Principal Component Analysis

CS771: Introduction to Machine Learning

Purushottam Kar

Recap of Last Lecture

Usefulness of treating matrices/vectors as maps in understanding

Several jargon terms related to linear algebra: column space, rank, column rank, (sub)space, affine space, (affine) hyperplane, nullspace

Singular Value Decomposition: a way to represent **any** rectangular or square matrix $A \in \mathbb{R}^{m \times n}$ as a product of two orthonormal (rotation+flip+swap) matrices and a scaling matrix $A = U \Sigma V^\top$

- *If* $k = \min\{m, n\}$ *then* $A = \sum_{i=1}^{k} \sigma_i \mathbf{u}^i \left(\mathbf{v}^i\right)^\top$ *using singular triplets* $\{\sigma_i, \mathbf{u}^i, \mathbf{v}^i\}$
- *Shows us that any linear map can be thought of as a "rotation" followed by a scaling operation (which may drop some coordinates or add some new ones) followed by another "rotation" (replace "rotation" with orthonormal map)*
- *Helps us gain an intuitive understanding of several concepts: rank, trace, determinant, inverse, pseudo inverse*

Eigenvectors and Eigenvalues

Can be considered quite enigmatic when taught in an MTH course

- *ML perspective is a bit different/easier (MTH people may dislike this – sorry ☹)*
- *SVD is clearly an extremely important (central even) concept in LinAlg and ML*
- *Eigenblah can be seen as simply a handy way of obtaining the SVD of a matrix*

Consider a feature matrix $X \in \mathbb{R}^{n \times d}$ – want to find its SVD $X = U\Sigma V^\top$

Instead, consider $X^\top X = V\Sigma^\top U^\top U\Sigma V^\top = V\Sigma^\top\Sigma V^\top \triangleq V\Lambda V^\top$

- *Nice as this matrix is square, symmetric and lets us focus on finding V – how?*
- *Let $V = [\mathbf{v}^1, \ldots, \mathbf{v}^d]$ (recall that columns of V are orthonormal). Then we have*

$$X^\top X\mathbf{v}^j = V\Lambda V^\top \mathbf{v}^j = V\Lambda\mathbf{e}_j = V\left(\lambda_j \cdot \mathbf{e}_j\right) = \lambda_j \cdot V\mathbf{e}_j = \lambda_j \cdot \mathbf{v}^j$$

where $\Lambda = [\lambda_1, \ldots, \lambda_n]$. Thus, $X^\top X$ merely scales right singular vectors of X

To handle U, simply take $XX^\top = U\Sigma V^\top V\Sigma^\top U^\top = U\Sigma\Sigma^\top U^\top \triangleq U\tilde{\Lambda}U^\top$

- *Will see that $XX^\top\mathbf{u}^i = \tilde{\lambda}_i\mathbf{u}^i$ i.e. $XX^\top$ merely scales left singular vectors of X*

Eigenvectors and Eigenvalues

For a square symm. matrix $A \in \mathbb{R}^{d\times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that
$A\mathbf{v} = \lambda \cdot \mathbf{v}$

> ***Caution**: $A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication*
>
> *The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc*
>
> *Eigenvalues may be negative or zero too*
>
> *In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$*

Eigenvectors and Eigenvalues

For a square symm. matrix $A \in \mathbb{R}^{d\times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

__Caution__: $A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication

The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc

Eigenvalues may be negative or zero too

In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$

$$A = \frac{1}{4} \cdot \begin{bmatrix} 7 & -3 \\ -3 & 7 \end{bmatrix}$$

Eigenvectors and Eigenvalues

For a square symm. matrix $A \in \mathbb{R}^{d \times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that
$A\mathbf{v} = \lambda \cdot \mathbf{v}$

- ***Caution***: *$A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication*
- *The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc*
- *Eigenvalues may be negative or zero too*
- *In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$*

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenvectors and Eigenvalues

For a square symm. matrix $A \in \mathbb{R}^{d \times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

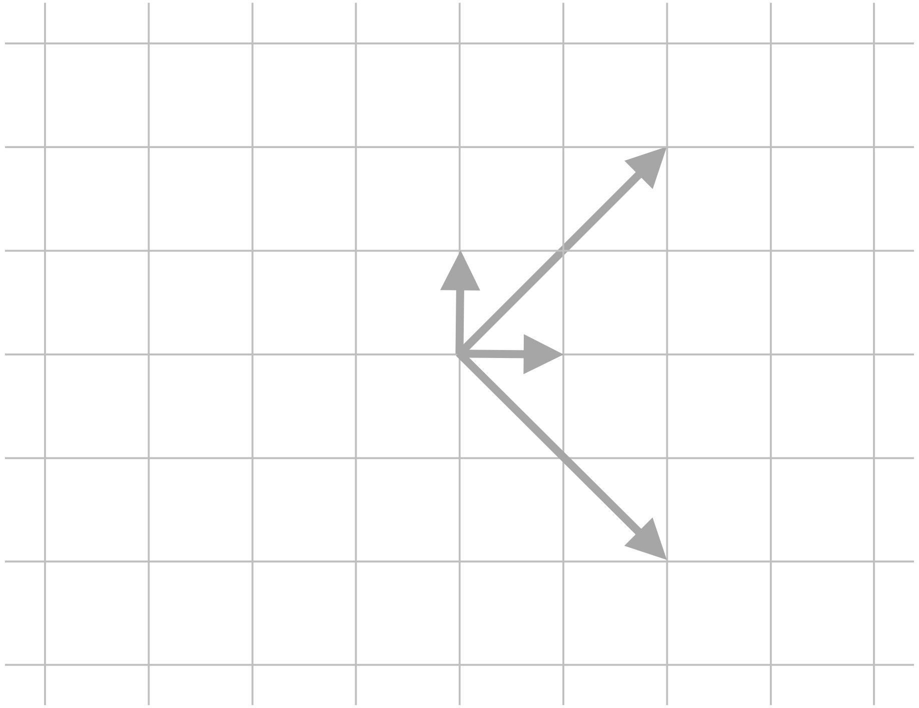
> ***Caution***: *$A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication*
>
> *The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc*
>
> *Eigenvalues may be negative or zero too*
>
> *In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$*

