{ matrix)}$$

$$U^T \Sigma U = \Lambda \text{ (an } M \times M \text{ matrix)}$$

$$U^T U = I_{M \times M}$$

… but $UU^T \neq I_{D \times D}$.

PCA = eigenvectors corresponding to nonzero eigenvalues

When Σ is positive semi-definite, it's most useful to define PCA for the nonzero eigenvalues (you can define PCA for the other eigenvectors, but it's not really useful). Thus $\vec{y}$ is an M-dimensional vector:

$$\vec{y} = U^T(\vec{x} - \vec{\mu}) = \begin{bmatrix} \vec{u}_1{}^T(\vec{x} - \vec{\mu}) \\ \dots \\ \vec{u}_M{}^T(\vec{x} - \vec{\mu}) \end{bmatrix}$$

Pseudo-inverse of a positive semi-definite matrix

For a positive semi-definite matrix, it's useful to define a "pseudo-inverse" :

$$\Sigma^{\dagger} = U\Lambda^{-1}U^{T}$$

The dagger † is a special character that means "pseudo-inverse."

- It's not a true inverse ($\Sigma\Sigma^{\dagger} \neq I$)…
- But it has some other properties that make it behave almost like an inverse. For example,
  - $\Sigma^{\dagger}\Sigma\Sigma^{\dagger} = \Sigma^{\dagger}$
  - $\Sigma\Sigma^{\dagger}\Sigma = \Sigma$.

Mahalanobis distance uses pseudo-inverse, since the true inverse doesn't exist

…in particular, the only sensible definition of Mahalanobis distance, in this case, is

$$d_{\Sigma}^{2}(\vec{x},\vec{\mu}) = (\vec{x}-\vec{\mu})^{T}\Sigma^{\dagger}(\vec{x}-\vec{\mu})$$

$$= \vec{y}^{T}\,\Lambda^{-1}\vec{y}$$

$$= \frac{{y_1}^2}{\lambda_1} + \cdots + \frac{{y_M}^2}{\lambda_M}$$

Notice what this means. It means that any component of $(\vec{x}-\vec{\mu})$ in a direction outside of the M-dimensional space $U = [\vec{u}_1, \ldots, \vec{u}_M]$ is just completely ignored. $(\vec{x}-\vec{\mu})$ is first projected into that subspace as $\vec{y} = U^{T}(\vec{x}-\vec{\mu})$, then $d_{\Sigma}^{2}(\vec{x},\vec{\mu})$ just calculates distance in the subspace.

PCA = left singular vectors of the data matrix

Normalized Data Matrix: Outer product = sample covariance matrix

Define the "normalized data matrix" to be

$$\tilde{X} = \frac{1}{\sqrt{N-1}}[\vec{x}_1 - \vec{\mu}, \vec{x}_2 - \vec{\mu}, \dots, \vec{x}_N - \vec{\mu}]$$

That way we get the unbiased sample covariance matrix as

$$\Sigma = \frac{1}{N-1}\sum_{n=1}^{N}(\vec{x}_n - \vec{\mu})(\vec{x}_n - \vec{\mu})^T = \tilde{X}\tilde{X}^T$$

Σ is a $D \times D$ matrix. Its $(c,d)^{th}$ element is the sample covariance of x_c and x_d

$$\sigma_{cd} = \frac{1}{N-1}\sum_{n=1}^{N}(x_{cn} - \mu_c)(x_{dn} - \mu_d) \approx E[(x_c - \mu_c)(x_d - \mu_d)]$$

Inner Product = Gram Matrix

Instead of the outer product $\tilde{X}\tilde{X}^T$, suppose we compute the inner product $\tilde{X}^T\tilde{X}$. That's called the "gram matrix:"

$$\Gamma = \tilde{X}^T\tilde{X}$$

Γ is an $N \times N$ matrix. Its $(m,n)^{th}$ element is the dot product of $(\vec{x}_m - \vec{\mu})$ and $(\vec{x}_n - \vec{\mu})$:

$$\gamma_{mn} = \frac{1}{N-1}(\vec{x}_m - \vec{\mu})^T(\vec{x}_n - \vec{\mu})$$

Eigenvalues of the Gram Matrix and the Sample Covariance

Both the gram matrix and the sample covariance are symmetric positive semi-definite matrices, so we can write

$$\Sigma = U\Lambda U^T, \Gamma = V\mathrm{K}V^T$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & \dots \\ 0 & \dots & \lambda_M \end{bmatrix}, \mathrm{K} = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & \dots \\ 0 & \dots & \kappa_M \end{bmatrix}$$

$$U = [\vec{u}_1, \dots, \vec{u}_M] \text{ is } D \times M$$

$$V = [\vec{v}_1, \dots, \vec{v}_M] \text{ is } N \times M$$

$$U^T U = V^T V = I_{M \times M}$$

But $UU^T \neq I_{D \times D}$ and $VV^T \neq I_{N \times N}$

Square Root Matrix

Define the square root matrix of Λ to be some matrix $\Lambda^{1/2}$ such that

$$\Lambda^{1/2}\Lambda^{1/2} = \Lambda$$

In fact, for a diagonal matrix like Λ, the square root matrix is easy to find. It's just

$$\Lambda^{1/2} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \sqrt{\lambda_2} & \dots \\ 0 & \dots & \sqrt{\lambda_M} \end{bmatrix}$$

