TANA09 Lecture 6

The *QR* decomposition

- Projections. Computer graphics.

The singular value decomposition (SVD).

- Definition. Properties. Fundamental subspaces.
- Linear systems of equations. Low rank approximation.
- Application - Automatic character identification.

Projections

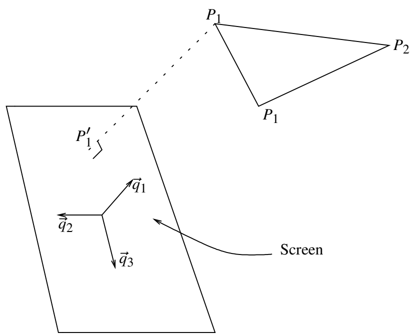
Lemma Suppose $Q_1 = (q_1, \ldots, q_k)$ has *orthogonala* columns. The *orthogonal projection* onto the subspace $\mathrm{span}(q_1, \ldots, q_k)$ is

$$P = Q_1 Q_1^T.$$

Application In computer graphics objects are represented by polygons. Each corner has coordinates in $\mathbb{R}^3$. To draw the object we need to compute projections onto the screen.

Draw polygons in the order *closest last* and we get the correct image. We need the distance z from the screen. This is called z-buffer.

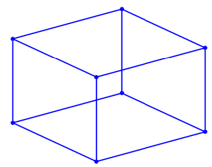
How to carry out the computations?



The polygon has corners $\{P_1, P_2, P_3\}$ to be projected on the screen $\mathrm{span}(\vec{q}_2, \vec{q}_3)$ in the direction given by the normal vector $\vec{q}_1$.

We have

$$P'_k = (q_2^T P_k) q_2 + (q_3^T P_k) q_3, \quad \text{och} \quad z_k = (\vec{q}_1^T P_k).$$



Example We create a matrix P containing the corners of a cube.

```
>> P =
     0     1     0     1     0     1     0     1
     0     0     1     1     0     0     1     1
     0     0     0     0     1     1     1     1
>> ind=[1 2 4 3 1 5 6 2 6 8 4 8 7 5 7 3];
>> plot3(P(1,ind),P(2,ind),P(3,ind),'b-*');
```

Can we recreate the same figure using projections and a 2*D* plot?

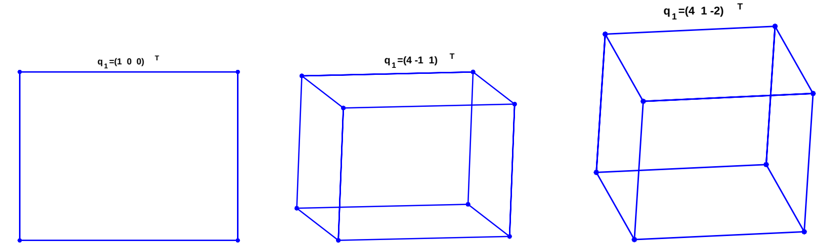
Let S_k be the projection of P_k onto the screen and q_1 be the normal vector. We obtain

```
>> q1=[1 1 0]';[Q,R]=qr(q1);
>> for k=1:8, S(:,k)=Q(:,2:3)'*P(:,k);,end
>> ind=[1 2 4 3 1 5 6 2 6 8 4 8 7 5 7 3];
>> plot(S(1,ind),S(2,ind),'b-*');
```

The distance from the point to the screen is

```
>> for k=1:8, z(k)=Q(:,1)'*P(:,k);,end
```

In this case we draw hidden lines as well.



We look at the cube from different directions q_1. The figures show the projection on the screen along the direction q_1.

If we were to draw surfaces we would sort according to the distance from the screen.

The singular value decomposition

Proposition Every matrix $A \in \mathbb{R}^{m\times n}$ has a decomposition

$$A = U\Sigma V^T,$$

where U and V are orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ is *diagonal* with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(n,m)} \geq 0$

Remark The equivalent formula

$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

writes A as a sum of rank one matrices.

Example Compute the SVD in Matlab using

```
>> A=[1 -2 3 ; -2 3 1 ; 2 -4 6 ; -1 2 -3];
>> [U,S,V]=svd(A)
U =                                S =
 -0.402  0.068  0.912 -0.000        9.278      0      0
  0.167  0.985 -0.000  0.000            0  3.452      0
 -0.805  0.136 -0.365  0.447            0      0  0.000
  0.402 -0.068  0.182  0.894            0      0      0

V =
 -0.296 -0.452  0.841
  0.574  0.619  0.535
 -0.762  0.642  0.076
```

The matrix A has two linearly independent rows, i.e. $\text{rank}(A) = 2$.