$$A = \overset{\text{counter clockwise rotation 45}}{\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}} \cdot \overset{\text{scaling}}{\begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}} \cdot \overset{\text{clockwise rotation 45}}{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}$$

Eigenvectors and Eigenvalues

For a square symm. matrix $A \in \mathbb{R}^{d\times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

- ***Caution***: *$A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication*
- *The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc*
- *Eigenvalues may be negative or zero too*
- *In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$*



$$A = \overset{\text{counter clockwise rotation 45}}{\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}} \cdot \overset{\text{scaling}}{\begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}} \cdot \overset{\text{clockwise rotation 45}}{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}$$

Eigenvectors and Eigenvalues

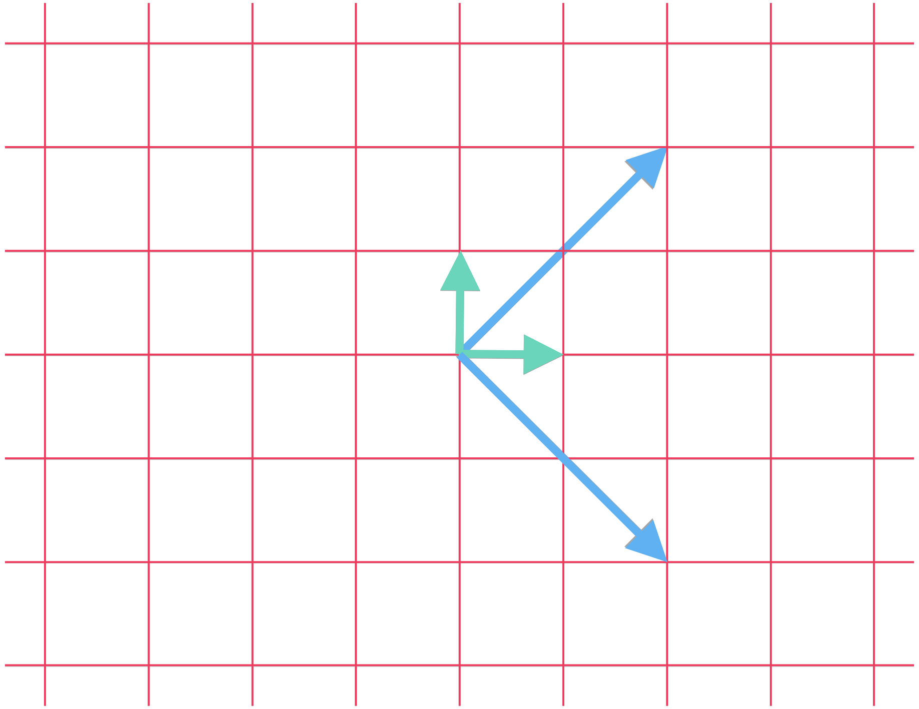
For a square symm. matrix $A \in \mathbb{R}^{d\times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

