Linear Algebra

Simple Matrix Rules, Eigenvalues & Eigenvectors

Cornelius Weber, WTM

Matrix Computations

The elements of a matrix $C = AB$ are obtained as:

$$c_{mn} = \sum_{q}^{Q} a_{mq} b_{qn}$$

In matrix notation:

$$\begin{pmatrix} & & Q \\ & A & \\ M & & \end{pmatrix} \begin{pmatrix} & B & N \\ Q & & \end{pmatrix} = \begin{pmatrix} & & N \\ & C & \\ M & & \end{pmatrix}$$

$\#$ columns of $A = \#$ rows of B

Special Cases of Matrix Computations 1/2

- Dot product

$$AB = \vec{a}^{\mathsf{T}}\vec{b} = (a_1, \ldots, a_Q) \begin{pmatrix} b_1 \\ \vdots \\ b_Q \end{pmatrix} = c \quad \text{(scalar)}$$

- Tensor product

$$AB = \vec{a}\vec{b}^{\mathsf{T}} = \begin{pmatrix} a_1 \\ \vdots \\ a_M \end{pmatrix} (b_1, \ldots, b_N) = \begin{pmatrix} & & N \\ & C & \\ M & & \end{pmatrix}$$

Simple Matrix Rules

$$A + B = B + A$$

$$AB \neq BA \quad \text{(in general)}$$

$$(AB)^\mathsf{T} = B^\mathsf{T} A^\mathsf{T} \quad \text{(mind the order!)}$$

Special Cases of Matrix Computations 2/2

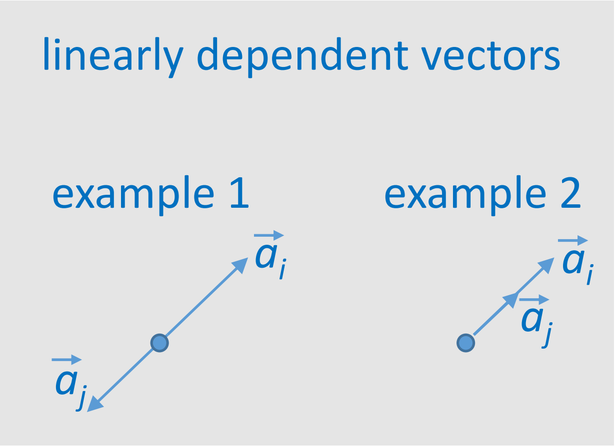
- Scalar a as a 1×1-matrix

$$\vec{c} = \vec{b}a \qquad \text{or} \qquad \vec{c}^T = a\vec{b}^T$$

- Linear combination of column-vectors

$$\begin{aligned} A\vec{x} &= \left(\begin{pmatrix} \cdot \\ \vec{a}_1 \\ \cdot \end{pmatrix} \cdots \begin{pmatrix} \cdot \\ \vec{a}_Q \\ \cdot \end{pmatrix} \right) \begin{pmatrix} x_1 \\ \cdot \\ x_Q \end{pmatrix} \\ &= \begin{pmatrix} \cdot \\ \vec{a}_1 \\ \cdot \end{pmatrix} x_1 + \ldots + \begin{pmatrix} \cdot \\ \vec{a}_1 \\ \cdot \end{pmatrix} x_Q \end{aligned}$$

