T-61.3050 Machine Learning: Basic Principles
Solution of Prerequisite Knowledge Test 2013

Md Mohsin Ali Khan(Student No: 336790)

September 13, 2013

1 Algebra, Probabilities:

a)

First, assume another function $g : \Omega \to \mathbb{R}$. Now as per the definition of the operator E, we write:

$$\begin{aligned}
E[f(\omega) + g(\omega)] &= \sum_{\omega \in \Omega} P(\omega)(f(\omega) + g(\omega)) \\
&= \sum_{\omega \in \Omega} P(\omega)f(\omega) + P(\omega)g(\omega) \\
&= \sum_{\omega \in \Omega} P(\omega)f(\omega) + \sum_{\omega \in \Omega} P(\omega)g(\omega) \\
&= E[f(\omega)] + E[g(\omega)]
\end{aligned}$$

Second, assume that t is a scalar, Now as per the definition of the operator E, we write:

$$\begin{aligned}
E[tf(\omega)] &= \sum_{\omega \in \Omega} P(\omega)(tf(\omega)) \\
&= \sum_{\omega \in \Omega} tP(\omega)(f(\omega)) \\
&= t \sum_{\omega \in \Omega} P(\omega)(f(\omega)) \\
&= tE[f(\omega)]
\end{aligned}$$

So, we can colclude that $E[\cdot]$ is a linear operator

b)

According to the definition of variance:

$$\begin{aligned}
Var[f(\omega)] &= E[(f(\omega) - E[f(\omega)])^2] \\
&= E[f(\omega)^2 - 2f(\omega)E[f(\omega)] + E[f(\omega)]^2] \\
&= E[f(\omega)^2] - E[2f(\omega)E[f(\omega)]] + E[E[f(\omega)]^2] \text{ //As } E[\cdot] \text{ is a linear operator} \\
&= E[f(\omega)^2] - 2E[f(\omega)]E[f(\omega)] + E[f(\omega)]^2 E[1] \text{ //} E[f(\omega)] \text{ is scalar value} \\
&= E[f(\omega)^2] - E[2f(\omega)E[f(\omega)]] + E[f(\omega)]^2 \sum_{\omega \in \Omega} P(\omega) \\
&= E[f(\omega)^2] - 2E[f(\omega)]^2 + E[f(\omega)]^2 \text{ //It is given that } \sum_{\omega \in \Omega} P(\omega) = 1 \\
&= E[f(\omega)^2] - E[f(\omega)]^2
\end{aligned}$$

2 Matrix Calculus:

It is given that,

$$\mathbf{B} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i {\mathbf{v}_i}^T \tag{1}$$

Multiplying both side of (1) by $\mathbf{v}_j$ where $j \in 1 \dots n$, we can write

$$\begin{aligned}
\mathbf{B}\mathbf{v}_j &= \left(\sum_{i=1}^{n} \lambda_i \mathbf{v}_i {\mathbf{v}_i}^T \right) \mathbf{v}_j \\
\mathbf{B}\mathbf{v}_j &= (\lambda_1 \mathbf{v}_1 {\mathbf{v}_1}^T + \lambda_2 \mathbf{v}_2 {\mathbf{v}_2}^T + \dots + \lambda_n \mathbf{v}_n {\mathbf{v}_n}^T) \mathbf{v}_j \\
\mathbf{B}\mathbf{v}_j &= \lambda_1 \mathbf{v}_1 {\mathbf{v}_1}^T \mathbf{v}_j + \lambda_2 \mathbf{v}_2 {\mathbf{v}_2}^T \mathbf{v}_j + \dots + \lambda_j \mathbf{v}_j {\mathbf{v}_j}^T \mathbf{v}_j + \dots + \lambda_n \mathbf{v}_n {\mathbf{v}_n}^T \mathbf{v}_j \\
\mathbf{B}\mathbf{v}_j &= \lambda_j \mathbf{v}_j {\mathbf{v}_j}^T \mathbf{v}_j \text{ //As } {\mathbf{v}_i}^T \mathbf{v}_j = \delta_{i,j} \text{ where } \delta_{i,j} = 1 \text{ when } i = j \text{ and } \delta_{i,j} = 0 \text{ when } i \neq j \text{)} \\
\mathbf{B}\mathbf{v}_j &= \lambda_j \mathbf{v}_j
\end{aligned}$$

