linear algebra

Saad Jbabdi

- Matrices & GLM
- Eigenvectors/eigenvalues
- PCA

linear algebra

Saad Jbabdi

- Matrices & GLM
- Eigenvectors/eigenvalues
- PCA

The GLM

$$y = M * x$$

There is a linear relationship between M and y

find x?

$$\underbrace{\begin{bmatrix} 0.4613 \\ 0.8502 \\ -0.3777 \\ 0.5587 \\ 0.3956 \\ -0.0923 \end{bmatrix}}_{y} \quad \underbrace{\begin{bmatrix} 1.0000 & 0.5377 \\ 1.0000 & 1.8339 \\ 1.0000 & -2.2588 \\ 1.0000 & 0.8622 \\ 1.0000 & 0.3188 \\ 1.0000 & -1.3077 \end{bmatrix}}_{M} \quad \underbrace{\begin{bmatrix} ? \\ ? \end{bmatrix}}_{x?}$$

Simultaneous equations

$$1.0000\, x_1 + 0.5377\, x_2 = 0.4613$$
$$1.0000\, x_1 + 1.8339\, x_2 = 0.8502$$
$$1.0000\, x_1 + -2.2588\, x_2 = -0.3777$$
$$1.0000\, x_1 + 0.8622\, x_2 = 0.5587$$
$$1.0000\, x_1 + 0.3188\, x_2 = 0.3956$$
$$1.0000\, x_1 + -1.3077\, x_2 = -0.0923$$

Examples

$$y = M*x$$

y	FMRI Time series from one voxel	some measure across subjects from one voxel	Behavioural scores across subjects
M	"regressors" (e.g.: the task)	"regressors" (e.g.: group membership)	Age, #Years at school
x	PEs (parameter estimates)	PEs (parameter estimates)	PEs (parameter estimates)

The GLM

$$y = M*x$$

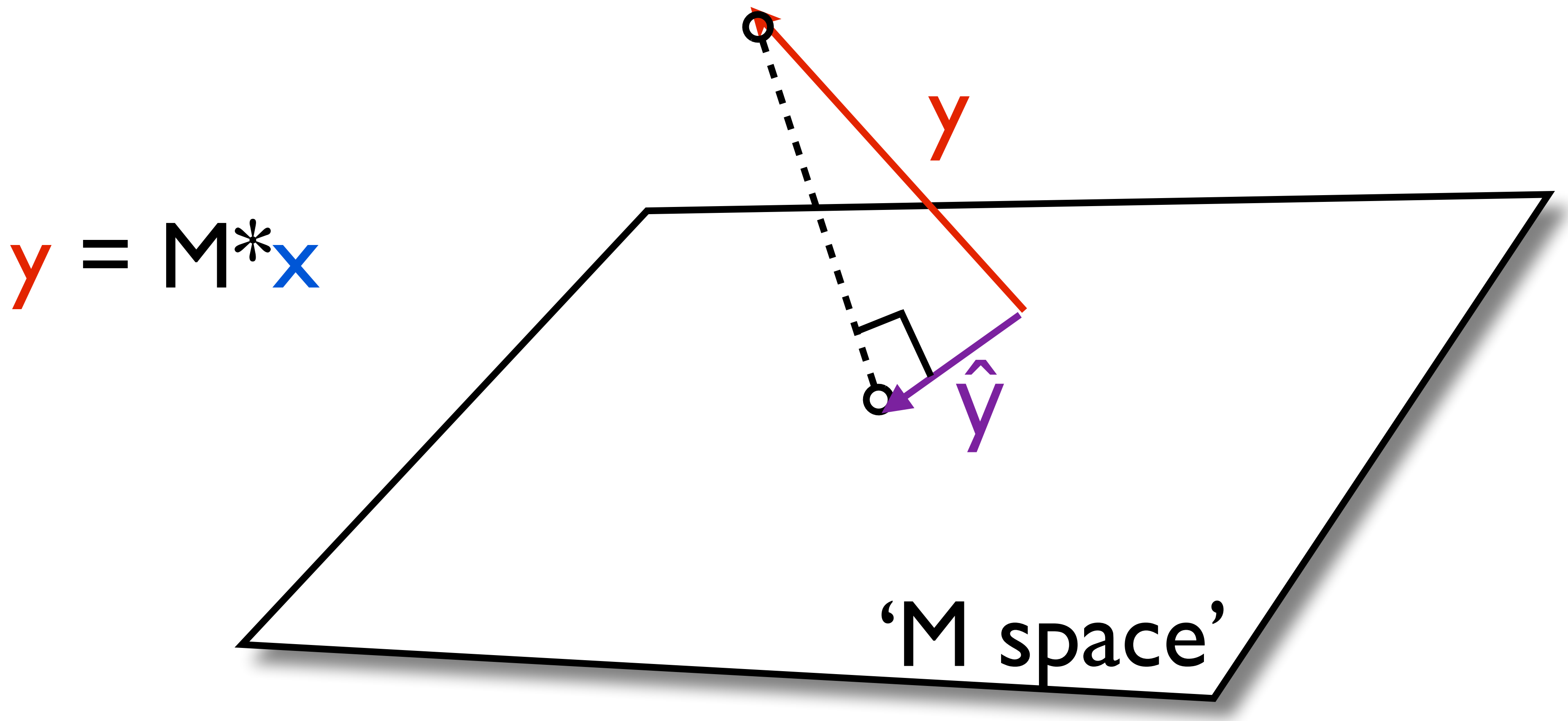
There is a linear relationship between M and y

find x?

solution : $x = \text{pinv}(M)*y$

(the actual matlab command)

what is the pseudo-inverse pinv ?



Must find the best (x,ŷ) such that ŷ =M*x (we can't get out of **M** space)

ŷ is the projection of y onto the 'M space'

x are the coordinates of ŷ in the 'M space'

pinv(M) *is used to project* y *onto the* 'M space'

This section is about the 'M space'

In order to understand the ‘M space’, we need to talk about these concepts:

- vectors, matrices
- dimension, independence
- sub-space, rank

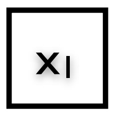
definitions

- Vectors and matrices are *finite* collections of "*numbers*"
- Vectors are columns of numbers
- Matrices are rectangles/squares of numbers

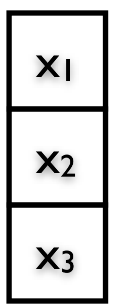
x_1
x_2
x_3
x_4
x_5

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}
x_{41}	x_{42}	x_{43}
x_{51}	x_{52}	x_{53}

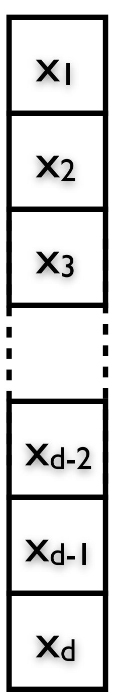
vectors



vector in a 1-dimensional space



vector in a 3-dimensional space

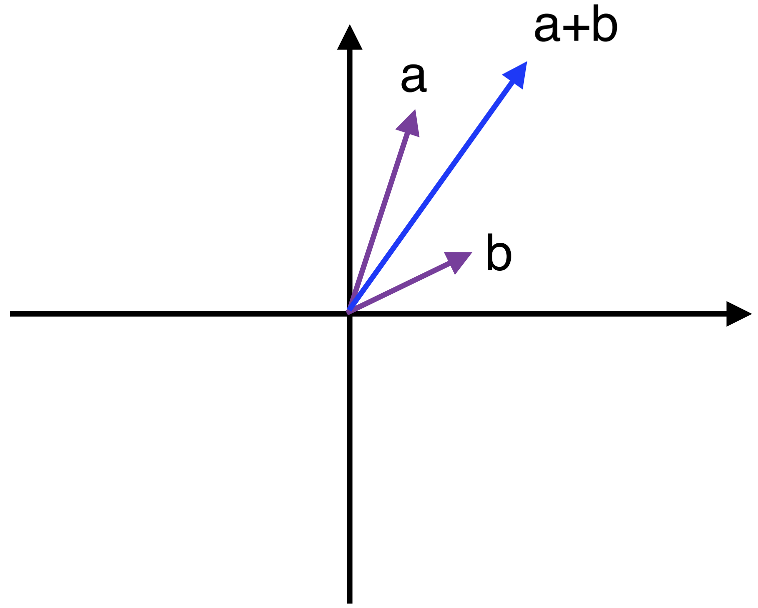


vector in d-dimensional space

vectors

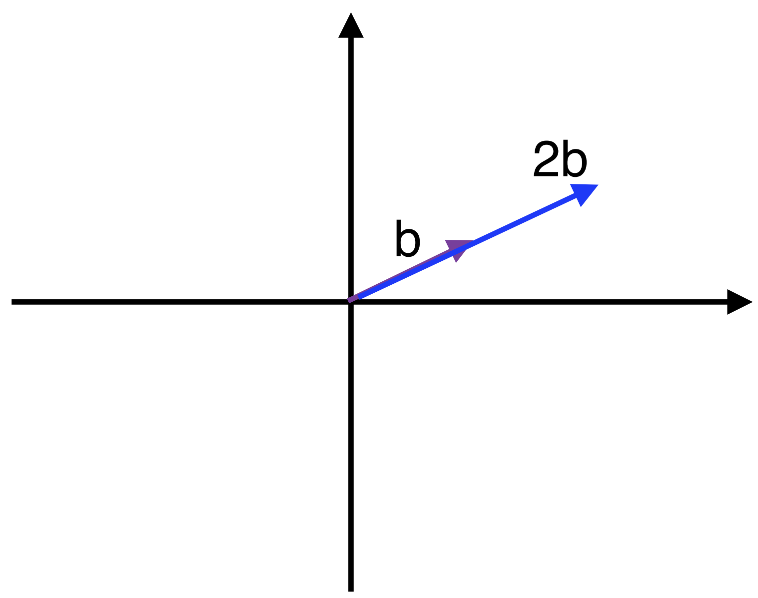
- Adding vectors
  - add element-wise

$$\mathbf{a}+\mathbf{b}=\begin{bmatrix}1\\3\end{bmatrix}+\begin{bmatrix}2\\1\end{bmatrix}=\begin{bmatrix}1+2\\3+1\end{bmatrix}=\begin{bmatrix}3\\4\end{bmatrix}$$



- Scaling of vectors
  - multiply element-wise

$$c\mathbf{b}=2\times\begin{bmatrix}2\\1\end{bmatrix}=\begin{bmatrix}2\times2\\2\times1\end{bmatrix}=\begin{bmatrix}4\\2\end{bmatrix}$$



- Linear combinations of vectors

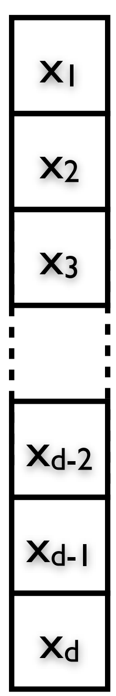
$\mathbf{c} = g.\mathbf{a}+h.\mathbf{b}$

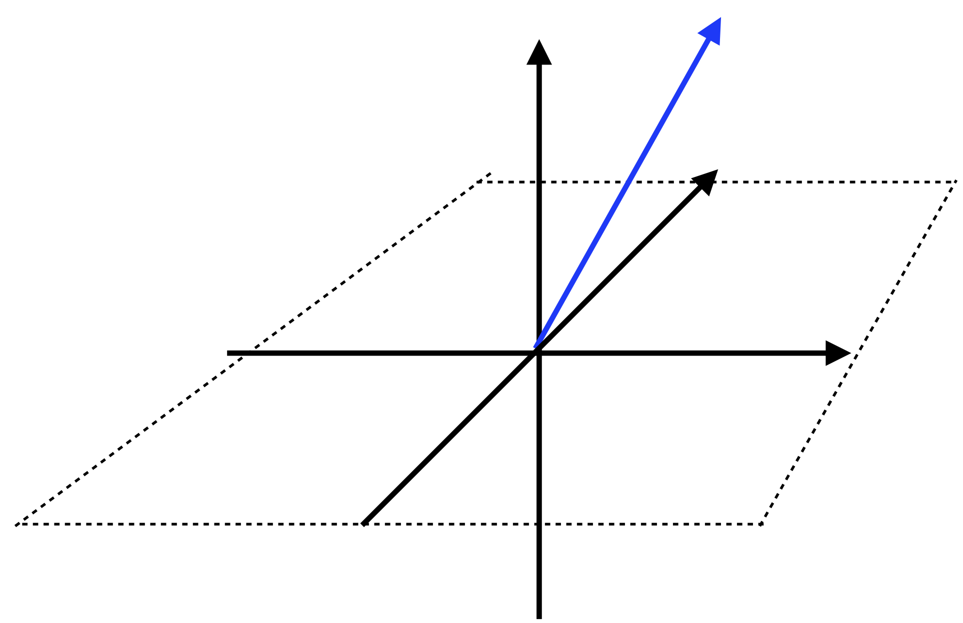
a, **b** and **c** in the same d-dimensional space

vectors

About d-dimensional vectors

The “arrow” picture is also useful in d-dimensions, as any vector is in effect one-dimensional.

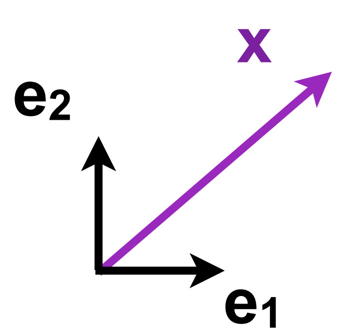


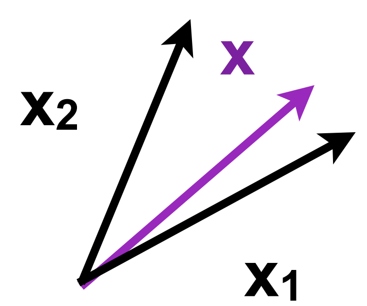


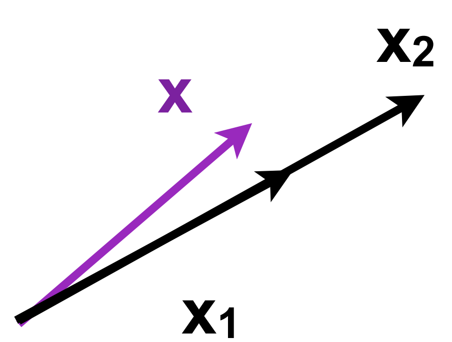
vectors

- Linear combinations of vector

$$\mathbf{c} = g.\mathbf{a}+h.\mathbf{b}$$







$\mathbf{x}=a^*\mathbf{e}_1+b^*\mathbf{e}_2$

any 2D vector is a linear combination of $\mathbf{e}_1$ and $\mathbf{e}_2$

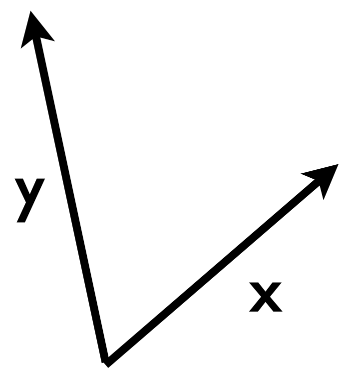
$\mathbf{x}=a^*\mathbf{x}_1+b^*\mathbf{x}_2$?

what about a linear combination of any 2 vectors?

