Lecture 2b - Linear Algebra Revision

OBJECTIVE: In this lecture, we will revise all the definitions and linear algebra facts that we need in order to understand the learning algorithms in later sections of the course.

◇ FAMILIAR DEFINITIONS

Let $\mathbf{x}$ be an n-dimensional column vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

Let $\mathbf{A}$ be an $m \times n$ matrix (m rows, n columns)

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

If $\mathbf{b} = \mathbf{A}\mathbf{x}$, then $\mathbf{b} \in \mathbb{R}^m$ where each component of $\mathbf{b}$,

$$b_i = \sum_{j=1}^{n} a_{ij} x_j \qquad i = 1, 2, \dots, m.$$

We can view $\mathbf{x} \rightarrow \mathbf{A}\mathbf{x}$ as a *linear map*. i.e., for any (vectors) $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and any (scalar) $\alpha \in \mathbb{R}$,

$$\begin{aligned} \mathbf{A}(\mathbf{x} + \mathbf{y}) &= \mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{y} \\ \mathbf{A}(\alpha\mathbf{x}) &= \alpha\mathbf{A}\mathbf{x} \end{aligned}$$

Question: Which side is more expensive to compute?

◇ MATRIX-VECTOR MULTIPLICATION

Let $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix}$ i.e., $\mathbf{a}_j \in \mathbb{R}^m$ is the j^{th} column of $\mathbf{A}$. Then, $\mathbf{b} = \mathbf{A}\mathbf{x} = \sum_{j=1}^{n} x_j \mathbf{a}_j$ i.e., $\mathbf{b}$ *is a linear combination of the columns of* $\mathbf{A}$.

$$\begin{bmatrix} \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= x_1 \begin{bmatrix} \mathbf{a}_1 \end{bmatrix} + x_2 \begin{bmatrix} \mathbf{a}_2 \end{bmatrix} + \dots + x_n \begin{bmatrix} \mathbf{a}_n \end{bmatrix}$$

Note 1 *This is nothing but a change of viewpoint (and notation).*
Instead of viewing $\mathbf{A}\mathbf{x} = \mathbf{b}$ *as "*$\mathbf{A}$ *acting on* $\mathbf{x}$ *to give* $\mathbf{b}$*", we view as "*$\mathbf{x}$ *acting on* $\mathbf{A}$ *to produce* $\mathbf{b}$*".*

◇ DETERMINANTS, INDEPENDENCE AND RANK

★

◇ EIGEN-DECOMPOSITIONS

The intuition is to find a scalar λ that has the same effect as $\mathbf{A}$ on $\mathbf{x}$.

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

⋆

⋆ Example:

⋆ Example:

◇ SPECTRAL PROPERTIES

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ be full rank, then

(i) $\mathbf{A}^{-1}$ has eigenvalues $1/\lambda_1, \ldots, 1/\lambda_m$.

⋆ Proof:

(ii) $\mathbf{A} - k\mathbf{I}$ has eigenvalues $\lambda_1 - k, \ldots, \lambda_m - k$.

⋆ Proof:

(iii) $\mathbf{A}^n$ has eigenvalues $\lambda_1^n, \ldots, \lambda_m^n$.

⋆ Proof:

(iv) *Spectral Mapping theorem:*

Theorem 1 *The matrix* $k_n\mathbf{A}^n + k_{n-1}\mathbf{A}^{n-1} + \ldots + k_{\cdot}\mathbf{A} + k_0\mathbf{I}$ *has eigenvalues* $k_n\lambda_j^n + k_{n-1}\lambda_j^{n-1} + \ldots + k_1\lambda_j^1 + k_0$ *for* $j = 1 \ldots m$.

The proof is question 1 of the homework.

