Scribe Notes: Multivariate Gaussians and PCA

Brenton Partridge

December 6, 2010

Contents

1 Overview

The idea of dimension reduction is to find a hyperplane in high-dimensional space such that an objective function [of the data with respect to its projection on the hyperplane] is optimized. In certain cases, we can interpret the dimensions that are obtained; a case study where this has worked is the ideal point model in voting. A commonly used method for dimension reduction is principal component analysis, or its more general form, factor analysis. To develop this method, we will begin by specifying a distribution over p-vectors, then describe variables in a graphical model in terms of this distribution.

2 Multivariate Gaussian

A multivariate Gaussian distribution (MVG) is a distribution over p-vectors, which are vectors of p real components ($\vec{x} \in \mathbb{R}^p$). The distribution takes two parameters:

1. $\vec{\mu} \in \mathbb{R}^p$ is the $p \times 1$ mean vector such that $\mu_i = E[X_i]$.
2. $\Sigma \in \mathbb{R}^{p \times p}$ is the $p \times p$ covariance matrix such that its elements are the covariances $\sigma_{ij} \triangleq \text{Cov}(X_i, X_j)$. It has a few notable properties:
   (a) $\sigma_{ij} = E[X_i X_j] - E[X_i]E[X_j]$
   (b) $\sigma_{ii} = E[X_i^2] - E[X_i]^2 \triangleq \text{Var}(X_i)$
   (c) Σ is positive definite and symmetric. Note[1] that some of its elements can be negative (implying negative correlation between component pairs), but the overall matrix must be positive definite, meaning $\vec{z}^T \Sigma \vec{z} > 0 \, \forall \vec{z} \neq \vec{0} \in \mathbb{R}^p$.

The pdf is:

$$p(\vec{x}|\vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu}) \right\} \tag{2.1}$$

where the fractional term in the beginning is a constant so the pdf integrates to 1 (note that $|\Sigma|$ is the determinant of Σ), and the exponent takes a quadratic form.

> *N.B. from recitation: The inverse covariance matrix Σ^{-1} (as in $(x - \mu)^T \Sigma^{-1} (x - \mu)$) has the fascinating property that if $(\Sigma^{-1})_{ij} = 0$ then $X_i \perp\!\!\!\perp X_j | \{X_k \, \forall \, k \neq i, j\}$; they are conditionally independent given all other components. We can interpret this as an undirected graphical model, where each component in the MVG is a node, and Σ^{-1} is a weighted adjacency matrix such that an edge only exists if $(\Sigma^{-1})_{ij} \neq 0$.*

Now consider the function in the exponent, $f(\vec{x}) = -\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})$. This defines contours of constant probability, such that if $f(\vec{x})$ is fixed, the values of $\vec{x}$ form an ellipse. Cases for these ellipses are shown in Figure 1.

If all pairs of components in a multivariate Gaussian distribution are uncorrelated (i.e. Σ is diagonal), then the components are independent.[2]

2.1 Maximum Likelihood Estimate of MVG

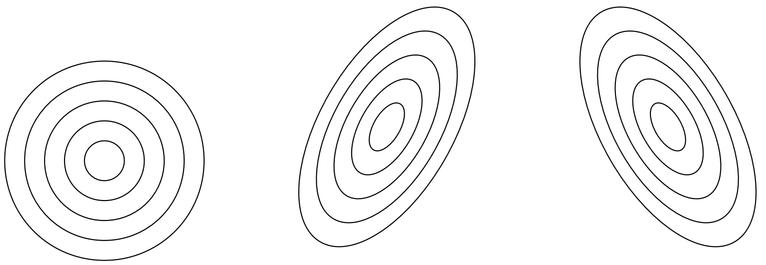
A new subscript notation is introduced, where $n = 1..N$ represent replication of data, and $\vec{x}_n \in \mathbb{R}^p \, \forall n$. Thus, $\mathscr{D} = \{\vec{x}_n\}_{n=1}^N$, and the M.L.E. $\arg\max_{\vec{\mu}, \Sigma} \log \prod_{n=1}^N p(\vec{x}_n|\vec{\mu}, \Sigma)$ yields the

[1] This point was clarified midway through the lecture.