Caution*: $A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication*

The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc

Eigenvalues may be negative or zero too

In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$

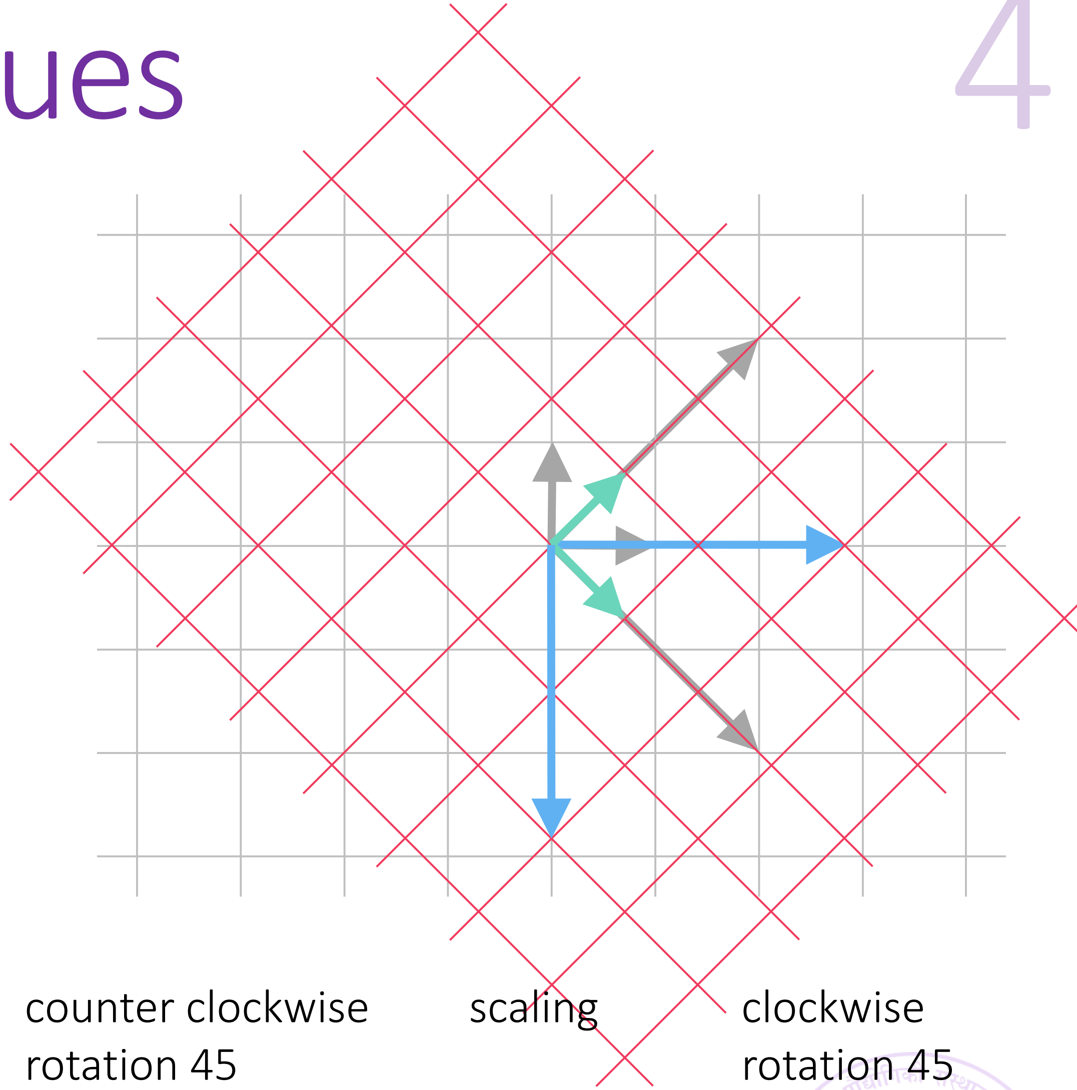


$$A = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}_{\text{counter clockwise rotation 45}} \cdot \underbrace{\begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}}_{\text{scaling}} \cdot \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}_{\text{clockwise rotation 45}}$$

Eigenvectors and Eigenvalues

For a square symm. matrix $A \in \mathbb{R}^{d \times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

- ***Caution***: *$A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication*
- *The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc*
- *Eigenvalues may be negative or zero too*
- *In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$*



counter clockwise rotation 45

scaling

clockwise rotation 45

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenvectors and Eigenvalues

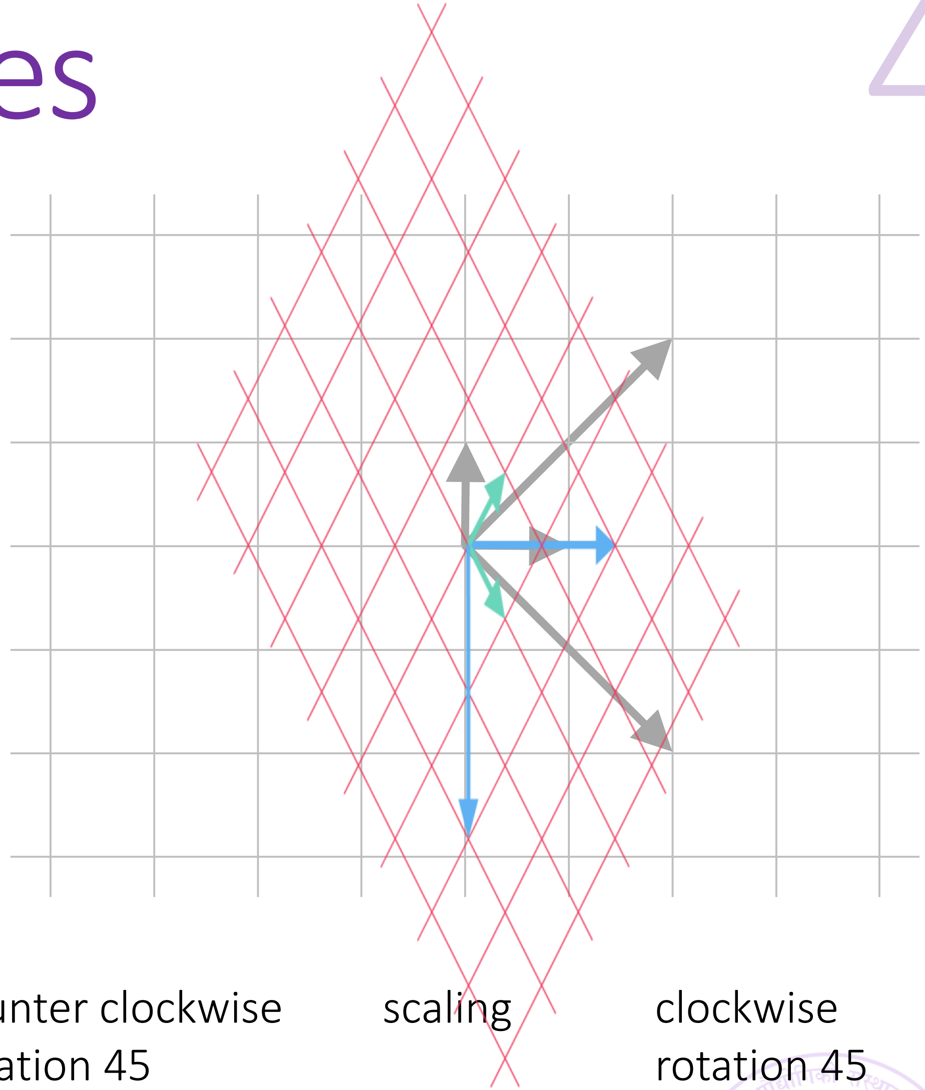
For a square symm. matrix $A \in \mathbb{R}^{d \times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