Example Suppose that $A \in \mathbb{R}^{4\times 3}$. Then

$$A = \begin{pmatrix} u_1 & u_2 & u_3 & u_4 \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^T.$$

Remark The diagonal elements σ_j are called *singular values*. The vectors u_i and v_j are *singular vectors*.

Question How to compute the condition number $\kappa_2(A)$?

Lemma Let $A \in \mathbb{R}^{m\times n}$. Then $\|A\|_2 = \sigma_1$.

Lemma Let $A \in \mathbb{R}^{n\times n}$. Then if $\sigma_n > 0$, the inverse A^{-1} exists, and $\|A^{-1}\|_2 = \sigma_n^{-1}$.

Remark The condition number is $\kappa_2(A) = \sigma_1/\sigma_n$.

The singular value decomposition is significantly more time consuming to compute than the *LU* decomposition.

The Fundamental Subspaces

Lemma If $\sigma_k > 0$ and $\sigma_{k+1} = 0$ then $\mathrm{Rank}(A) = k$.

Remark This means that

$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^T.$$

Lemma If $\mathrm{rank}(A) = k$ then $\mathrm{Range}(A) = \mathrm{span}\{u_1, \ldots, u_k\}$.

Question How to write a basis for $\mathrm{null}(A)$?

Lemma If $\mathrm{rank}(A) = k$ then $\mathrm{null}(A) = \mathrm{span}\{v_{k+1}, \ldots, v_n\}$.

Example Let $Ax = b$. It is often useful to split x and b into components, e.g.

$$x = x_1 + x_2, \quad \text{where} \quad x_1 \in \mathrm{null}(A)^\perp \text{ and } x_2 \in \mathrm{null}(A).$$

Remark It holds that $A^T = V\Sigma U^T$ so $\mathrm{range}(A)^\perp = \mathrm{null}(A^T)$.

Linear Systems of Equations

Lemma If $A \in \mathbb{R}^{m\times n}$ then $Ax = b$ has a solution if $b \in \text{range}(A)$. The solution is unique if $\text{rank}(A) = n$.

Remark If $\text{rank}(A) = k$ and $b \in \text{range}(A)$ then the general solution of $Ax = b$ is

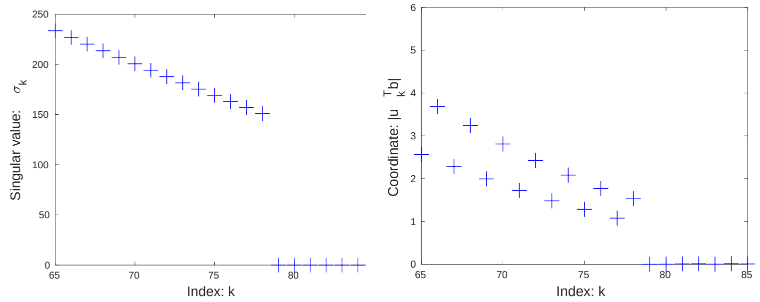
$$x = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i + \sum_{i=k+1}^{n} c_i v_i.$$

where $c_{k+1}, \ldots, c_n$ are undetermined parameters.

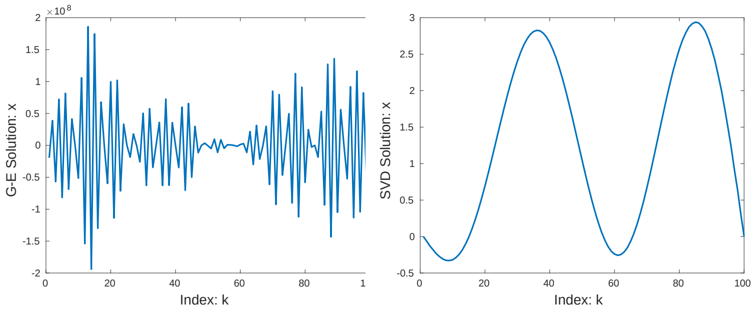
Question How to verify $b \in \text{range}(A)$?

Example In an application we have an 500×100 matrix A and want to solve a linear system $Ax = b$. Since b is obtained by measurements and we know the model is valid $b \in \text{range}(A)$.

In Matlab Compute the SVD and plot the singular values and also the coefficients $|u_i^T b|$.



Remark We see that $\sigma_{78} = 300.3492$ and $\sigma_{79} = 2.3 \cdot 10^{-10}$ so the rank is $k = \text{rank}(A) = 78$.