$\mathbf{x}=a^*\mathbf{x}_1+b^*\mathbf{x}_2$?

what if the two vectors are co-linear?

"spanning"



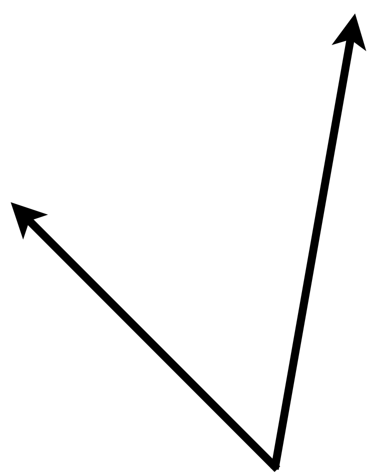
spanning means covering using linear combinations

E.g.: $\mathbf{z} = a * \mathbf{x} + b * \mathbf{y}$

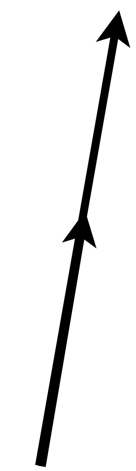
space covered by **z** <u>for all *a* and *b*</u> is the space that **x** and **y** <u>span</u>

→ **x** and **y** span a 2D space

“spanning”



these two vectors span 2 dimensions
can they span 3?

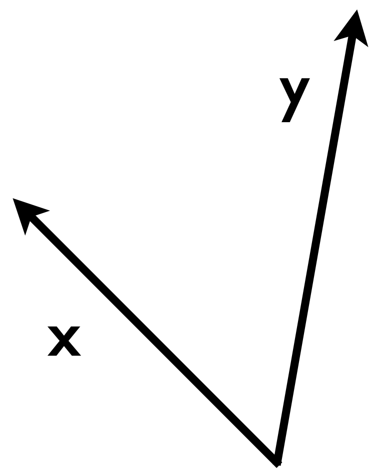


these two vectors
span 1 dimension

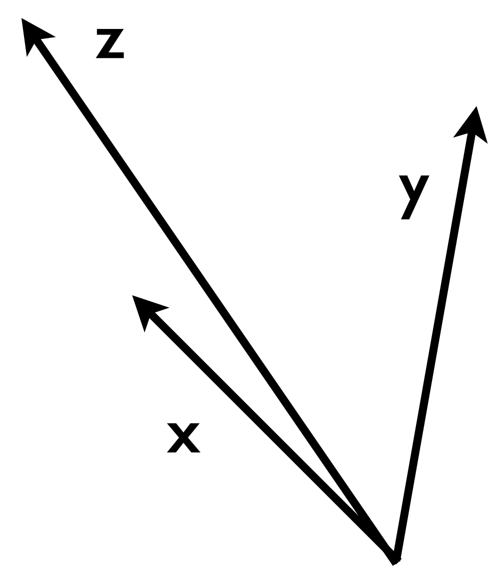
vectors can span a “sub-space”

dimensions of the sub-space relates to “linear independence”

linear independence



can we write $\mathbf{x}=a.\mathbf{y}$?



can we write $\mathbf{x}=a.\mathbf{y}+b\mathbf{z}$?

the vectors $\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3 \ldots \mathbf{x}_n$ are linearly independent if none of them is a linear combination of the others

In higher dimensions

$$x_1 = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 2 \\ 2 \end{bmatrix} \quad x_2 = \begin{bmatrix} 2 \\ 6 \\ 10 \\ 14 \\ 4 \\ 4 \end{bmatrix}$$

these two vectors
are *not* linearly independent
($x_2=2*x_1$)

what about these?
how many "linearly independent" vectors?

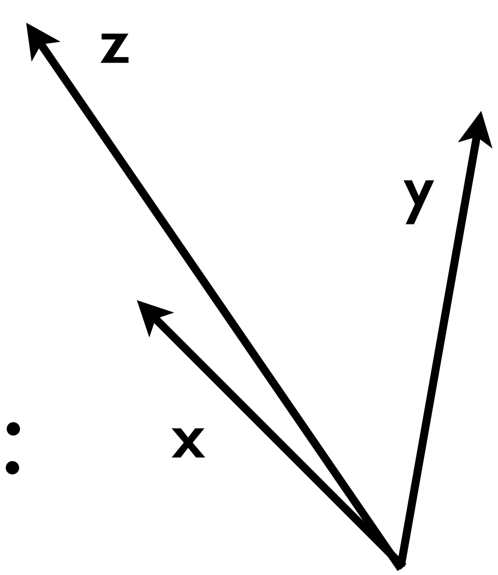
x_1	x_2	x_3	x_4	x_5	x_6
0.9298	1.1921	1.0205	-2.4863	0.0799	0.8577
0.2398	-1.6118	0.8617	0.5812	-0.9485	-0.6912
-0.6904	-0.0245	0.0012	-2.1924	0.4115	0.4494
-0.6516	-1.9488	-0.0708	-2.3193	0.6770	0.1006

hard to tell, but there can't be more than 4

Theorem

The number of independent vectors is smaller than the dimension of the space

2D example:



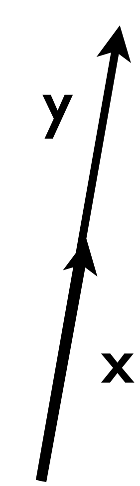
Theorem

Given a collection of vectors, the space of all linear combinations has dimension equal to the number of linearly independent vectors

This space is called a "sub-space"

2D example:

(dimension of sub-space spanned by {x,y} is 1)



Matrices

Matrices

- Matrices, what are they?
  - A matrix is a rectangular arrangement of values and is usually denoted by a **BOLD UPPER CASE** letter, e.g.

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}$$

Is an example of a 2-by-2 matrix

and

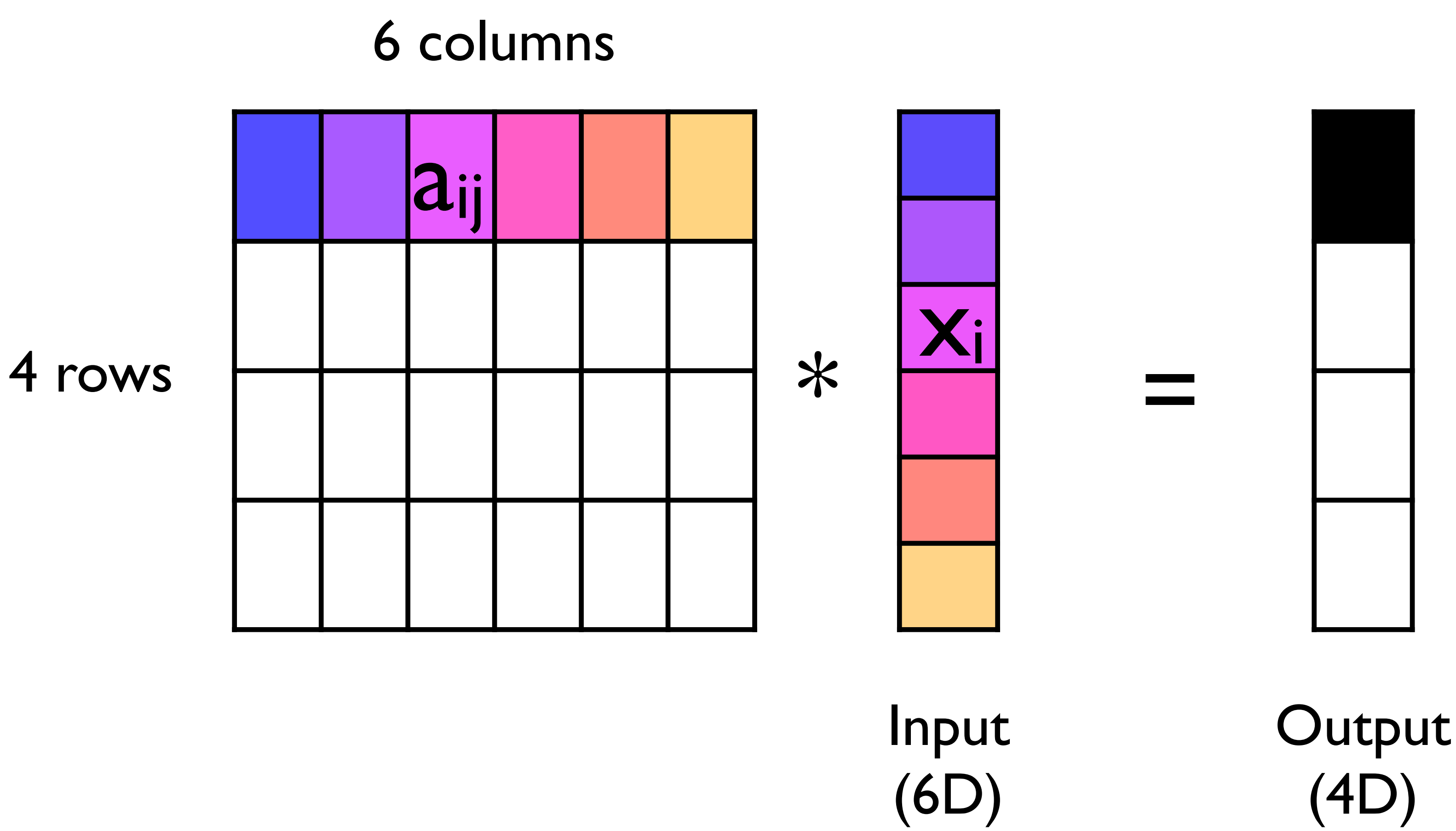
$$\mathbf{B} = \begin{bmatrix} 4 & 7 & 6 \\ 4 & 1 & 5 \end{bmatrix}$$