Caution: $A\mathbf{v}$ *is mat-vec multiplication whereas* $\lambda \cdot \mathbf{v}$ *is scalar multiplication*

The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc

Eigenvalues may be negative or zero too

In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ *is an eigenpair,* $(c \cdot \mathbf{v}, \lambda)$ *is also an eigenpair for every* $c \in \mathbb{R}$



counter clockwise rotation 45

scaling

clockwise rotation 45

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenvectors and Eigenvalues

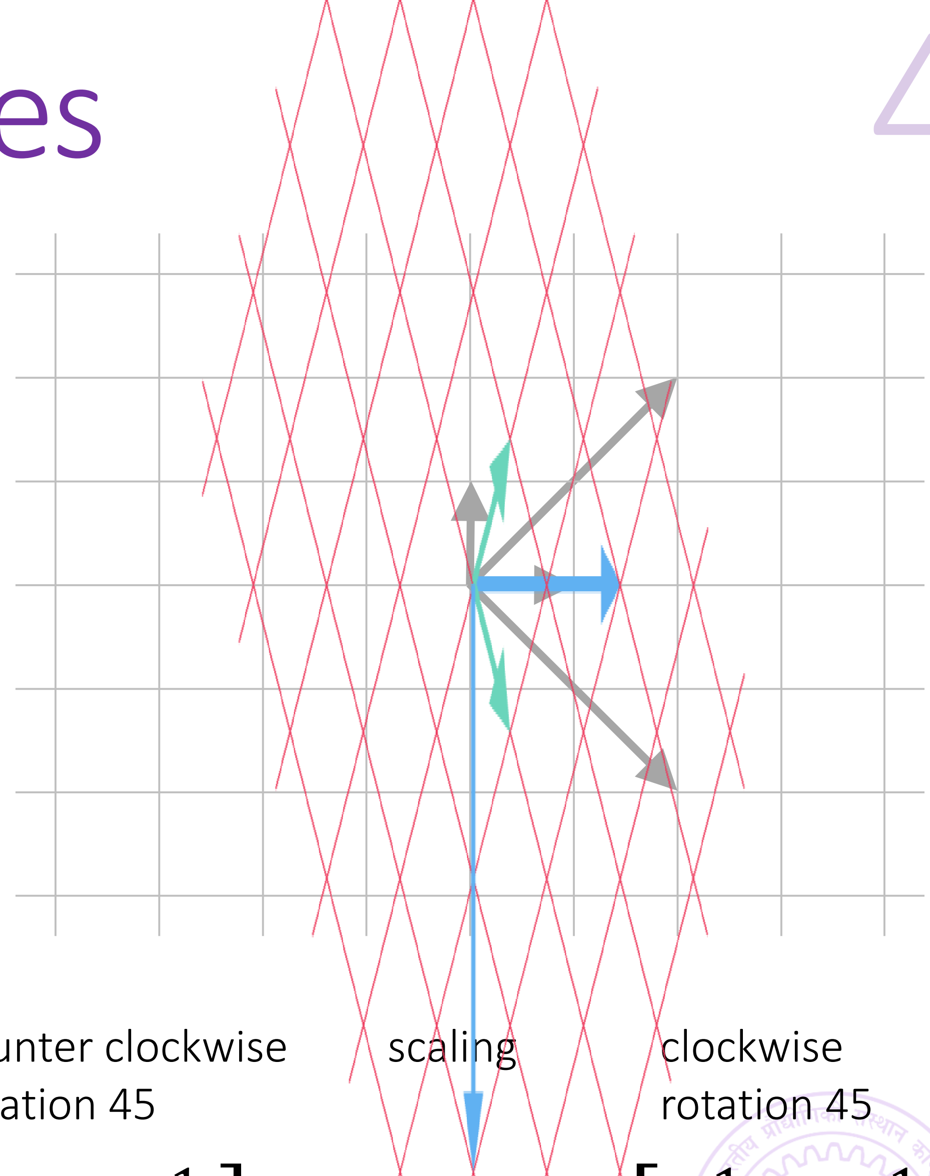
For a square symm. matrix $A \in \mathbb{R}^{d \times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