[2] Proof of this relationship was not provided in lecture, although it was discussed at length. $E[X_i X_j] = E[X_i]E[X_j]$ does not suffice because all other moments would need to agree. A proof may arise from the sufficient statistics (meaning the proof would apply to the general exponential family), but this was not confirmed.

Note that in general, although independence implies zero correlation, the reverse is not true. Consider the case where $X \sim \mathcal{N}(0, 1)$ and $Y = XZ$ where Z is 1 with probability $\frac{1}{2}$ and -1 with probability $\frac{1}{2}$. Then $Y \sim \mathcal{N}(0, 1)$ marginally, and $\text{Cov}(X, Y) = 0$, but X and Y are clearly dependent. Such a case, though, would be impossible in a multivariate (joint) Gaussian distribution, were $[X, Y] \sim \mathcal{N}_2(\vec{0}, I)$. While not a formal proof, this may shed some light on the subject.

Figure 1: Contours of constant probability in the exponent of a bivariate Gaussian distribution.



(a) circles → Σ diagonal, components $X \perp\!\!\!\perp Y$ are uncorrelated

(b) X, Y are positively correlated, $\sigma_{xy} > 0$

(c) X, Y are negatively correlated, $\sigma_{xy} < 0$

following:

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} \vec{x}_n \tag{2.2}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (\vec{x}_n - \hat{\mu})(\vec{x}_n - \hat{\mu})^T \tag{2.3}$$

where $\hat{\Sigma}$ can be interpreted as the average outer product of differences from the mean.

2.2 Decomposing a MVG distribution into multiple MVGs

We can divide $\vec{x}$ into $\langle \vec{x}_1, \vec{x}_2 \rangle$ elementwise (i.e. $\vec{x}_1 = \vec{x}_{[1..q]}$ and $\vec{x}_2 = \vec{x}_{[q+1..p]}$). The parameters can be decomposed accordingly, such that $\vec{\mu} = \langle \vec{\mu}_1, \vec{\mu}_2 \rangle$[3] and Σ is the block matrix,

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

Consider the joint distribution using the chain rule,

$$p(\vec{x}_1, \vec{x}_2) = p(\vec{x}_2) p(\vec{x}_1 | \vec{x}_2).$$

The marginal $p(\vec{x}_2)$ is a MVG with:

$$\vec{\mu} = \vec{\mu}_2 \tag{2.4}$$

$$\Sigma = \Sigma_{22} \tag{2.5}$$

[3] For more about the significance of this decomposition, refer to the film http://www.imdb.com/media/rm372218880/tt0190641.

The conditional $p(\vec{x}_1|\vec{x}_2)$ is a MVG with:

$$\vec{\mu} = \vec{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}\left(\vec{x}_2 - \vec{\mu}_2\right) \tag{2.6}$$

$$\Sigma = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{2.7}$$

These results will be proved in the reading which will be sent out; note for now that the dimensions agree. To generalize these results to exponential families, the results should be available in Brown, L., *Exponential Families.*

3 Factor Analysis & PCA

To define a generative model for a single vector in p-space, we first choose q independent components to form a vector in q-space:

$$\vec{Z} \sim \mathcal{N}_q\left(\vec{0}, I\right)$$

Then, we generate a vector $\vec{x} \in \mathbb{R}^p$ from these components:

$$\vec{X}|\vec{Z} \sim \mathcal{N}_p\left(\vec{\mu} + \Lambda\vec{Z}, \Psi\right)$$

Here, $\vec{\mu} \in \mathbb{R}^p, \Lambda \in \mathbb{R}^{p\times q}, \vec{Z} \in \mathbb{R}^q, \Psi \in \mathbb{R}^{p\times p}$. Ψ, the covariance matrix, is a diagonal matrix of noise/error amounts; since it is diagonal, the components have zero covariance. In PCA, these Ψ_i along the diagonal must be the same for all components, such that $\Psi = I\sigma^2$ where σ^2 is a scalar variance value. However, in factor analysis, the Ψ_i can be different.