Is an example of a 2-by-3 matrix

Matrices

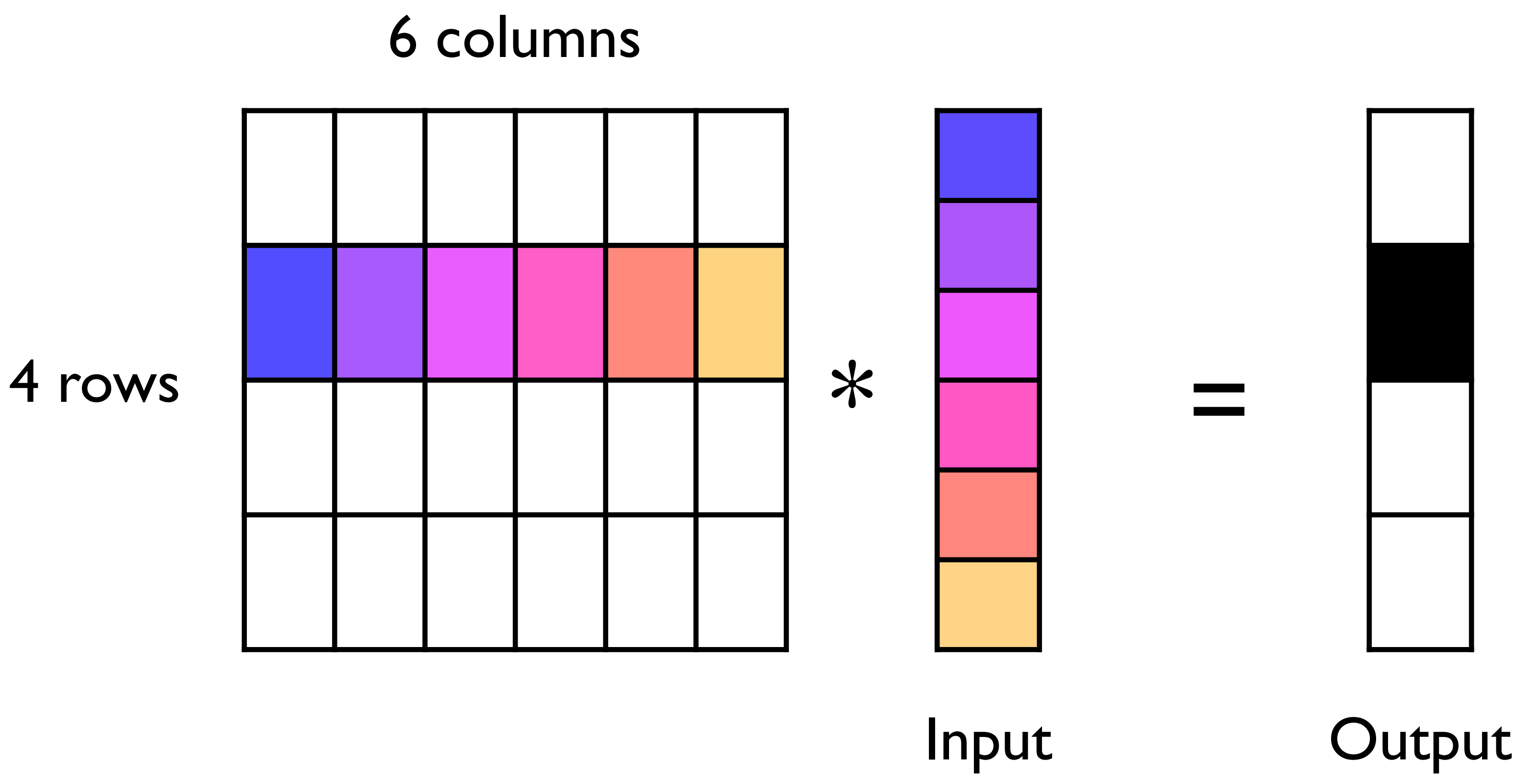
Multiplying a matrix by a vector

$6\text{dimensions} \longmapsto 4\text{dimensions}$



Matrices

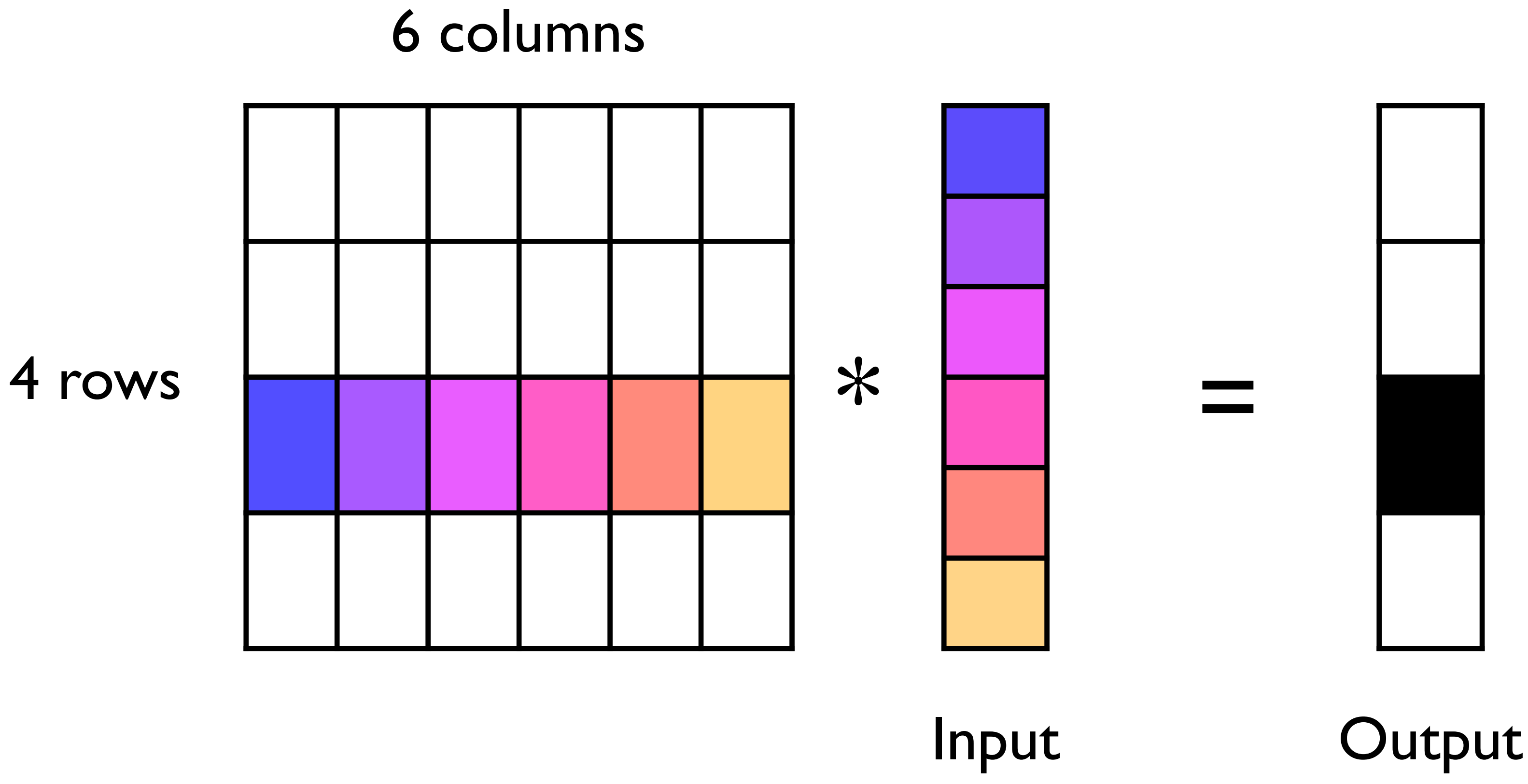
Multiplying a matrix by a vector
$6\text{dimensions} \longmapsto 4\text{dimensions}$



Matrices

Multiplying a matrix by a vector

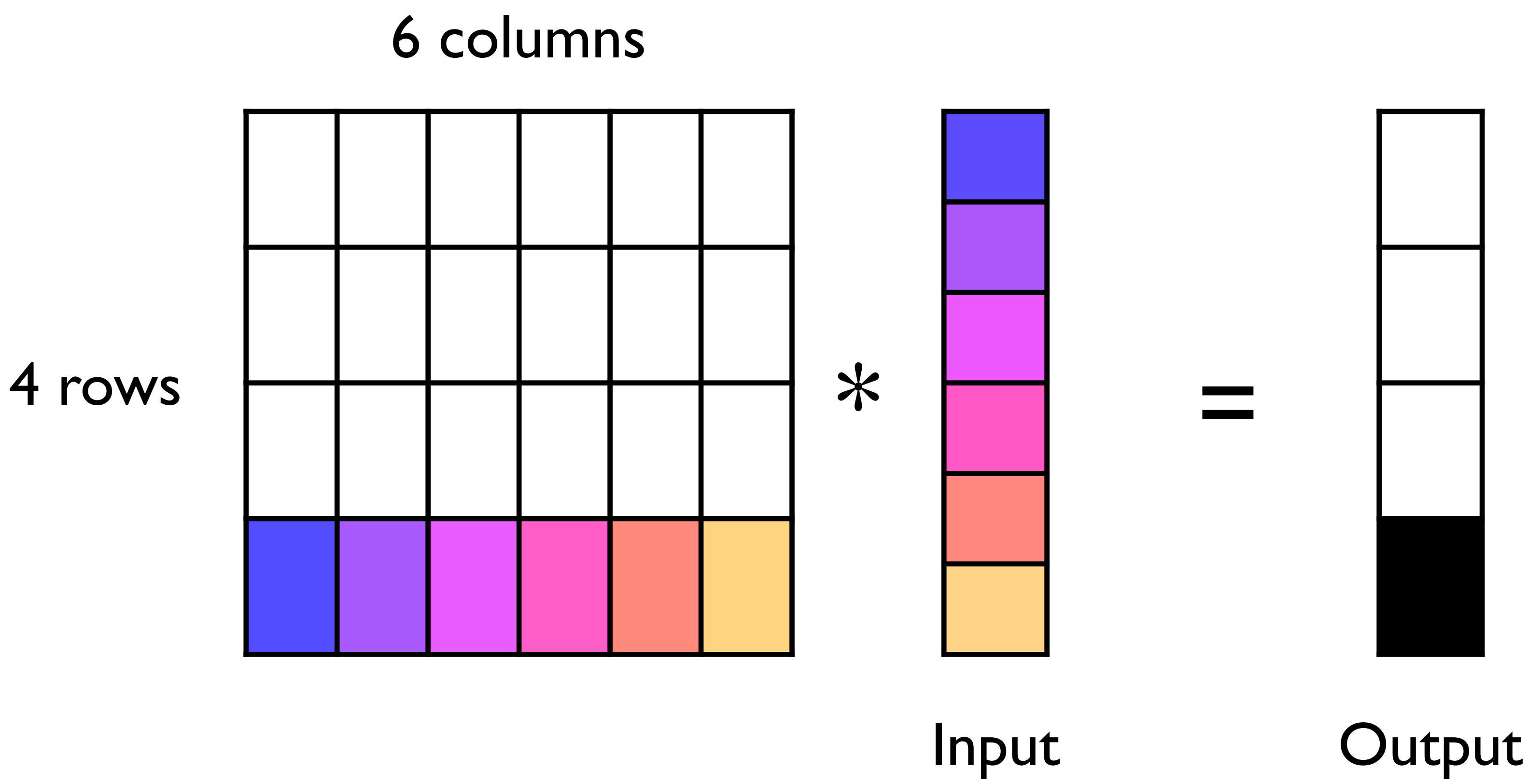
6dimensions $\longmapsto$ 4dimensions



Matrices

Multiplying a matrix by a vector

$$6\text{dimensions} \longmapsto 4\text{dimensions}$$



definitions

- Matrix multiplication as linear combinations of vectors
  - Let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = [\mathbf{a}_1 \quad \mathbf{a}_2] \text{ and } \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

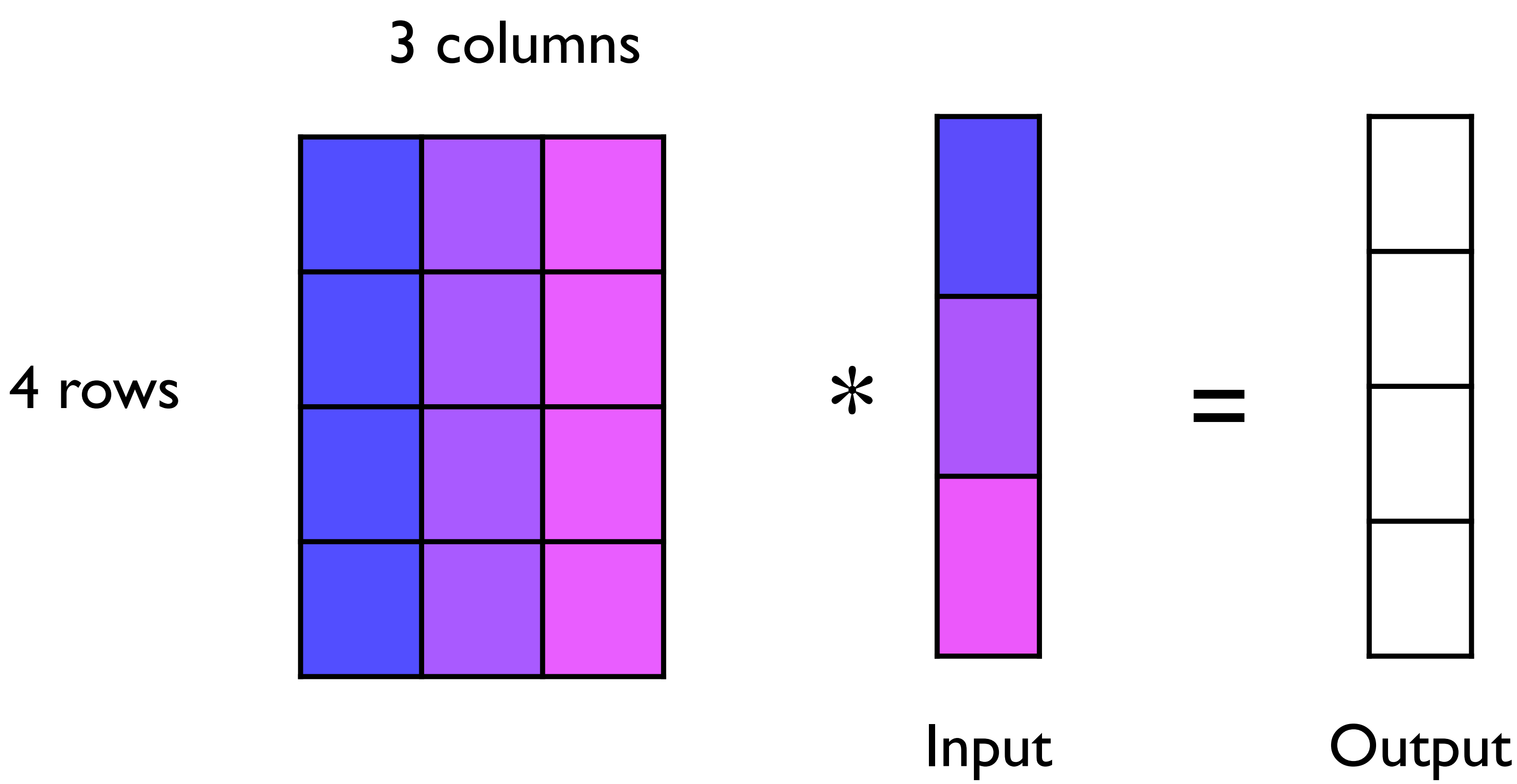
  - then

i.e. the vector **Ab** is a linear combination of the vectors constituting the columns of **A**, i.e. it lies in the “column space” of **A**.

$$\mathbf{Ab} = b_1\mathbf{a}_1 + b_2\mathbf{a}_2 = b_1\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} + b_2\begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix}$$

what does this imply?