__Caution__: $A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication

The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc

Eigenvalues may be negative or zero too

In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$



counter clockwise rotation 45

scaling

clockwise rotation 45

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenvectors and Eigenvalues

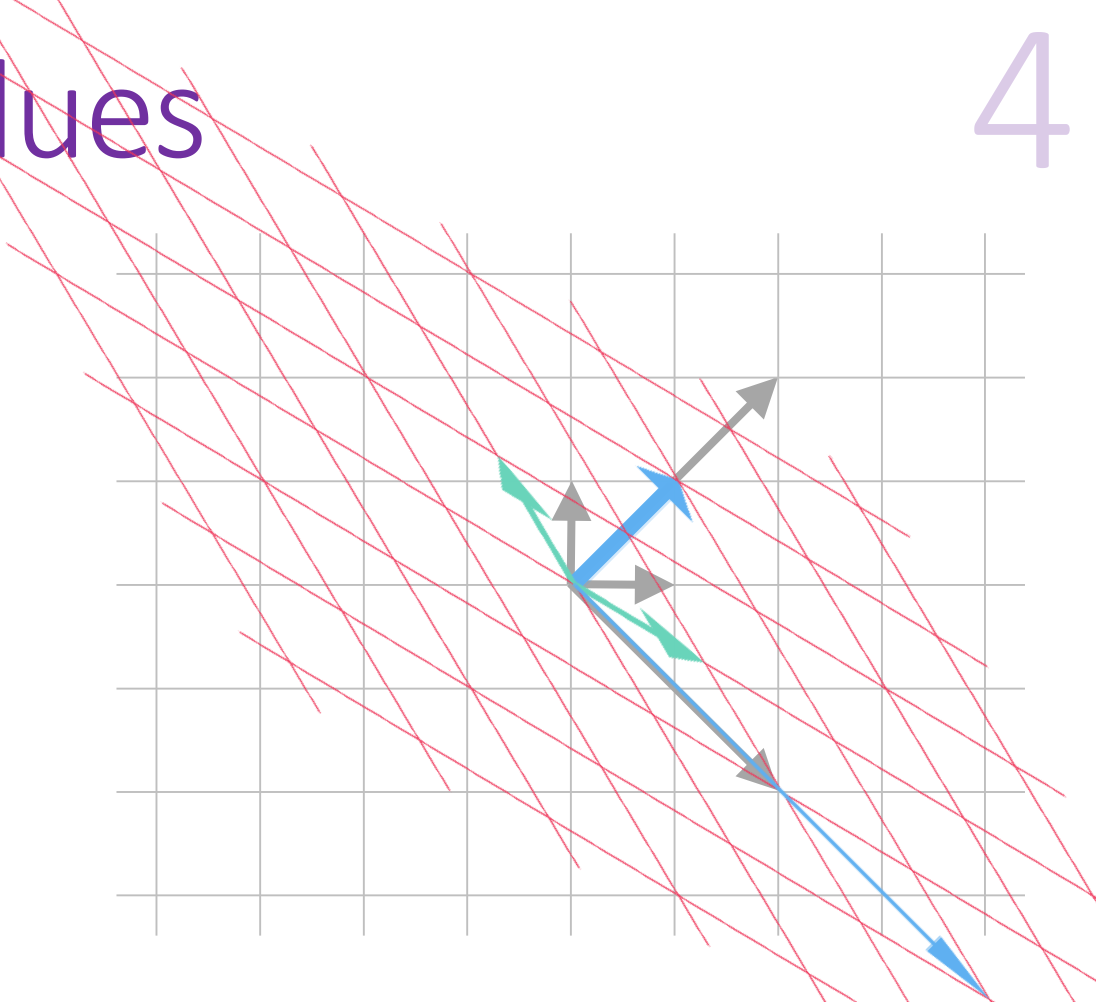
For a square symm. matrix $A \in \mathbb{R}^{d\times d}$, an *eigenpair* for this matrix is a pair of a vector and a scalar $(\mathbf{v}, \lambda)$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$

__Caution__: $A\mathbf{v}$ is mat-vec multiplication whereas $\lambda \cdot \mathbf{v}$ is scalar multiplication

The vector in the eigenpair (eigenvector) is merely scaled by the scalar in the pair (eigenvalue) and not rotated etc