$A\vec{x} = 0 \qquad$ with $\qquad \vec{x} \neq 0$

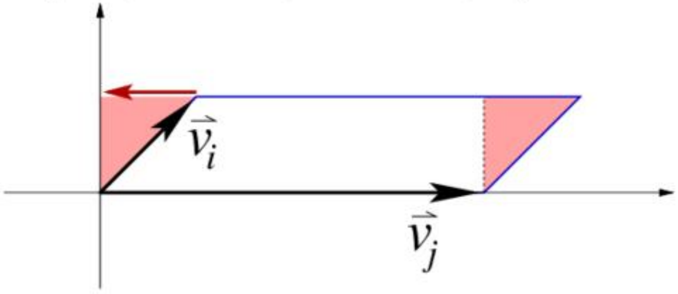


$\Rightarrow$ column vectors of A linearly dependent

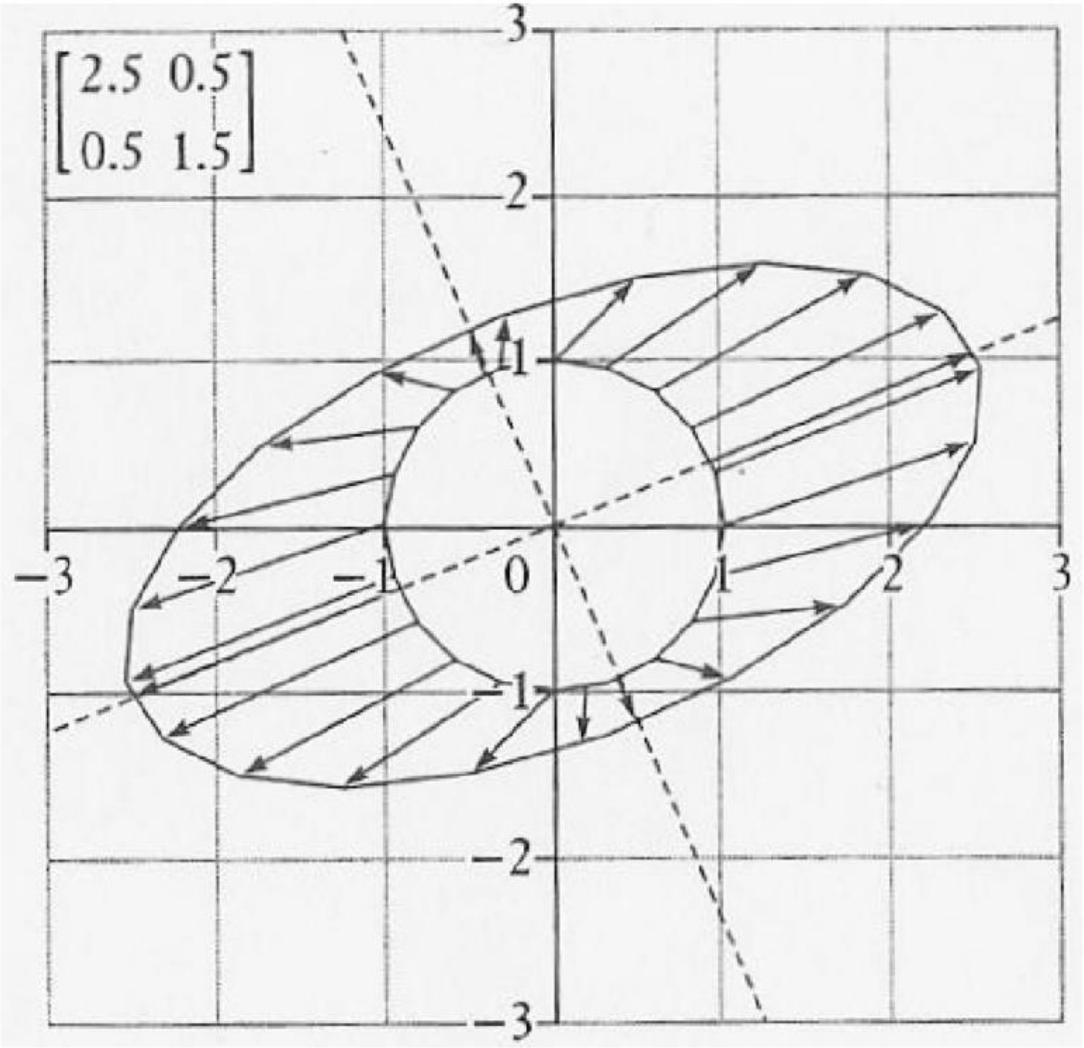
Implication for a square $N \times N$-matrix:
column vectors do **not** span N-dimensional space.

This is equivalent to that the determinant $det(A) = 0$.

The determinant of a square matrix is the area of the parallelogram spanned by its row (or column-) vectors (neglecting the sign).



Symmetric Matrix



The Eigenvalue Equation

V is a M-dimensional vector space $\mathbb{R}$.
A: $V \to V$ is a linear function ($\approx M \times M$-matrix).

A real number λ_m is an eigenvalue of A, if there exists a vector $\vec{v}_m \in V$ with $\vec{v}_m \neq 0$, so that:

$$A\vec{v}_m = \lambda_m \vec{v}_m$$

$\boxed{\text{The } \vec{v}_m \text{ are those vectors which are scaled by } A\text{, but not rotated.}}$

$\vec{v}_m$ is eigenvector $\Rightarrow c\,\vec{v}_m$ is eigenvector

The Eigenvalue Equation in Matrix Notation

for all M eigenvalues and eigenvectors:

$$AU = U\Lambda$$

or

$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} \left(\begin{pmatrix} \vec{v}_1 \\ \cdot \\ \cdot \end{pmatrix} \cdots \begin{pmatrix} \vec{v}_M \\ \cdot \\ \cdot \end{pmatrix} \right)$$

$$= \left(\begin{pmatrix} \vec{v}_1 \\ \cdot \\ \cdot \end{pmatrix} \cdots \begin{pmatrix} \vec{v}_M \\ \cdot \\ \cdot \end{pmatrix} \right) \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_M \end{pmatrix}$$

The Eigenvalue Equation in Matrix Notation

for all M eigenvalues and eigenvectors:

$$AU = U\Lambda$$

- Λ is a diagonal matrix, with the eigenvalues on the diagonal.
- U is a matrix, with the eigenvectors as its columns.
- Mind the order on both sides of the equation!
- M eigenvectors. Reasonable, because in an M-dimensional vector space there are maximally M linear independent vectors.