The output is a linear combination of the <u>columns</u>
The output sub-space is the space spanned by the <u>columns</u>
The <u>dimension</u> of the output sub-space is smaller or equal to the number of <u>columns</u>

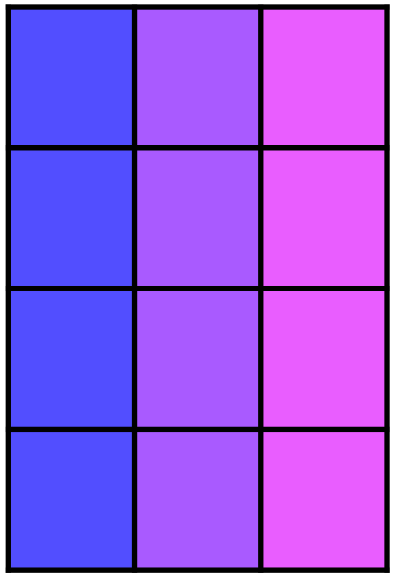


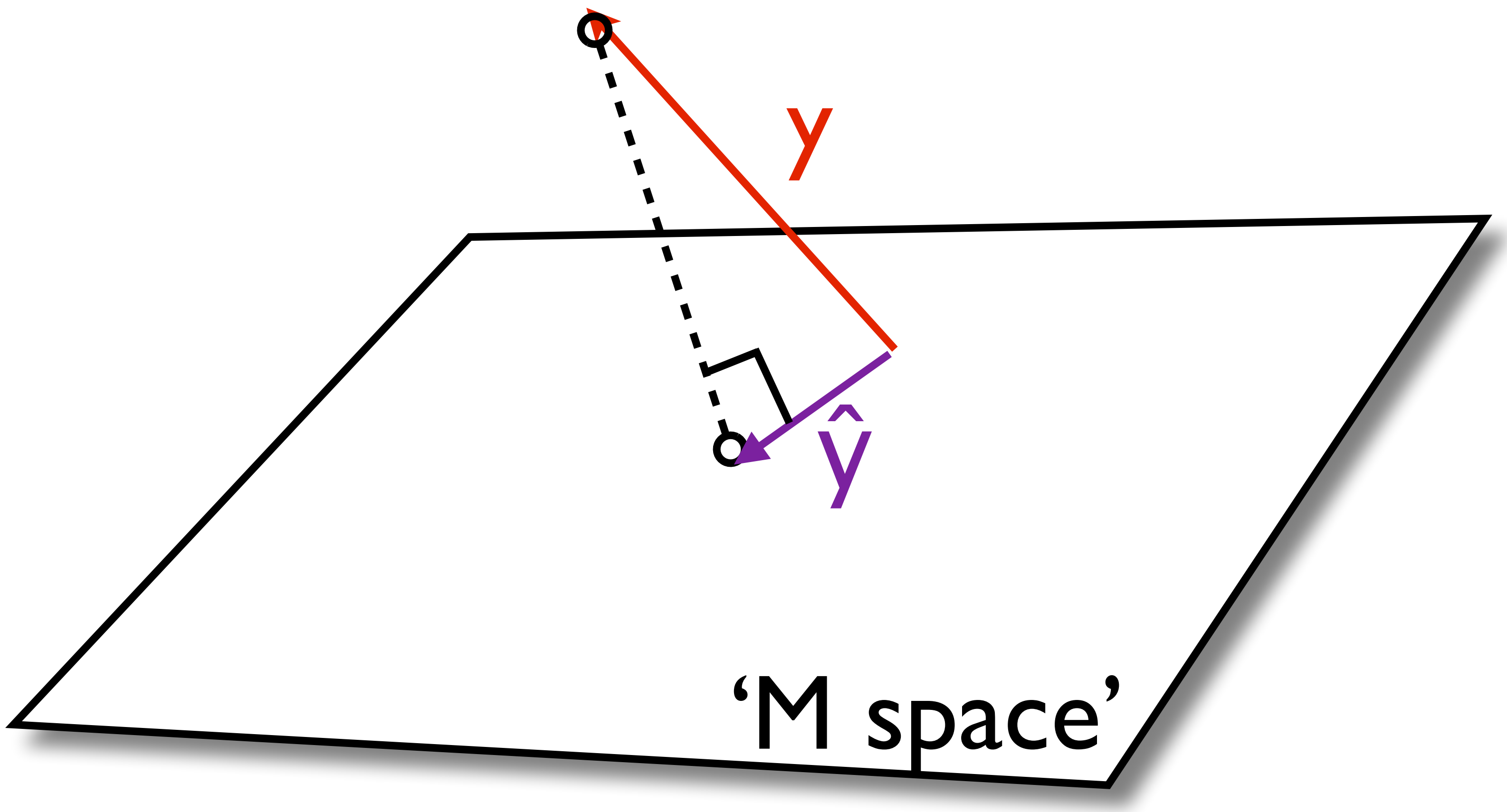
rank

The rank of a matrix is the number of independent columns

Full rank means equal to the maximum possible

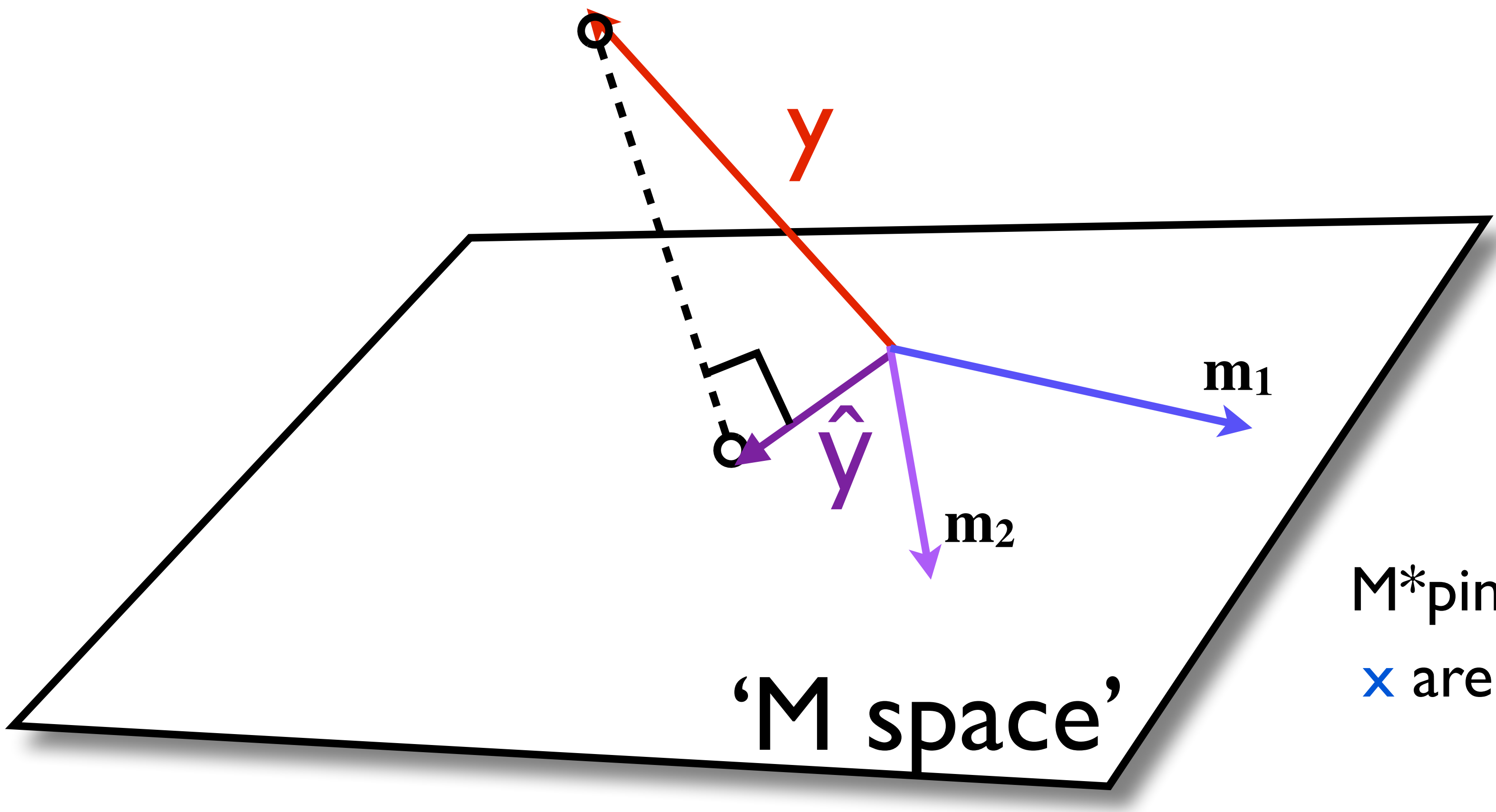
Otherwise it is said to be rank deficient





M*pinv(M) *is the projector on the* 'M space'

x are the coordinates of ŷ in the 'M space'

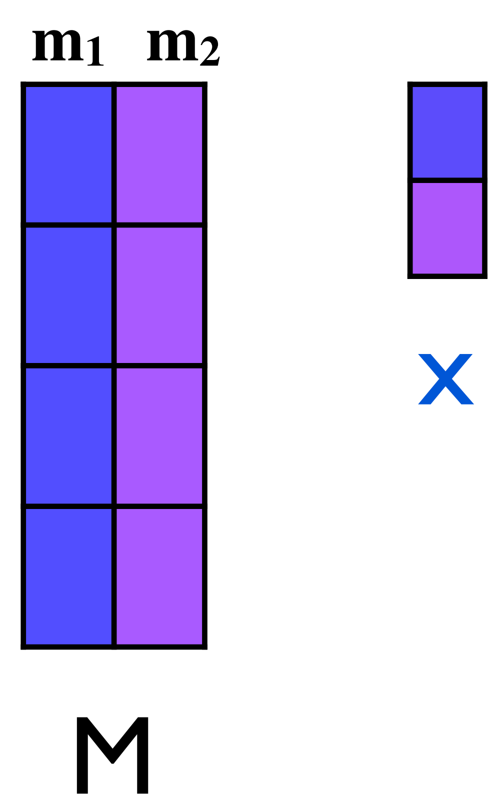


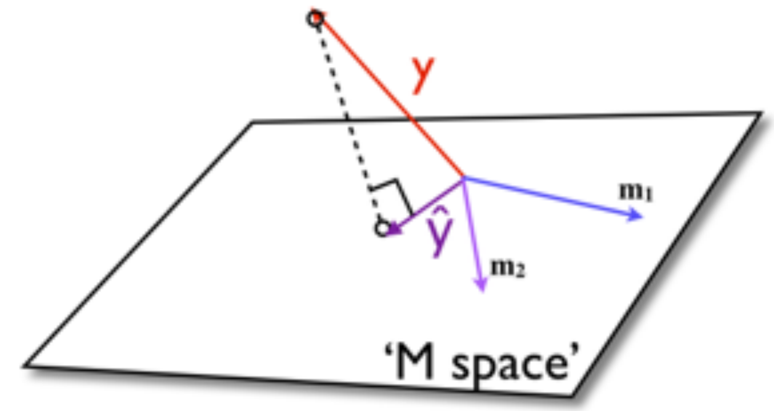
M*pinv(M) is the projector on the 'M space'

x are the coordinates of $\hat{y}$ in the 'M space'

$$x = pinv(M) * y$$

$$\hat{y} = M*x = \underbrace{M*pinv(M)}_{\text{projector}}*y$$





- x are the coordinates of ŷ in the space spanned by the columns of M
- x tells us "how much" of each column we need to approximate y
- the best approximation we can get is the projection onto the 'M space'
- we cannot get closer (out of 'M space') because that is what the columns of M span
- But if y is already in M-space, we get a perfect fit

End of part one

- The columns of the design matrix span the space "available" from the regressors (M-space)
- The pseudo-inverse finds the best vector of the M-space

- Next: Eigen-values/Eigen-vectors

Part two.

- Eigenvectors and eigenvalues
- PCA

$$y = Mx$$

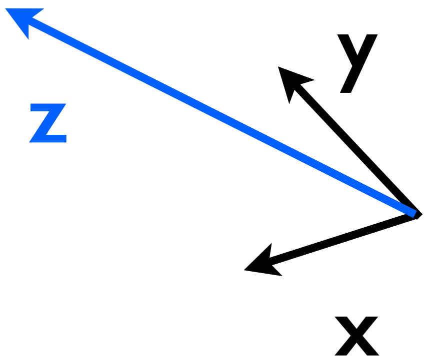
output
linear combination of columns of M

input
coefficients of the linear combination

$$\begin{bmatrix} 0.4613 \\ 0.8502 \\ -0.3777 \\ 0.5587 \\ 0.3956 \\ -0.0923 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.5377 \\ 1.0000 & 1.8339 \\ 1.0000 & -2.2588 \\ 1.0000 & 0.8622 \\ 1.0000 & 0.3188 \\ 1.0000 & -1.3077 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}$$

y M x

Special vectors

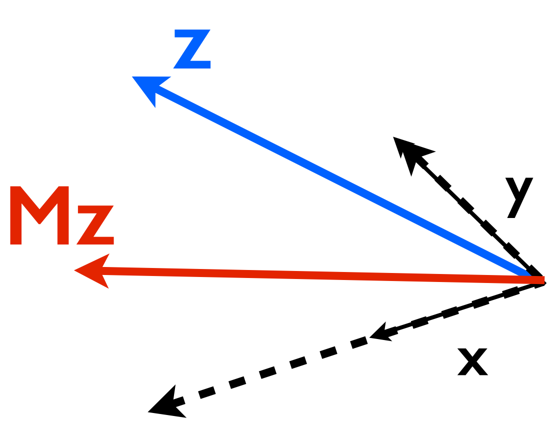


z = x+2*y

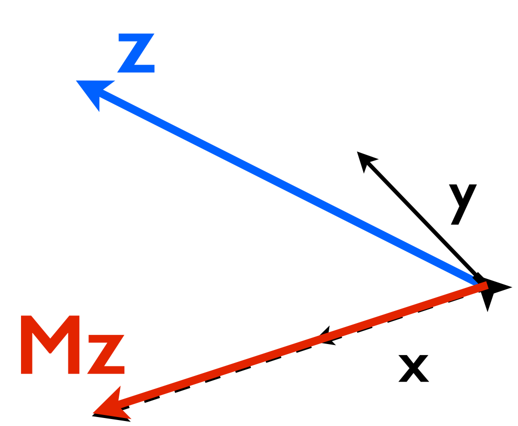
Mz = Mx+2*My

If x and y are such that: Mx=ax
My=by

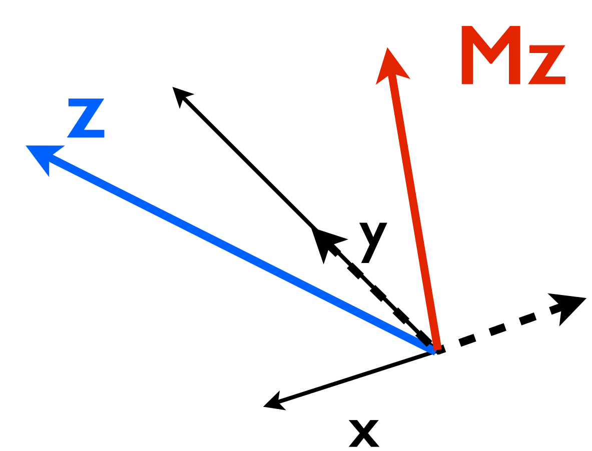
Then: Mz = ax+2*by



a=2, b=1



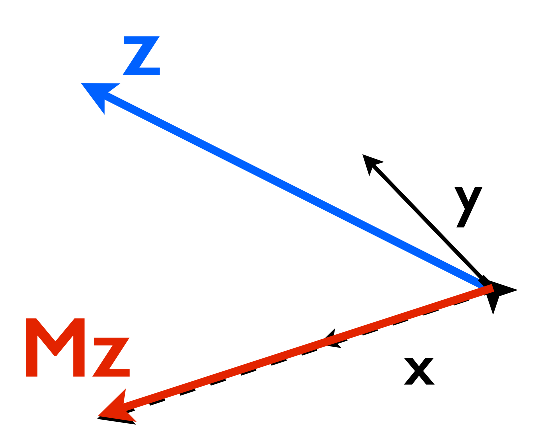
a=2, b=0



a=-1, b=2

Special vectors

- (**x**,a) and (**y**,b) are "intrinsic properties" of **M** that tell us how to transform any vector
- Easy to see what happens if an eigenvalue dominates the others
- Intuition for why small eigenvalue means close to rank deficiency



a=2, b=0.0001

need huge input to create output along weaker eigenvectors

Mz = ax+2*by

Eigenvector

$$\mathbf{Mx} = \lambda\,\mathbf{x}$$

Eigenvalue

Special vectors

- $\mathbf{Mx} = \lambda \mathbf{x}$ This means **M** is a square matrix

How do we find **x**?