Eigenvalues may be negative or zero too

In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ is an eigenpair, $(c \cdot \mathbf{v}, \lambda)$ is also an eigenpair for every $c \in \mathbb{R}$



counter clockwise rotation 45

scaling

clockwise rotation 45

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenve

For a square

an *eigenpair* for this matrix is a pair of

a vector a

$\mathbf{A}\mathbf{v} = \lambda \cdot$

Caution

wherea

The vector in the eigenpair (eigenvector)

is merely scaled by the s

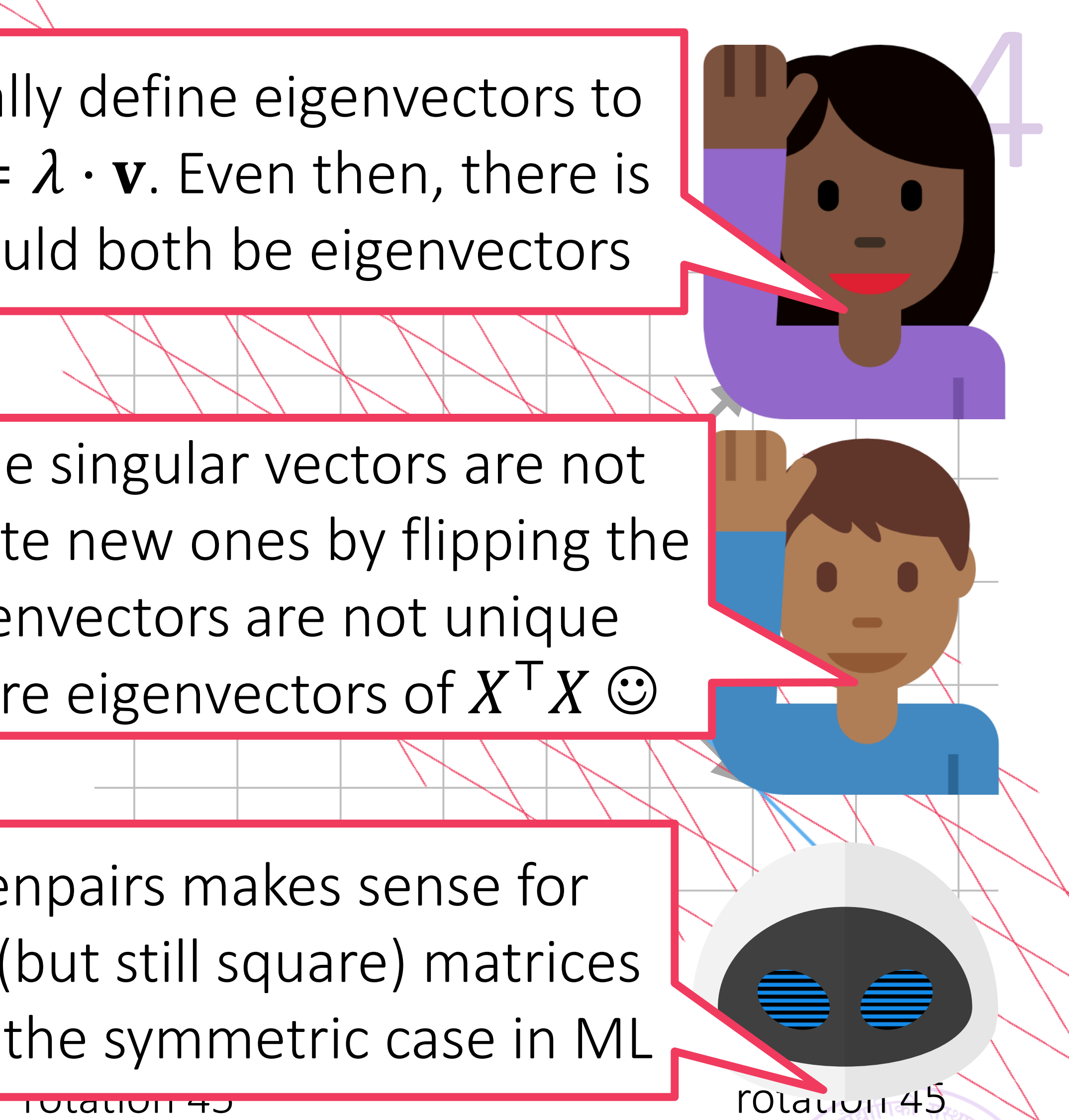
(eigenvalue) and not rot

Eigenvalues may be neg

In general, matrices have an infinite number of such eigenpairs – whenever $(\mathbf{v}, \lambda)$ *is an eigenpair,* $(c \cdot \mathbf{v}, \lambda)$ *is also an eigenpair for every* $c \in \mathbb{R}$

To get around this issue, people usually define eigenvectors to be only unit vectors $\mathbf{v}$ such that $A\mathbf{v} = \lambda \cdot \mathbf{v}$. Even then, there is still some ambiguity as $\mathbf{v}$ and $-\mathbf{v}$ would both be eigenvectors

Recall that we commented that even the singular vectors are not really unique since we can always generate new ones by flipping the sign etc. so it is not surprising that eigenvectors are not unique either since right singular vectors of X are eigenvectors of $X^\top X$ ☺



The concept of eigenpairs makes sense for even non-symmetric (but still square) matrices but we will only need the symmetric case in ML

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Eigenpairs and Singular Triplets

We have seen how right singular vectors of X are eigenvectors of $X^\top X$ and left singular vectors of X are eigenvectors of $XX^\top$

