# Computational Linear Algebra, Module 13

Maya Shende

Due: May 2nd, 2018

1. The dimension of the nullspace is $n - r$.

2. Let $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ c_1 & c_2 & \dots & c_n \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}$.
   Then, $B = A^T A =$
   $\begin{bmatrix} \dots & c_1 & \dots \\ \dots & c_1 & \dots \\ \vdots & \vdots & \vdots \\ \dots & c_n & \dots \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ c_1 & c_2 & \dots & c_n \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} = \begin{bmatrix} c_1 \cdot c_1 & c_1 \cdot c_2 & c_1 \cdot c_3 & \dots & c_1 \cdot c_n \\ c_2 \cdot c_1 & c_2 \cdot c_2 & c_2 \cdot c_3 & \dots & c_2 \cdot c_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n \cdot c_1 & c_n \cdot c_2 & c_n \cdot c_3 & \dots & c_n \cdot c_n \end{bmatrix}$
   and we know that $c_1 \cdot c_2 = c_2 \cdot c_1$ (dot product is commutative), therefore we can see that $B$ is symmetric.

3. The last step of $\mathbf{w}_i \cdot \mathbf{w}_i = \mathbf{w}_i^T \cdot \mathbf{w}_i = \lambda_i \mathbf{v}_i^T \mathbf{v}_i = \lambda_i$ is true because we know that the $\mathbf{v}_i$ vectors are orthogonal, this $\mathbf{v}_i \cdot \mathbf{v}_i = 1$ leading to $\lambda_i \mathbf{v}_i^T \mathbf{v}_i = \lambda_i$. Therefore, $\mathbf{w}_i \cdot \mathbf{w}_i = \lambda_i$.

4. $V'$ has $n$ rows and $U'$ has $m$ rows

5. 
   $$\begin{aligned} AV' &\implies (m \times n)(n \times r) = (m \times r) \\ U'\Sigma' &\implies (m \times r)(r \times r) = (m \times r) \end{aligned}$$

   Therefore the matrix multiplication is size-compatible since the dimensions are the same.

6. Since $V$ is orthogonal, $V^T = V^{-1}$ and $VV^{-1} = I$.

7. Confirmed:

```
U: Matrix, numRows=3 numCols=2:
    0.528   -0.235
    0.240   -0.881
    0.815    0.411

Sigma: Matrix, numRows=2 numCols=2:
    9.060    0.000
    0.000    1.710

VT: Matrix, numRows=2 numCols=5:
    0.413    0.238    0.651   -0.063    0.587
   -0.068    0.344    0.276   -0.756   -0.479

C: Matrix, numRows=3 numCols=5:
    2.000    1.000    3.000    0.000    3.000
    1.000    0.000    1.000    1.000    2.000
    3.000    2.000    5.000   -1.000    4.000
```

8. The vector $u_k$ has length $m$ and the vector $v_k$ has length $n$, thus to store the two vectors in memory, it only requires $m+n$ space to store the $m+n$ combined elements of the two vectors. For $p$ such terms, the total storage is $p(m+n)$. Suppose $m=2$ and $n=3$, then $m\times n=6$ and $m+n=5$, so $m+n<m\times n$ when $p=1$, but $m+n>m\times n$ when $p>1$. Suppose $m=4$ and $n=4$, then $m\times n=16$ and $m+n=8$, so $m+n\leq m\times n$ when $p\leq 2$ and $m+n>m\times n$ when $p>2$

9. There is a very poor approximation with $p=1$

   $p=1:$

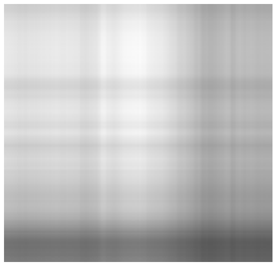
```
C: Matrix, numRows=3 numCols=5:
    2.000    1.000    3.000    0.000    3.000
    1.000    0.000    1.000    1.000    2.000
    3.000    2.000    5.000   -1.000    4.000

D: Matrix, numRows=3 numCols=5:
    1.973    1.138    3.111   -0.303    2.808
    0.898    0.518    1.416   -0.138    1.278
    3.048    1.758    4.806   -0.468    4.337
```