We now wish to generate N such vectors. For simplicity, we can assume (throughout the rest of these notes) that the data is centered, that is, $\vec{\mu} = 0$ across the data points. This can be ensured during pre-processing by finding the component-wise mean $\vec{\mu}$ and subtracting it from each datum, such that $\mathscr{D}_{\text{centered}} = \{\vec{x}_n - \vec{\mu}\}_{n=1}^N$. Then, we have:

$$\vec{Z}_n \sim \mathcal{N}_q\left(\vec{0}, I\right) \tag{3.1}$$

$$\vec{X}_n|\vec{Z}_n \sim \mathcal{N}_p\left(\Lambda\vec{Z}_n, \Psi\right) \tag{3.2}$$

This leads to the graphical model in Figure 2.

Note that the vector $\Lambda\vec{Z}$ can be considered as a linear combination of the q columns $\vec{\lambda}_i$ of Λ, where the coefficients are the scalar elements of $\vec{Z}$:

$$E\left[\vec{X}|\vec{Z}\right] = \Lambda\vec{Z} = Z_1\vec{\lambda}_1 + Z_2\vec{\lambda}_2 + \ldots + Z_q\vec{\lambda}_q \tag{3.3}$$

Thus, some data points can use some factors more than others. The idea is that the $\vec{\lambda}_i$'s represent common sources of variation throughout the data.

Figure 2: Graphical model for PCA and FA

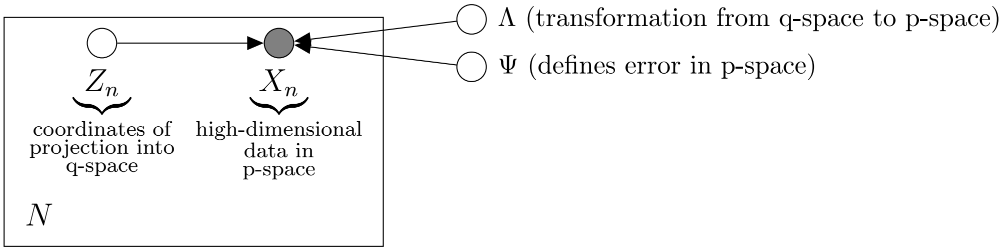
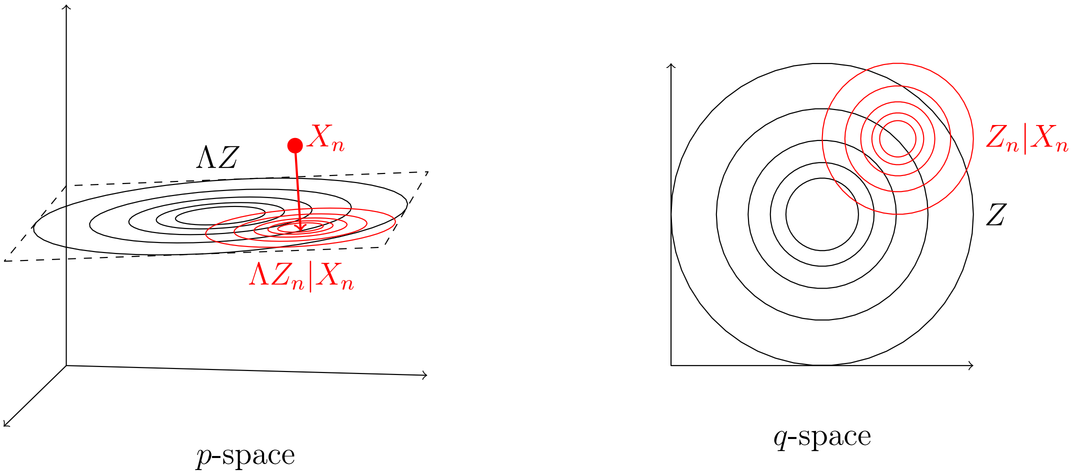

Z_n
coordinates of projection into q-space
X_n
high-dimensional data in p-space
N
Λ (transformation from q-space to p-space)
Ψ (defines error in p-space)