It turns out that the square roots of the eigenvalues of $X^\top X$ give us all the non-zero singular values of X

In fact, the square roots of the eigenvalues of $XX^\top$ also do the same thing because $XX^\top$ and $X^\top X$ share non-zero eigenvalues

***Proof**: We have $X^\top X = V\Sigma^\top\Sigma V^\top$ and $XX^\top = U\Sigma\Sigma^\top U^\top$. We defined $\Lambda = \Sigma^\top\Sigma$ which is a $d \times d$ diagonal matrix and $\tilde{\Lambda} = \Sigma\Sigma^\top$ which is an $n \times n$ diagonal matrix. It is easy to verify the following simple facts*

Λ and $\tilde{\Lambda}$ share their non-zero (diagonal) entries

Those diagonal entries are exactly the eigenvalues corresponding to the eigenvectors

Those diagonal entries are exactly squares of the singular values of X

Definiteness

Note that $\forall\, \mathbf{x} \in \mathbb{R}^d$, we have $\mathbf{x}^\top X^\top X\mathbf{x} = \|X\mathbf{x}\|_2^2 \geq 0$ i.e. $X^\top X$ is PSD

We are now ready to see when exactly is a matrix $A \in \mathbb{R}^{d\times d}$ PSD. We will restrict ourselves to only square symmetric matrices

Non-symmetric case gets a bit complicated due to diagonalizability *issues*

Claim without proof: every square symmetric matrix A can be written as $A = QSQ^\top$ where $Q \in \mathbb{R}^{d\times d}$ is orthonormal and S is diagonal matrix

As before, we can write $A = \sum_{j=1}^d s_j \mathbf{q}^j \left(\mathbf{q}^j\right)^\top$ *where* $Q = [\mathbf{q}^1, \dots, \mathbf{q}^d]$ *and* $S = \mathbf{diag}(s_1, \dots, s_d)$. *It is easy to see that* $\left(\mathbf{q}^j, s_j\right)$ *are eigenpairs for* A

Thus, for every $\mathbf{x} \in \mathbb{R}^d$, *we have* $\mathbf{x}^\top A\mathbf{x} = \sum_{j=1}^d s_j \left(\mathbf{x}^\top \mathbf{q}^j\right)^2$

If all $s_j \geq 0$ *then easy to see that we always have* $\mathbf{x}^\top A\mathbf{x} \geq 0$ *i.e. PSD*

If even one $s_j < 0$, *then take* $\mathbf{x} = \mathbf{q}^j$ *to get* $\mathbf{x}^\top A\mathbf{x} < 0$ *i.e. non PSD*

Defin

A square symmetric matrix is PSD if and only if (aka iff) none of its eigenvalues is negative. We use the term *Positive Definite (PD)* to refer to matrices all of whose eigenvalues are strictly positive. For these matrices, $\mathbf{x}^\top A\mathbf{x} > 0$ unless $\mathbf{x} = \mathbf{0}$ in which case obviously we must have $\mathbf{x}^\top A\mathbf{x} = 0$

Note th

We are now ready to see when exactly is a matrix $A \in \mathbb{R}^{d\times d}$ PSD. We will restrict ours

Non-symmetric

Can you see that a square symmetric matrix has one or more zero eigenvalues iff it is not full rank? A full rank square symmetric matrix will have only non-zero eigenvalues

Claim without proof: every square symmetric matrix A can be w as $A = QSQ^\top$ where $Q \in \mathbb{R}^{d\times d}$ is orthonormal and S is diagonal matrix

It is illuminating to see this work when the square symmetric matrix is $X^\top X$ or $XX^\top$. All eigenvalues are squares of singular values of X which means that

1. All its eigenvalues must be non-negative i.e. $X^\top X$ is always PSD
2. If $d < n$, then $X^\top X$ can have a zero eigenvalue iff X has a zero singular value. This means that $X^\top X$ is full rank iff X is full rank in the case $d < n$
3. If $d \geq n$, then $XX^\top$ can have a zero eigenvalue iff X has a zero singular value. This means that $XX^\top$ is full rank iff X is full rank in the case $d \geq n$

If even one $s_j < 0$, then take $\mathbf{x} = \mathbf{q}^j$ to get $\mathbf{x}^\top A\mathbf{x} < 0$ i.e. non PSD

Singular Blah vs Eigenblah

Singular Value Decomposition (SVD): write a (rect) matrix $A = U\Sigma V^\top$

Eigendecomposition (ED): write a square symm matrix as $B = QSQ^\top$

- *Be aware of the differences between SVD and ED – do not get confused*
- *SVD always exists, no matter whether matrix is square/rect, symm/non-symm*
- *ED always exists for square symm mats, may not exist (may require complex S or non-ortho Q) for non symm. mats. ED does not make sense for rect mats.*
- *Singular values are always non-negative, eigenvalues can be pos/neg/complex*

For a PSD square symmetric matrix, its SVD is its ED and vice versa

For a nonPSD square symm. matrix, its ED can be used to obtain its SVD

- *Get ED* $A = QSQ^\top$. *Let* $N = \text{diag}\left(\text{sign}(s_1), \ldots, \text{sign}(s_d)\right)$. *Then, let* $U = QN$, $V = Q$ *and* $\Sigma = |S|$ *(or else take* $U = Q, V = NQ$*) to get the SVD* $A = U\Sigma V^\top$
- *Note that* U, V *still orthonormal but* $U \neq V$

Sin

Symmetric square matrices always have real eigenvalues. It is only in the non-symmetric case that funny things start happening. Fortunately, in most ML situations, whenever we encounter square matrices, they are symmetric too.