Matlab: [V,D]=eig(**M**)

Eigenvectors

Eigenvalues

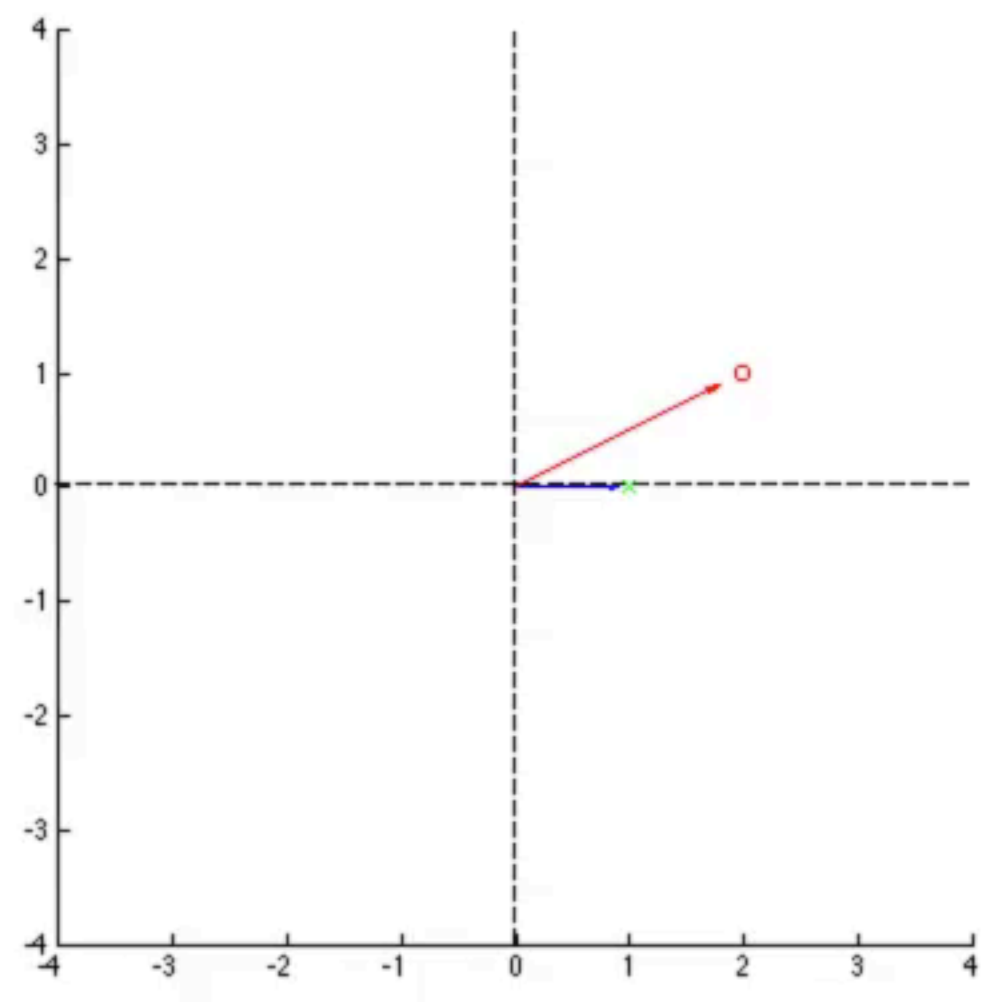
Examples in 2D

Positive definite matrix

$$M = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}$$

$\lambda_1 = 3$

$\lambda_2 = 2$



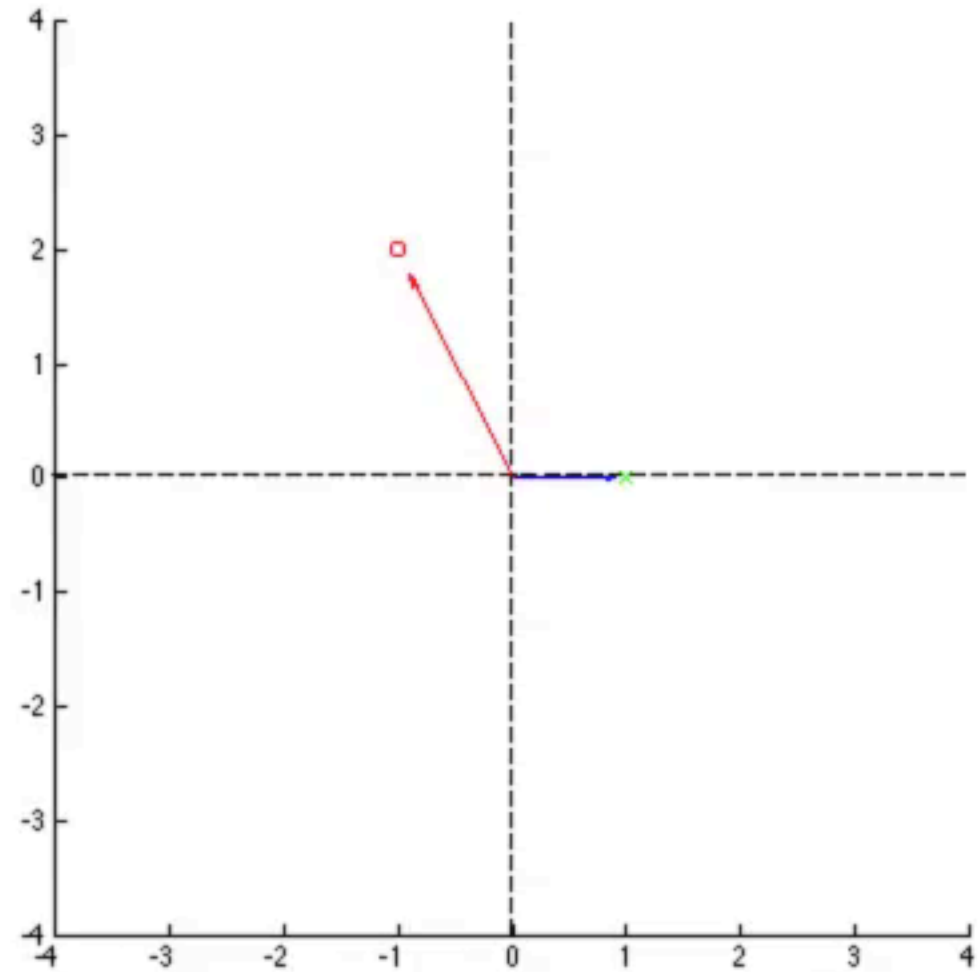
Examples in 2D

Negative eigenvalue

$$M = \begin{pmatrix} -1 & 1 \\ 2 & 3 \end{pmatrix}$$

$\lambda_1 = -1.4495$

$\lambda_2 = 3.4495$



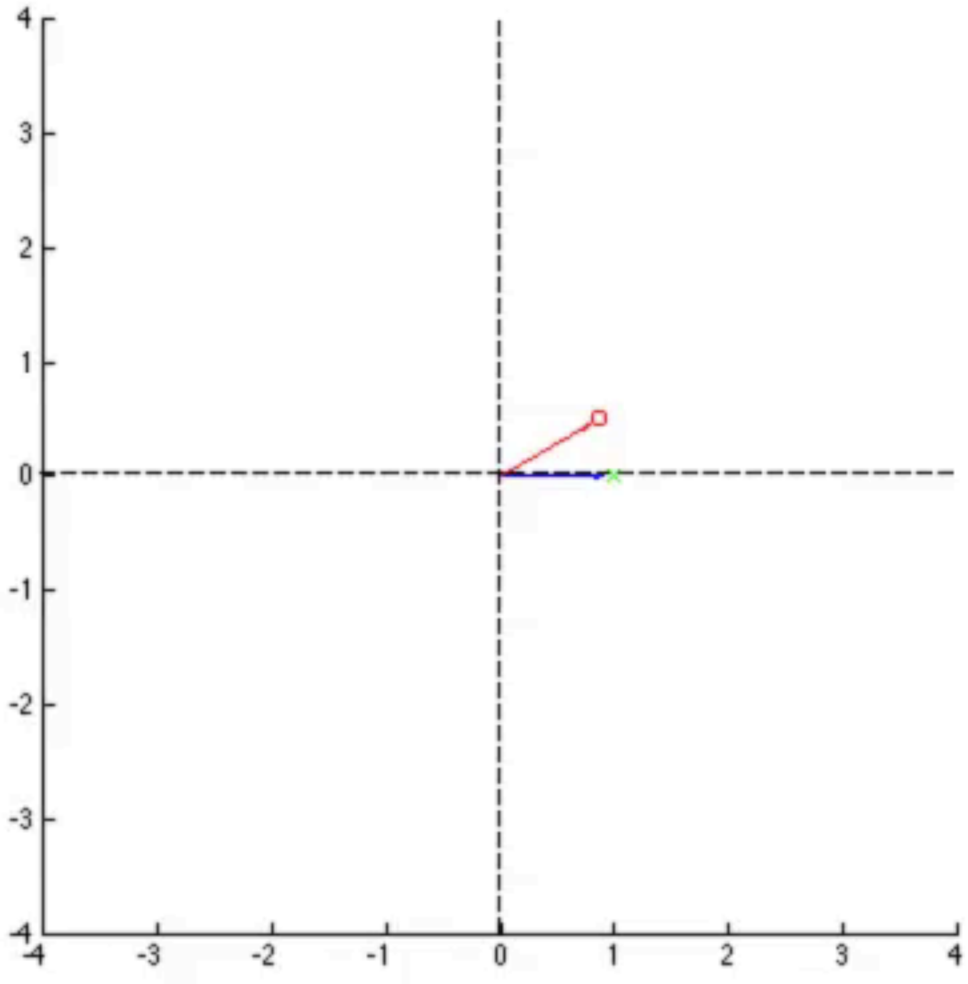
Examples in 2D

Rotation matrix

$$M = \begin{pmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{pmatrix}$$

λ_1 = ???

λ_2 = ???