Let U^{-1} be the inverse matrix of U (i.e. $U^{-1}U = \mathbb{1}$):

$$U^{-1}AU = \Lambda \qquad \Leftrightarrow \qquad AU = U\Lambda \qquad \Leftrightarrow \qquad A = U\Lambda U^{-1}$$

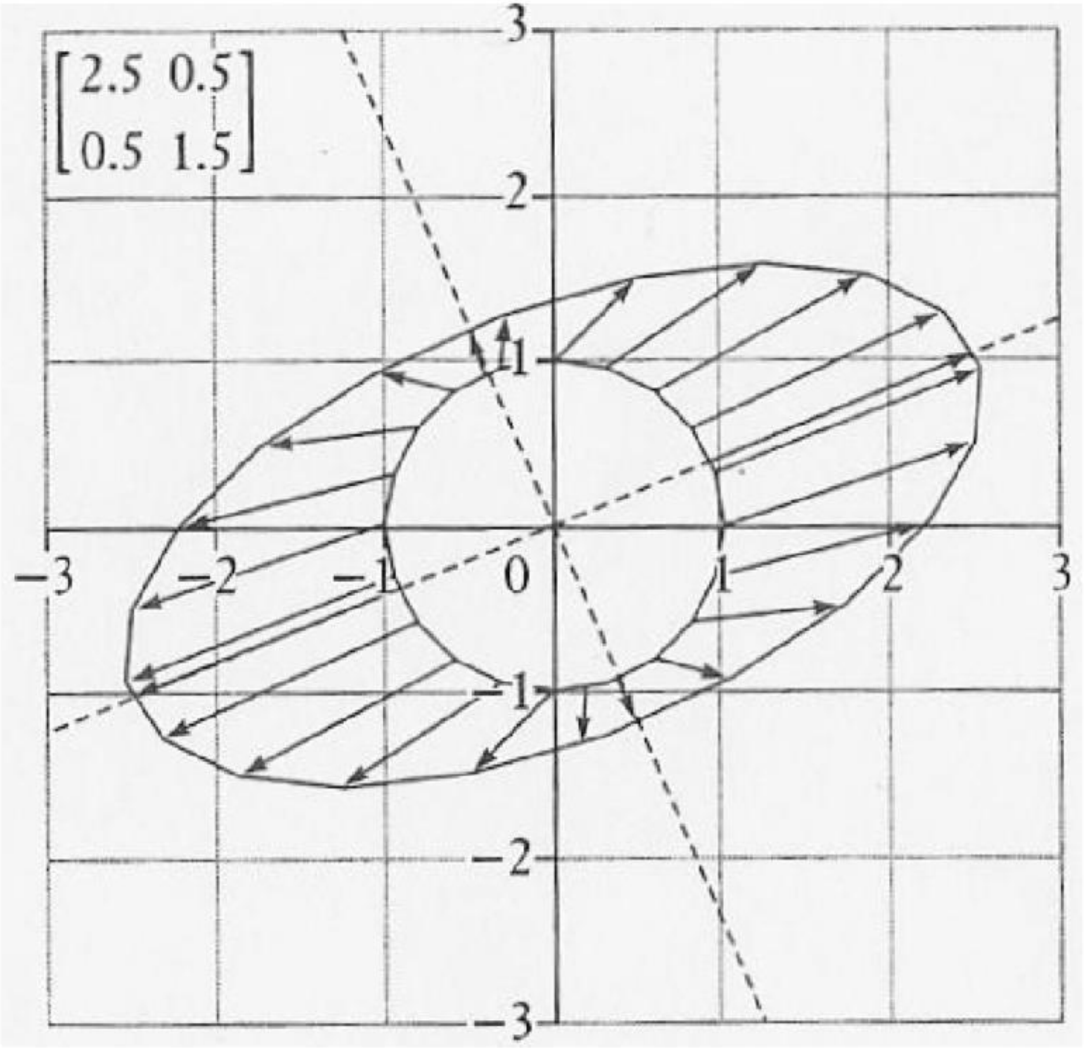
Relevance of the Eigenvalue Equation

We have: $\vec{y} = A\vec{x}$

We want: $\tilde{y} = \Lambda\,\tilde{x}$ (basis system of eigenvectors)

