3

This means that the columns of $\underset{\sim}{U}$ and $\underset{\sim}{V}$ are *also* orthogonal!

This is easy to show. Let's look at the product:

$$\underset{\sim}{C} = \underset{\sim}{U}^T \underset{\sim}{U} = \underset{\sim}{I}$$

Now the element c_{ij} is the inner product of the i^{th} column of $\underset{\sim}{U}$ (e.g., $\underset{\sim}{u}_i$) w/ the j^{th} column!

$$C_{ij} = \underset{\sim}{u}_i^T \underset{\sim}{u}_j = I_{ij}$$

But if $i \neq j$ this is an off-diag. element of the identity matrix!

$$\text{So} : \quad \underset{\sim}{u}_i^T \underset{\sim}{u}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

The same applies to $\underset{\sim}{V}$!

We can use the values (magnitude) of σ_{ii} to explore the sensitivity of $\underset{\sim}{z}$ to variations in $\underset{\sim}{X}$.

Suppose the model $\underset{\sim}{z} = \underset{\sim}{f}(\underset{\sim}{x})$ represents a multiple input / multiple output model of plant operations.

$\underset{\sim}{x}$ might be the mass flow rate of input streams

$\underset{\sim}{z}$ might be the resulting products needed elsewhere in the plant.

In general, you want to minimize $\underset{\sim}{z}'$ - the deviation from some optimal output $\underset{\sim}{z}_0$.

Question: what fluctuations $\underset{\sim}{x}'$ give rise to the <u>maximum</u> change in $\underset{\sim}{z}'$?

This is addressed by SVD, looking at the <u>largest</u> singular value σ_{11}.

The most dangerous mode is any vector of disturbances $\underset{\sim}{x}'$ proportional to the <u>first</u> <u>column</u> of $\underset{\approx}{\tilde{V}}$

The resulting output vector is proportional to the first column of $\underset{\sim}{U}$

The amplification factor is the corresponding singular value σ_{11}

If the i^{th} singular value is zero, then ~~that~~ a disturbance vector $\underset{\sim}{x}' \sim V(:,i)$ (i^{th} column of $\underset{\sim}{V}$) has no effect on $\underset{\sim}{z}$!

In the homework project 3 you are doing the same thing, only in reverse: what is the magnitude of $\underset{\sim}{x}$ to get some output $\underset{\sim}{z}$.

Thus, you are interested in the smallest singular values, rather than (as in the usual case) the largest.

This all is an example of Principal Component Analysis, or PCA.

Data Compression

73

Let $\underset{\sim}{u}_i$ be the columns of $\underset{\approx}{U}$
and let $\underset{\sim}{v}_i$ be the columns of $\underset{\approx}{V}$

(or the rows of $\underset{\approx}{V}^T$)

Then:

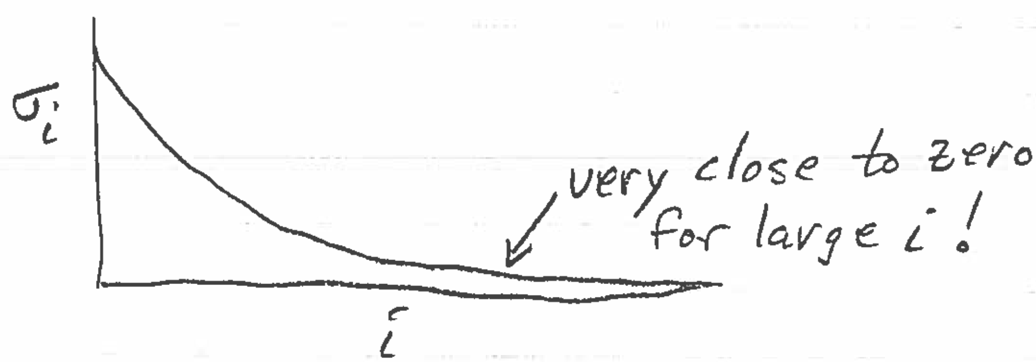
$$\underset{\approx}{A} = \sum_{i=1}^{n} \sigma_i \, \underset{\sim}{u}_i \, \underset{\sim}{v}_i^T$$

vector composition or outer product

Suppose $\underset{\approx}{A}$ is a huge matrix, say

$\underset{\approx}{A} \equiv 512 \times 512$ (usual picture size)

If we plot σ_i vs. i we get:



Thus if we truncate picture after $i \cong 20$ we can reproduce the original picture!

Take $$\underset{\approx}{A} \cong \sum_{i=1}^{20} \sigma_i \underset{\sim}{u}_i \underset{\sim}{v}_i^T$$

This has a total of

$$20 \times (1 + 512 + 512) = 20{,}500$$

vs. 262,144 elements!

For larger matrices you get even greater reduction in size

Now SVD requires a large # of computations, so it's a tradeoff between storage/transmission considerations and comp. speed!

Recently, transmission rates have not kept up w/ processor speed, so equation has shifted toward computationally expensive compression

An extreme example: the Gallileo Jupiter probe

- stuck high gain antenna, so must use wimpy low gain antenna
- transmission rate = $O(10 \text{ bps})$!

$\Rightarrow$ Use on-board computers to compress images, then transmit very slowly!

They are using JPEG compression - not quite SVD, but related. It's using fourier transform of data & only transmitting some of the modes.

You really appreciate data compression when downloading images over a modem!

~~Run matlab example~~

we'll look at SVD again later

Systems of linear equations

Summary of Solution approaches

$$A x = b$$

1) If A is square & well conditioned

   a) PLU factorization (Gaussian elimination

   b) A^{-1} so $x = A^{-1} b$

      - convenient, but
      - squares cond(A) & takes 3x longer...

   c) If A is sparse special techniques are much fas

2) If A is square but close to singularity

   a) QRP factorization
      (zero out rows of R which are very small)

   b) SVD
      (zero out singular values which are very small)

      ⇒ computationally intensive, but guaranteed to get sol'n w/ min 2-norm

3) If $\underset{\approx}{A}$ is overdetermined $(n > m)$
- often occurs in regression

a) QRP

b) $\underset{\approx}{A}^T \underset{\approx}{A} \underset{\sim}{x} = \underset{\approx}{A}^T \underset{\sim}{b}$

both minimize 2-norm of residual

- square. $m \times m$ problem
- solve via LU, inverse, etc.
- if m is large, cond maybe a problem

4) If $\underset{\approx}{A}$ is underdetermined $(m > n)$

- use SVD (min 2-norm of $\underset{\sim}{x}$)

SVD: Uses

1) Solve linear equations, particularly those close to singularity

2) Get <u>approx</u> solutions which yield more robust control by zeroing out smallest singular values

3) Analyze transfer functions:

$\underset{\approx}{U}$ ≡ orthonormal basis set of output vectors

$\underset{\approx}{V}$ = orthonormal basis set of input vectors

$\underset{\approx}{\Sigma}$ = amplification factors

4) Compress images:

outer product yields matrix

singular values

$$\underset{\approx}{A} = \operatorname*{sum}_{i} \left\{ \underset{\sim}{u}_i \, \underset{\sim}{v}_i^T \, \sigma_{ii} \right\}$$

col. of $\underset{\approx}{U}$ col of $\underset{\approx}{V}$

– computationally expensive, but good for very large $\underset{\approx}{A}$

5) Face Detection:

The columns of U are an ordered (by importance) set of eigenfaces

6). Principal Component Analysis:
By examining the <u>projection</u> of a data set on $\underset{\approx}{U}$, can distinguish pop.