Figure 3: Projection from three dimensions onto principal components



3.1 Geometric Interpretation

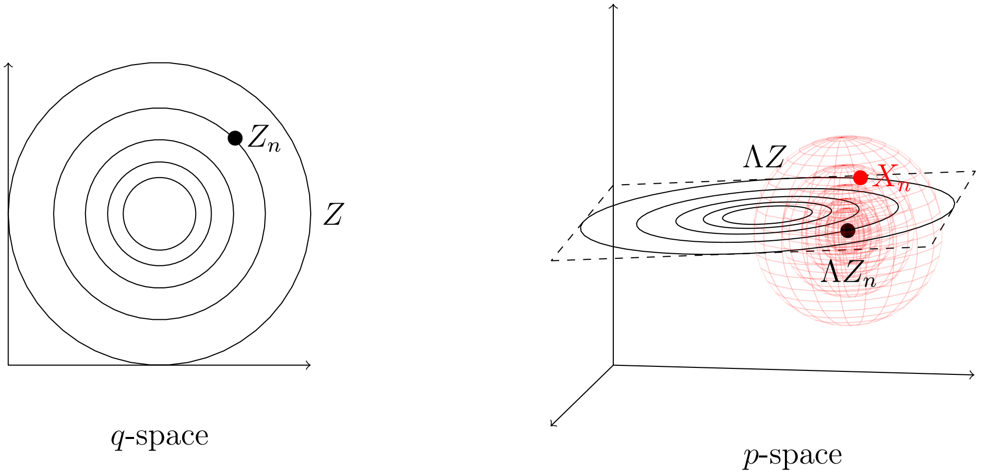
In the graphical model in Figure 2, we consider $\{X_n\}$ as data in a high-dimensional p-space, and Λ defining a hyperplane (or subspace) of that p-space; the columns of Λ, $\left\{\vec{\lambda}_i \in \mathbb{R}^p : i = 1..q\right\}$, span the subspace. Thus, given a specific Λ, the projection of any point X_n onto the subspace can be specified by a set of coordinates Z_n, as per the linear combination shown in Equation 3.3. Figure 3 shows the process by which these coordinates would be determined, where the circles represent a Bayesian view of the values. In this case, the distribution $Z_n|X_n$ is centered on the closest point in the subspace to X_n.[4]

We can now geometrically interpret the generative model that would create the observed variables $\{X_n\}$. For each of N iterations:

1. Draw a point from q-space based on the distribution $\vec{Z}$, and transform it into p-space.

[4] Contrast this with regression, in which the closest point in the direction of an axis defined *a priori* is used.

Figure 4: Generating three-dimensional data from a PCA model



q-space

p-space

2. In p-space, consider a <u>"ball" of noise</u> centered on that point, where points closer to the center of the ball have a higher probability of occurring. The "size" of the ball is determined by the elements of Ψ. For factor analysis in general, the ball may be any ellipsoid whose axes are in the directions of the columns of Λ. For PCA specifically, the ball is a sphere.

3. Draw X_n from that ball of noise in p-space.

Figure 4 shows this process graphically for PCA, with a spherical ball of noise. See the Appendix for the code used to generate the figure.

3.2 Reversing the Generative Process

Our next goal is, given the generative process, to find a distribution or estimate on $\vec{Z}|\vec{X}$, the posterior. We begin by expressing the joint distribution of $\langle Z, X\rangle$ as a $(p+q)$-variate Gaussian with parameters:

$$\vec{\mu} = \left\langle \underbrace{\vec{0}}_{\text{q-vector}}, \underbrace{\vec{0}}_{\text{p-vector}} \right\rangle \tag{3.4}$$

$$\Sigma = \begin{bmatrix} \overbrace{I}^{\text{Var}(Z)} & \overbrace{\Lambda^T}^{\text{Cov}(X,Z)} \\ \underbrace{\Lambda}_{\text{Cov}(Z,X)} & \underbrace{\left(\Lambda\Lambda^T + \Psi\right)}_{\text{Var}(X)} \end{bmatrix} \tag{3.5}$$