Inserting an Identity Matrix: Covariance

Remember that $V^T V = I$. Therefore we can write

$$\begin{aligned}
\tilde{X}\tilde{X}^T &= \Sigma \\
&= U\Lambda U^T \\
&= U\Lambda^{1/2}\Lambda^{1/2}U^T \\
&= U\Lambda^{1/2} I \Lambda^{1/2}U^T \\
&= U\Lambda^{1/2}V^T V\Lambda^{1/2}U^T \\
&= (U\Lambda^{\frac{1}{2}}V^T)(U\Lambda^{\frac{1}{2}}V^T)^T
\end{aligned}$$

Inserting an Identity Matrix: Gram Matrix

Remember that $U^T U = I$. Therefore we can write

$$\begin{aligned}
\tilde{X}^T \tilde{X} &= \Gamma \\
&= V \mathrm{K} V^T \\
&= V \mathrm{K}^{1/2} \mathrm{K}^{1/2} V^T \\
&= V \mathrm{K}^{1/2} I \mathrm{K}^{1/2} V^T \\
&= V \mathrm{K}^{1/2} U^T U \mathrm{K}^{1/2} V^T \\
&= (U \mathrm{K}^{\frac{1}{2}} V^T)^T (U \mathrm{K}^{\frac{1}{2}} V^T)
\end{aligned}$$

Singular Value Decomposition (SVD)

Thus we have

$$\tilde{X}^T\tilde{X} = (U\mathrm{K}^{\frac{1}{2}}V^T)^T(U\mathrm{K}^{\frac{1}{2}}V^T)$$

And

$$\tilde{X}\tilde{X}^T = (U\Lambda^{\frac{1}{2}}V^T)(U\Lambda^{\frac{1}{2}}V^T)^T$$

The only way these things can both be true is if

$$\tilde{X} = USV^T$$

Where U $(D \times M)$ and V $(N \times M)$, are both inner-orthonormal, and

$$S = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & \dots \\ 0 & \dots & s_M \end{bmatrix}, s_j = \sqrt{\lambda_j} = \sqrt{\kappa_j}$$

Singular Value Decomposition (SVD)

$\tilde{X} = USV^T$, where U $(D \times M)$ and V $(N \times M)$, are both inner-orthonormal, and

$$S = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & \dots \\ 0 & \dots & s_M \end{bmatrix}, s_j = \sqrt{\lambda_j} = \sqrt{\kappa_j}$$

- U and V are called the left and right singular vectors of $\tilde{X}$
- s_j are called the singular values
- There's nothing special about $\tilde{X}$. EVERY matrix has singular values and singular vectors, but…
- The only way to find the singular values is by finding the eigenvalues of $\tilde{X}^T\tilde{X}$ or $\tilde{X}\tilde{X}^T$, whichever is smaller.

Why is this useful for MP2?

- The only way to find the singular values is by finding the eigenvalues of $\tilde{X}^T\tilde{X}$ or $\tilde{X}\tilde{X}^T$, whichever is smaller.
- In MP2, the sample covariance is $D \times D$, which is huge. The gram matrix is $N \times N$, which is actually a lot smaller. So you want to start out by computing $\Gamma = V\Lambda V^T$, not $\Sigma = U\Lambda U^T$
- To find the principal components, though, you need U. How do you find it, if you already know V and Λ? Answer: use the data matrix:

$$\tilde{X} = USV^T$$
$$\tilde{X}V = USV^TV = US$$
$$\tilde{X}VS^{-1} = U$$

Why is this useful for MP2?

- But actually, in past semesters we've discovered that, instead of using the orthonormal PCA, you actually get better results if you use the S-weighted PCA:

$$US = [s_1\vec{u}_1, \ldots, s_M\vec{u}_M]$$

Summary

- Principal components are what you get by projecting the data onto the principal component directions
- Principal component directions are eigenvectors of the covariance
- Variance of a PC is the eigenvalue of the covariance, which is also the eigenvalue of the gram matrix. Standard deviation of a PC is the singular value of the data matrix (square root of the variance).
- Once you have eigenvalues and eigenvectors of the gram matrix, you can find the eigenvectors of the covariance by multiplying through the data matrix.