Quadratic forms and Ellipses

Max Turgeon

19/09/2019

In these notes, I want to clarify a few concepts that were discussed in class.

Let A be a $p \times p$ positive definite matrix. Let $\lambda_1 \geq \cdots \geq \lambda_p$ be its eigenvalues, with corresponding eigenvectors $v_1, \ldots, v_p$; we assume all eigenvectors have unit norm. The matrix A induces a metric on $\mathbb{R}^p$ called the *Mahalanobis distance*:

$$d(x, y) = \sqrt{(x-y)^T A^{-1}(x-y)}.$$

Let $\mu \in \mathbb{R}^p$ be a point of interest. For a fixed constant $c > 0$, the points x that are at a distance c from μ form a *hyperellipsoid* in $\mathbb{R}^p$. Equivently, we can define this hyperellipsoid as

$$\left\{x \in \mathbb{R}^p \mid (x-\mu)^T A^{-1}(x-\mu) = c^2\right\}.$$

As a hyperellipsoid is completely determined by its axes, yet another equivalent definition is that this hyperellipsoid has axes

$$c\sqrt{\lambda_j} v_j, \qquad \text{for } j = 1, \ldots, p.$$

Now, let $A^{-1} = LL^T$ be the Cholesky decomposition of A^{-1}. We then have

$$\begin{aligned}(x-\mu)^T A^{-1}(x-\mu) = c^2 &\iff (x-\mu)^T (LL^T)(x-\mu) = c^2 \\ &\iff (L^T(x-\mu))^T (L^T(x-\mu)) = c^2\end{aligned}$$

In other words, x falls on the hyperellipsoid centered around μ if and only if $y = L^T(x-\mu)$ falls on a hypershpere of radius c centered around the origin.

Therefore, to generate points on the hyperellipsoid, we can

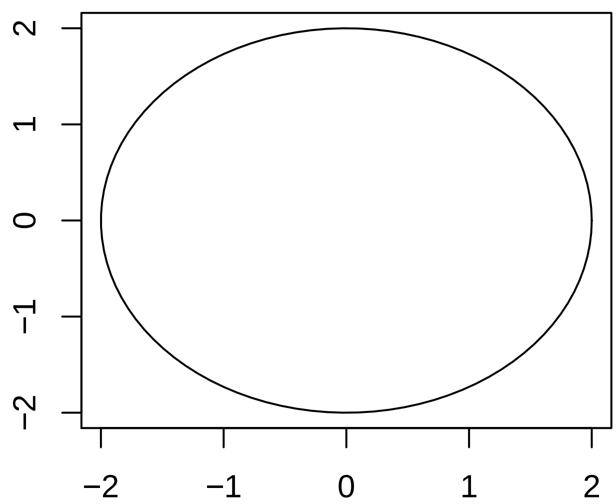
1. Generate points u on the hypersphere of radius c centered around the origin.
2. Transform $v = (L^T)^{-1}u + \mu$.

R code example

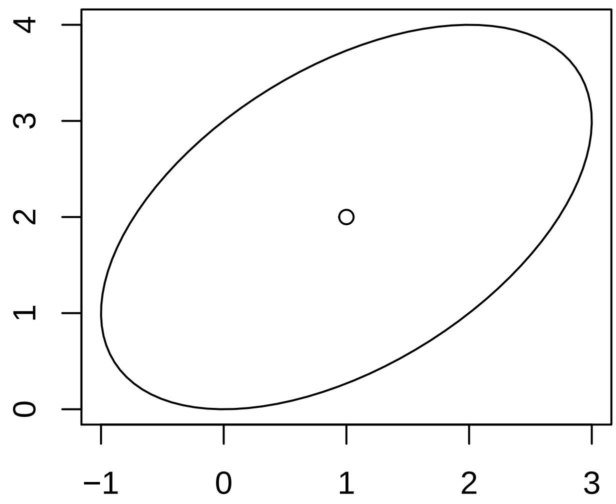
In this section, I give an example of transforming a circle into an ellipse, and I demonstrate that we can get the axes from the eigendecomposition.

```r
# Pick a positive definite matrix in two dimensions
A <- matrix(c(1, 0.5, 0.5, 1), ncol = 2)

# We also pick a point and a radius
mu <- c(1, 2)
c <- 2
```

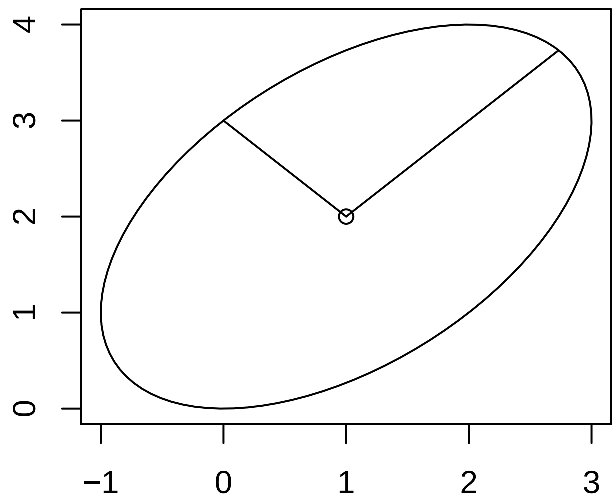
```r
# First create a circle of radius c
theta_vect <- seq(0, 2*pi, length.out = 100)
circle <- c * cbind(cos(theta_vect), sin(theta_vect))

plot(circle, type = 'l',
     xlab = "", ylab = "")
```




```r
# Compute inverse Cholesky
transf_mat <- solve(chol(solve(A)))

# Then turn circle into ellipse
ellipse <- circle %*% t(transf_mat)
# Then translate
ellipse <- t(apply(ellipse, 1, function(row) row + mu))

plot(ellipse, type = 'l',
     xlab = "", ylab = "")
points(mu[1], mu[2])
```




```r
# Compute the eigendecomposition
decomp <- eigen(A, symmetric = TRUE)

first_axis <- c*sqrt(decomp$value[1])*decomp$vectors[,1]
second_axis <- c*sqrt(decomp$value[2])*decomp$vectors[,2]

# Plot everything together
plot(ellipse, type = 'l',
     xlab = "", ylab = "")
points(mu[1], mu[2])

lines(x = c(mu[1], first_axis[1] + mu[1]),
      y = c(mu[2], first_axis[2] + mu[2]))

lines(x = c(mu[1], second_axis[1] + mu[1]),
      y = c(mu[2], second_axis[2] + mu[2]))
```



You can find an animation of this process (i.e. circle to ellipsis followed by a translation) on the course website: https://www.maxturgeon.ca/f19-stat4690/ellipse.gif