We can now use the results from Section 2.2 to find $p(\vec{X})$ and $p(\vec{Z},\vec{X})$ based on the distribution above. Plugging in Equations 3.4 and 3.5 to Equations 2.4, 2.5, 2.6, and 2.7, we find:

$$\vec{X} \sim \mathcal{N}_p\left(\vec{0}, \Lambda\Lambda^T + \Psi\right) \tag{3.6}$$

$$\vec{Z}|\vec{X} \sim \mathcal{N}_q\left(\Lambda^T\left(\Lambda\Lambda^T + \Psi\right)^{-1}\vec{X}, \left(I + \Lambda^T\Psi^{-1}\Lambda\right)^{-1}\right) \tag{3.7}$$

4 Conclusion

Principal component analysis and factor analysis are powerful tools for reducing the dimensionality of data. A generative graphical model for the data with an intuitive geometric interpretation is described. Using properties of the multivariate Gaussian distribution, this generative model can be reversed to find the posterior distribution of the hidden variables. Having thus reversed the generative process, we will turn our attention to efficient methods of computing the posterior, namely Expectation-Maximization, in future lectures.

Appendix A: Sketch Code for Figure 4

The following code for Sketch, a 3D graphics precompiler for LaTeX available at http://www.frontiernet.net/∼eugene.ressler/, was used to generate Figure 4.

```
def O (0,0,0) % origin
def ax (1,0,0)
def ay (0,1,0)
def az (0,0,1)

def circles {
    def n_circle 50
    repeat { 5, scale(0.7) }
        sweep[cull=false]
            {n_circle, rotate(360 / n_circle, (0,0,0), [0,0,1]) }
            (0.25,0,0)
}

def redcircles {
    def n_circle 50
    repeat { 5, scale(0.7) }
        sweep[cull=false,draw=red]
            {n_circle, rotate(360 / n_circle, (0,0,0), [0,0,1]) }
            (0.25,0,0)
}

def redsphere {
    def n_circle 20 def n_sphere 20
    sweep[draw=red,fill=none,draw opacity=0.10]
        {n_sphere, rotate(-360/n_sphere, (O), [0,1,0])}
        sweep {n_circle, rotate(180/n_circle, (O), [0,0,1])}
            (0,1,0)
}

def redspheres {
    repeat { 5, scale(0.7) } {redsphere}
}

def pspace_plane {
    %plane
    polygon[style=dashed,fill=none](0,0,1)(1,0,1)(1,0,0)(0,0,0)
    %special |\path #1 node[right] {$\leftarrow \Lambda$};|(1,.5,.5)

    put { scale(2) then rotate(90, (O), [1,0,0])
        then translate([0.5,0,0.5]) } {circles}
    special |\path #1 node[above] {$\Lambda Z$};|(.5,.1,.5)

    dots[style=ultra thick](.75,0,.75)
    special |\path #1 node[below] {$\Lambda Z_n$};
        |(.75,-.05,.75)

    put { scale(0.25) then translate([0.75,0,0.75]) } {redspheres}

    dots[fill=red,draw=red,style=ultra thick](.8,.15,.8)
    special |\path #1 node[right,red] {$X_n$};|(.8,.15,.8)
}

def pspace {
    %axes
    line[arrows=<->] (ax)(O)(ay)
    line[arrows=->] (O)(az)

    put { rotate(5, (O), [1,0,1]) then translate([0,0.5,0]) } {pspace_plane}

    special |\node at #1 {$p$-space};| (0.5,-0.25,0)
}

put { scale(1.5) then view((5,5,30)) then perspective(100) } {pspace}

def qspace {
    %axes
    line[arrows=<->] (ax)(O)(ay)

    put { scale(2) then translate([0.5,0.5,0]) } {circles}
    special |\path #1 node[right] {$Z$};|(1,0.5,0)

    dots[style=ultra thick](.75,.75,0)
    special |\path #1 node[right] {$Z_n$};|(.75,0.75,0)

    special |\node at #1 {$q$-space};| (0.5,-0.25,0)
}

put { scale(4) then translate([-8,0,0]) } {qspace}

global { language tikz }
```