(v) Trace and determinant:

⋆

◇ TRANSPOSE

Definition 1 *The* transpose $\mathbf{A^T}$ *of an* $m \times n$ *matrix* $\mathbf{A}$ *is an* $n \times m$ *matrix where the (i,j) entry of* $\mathbf{A^T}$ *is the (j,i) entry of* $\mathbf{A}$
$\leftrightarrow$ *interchange the rows with the columns*

⋆
e.g., If $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$, then $\mathbf{A^T} =$

If $\mathbf{A} = \mathbf{A^T}$ (so $\mathbf{A}$ has to be square!) then $\mathbf{A}$ is said to be *symmetric.*

◇ SPD MATRICES

Definition 2 *A matrix* $\mathbf{A}$ *is* symmetric positive definite *(SPD) if it is symmetric and*

$$\mathbf{x^T A x} > 0, \qquad \forall \mathbf{x} \neq \mathbf{0}.$$

Theorem 2 *If* $\mathbf{A}$ *is SPD, its eigenvalues are positive.*

⋆ Proof:

◇ INNER PRODUCT

Definition 3 *Let* $\mathbf{x}$,$\mathbf{y} \in \mathbb{R}^m$. *Then, the* inner product *of* $\mathbf{x}$ *and* $\mathbf{y}$ *is a* <u>*scalar*</u>

$$\mathbf{x^T y} = \sum_{i=1}^{m} x_i y_i$$

The (Euclidean) length *of a vector* $\mathbf{x}$ *is written as* $\|\mathbf{x}\|$ *and can be defined as the square root of the inner product of the vector with itself*

$$\|\mathbf{x}\| = \sqrt{\mathbf{x^T x}} = \left(\sum_{i=1}^{m} x_i^2 \right)^{\frac{1}{2}}$$

Also, if the angle between vectors $\mathbf{x}$ *and* $\mathbf{y}$ *is* α, *we have*

$$\cos\ \alpha = \frac{\mathbf{x^T y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

◇ ORTHOGONAL MATRICES

A square matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$ is *orthogonal* if

$$\mathbf{Q^T} = \mathbf{Q^{-1}}.$$

i.e., $$\mathbf{Q^T Q} = \mathbf{Q Q^T} = \mathbf{I}$$

$$\begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_m^T \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_m \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

<u>NOTATION</u>

$$\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

δ_{ij} is called the *Kronecker delta*

Lecture 3 - *The Singular Value Decomposition (SVD)*

OBJECTIVE: The SVD is a matrix factorization that has many applications: e.g., information retrieval, least-squares problems, image processing.

◇ EIGENVALUE DECOMPOSITION

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$. If we put the eigenvalues of $\mathbf{A}$ into a diagonal matrix $\mathbf{\Lambda}$ and gather the eigenvectors into a matrix $\mathbf{X}$, then the eigenvalue decomposition of $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}.$$

But what if $\mathbf{A}$ is not a square matrix? Then the SVD comes to the rescue.

◇ FORMAL DEFINITION OF THE SVD

Given $\mathbf{A} \in \mathbb{R}^{m \times n}$, the SVD of $\mathbf{A}$ is a factorization of the form

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where $\mathbf{u}$ are the left **singular vectors**, σ are the **singular values** and $\mathbf{v}$ are the right singular vectors.

$\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal with positive entries (singular values in the diagonal).
$\mathbf{U} \in \mathbb{R}^{m \times n}$ with orthonormal columns.
$\mathbf{V} \in \mathbb{R}^{n \times n}$ with orthonormal columns.
($\Rightarrow \mathbf{V}$ is orthogonal so $\mathbf{V}^{-1} = \mathbf{V}^T$)

The equations relating the right singular values $\{\mathbf{v}_j\}$ and the left singular vectors $\{\mathbf{u}_j\}$ are

$$\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j \qquad j = 1, 2, \ldots, n$$

i.e.,

$$\mathbf{A}\begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

or $\mathbf{AV} = \mathbf{U\Sigma}$.