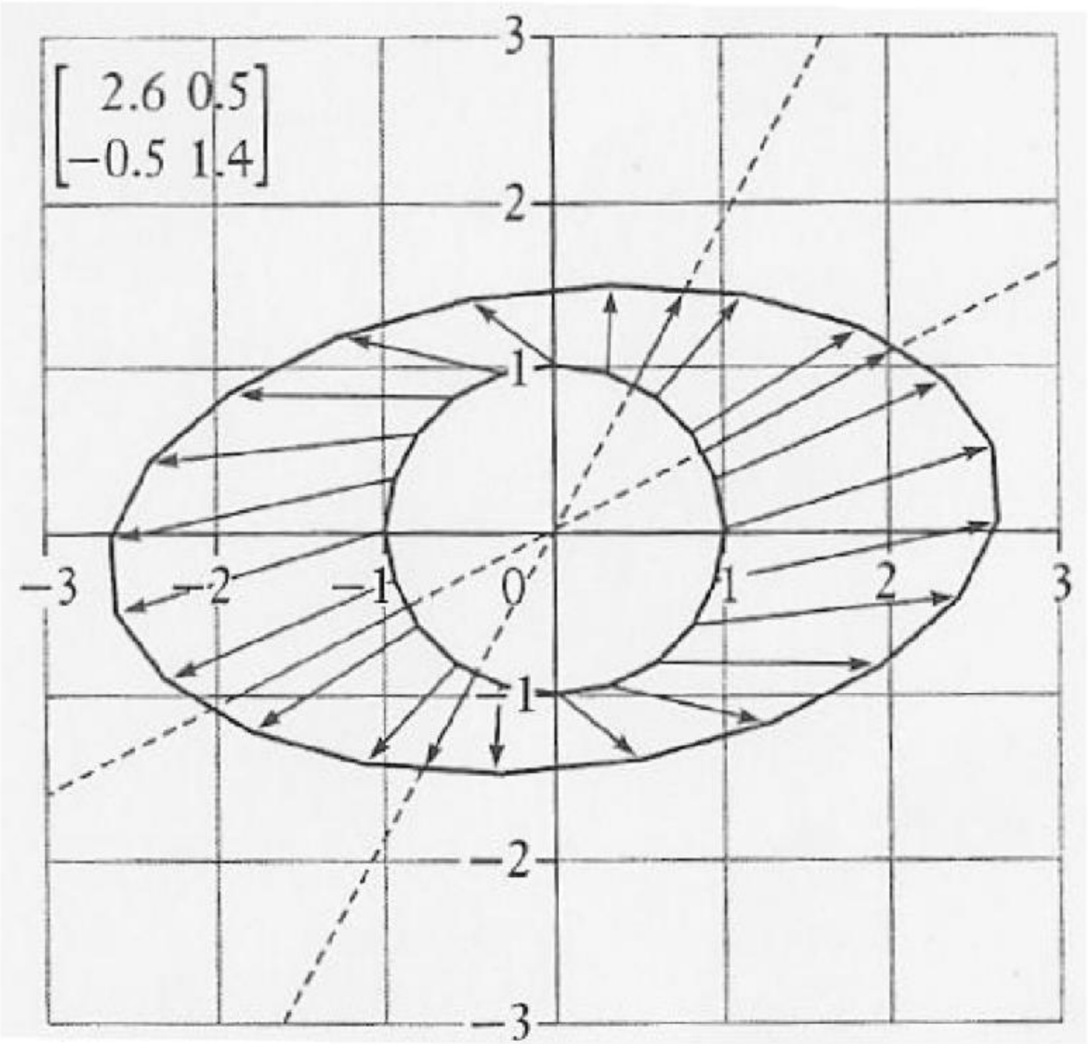
- How to get there: $\vec{y} = U\Lambda U^{-1}\,\vec{x}$
- Left-multiply by U^{-1}: $U^{-1}\vec{y} = \Lambda U^{-1}\,\vec{x}$
- Define: $\tilde{y} := U^{-1}\vec{y}$ and $\tilde{x} := U^{-1}\,\vec{x}$
- Back-transform: $\vec{y} = U\tilde{y}$ and $\vec{x} = U\,\tilde{x}$

