# Stat 243 Problem Set 7

*Eugene Yedvabny*

*11/14/2014*

## Q1

The answers to the Chen *et al.* section questions were submitted on Monday but are also attached in hard copy to the end of this report.

## Q2

### a

By definition, if matrix $A$ is eigen-decomposable, $A\vec{v} = \lambda\vec{v} = (\lambda I)\vec{v}$ or alternatively $(A - \lambda I)\vec{v} = 0$ for *any* non-zero $\vec{v}$ if $\lambda$ is an eigenvalue of $A$. This condition is only true when $A - \lambda I$ is singular, which in turn has the property $det(A - \lambda I) = p(\lambda) = 0$, meaning the eigenvalues are *roots* of the characteristic polynomial $p(\lambda)$. For a generic value $x$, $p(x)$ can thus be written in terms of eigenvalues as $p(x) = (\lambda_1 - x)(\lambda_2 - x)...(\lambda_n - x)$. Putting it all together we get

$$det(A) = det(A - 0I) = p(0) = (\lambda_1 - 0)(\lambda_2 - 0)...(\lambda_n - 0) = \prod \lambda_i$$

### b

$$\|A\| = sup\sqrt{(Az)^T Az} = sup\sqrt{(\Gamma\Lambda\Gamma^T z)^T(\Gamma\Lambda\Gamma^T z)} = sup\sqrt{(z^T\Gamma\Lambda\Gamma^T)(\Gamma\Lambda\Gamma^T z)} = sup\sqrt{(z^T\Gamma\Lambda^2\Gamma^T z)} = sup\sqrt{(y^T\Lambda^2 y)}$$

$$\|y\| = sup\sqrt{(\Gamma^T z)^T\Gamma^T z} = sup\sqrt{z^T\Gamma\Gamma^T z} = sup\sqrt{z^T z} = \|z\| = 1$$

The above transformation is possible since $\Gamma$ is orthogonal, so $\Gamma^T = \Gamma^{-1}$ thus $\Gamma^T\Gamma = I$.

$$\|A\| = sup\sqrt{(y^T\Lambda^2 y)} = sup\sqrt{\sum \lambda_i^2 * y_i^2}$$

The norm of A is thus the max of the square root of the above sum. Since $\|y\| = 1$ the sum can be expressed simply as $\lambda_n^2 * 1 + \sum_{i \neq n} \lambda_i^2 * 0$ for any eigenvalue $\lambda_n$ of A. The *maximum* of the sum is therefore the *largest* squared eigenvalue, meaning $\|A\|$ is the largest absolute value (square root of a square) eigenvalue of A.

## Q3

First some naive calculations to check for validity:

```
library(microbenchmark)
n <- 1000
D <- diag(sample(1:9,n,replace=T)) # 10x10 diagonal random matrix
X <- replicate(n,sample(1:9,n,replace=T)) # 10x10 dense random matrix
summary(microbenchmark(DX <- D%*%X, unit='ms'))$median
```

```
## [1] 39.09281
```

```r
summary(microbenchmark(XD <- X%*%D, unit='ms'))$median
```

```
## [1] 42.28403
```

In both cases these are $O(n^3)$ operations.

#### a

DX translates to multiplying every value in the nth *row* of X by the nth diagonal element of D. We can extract just the vector of the diagonal elements from D and do an element-by-element multiplication as $O(n^2)$

```r
# diag(D) returns a vector of diagonal elements
summary(microbenchmark(DX_2 <- X * diag(D),unit='ms'))$median
```

```
## [1] 3.824323
```

```r
all.equal(DX_2,DX)
```

```
## [1] TRUE
```

#### b

XD is now a multiplication of every value in the nth *column* of X by the nth diagonal element of D. Since R is column-major, matrix*vector calculations won't work by default here. The easiest solution is to transpose X, multiply by diag vector of D and transpose again. The multiplication is again $O(n^2)$ and transpose is technically 'free' (depending on the implementation of course).

```r
summary(microbenchmark(XD_2 <- t(t(X)*diag(D)),unit='ms'))$median
```

```
## [1] 12.24624
```

```r
all.equal(XD_2,XD)
```

```
## [1] TRUE
```

## Q4

## Q5