   $p=r:$

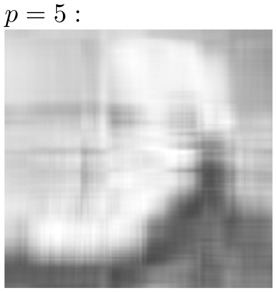
```
C: Matrix, numRows=3 numCols=5:
    2.000    1.000    3.000    0.000    3.000
    1.000    0.000    1.000    1.000    2.000
    3.000    2.000    5.000   -1.000    4.000

D: Matrix, numRows=3 numCols=5:
    2.000    1.000    3.000    0.000    3.000
    1.000    0.000    1.000    1.000    2.000
    3.000    2.000    5.000   -1.000    4.000
```

10. With $p=10$, we start to see facial features, but with $p=50$, we see a clear image.

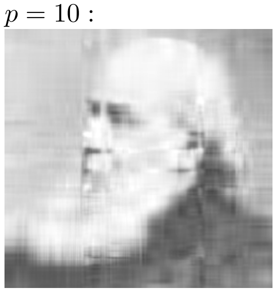
    $p=1:$



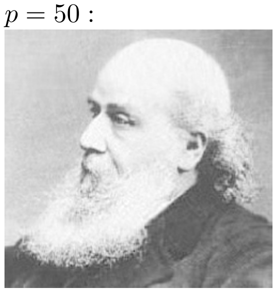
$p = 5:$



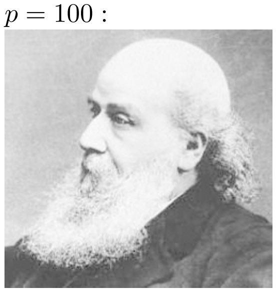
$p = 10:$



$p = 50:$



$p = 100:$



11. We are using the $p$ best values, so we are going to get $p$ components instead of all $r$, thus the rank is $p$. Setting the extra $\sigma$ terms to 0 just ensures that those respective terms of $u$ and $v$ are zeroed out as well.

12. $U$ is orthogonal, so $U^T = U^{-1}$

13. Wikinews dataset:
    $U : (64, 6)$
    $\Sigma : (6, 6)$
    $V^T : (6, 7)$
    Covariance computed with changed coordinates shows clustering among the data.

```
Data matrix created
 0.000000 0.000000 1.000000 1.000000  cost
 1.000000 0.000000 1.000000 1.000000  matrix
 1.000000 1.000000 0.000000 0.000000  multiplication
 0.000000 0.000000 1.000000 1.000000  shows
 1.000000 1.000000 0.000000 0.000000  vector
X: Matrix, numRows=5 numCols=4:
   -0.500   -0.500    0.500    0.500
    0.250   -0.750    0.250    0.250
    0.500    0.500   -0.500   -0.500
   -0.500   -0.500    0.500    0.500
    0.500    0.500   -0.500   -0.500

Covariance matrix without SVD
   1.000   0.700  -0.800  -0.800
   0.700   1.000  -0.500  -0.500
  -0.800  -0.500   1.000   1.000
  -0.800  -0.500   1.000   1.000
U: Matrix, numRows=5 numCols=2:
   -0.481    0.136
   -0.272   -0.962
    0.481   -0.136
   -0.481    0.136
    0.481   -0.136

S: Matrix, numRows=2 numCols=2:
    2.070    0.000
    0.000    0.683

VT: Matrix, numRows=2 numCols=4:
    0.432    0.564   -0.498   -0.498
   -0.751    0.658    0.047    0.047

Y: Matrix, numRows=2 numCols=4:
    0.894    1.166   -1.030   -1.030
   -0.513    0.449    0.032    0.032

Covariance matrix in new coords:
   1.000   1.000  -1.000  -1.000
   1.000   1.000  -1.000  -1.000
  -1.000  -1.000   1.000   1.000
  -1.000  -1.000   1.000   1.000
```