Symmetric Matrix



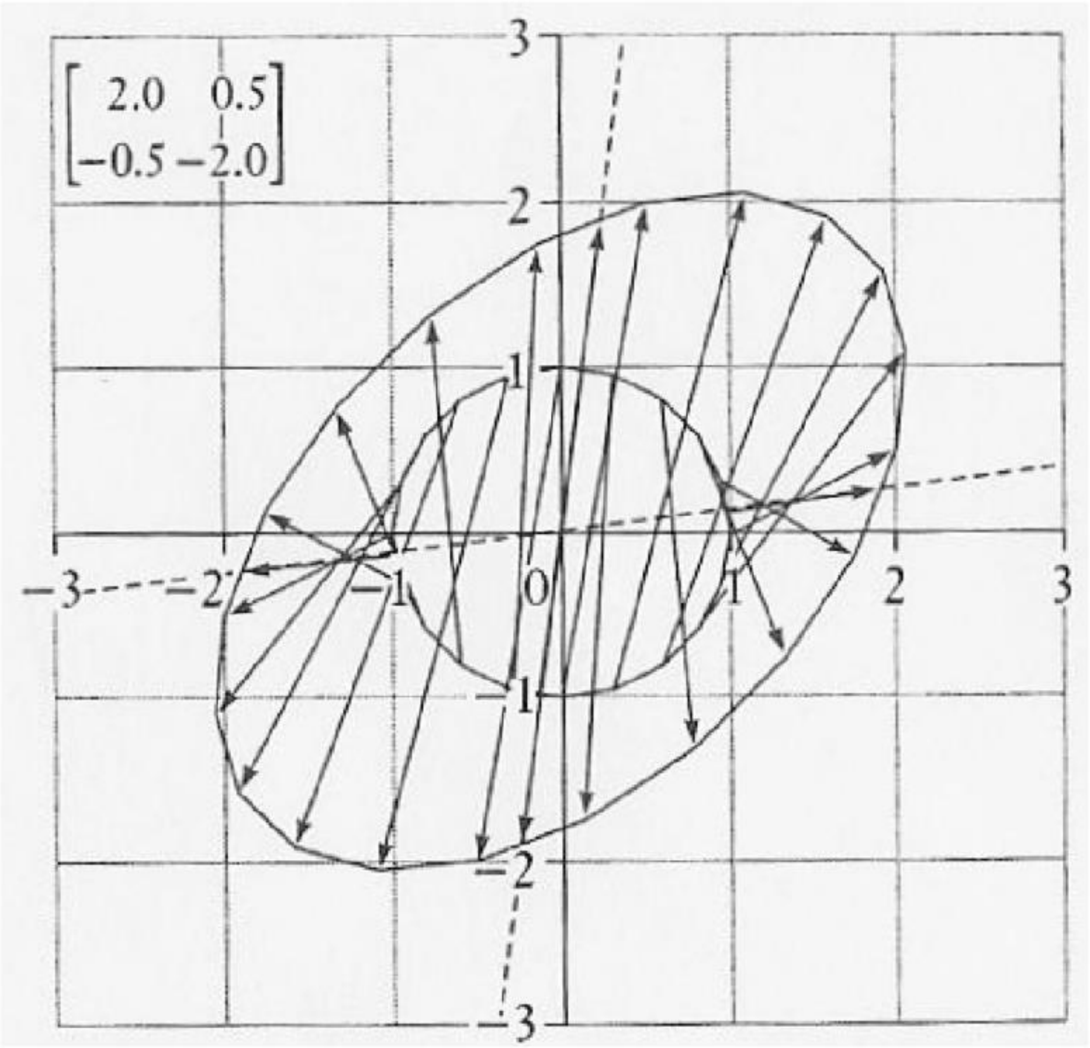
orthogonal eigenvectors

Non-Symmetric Matrix



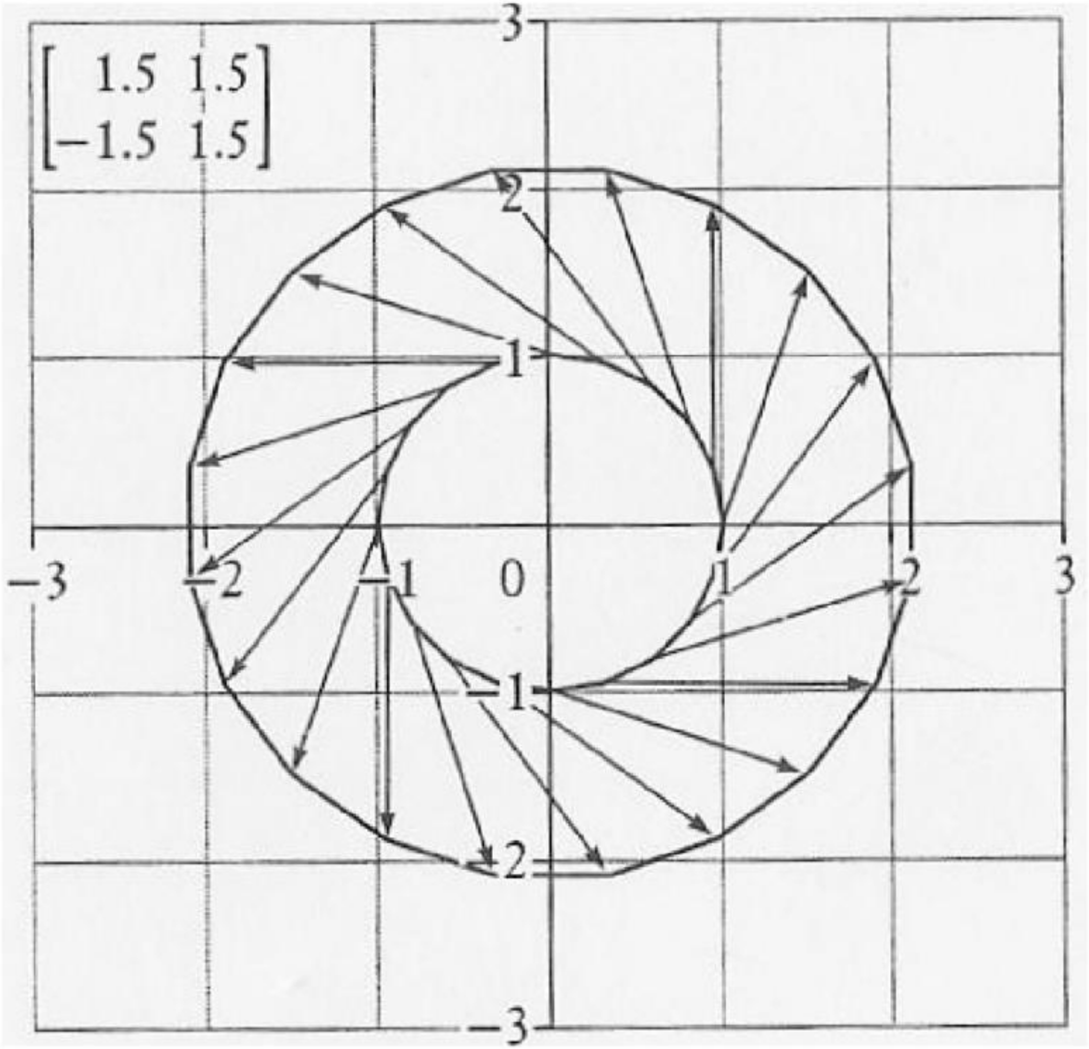
non-orthogonal eigenvectors

Arbitrary Matrix Operation



one positive, one negative eigenvalue

Rotation Matrix



no eigenvector

Orthonormal Vectors 1/2

M vectors $\in \mathbb{R}^M$ are orthonormal if

$$\vec{v}^i \cdot \vec{v}^j = \delta_{ij}$$

In matrix notation:

$$\begin{pmatrix} (\ldots \vec{v}^1 \ldots) \\ \cdot \\ (\ldots \vec{v}^M \ldots) \end{pmatrix} \left(\begin{pmatrix} \vec{v}^1 \\ \cdot \\ \cdot \end{pmatrix} \cdots \begin{pmatrix} \vec{v}^2 \\ \cdot \\ \cdot \end{pmatrix} \right) = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$$

Orthonormal Vectors 2/2

same as

$$SS^\intercal = \mathbb{1}_M$$

equivalent to (by left-multiplying with S^{-1})

$$S^\intercal = S^{-1}$$

Orthogonal eigenvectors may always be scaled to be orthonormal. Relevant for our eigenvectors, whether they span an orthogonal base of $\mathbb{R}^M$.

A Symmetric Matrix has Orthogonal Eigenvectors

Let $A\vec{v}_j = \lambda_j \vec{v}_j$ and $A\vec{v}_i = \lambda_i \vec{v}_i$
Then:

$$\lambda_j \vec{v}_i^{\mathsf{T}} \vec{v}_j = \vec{v}_i^{\mathsf{T}} \lambda_j \vec{v}_j = \vec{v}_i^{\mathsf{T}} A \vec{v}_j = (A^{\mathsf{T}} \vec{v}_i)^{\mathsf{T}} \vec{v}_j \overset{\text{symm}}{=} (A\vec{v}_i)^{\mathsf{T}} \vec{v}_j = \lambda_i \vec{v}_i^{\mathsf{T}} \vec{v}_j$$

$\underline{\text{If } \lambda_j \neq \lambda_i}$ then it has to be $\vec{v}_i^{\mathsf{T}} \vec{v}_j = 0$.

Finding the Eigenvalues – in Theory