So it is clear that λ_j and $\mathbf{v}_j$ are Eigenvalues and Eigenvectors of the matrix $\mathbf{B}$

3 Algorithms:

The Algorithm:

GenFibonacciuptoN(n)
Begin

1. Declare an Array $Fib[1..n]$
2. $Fib[1] = 1, Fib[2] = 1$
3. For each $i \in \mathbb{N}$ starting from 3 to n in ascending order, do

   i) $Fib[i] = Fib[i-2] + Fib[i-1]$
4. Return the array $Fib[1..n]$. The i-th index holds the i-th Fibonacci Number

End

Time Complexity:

1 Worst case time complexity is: $T(n) = n - 2$ i.e. $T(n) = O(n)$

2 The algorithm has linear time complexity with the growth of input size

4 Basic Data Analysis, Software Tools:

Matlab Code:

```
fid = fopen('T-61_3050_data_set.txt');
A = fscanf(fid, '%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f', [16 inf]);
fclose(fid);

A = A';

x = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
xsquare = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
xvar = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
max = 0;
```

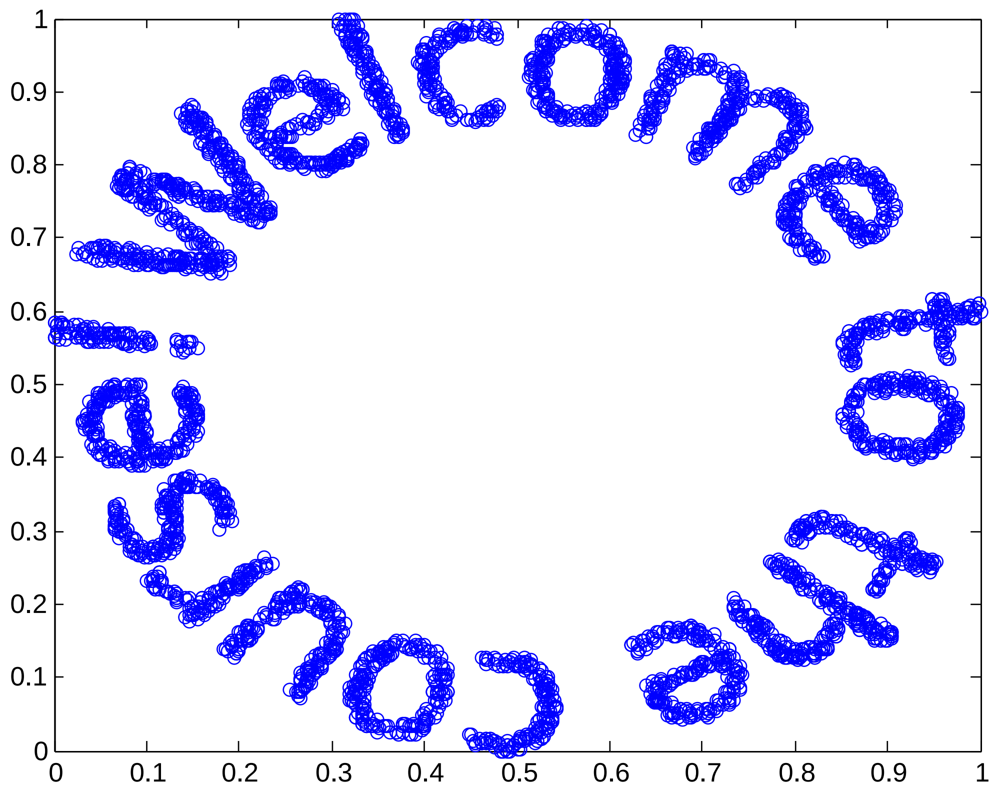
```
secondmax = 0;
maxi = 0;
secondmaxi = 0;

for i=1:16,
    for j=1:4000
        x(i) = x(i) + A(j,i);
        xsquare(i) = xsquare(i) + A(j,i)^2;
    end
xvar(i) = xsquare(i)/4000 - (x(i)/4000)^2;
if xvar(i) > max
    secondmax = max;
    secondmaxi = maxi;
    max = xvar(i);
    maxi = i;
end
end

maxvariancecolumn = A(:,maxi);
secondmaxvariancecolumn = A(:,secondmaxi);

plot(secondmaxvariancecolumn,maxvariancecolumn,'o');
```

Scatterplot of the the columns having largest variance:


Welcome to the course!
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1