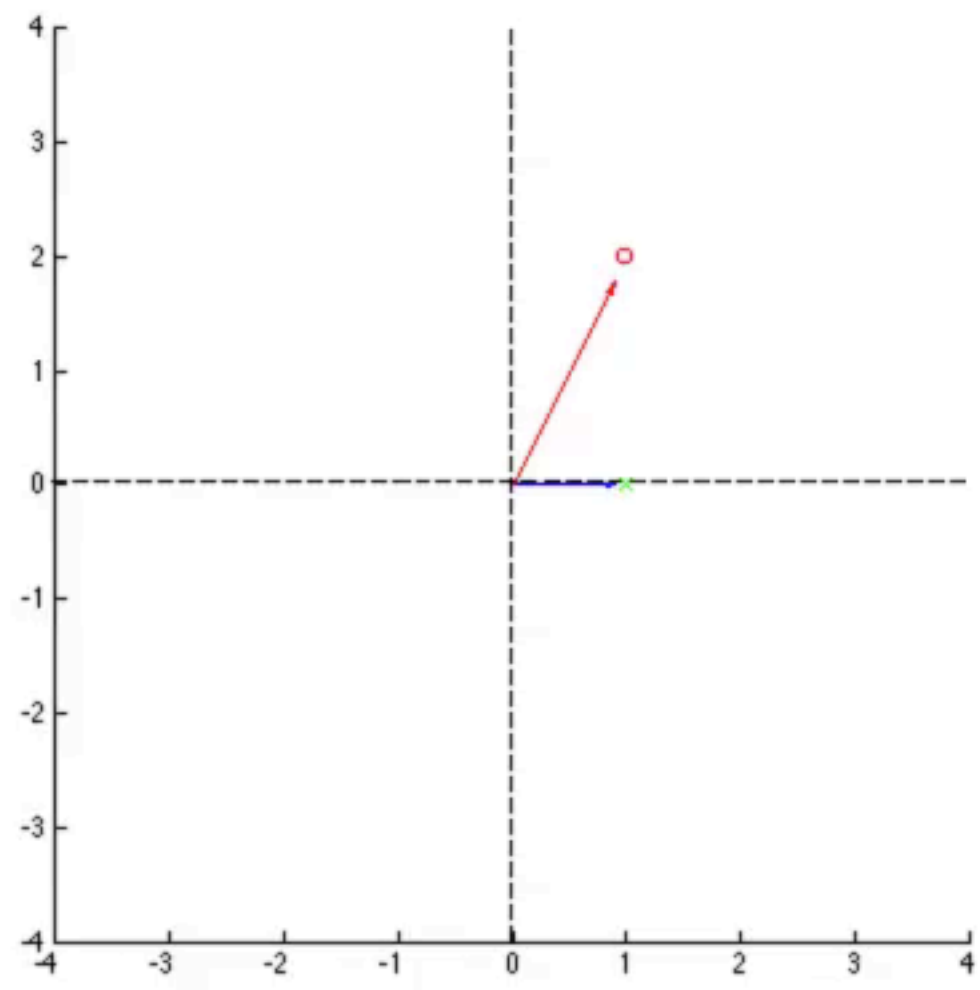


Examples in 2D

Rank deficient matrix

$$M = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

$\lambda_1 = 0$
$\lambda_2 = 5$

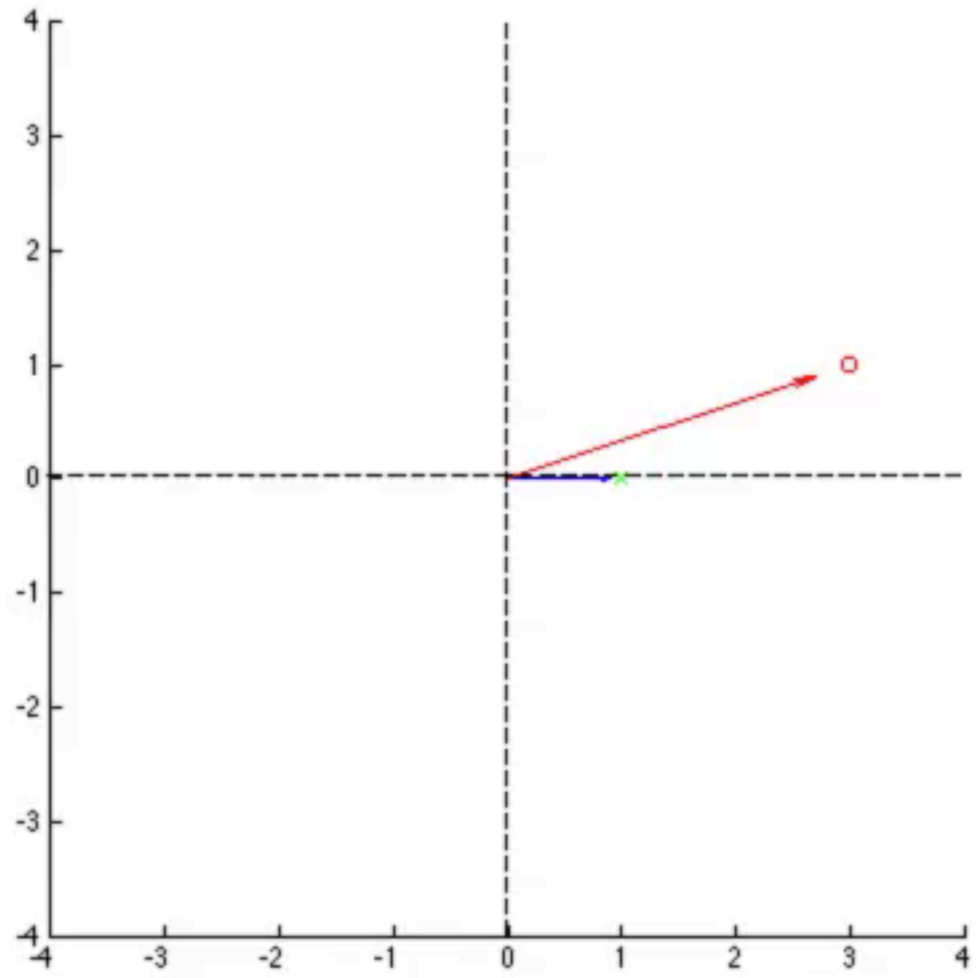


Examples in 2D

Symmetric matrix

$$M = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

$\lambda_1 = 2$
$\lambda_2 = 4$



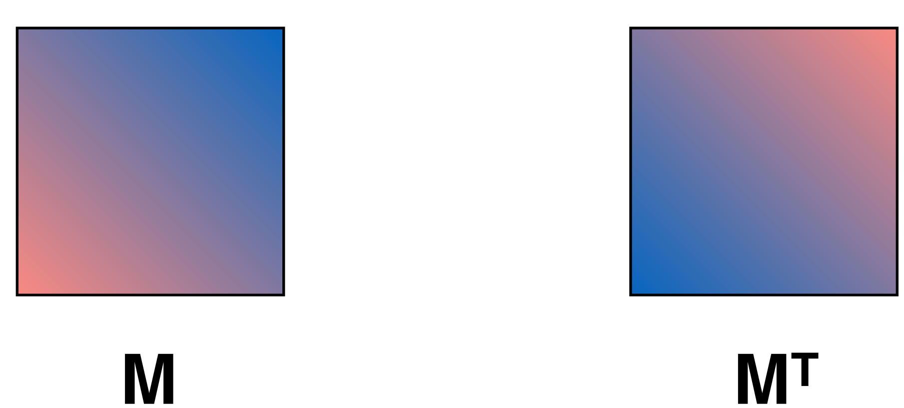
Symmetric matrices

examples

Covariance matrix

Correlation matrix

$\mathbf{M}\mathbf{M}^T$ and $\mathbf{M}^T\mathbf{M}$ for any rectangular matrix $\mathbf{M}$



Why is this interesting?

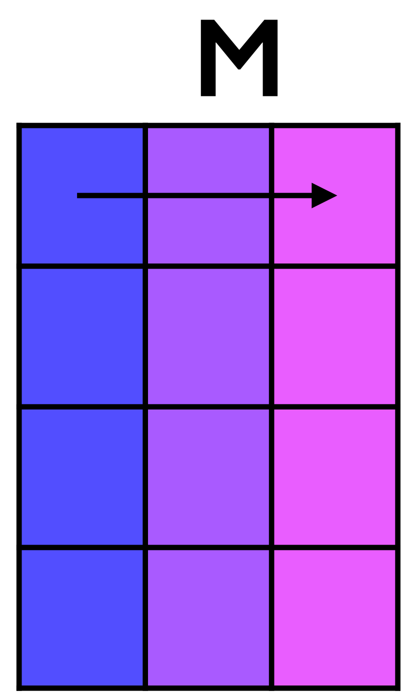
- We can generalise eigenvectors/values to rectangular matrices (by looking at $\mathbf{M}^T\mathbf{M}$ or $\mathbf{M}\mathbf{M}^T$)

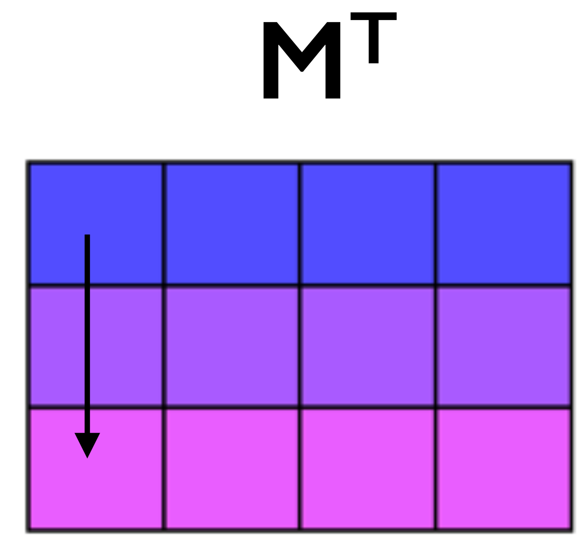
what is this for?

- Approximate rank
- Approximate matrix

Rank and transpose

Remember: the rank of a matrix is the dimension of the output sub-space





theorem

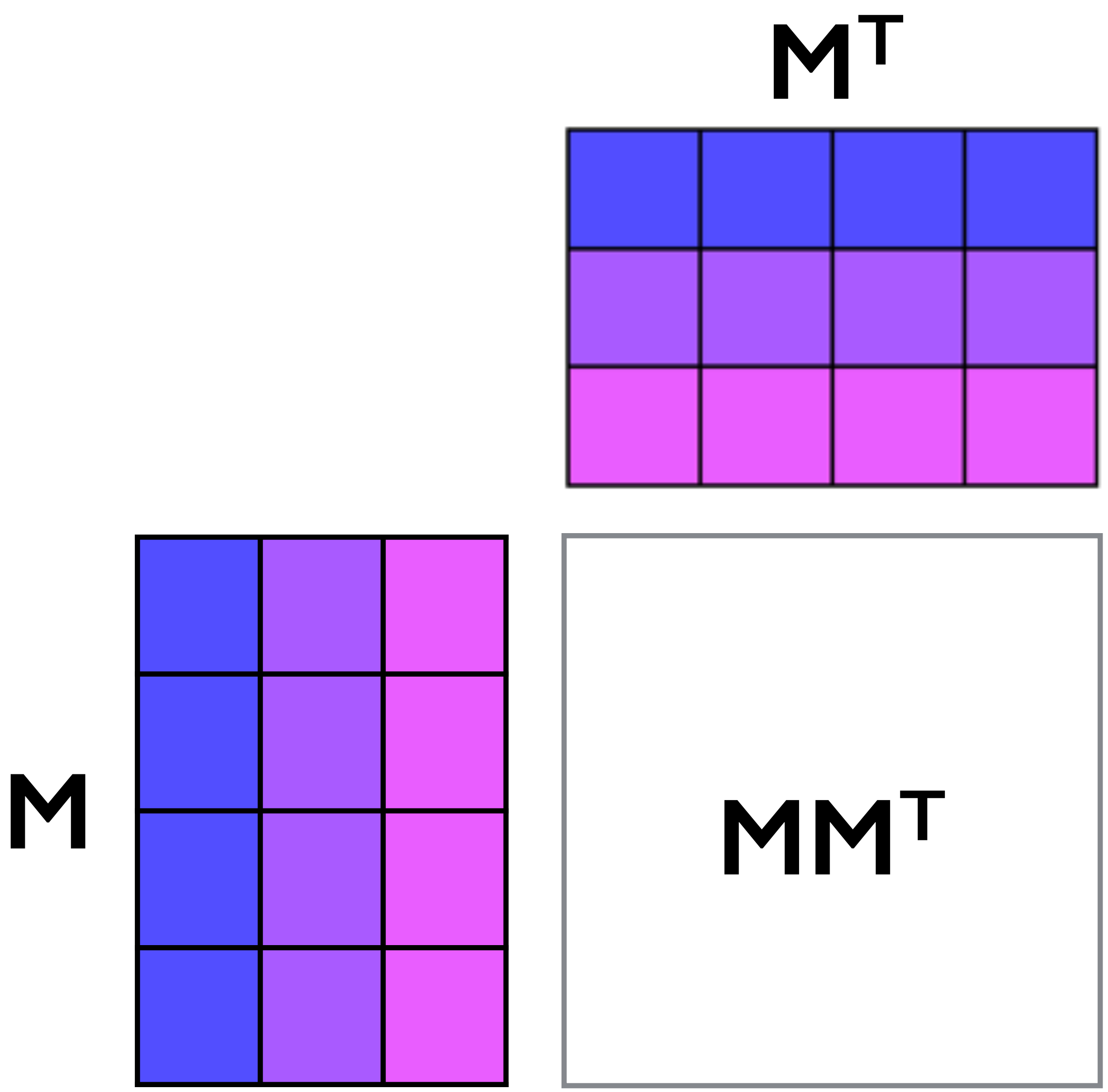
- rank(**M**) = rank(**M**^{T}) = rank(**MM**^{T}) = rank(**M**^{T}**M**)

Approximate the rank

I can calculate the eigenvalues of $\mathbf{MM}^T$ (always)

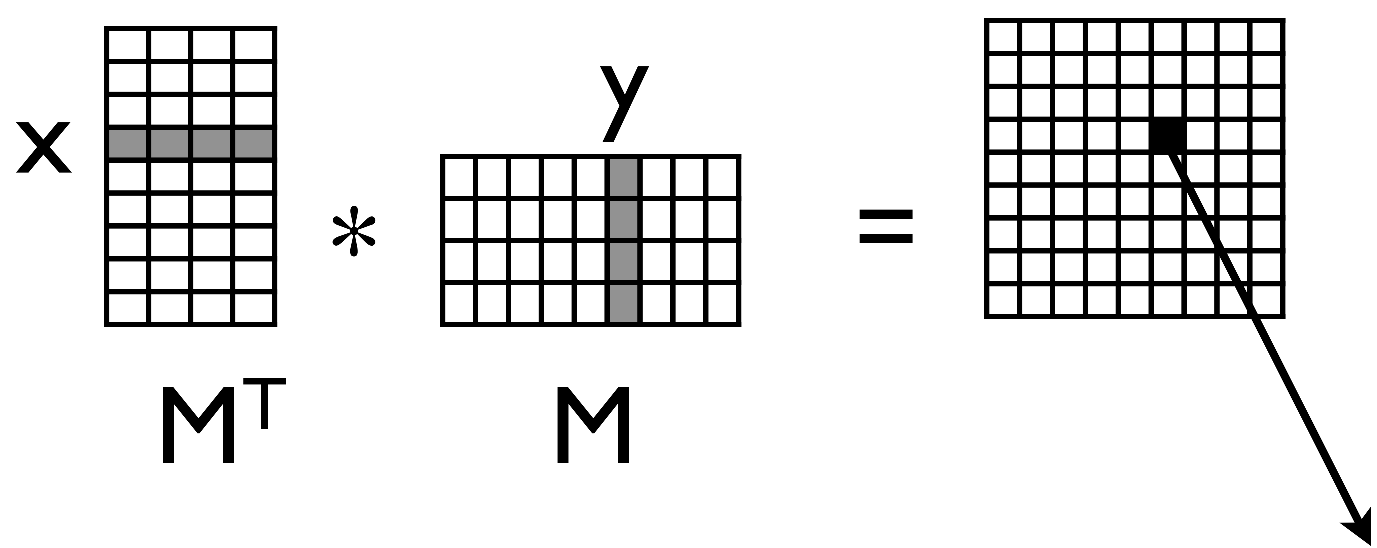
Let's say I find them to be 1.5, 2.0, 0.0001, 0