14. The eigen images correspond to the eigen vectors that go into the SVD. They are representative images of the training data. $U : (10000 \times 9)$.

```
R=9
#rows of U=10000 #cols of U=9
U: 10000 x 9
Best match for testimage0: image3 in the training data
```

15.

$$\sum^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \frac{1}{\sigma_n} \end{bmatrix}$$

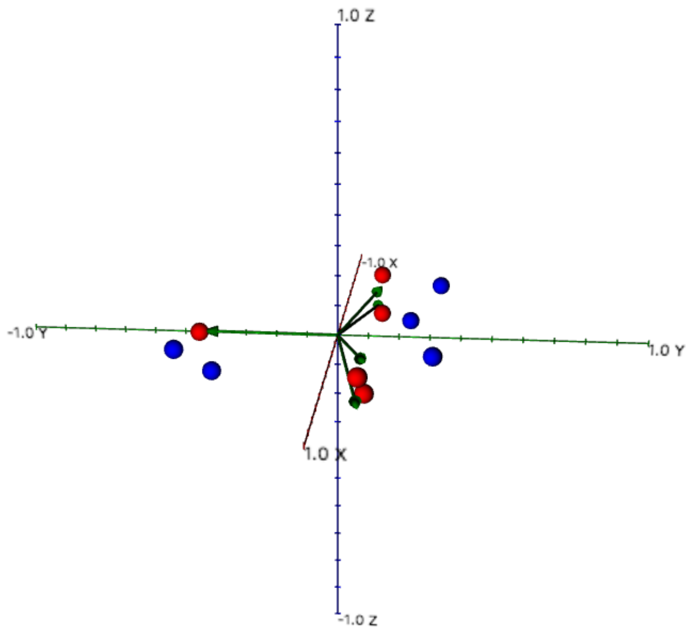
16. Start with $\mathbf{A} = \mathbf{U\Sigma V^T}$. So,

$$\begin{aligned} \mathbf{A} &= \mathbf{U\Sigma V^T} \\ \mathbf{AV} &= \mathbf{U\Sigma} && \text{(since } \mathbf{V} \text{ is orthogonal, } V^T = V^{-1}\text{)} \\ \mathbf{AV\Sigma^{-1}} &= \mathbf{U} \\ \mathbf{AV\Sigma^{-1}U^{-1}} &= \mathbf{I} \\ \mathbf{V\Sigma^{-1}U^{-1}} &= \mathbf{A^{-1}} \end{aligned}$$

17. This tells us that only the first dimension matters:

```
X: Matrix, numRows=3 numCols=5:
    0.620   -0.280   -1.180    0.120    0.720
    0.720   -0.180   -1.280   -0.080    0.820
    0.160    0.260   -0.040   -0.140   -0.240

C: Matrix, numRows=3 numCols=3:
    2.388    2.588   -0.116
    2.588    2.868   -0.066
   -0.116   -0.066    0.172

C2: Matrix, numRows=3 numCols=3:
    5.230    0.000    0.000
    0.000    0.180    0.000
    0.000    0.000    0.018
```

18. Here we get clusters when running SVD:




Output:

```
X: Matrix, numRows=3 numCols=5:
    0.320    0.220   -0.480   -0.380    0.320
   -0.480   -0.380    0.320    0.220    0.320
    0.040   -0.060    0.040   -0.060    0.040

C: Matrix, numRows=3 numCols=3:
    0.628   -0.372    0.016
   -0.372    0.628    0.016
    0.016    0.016    0.012

C2: Matrix, numRows=3 numCols=3:
    1.000    0.000    0.000
    0.000    0.258    0.000
    0.000    0.000    0.010

SVD rank: 3
Yp: Matrix, numRows=3 numCols=5:
    0.566    0.424   -0.566   -0.424   -0.000
    0.109    0.118    0.109    0.118   -0.454
   -0.050    0.049   -0.050    0.049    0.002
```