⋆

1. There is no assumption that $m \geq n$ or that $\mathbf{A}$ has full rank.
2. All diagonal elements of $\mathbf{\Sigma}$ are non-negative and in non-increasing order:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$$

where $p = \min(m, n)$

Theorem 3 *Every matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *has singular value decomposition* $\mathbf{A} = \mathbf{U\Sigma V}^T$
Furthermore, the singular values $\{\sigma_j\}$ *are uniquely determined.*
If $\mathbf{A}$ *is square and* $\sigma_i \neq \sigma_j$ *for all* $i \neq j$*, the left singular vectors* $\{\mathbf{u}_j\}$ *and the right singular vectors* $\{\mathbf{v}_j\}$ *are uniquely determined to within a factor of* ± 1.

◇ EIGENVALUE DECOMPOSITION

Theorem 4 *The nonzero singular values of* $\mathbf{A}$ *are the (positive) square roots of the nonzero eigenvalues of* $\mathbf{A}^T\mathbf{A}$ *or* $\mathbf{A}\mathbf{A}^T$ *(these matrices have the same nonzero eigenvalues).*

⋆ Proof:

◇ LOW-RANK APPROXIMATIONS

Theorem 5 $\|\mathbf{A}\|_2 = \sigma_1$, *where* $\|\mathbf{A}\|_2 = \max_{\mathbf{x}\neq 0} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\|\mathbf{x}\|\neq 1} \|\mathbf{A}\mathbf{x}\|$.

⋆ Proof:

Another way to understand the SVD is to consider how a matrix may be represented by a sum of rank-one matrices.

Theorem 6

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

where r is the rank of $\mathbf{A}$.

⋆ Proof:

What is so useful about this expansion is that the ν^{th} *partial sum captures as much of the "energy" of* $\mathbf{A}$ *as possible by a matrix of at most rank-ν*. In this case, "energy" is defined by the 2-norm.

Theorem 7 *For any ν with $0 \leq \nu \leq r$ define*

$$\mathbf{A}_\nu = \sum_{j=1}^{\nu} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

If $\nu = p = \min(m, n)$, define $\sigma_{\nu+1} = 0$.

Then,

$$\|\mathbf{A} - \mathbf{A}_\nu\|_2 = \sigma_{\nu+1}$$

Lecture 4 - *Fun with the SVD*

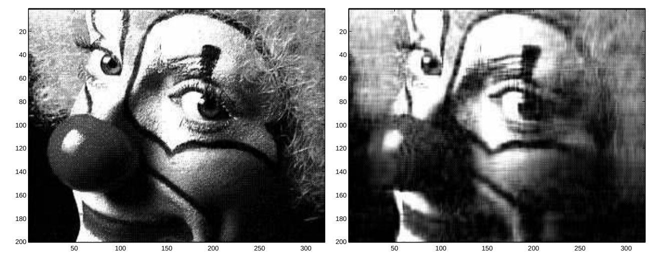
OBJECTIVE: Applications of the SVD to image compression, dimensionality reduction, visualization, information retrieval and latent semantic analysis.

◇ IMAGE COMPRESSION EXAMPLE

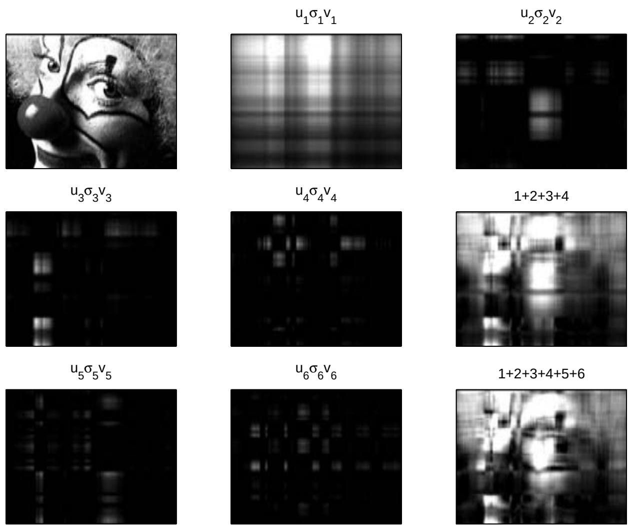
```
load clown.mat;
figure(1)
colormap('gray')
image(A);

[U,S,V] = svd(A);
figure(2)
k = 20;
colormap('gray')
image(U(:,1:k)*S(1:k,1:k)*V(:,1:k)');
```

The code loads a clown image into a 200×320 array $\mathbf{A}$;

displays the image in one figure; performs a singular value decomposition on $\mathbf{A}$; and displays the image obtained from a rank-20 SVD approximation of $\mathbf{A}$ in another figure. Results are displayed below:



The original storage requirements for $\mathbf{A}$ are $200 \cdot 320 = 64,000$, whereas the compressed representation requires $(200 + 300 + 1) \cdot 20 \approx 10,000$ storage locations.