The eigenvalue equation:

$$(A - \lambda_m \mathbb{1})\vec{v}_m = 0$$

- System of equations for $\vec{v}_m$
- Non-trivial solutions (i.e. $\vec{v}_m \neq 0$) can only exist if

$$det(A - \lambda_m \mathbb{1}) = 0$$

- Computing this determinant results in a polynomial in λ_m, the *characteristic polynomial* of A.
- Its solutions are the eigenvalues $\{\lambda_m\}$ of A.

Finding the Eigenvectors – in Theory

Assume we have found the Eigenvalues.
For each eigenvalue λ_m solve for $\vec{v}_m$:

$$(A - \lambda_m \mathbb{1})\vec{v}_m = 0$$

- System of equations for $\vec{v}_m$
- $det = 0 \Rightarrow$ solution not unique
- direction but not length of eigenvectors given

Finding Eigenvalues & Eigenvectors – Practical Example

```
python

import numpy
A = numpy.array([[1,2,3],[3,2,1],[1,0,-1]])
w, v = numpy.linalg.eig(A)

w
array([ 4.316624e+00, -2.316624e+00, 1.930415e-17])
v
array([[ 0.58428153, 0.73595785, 0.40824829],
        [ 0.80407569, -0.38198836, -0.81649658],
        [ 0.10989708, -0.55897311, 0.40824829]])
```

Towards Principal Component Analysis (PCA)

Given N random values $\{x_n\}$ with mean μ. The variance is:

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

Given N random M-dimensional vectors $\vec{x}_n$ with mean $\vec{\mu}$. The covariance between coordinate axes i and j is:

$$Cov_{ij} = \frac{1}{N}\sum_{n=1}^{N}(x_{in} - \mu_i) \cdot (x_{jn} - \mu_j)$$

All covariances make up the covariance matrix:

$$Cov := \frac{1}{N}\sum_{n=1}^{N}(\vec{x}_n - \vec{\mu})(\vec{x}_n - \vec{\mu})^T$$

In the following, let's assume for simplicity: $\vec{\mu} = 0$.