Results Solutions using `x=A\b` and `x=Vk*inv(Sk)*Uk'*b`.

After eliminating the small singular values the solution is very good.

Least squares problems

Recall Let $A \in \mathbb{R}^{m\times n}$. Define $A^+ = (A^T A)^{-1} A^T$ then $x = A^+ b$ is the vector that minimize $\|Ax - b\|_2$.

Definition If $A \in \mathbb{R}^{m\times n}$ and $\text{rank}(A) = k$ then

$$A^+ = \sum_{i=1}^{k} \frac{v_i u_i^T}{\sigma_i}.$$

Remark If $\text{rank}(A) = n$ then $(A^T A)^{-1}$ exists. Otherwise we get the minimum norm least squares solution.

Projections and the SVD

Lemma Suppose $V \in \mathbb{R}^{n\times k}$ has orthonormal columns. Then

$$P = VV^T,$$

is an *orthogonal projection* onto range(V).

Example Suppose $A = U\Sigma V^T$ and rank$(A) = k$. Partition

$$U = (U_k, U_{m-k}) \quad \text{and} \quad V = (V_k, V_{n-k}) .$$

where, e.g, $U_k = (u_1, \ldots, u_k)$.

Question What is the orthogonal projection onto $(\text{null}(A))^{\perp}$?

Example Suppose that the decomposition $A = U\Sigma V^T$ is available and we want to compute the distance from b to the subspace range(A), i.e. find the minimum of $\|Ax - b\|_2$.

How should we organize the computations?

Application: Low rank approximation

Theorem If $A \in \mathbb{R}^{m\times n}$ then

$$\min_{\text{rank}(B)=k} \|A - B\|_2 = \sigma_{k+1}, \quad B = \sum_{i=1}^{k} \sigma_i u_i v_i^T .$$

Remark If the number σ_n is small then A is close to rank deficient.

Definition Let $\varepsilon > 0$. The *numerical rank* of A is

$$\text{rank}(A, \epsilon) = \max_k \{\sigma_k > \varepsilon\}.$$

Remark Let μ be the machine precision. If A has full rank but rank$(A, \mu) < n$ its likely better to treat A as rank deficient.

Example Suppose we have a 256×256 image represented by a matrix A. To save memory we approximate the image. The error $\|A - A_k\|_2$ should be small.

Solution Let $A \in \mathbb{R}^{m \times n}$. Compute the decomposition $A = U\Sigma V^T$ and set

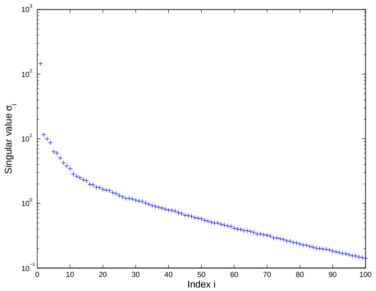
$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T.$$

The error is $\|A - A_K\| = \sigma_{k+1}$.

Instead of storing A we store the vectors u_i and v_i, $i = 1, \ldots, k$. Less storage needed if

$$k(m + n + 1) < mn.$$

How to pick the parameter k?



The first 100 singular values σ_i. Introduce an *energy*

$$E_k^2 = \frac{\sigma_2^2 + \cdots + \sigma_k^2}{\sigma_2^2 + \cdots + \sigma_n^2}$$

and pick k so that most of the total energy in the image is included.

Ignore the first singular value when computing the energy.

Approximation A_1 Approximation A_{10} Approximation A_{20}

Low rank approximations A_1, A_{10} och A_{20}. The energy is $E_1 = 0$, $E_{10} = 0.918$ and $E_{20} = 0.96$.

The approximation is optimal in $\| \cdot \|_2$. Other norms are better if you want to minimise the "visible" difference between imnages.

The Classification Problem

Suppose we study *objects* of a certain type and that objects occur in different variants, or *classes*. Given a new object we want to determine which class it belongs to.

We do the following

- We collect a large number of *reference objects* $\{R_k\}$ of known class. This is the *reference set*.
- Let S be unknown and R_k belong to the reference set. The *distance function* $d(S, R_k)$ measures the similarity between the two objects.

Example Incomming email can either be a spam mail or not.