Then the approximate rank of M is 2



Approximate a matrix

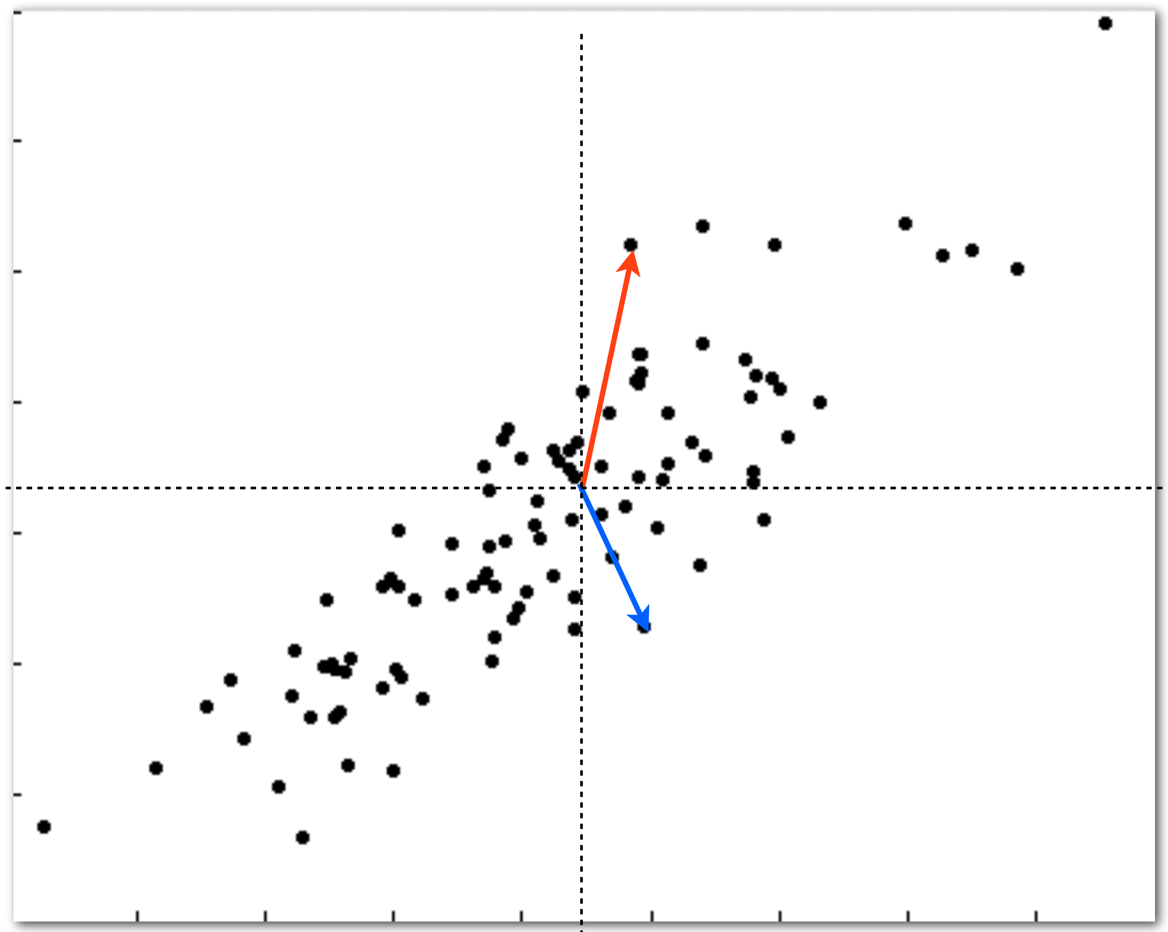
When the columns of M are demeaned, $M^T M$ is the covariance

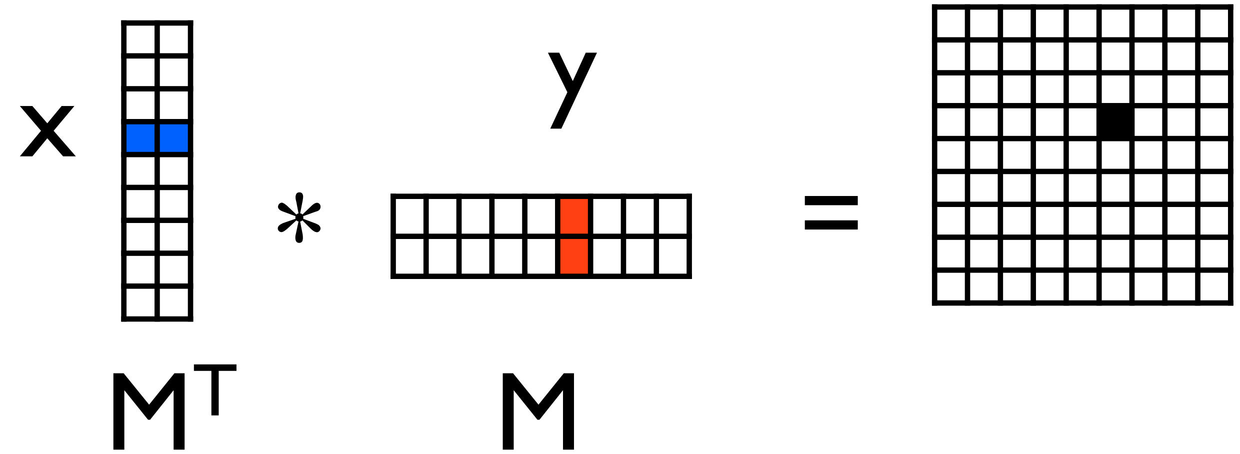


Sum{ (x_i-mean(x)) . (y_i-mean(y)) }

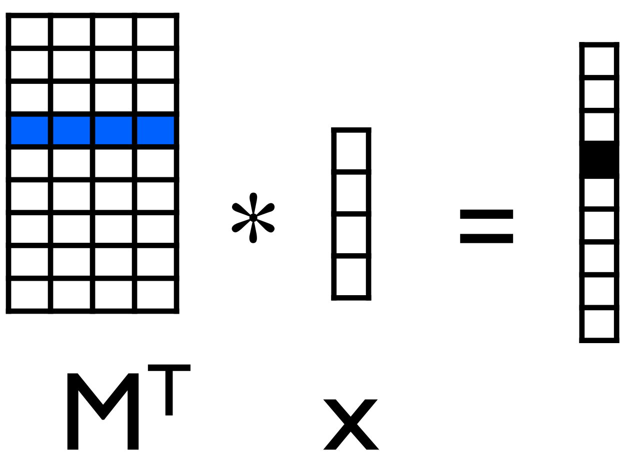
data

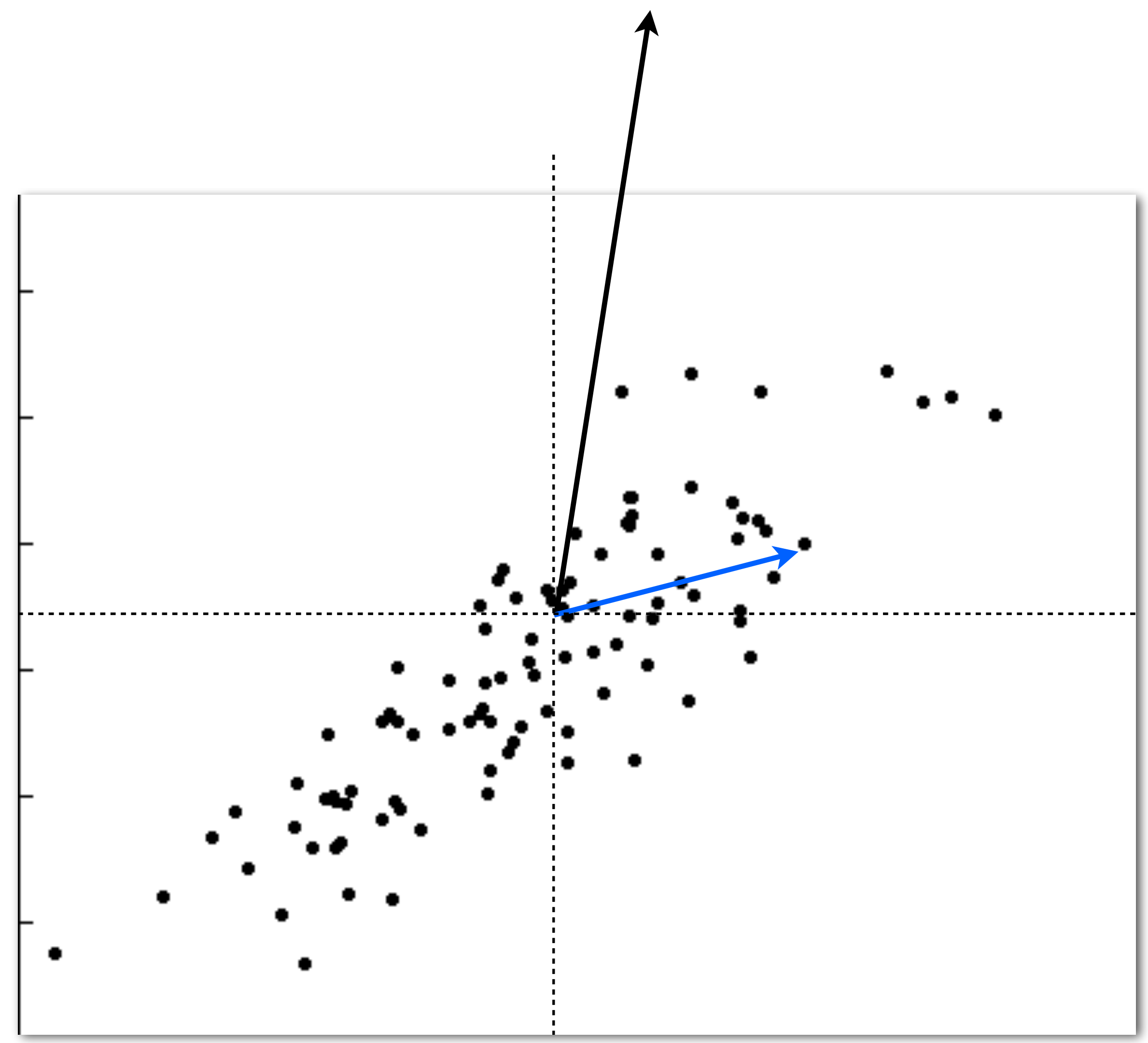
$$M = \begin{bmatrix} 1.3401 & 2.1599 & -0.4286 & 1.5453 & 1.2016 & 0.1729 & 0.7258 & 1.2167 & 3.2632 & 2.7515 & \\ 2.9208 & 3.0004 & 0.2012 & 2.3979 & 2.4349 & 0.5834 & 1.9231 & 2.7030 & 5.9159 & 4.1647 & \cdots \end{bmatrix}$$





Now what is $M^T x$?





We want an x that "looks like" most of the data points

i.e. <u>maximise</u> $|M^T x|$

(with $|x|=1$ for example, otherwise take $|x|=\text{infinity}!$)

Some maths

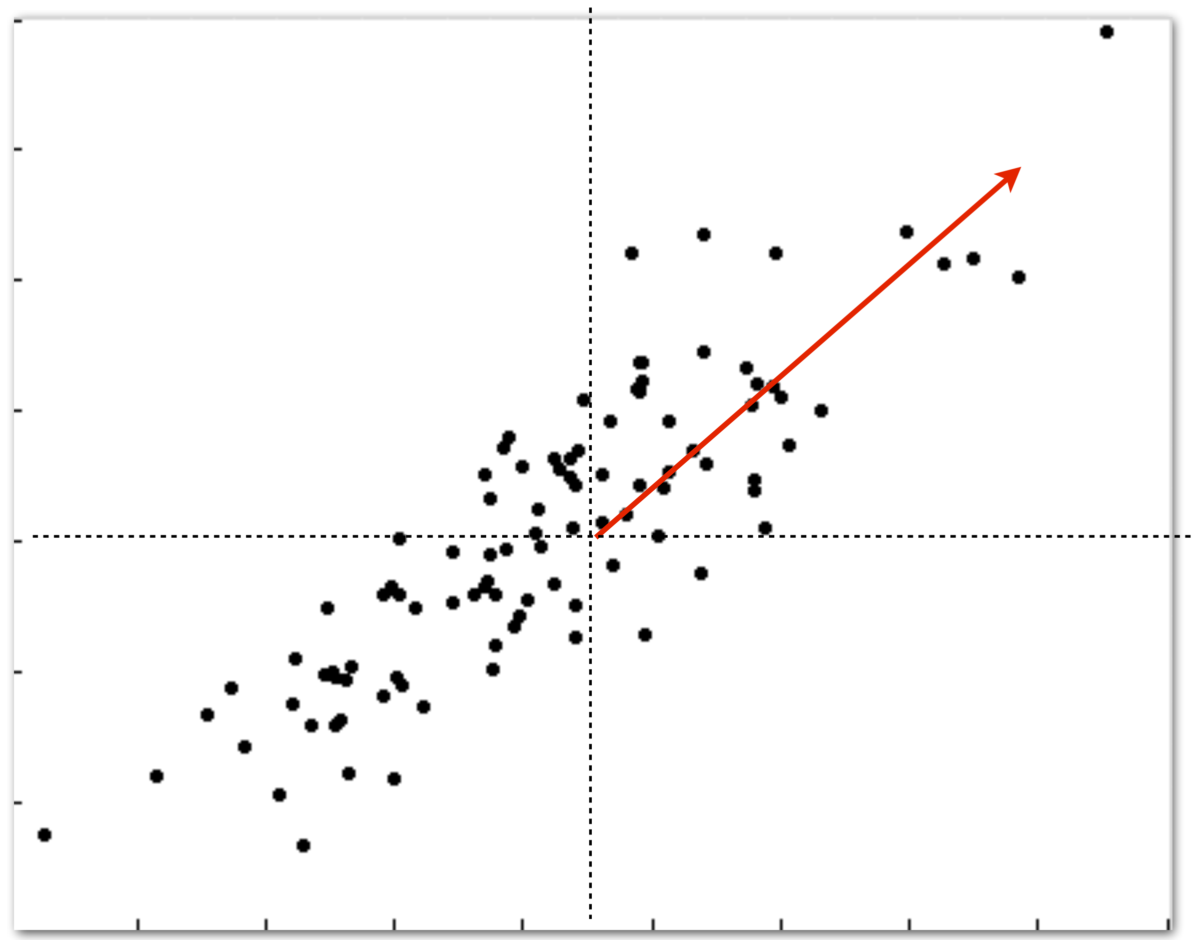
$$x^T M M^T x = (M^T x)^T (M^T x) = |M^T x|^2$$

max   max

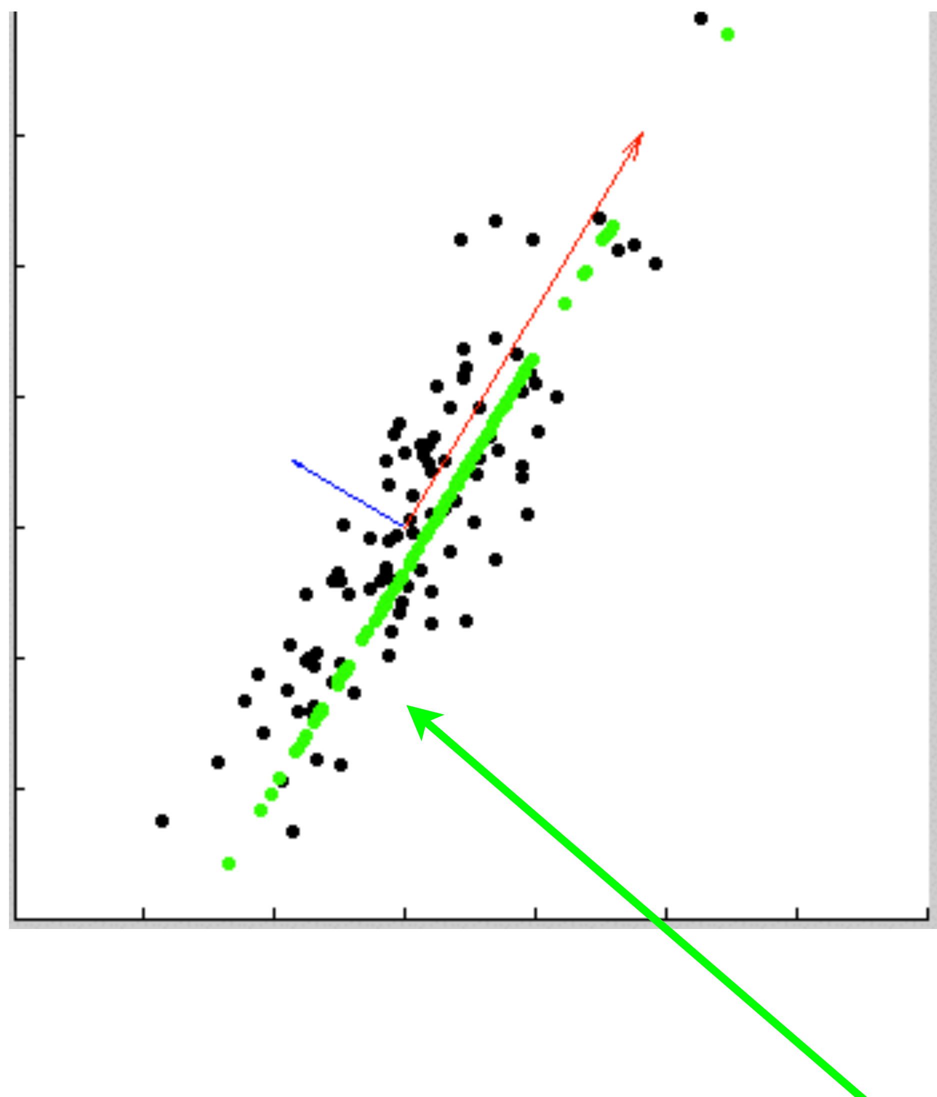
MM^T is symmetric! Max is along first eigenvector!