PCA – Cov Eigenvectors Form an Orthonormal Basis

The covariance matrix is symmetric, i.e. $Cov_{ij} = Cov_{ji}$.
$\Rightarrow$ Cov has orthogonal eigenvectors.

Let us normalize every eigenvector to length 1.
$\Rightarrow$ Arranging as column vectors to form an orthonormal matrix U.

The eigenvalue equation is (assume as solved):

$$Cov\, U = U\, \Lambda$$

The diagonal matrix Λ has the eigenvalues of Cov on the diagonal.

The column vectors of U are the orthonormal eigenvectors of Cov.

PCA – Express Data in Orthonormal Basis

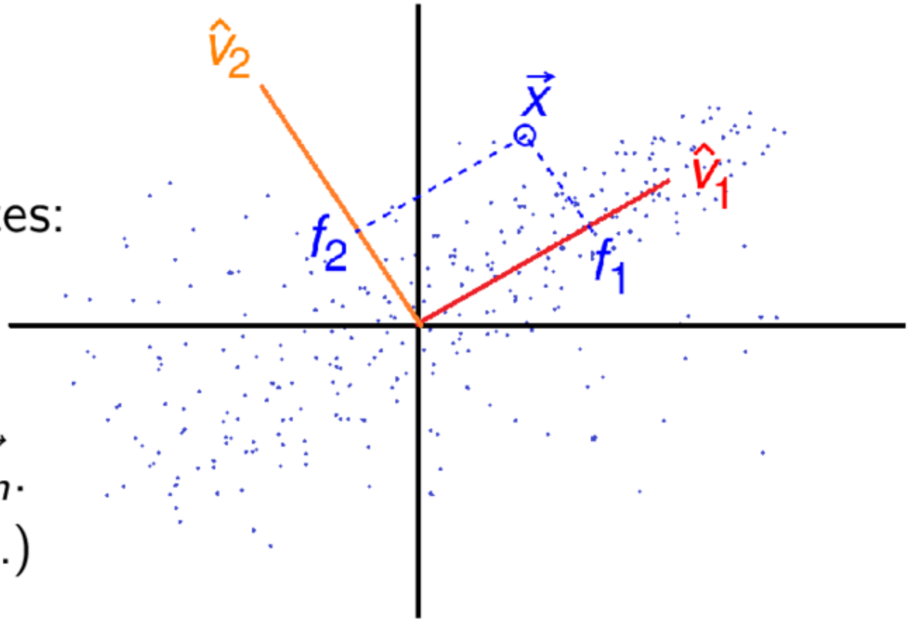
The normalized eigenvectors $\{\hat{v}_m\}$ set up an orthonormal base.
Each data vector $\vec{x}_n$ can be expressed by coordinates f_{mn} on the coordinate axes $\hat{v}_m$.
The coordinate on axis m is obtained by a projection (dot product) of the data point onto this axis:

$$f_{mn} = \hat{v}_m^T \vec{x}_n$$

This may be done for all coordinates:

$$\vec{f}_n = U^{\mathsf{T}} \vec{x}_n$$

and the backtransform is $\vec{x}_n = U\vec{f}_n$.
($U^{-1} = U^T$ for orthonormal matrices.)



PCA: Eigenvalues = Variances along the Principal Axes

Let's compute the variance of the data along an axis $\hat{v}_m$:

$$
\begin{aligned}
\sigma^2_{\hat{v}_m} &= \frac{1}{N}\sum_n f^2_{mn} \\
&= \frac{1}{N}\sum_n (\vec{x}_n^{\mathsf{T}}\hat{v}_m)(\vec{x}_n^{\mathsf{T}}\hat{v}_m) \\
&= \frac{1}{N}\sum_n \hat{v}_m^{\mathsf{T}}\vec{x}_n\vec{x}_n^{\mathsf{T}}\hat{v}_m \\
&= \hat{v}_m^{\mathsf{T}}\left(\frac{1}{N}\sum_n \vec{x}_n\vec{x}_n^{\mathsf{T}}\right)\hat{v}_m \\
&= \hat{v}_m^{\mathsf{T}}\, Cov\, \hat{v}_m \\
&= \lambda_m \hat{v}_m^{\mathsf{T}}\hat{v}_m \\
&= \lambda_m
\end{aligned}
$$