Nearest Neighbour Classification

Algorithm Let $\{R_k\}$ be the reference set and $d(\cdot,\cdot)$ be the distance function. Do

1. Find k such that $d(S, R_k) = \min_j d(S, R_j)$.

2. The object S is of the same class as R_k.

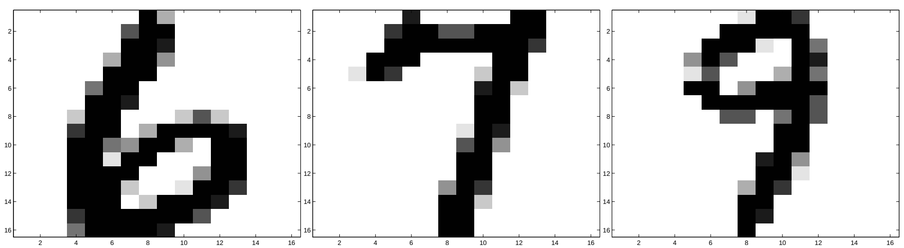
Remark This method is simple, but very accurate assuming the reference set is large enough. It is also too inefficient for practical use.

A good distance function is needed.

Classification of Handwritten Digits

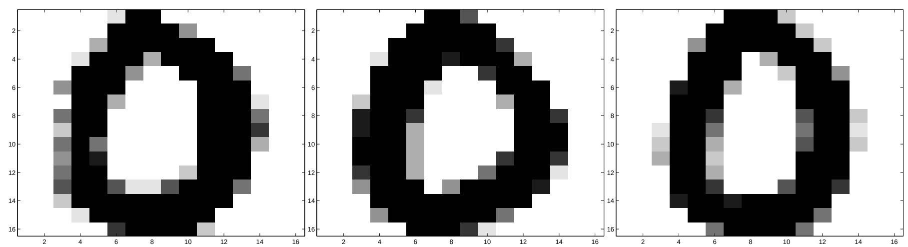
Example A *reference set* consists of $n = 1707$ digits taken from letters (postal codes). The images are stored as 16×16 pixels.

In Matlab `DisplayDigit( RefSet(:,1) );`



Measure distance using Euclidean norm $\|S - R_j\|_2$.

Example The digit S_1 and its two nearest neighbours R_{11} and R_{303}.



This is a successful classification. Of the 20 nearest there are 18 nines and 2 sevens.

Of a (very difficult) *Test Set* of size 2007 a total of 92.8% are classified correctly. Objects are vectors in $\mathbb{R}^{256}$ so have vector space structure.

Observation The reference set contains many examples of digits that are very similar.

Let $R^{(k)}$ be a matrix of size $256 \times n_k$ consisting of all reference digits of type k, $k = 0, 1, \ldots, 9$.

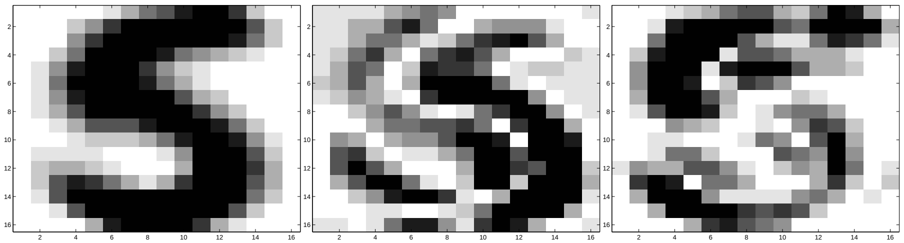
Approximation Compute $R^{(k)} = U^{(k)} \Sigma V^T$ and use

$$\text{span}(R_1^{(k)}, \ldots, R_{n_k}^{(k)}) \approx \text{span}(u_1^{(k)}, \ldots, u_m^{(k)})$$

where m is the dimension of the subspace.

Remark A low dimension m is sufficient to accurately describe the most common variations in writing style.

Example The first 3 basis vectors $u_k^{(5)}$. Created from a total of 88 5:s from the reference set.



Just 5-10 basis vectors very accurately describe the digit 5 and its variations.

For each type of digit we find a low rank approximating subspace $U_m^{(k)} = \{u_1^{(k)}, \ldots, u_m^{(k)}\}$, $k = 0, 1, \ldots, 9$.

Algorithm Classify an unknown object S by

1. Find k such that $d(S, U_m^{(k)}) = \min_j d(S, U_m^{(j)})$.

2. The object S is of class k.

The distance $d(S, U^{(k)})$ is the distance from S to the subspace. This is a least squares problem. The matrices U_m^k has orthogonal columns.

Using subspaces of dimension $m = 10$ we classify 93.2% of the test set correctly. Bad reference digits are removed.

Summary

- Projections, reflections, and other operations can be implemented using the QR decomposition
- The singular value decomposition (SVD) provides orthogonal bases for the fundamental subspaces of a matrix A. Can solve systems of equations, deal with ill-conditioning, and more.
- Application: Character recognition using the SVD.