Smaller eigenvectors capture high frequency variations (small brush-strokes).

◇ TEXT RETRIEVAL - LSI

The SVD can be used to cluster documents and carry out information retrieval by using concepts as opposed to word-matching. This enables us to surmount the problems of synonymy (car,auto) and polysemy (money bank, river bank). The data is available in a term-frequency matrix

★

If we truncate the approximation to the k-largest singular values, we have

$$\mathbf{A} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

So

$$\mathbf{V}_k^T = \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{A}$$

⋆

In English, $\mathbf{A}$ is projected to a lower-dimensional space spanned by the k singular vectors $\mathbf{U}_k$ (eigenvectors of $\mathbf{A}\mathbf{A}^T$). To carry out **retrieval**, a **query** $\mathbf{q} \in \mathbb{R}^n$ is first projected to the low-dimensional space:

$$\widehat{\mathbf{q}}_k = \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{q}$$

And then we measure the angle between $\widehat{\mathbf{q}}_k$ and the $\mathbf{v}_k$.

⋆

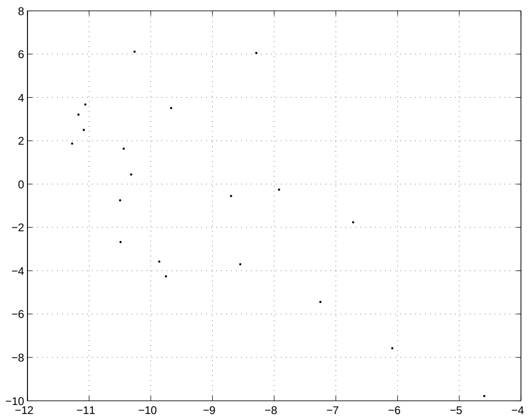
◇ PRINCIPAL COMPONENT ANALYSIS (PCA)

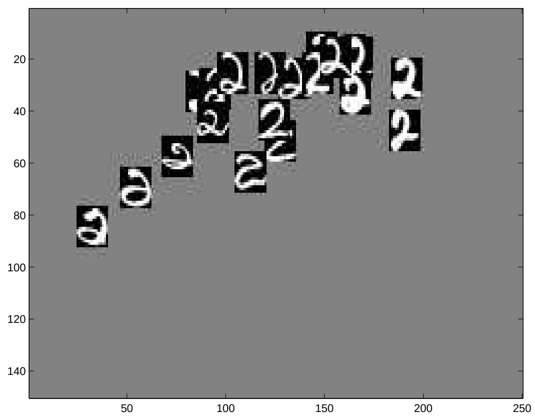
The columns of $\mathbf{U\Sigma}$ are called the **principal components** of $\mathbf{A}$. We can project high-dimensional data to these components in order to be able to visualize it. This idea is also useful for cleaning data as discussed in the previous text retrieval example.

⋆

For example, we can take several 16×16 images of the digit 2 and project them to 2D. The images can be written as vectors with 256 entries. We then from the matrix $\mathbf{A} \in \mathbb{R}^{n \times 256}$, carry out the SVD and truncate it to $k = 2$. Then the components $\mathbf{U}_k \mathbf{\Sigma}_k$ are 2 vectors with n data entries. We can plot these 2D points on the screen to visualize the data.

⋆


8
6
4
2
0
−2
−4
−6
−8
−10
−12
−11
−10
−9
−8
−7
−6
−5
−4


20
40
60
80
100
120
140
50
100
150
200
250