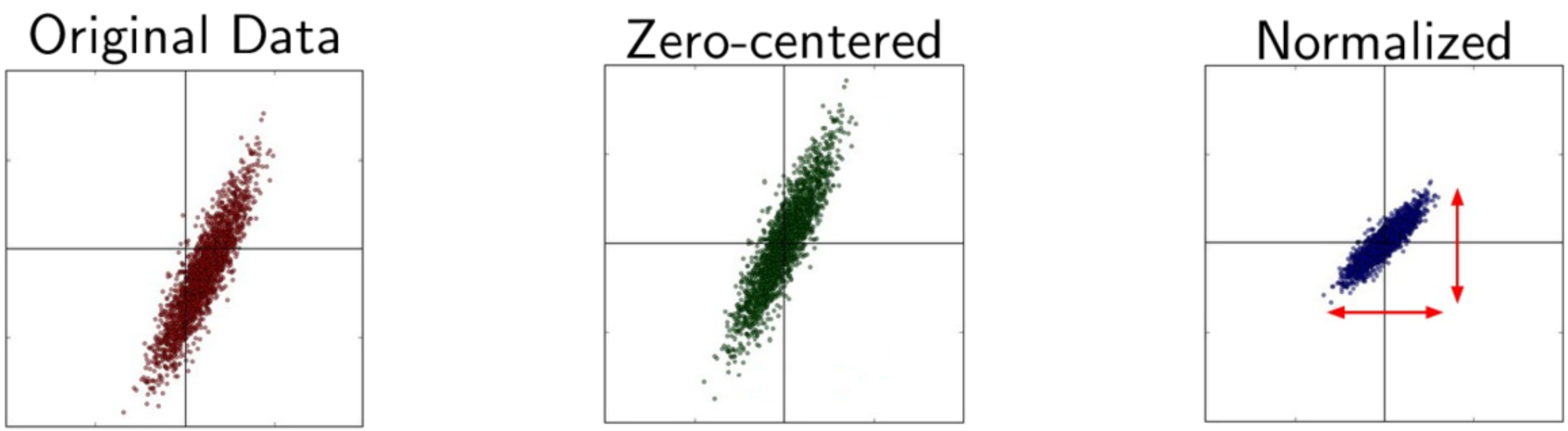
We used: $\hat{v}_m$ is eigenvector of Cov with eigenvalue λ_m.

Z-score Data Normalization

Many data-driven algorithms suffer from unequal ranges of individual data variables.

As simple solution apply Z-score normalization:

$$\vec{x} \mapsto \frac{\vec{x} - \vec{\mu}}{\sigma}$$



But variables are still highly correlated!

Figure source: http://cs231n.github.io/neural-networks-2/

Data Whitening

PCA provides us the eigendecomposition $Cov = U \Lambda U^{-1}$.

$U^{-1} = U^T$ projects into the coordinate system of eigenvectors.
There, covariances for $\hat{v}_m \neq \hat{v}_k$ are zero: variables are decorrelated.

Then divide by corresponding variance ($=$ eigenvalue) for each axis.

The full whitening transform is *(to be read from back to front)*:

$$\vec{x} \mapsto \Lambda^{-1} U^T (\vec{x} - \vec{\mu})$$

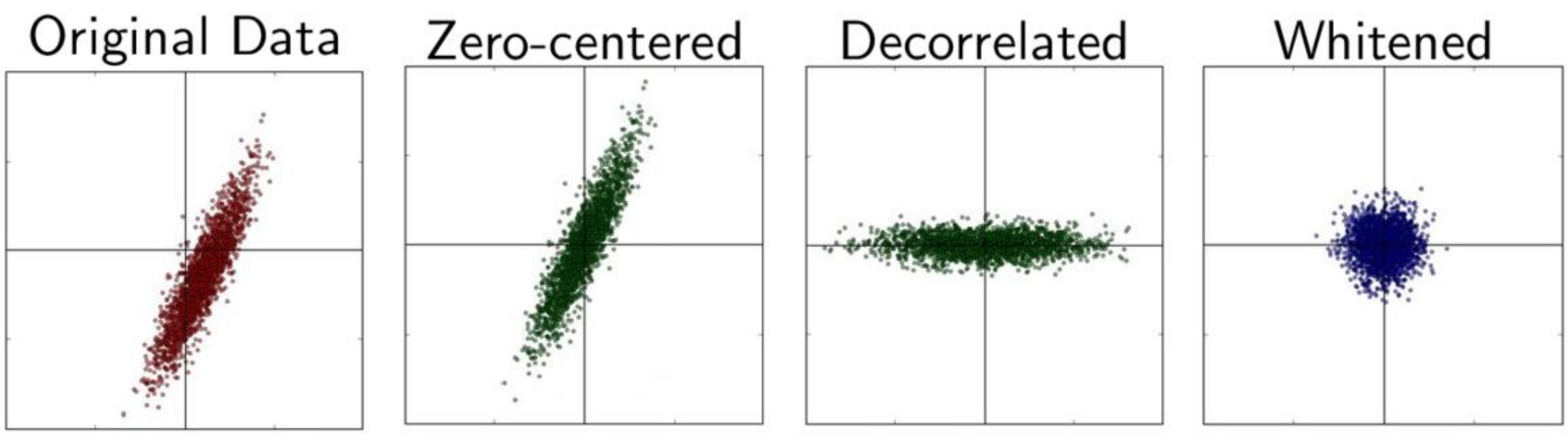


Figure source: http://cs231n.github.io/neural-networks-2/