"Principal" eigenvector of MM^T is best 1D approx to the data

$MM^T v = \lambda v$



Principal component analysis



reduced data = Mv

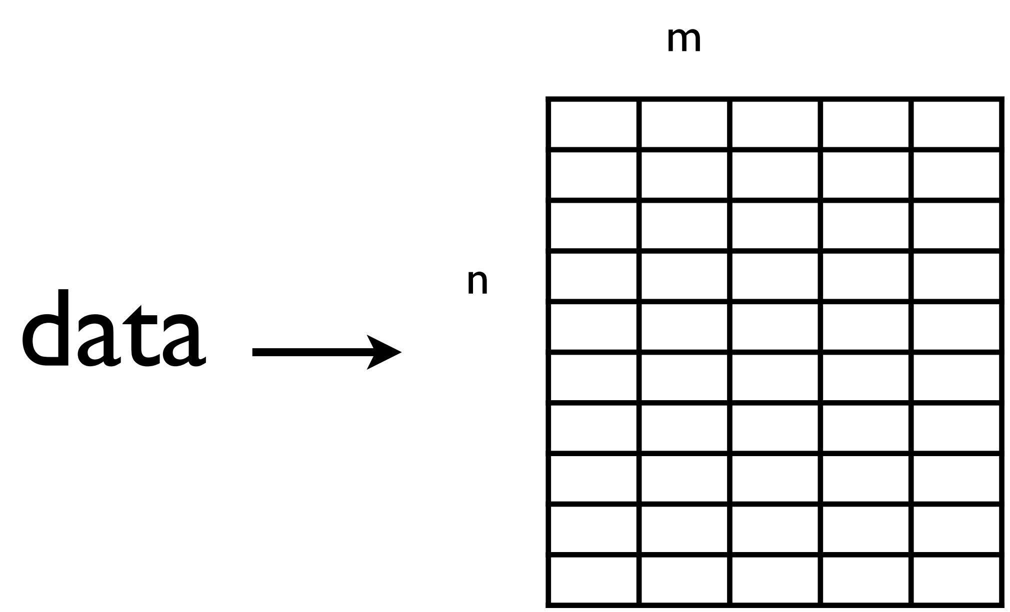
reduced data in original space = Mvv^T

Data projected onto first principal component

Principal component analysis

Identifying directions of large variance in data

. dimensionality reduction
. denoising
. finding patterns

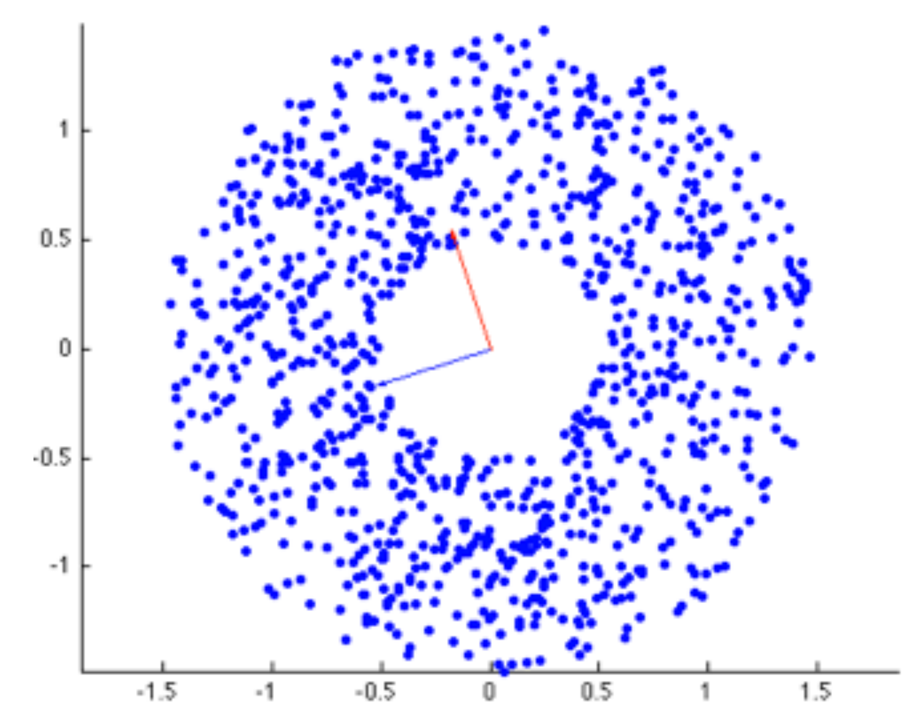


Assumptions in PCA

- The data is a linear combination of “interesting” components
- Variance is a good (sufficient?) feature
- Large variance is “interesting”

Assumptions in PCA

- The data is a linear combination of "interesting" components
- Variance is a good (sufficient?) feature
- Large variance is "interesting"

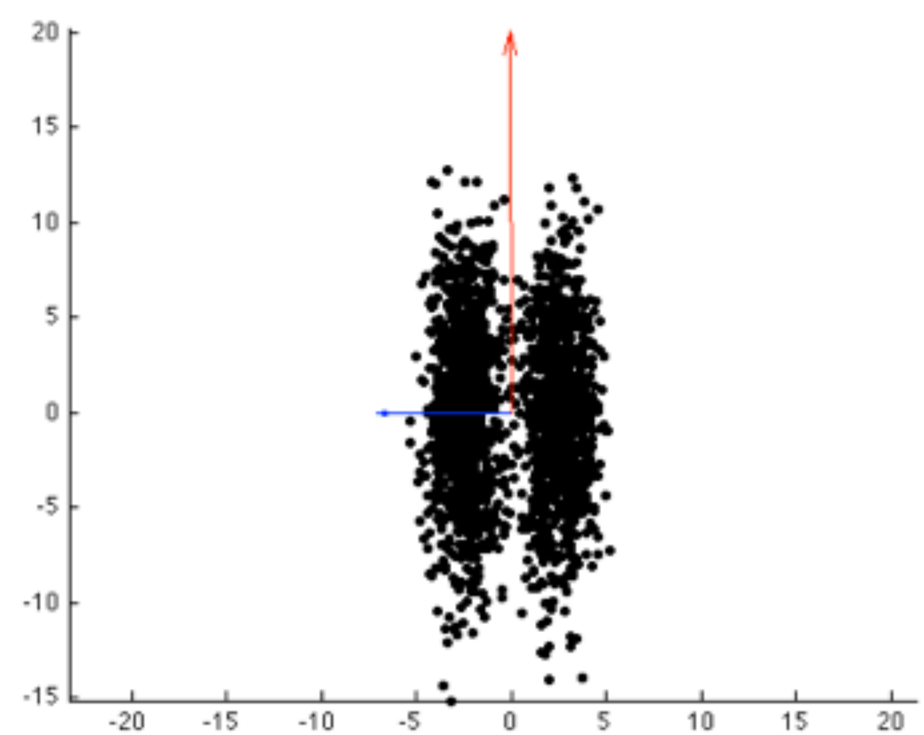


Alternatives:
Kernel PCA, MDS,Laplacian
eigenmaps, etc.

Assumptions in PCA

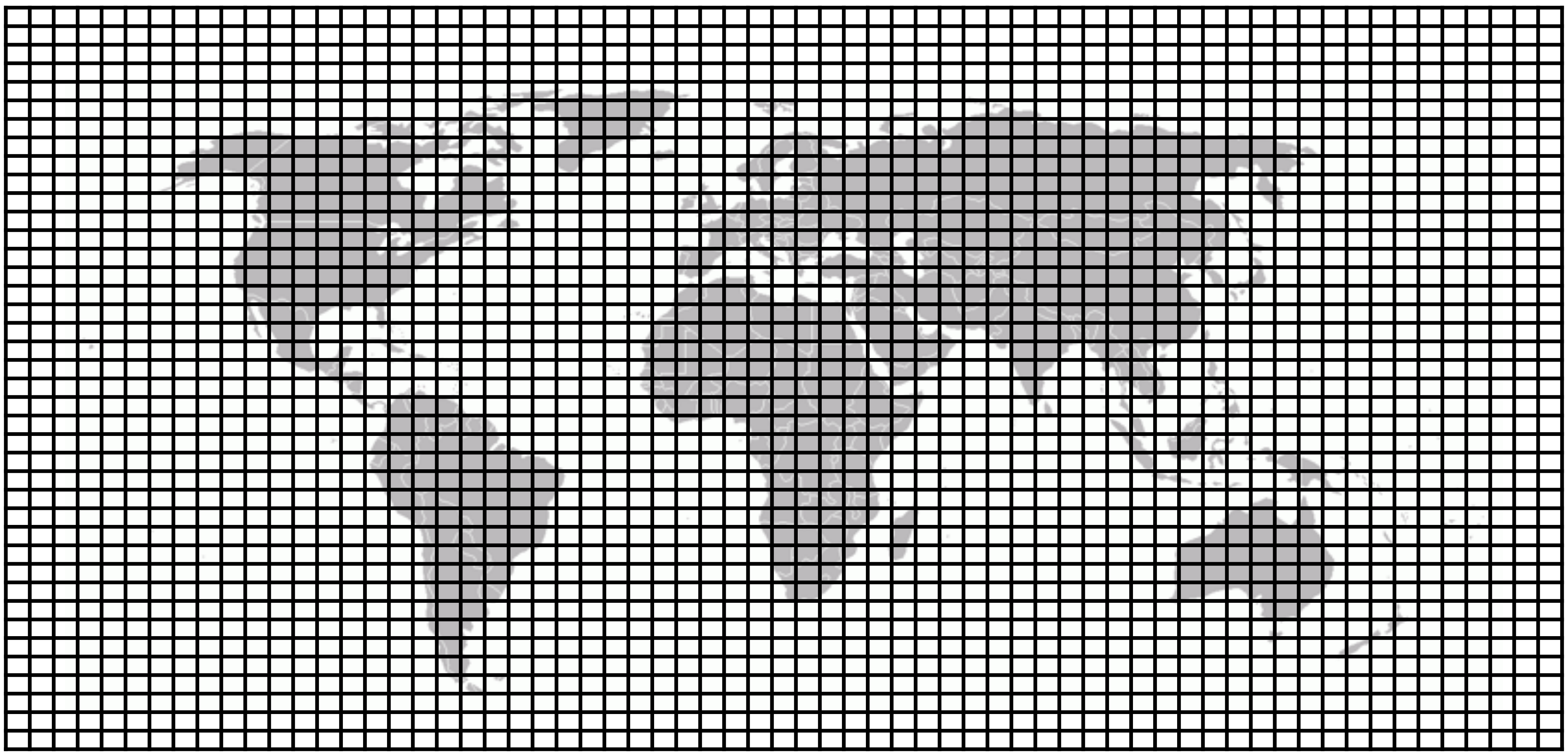
- The data is a linear combination of "interesting" components
- Variance is a good (sufficient?) feature
- Large variance is "interesting"

(Alternative: LDA)



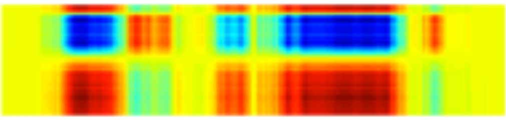
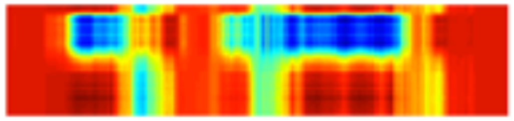
Example

PCA of the world



Example

PCA of the world

	new data	add mean	original data
1 (38%)			

ONBI – GLM Practical

www.fmrib.ox.ac.uk/~saad/ONBI/glm_practical.html

ONBI - GLM Practical. 2014/15

Practical Overview

This practical requires Matlab. Go through the page and execute the listed commands in the Matlab command window (you can copy-paste). Don't click on the "answers" links until you have thought hard about the question. Raise your hand should you need any help.

Contents:

Simple GLM

Let's start simple. Open matlab, and generate noisy data **y** using a linear model with one regressor **x** and an intercept. I.e. `y=a+b*x`

```
x = (1:20)';
intercept   = -10;
slope       = 2.5;
y = intercept + slope*x;
y = y + 10*randn(size(y)); % add some noise
```

Now plot the data against **x**:

```
figure
plot(x,y,'.');
xlabel('x');
ylabel('y');
```

Let's compare fitting a linear model with and without the intercept. First, set up two design matrices:

```
M1 = [x];                 % w/o intercept
M2 = [ones(size(x)) x];   % w/ intercept
```

That's all folks.