Sing

Rotation matrices (and orthonormal matrices in general) are where the difference between SVD and ED is most stark. Rotation matrices in general do not (actually cannot) have any eigenvectors. This is because they rotate every vector (except $\mathbf{0}$ which is a trivial case) so no vector is transformed with just a simple scaling. Thus, they have no ED. However, they do have an SVD (all matrices do) and actually they are their own SVD i.e. for a rotation matrix A, the SVD is $A = AII$

$QSQ^\top$

ed

on-

t mats:

Singular values are always non-negative, eigenvalues can be pos/neg/complex

However, the identity matrix (which is also a rotation matrix – rotation by 0 degrees) is also a bit weird in that every vector is its eigenvector since $I\mathbf{x} = \mathbf{x}$. The identity matrix has the same SVD and ED and that is $I = III$

For

For

Get ED $A = QSQ^\top$. *Let* $N = \mathrm{diag}(\mathrm{sign}(s_1), \ldots, \mathrm{sign}(s_d))$. *Then, let* U

$V = Q$ *and* $\Sigma = |S|$ *(or else take* $U = Q, V = NQ$*) to get the SVD* $A = U\Sigma V^\top$

Note that U, V *still orthonormal but* $U \neq V$

Principal Component Analysis

The largest singular value (resp. eigenvalue) of a matrix is called its *leading* singular value (resp. eigenvalue)

The corresponding left/right singular vector (resp. eigenvector) is called its leading *left/right singular vector (resp. eigenvector)*

In some cases, there may be more than one singular vector with the same singular value (resp. more than one eigenvector with the same eigenvalue)

Principal Component Analysis: the process of finding the top few singular values and corresponding singular vectors (left + right)

Given $X \in \mathbb{R}^{n \times d}$ *(assume* $d < n$*. The case* $d \geq n$ *similar) with SVD* $X = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$ *where* $\tilde{U} \in \mathbb{R}^{n \times d}$*,* $\tilde{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_d)$ *with* $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$ *and* $\tilde{V} \in \mathbb{R}^{d \times d}$

$\tilde{U}$ *and* $\tilde{V}$ *both have orthonormal columns,* $\tilde{V}$ *is square but* $\tilde{U}$ *is not!*

Note*: we dropped the last* $n - d$ *columns of* U *which were useless, to get* $\tilde{U}$

Want to find the leading triplets i.e. $\{\sigma_i, \mathbf{u}^i, \mathbf{v}^i\}$ *for* $i = 1, \ldots k$ *for some* $k \leq d$

Prin

Careful: since we have dropped some columns of U we need to be careful. When U was square i.e. $\in \mathbb{R}^{n\times n}$, its columns were orthonormal as were its rows i.e. $UU^\top = I = U^\top U$. However, now that we have removed some columns, whereas the remaining columns are still orthonormal, the rows are not necessarily orthonormal i.e. $\tilde{U}^\top\tilde{U} = I$ but maybe $\tilde{U}\tilde{U}^\top \neq I$

The l

leadi

The corresponding left/right singular vector (resp. eigenvector) is called its leading *left/right singular vector (resp. eigenvector)*

In some cases, there may be more than one singular vector with the same singular value (resp. more than one eigenvector with the same eigenvalue)

Principal Component Analysis: the process of finding the top few singular values and corresponding singular vectors (left + right)

Given $X \in \mathbb{R}^{n\times d}$ *(assume* $d < n$*. The case* $d \geq n$ *similar) with SVD* $X = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$ *where* $\tilde{U} \in \mathbb{R}^{n\times d}$, $\tilde{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ *with* $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$ *and* $\tilde{V} \in \mathbb{R}^{d\times d}$

$\tilde{U}$ *and* $\tilde{V}$ *both have orthonormal columns,* $\tilde{V}$ *is square but* $\tilde{U}$ *is not!*

Note*: we dropped the last* $n - d$ *columns of* U *which were useless, to get* $\tilde{U}$

Want to find the leading triplets i.e. $\{\sigma_i, \mathbf{u}^i, \mathbf{v}^i\}$ *for* $i = 1, \ldots k$ *for some* $k \leq d$

Principal Component Analysis

Suppose we wish to find the leading right singular vector of X

- *Same as finding the leading eigenvector of $X^\top X$*
- *To find the leading left singular vector of X, find leading eigenvector of $XX^\top$*

Denote $A = X^\top X = V\Lambda V^\top = \sum_{j=1}^{d} \lambda_i \cdot \mathbf{v}^i \left(\mathbf{v}^i\right)^\top$

- *Recall that $\lambda_i = \sigma_i^2$ and that we reorder things so that $\lambda_1 \geq \lambda_2 \geq \cdots$*
- *Assume for sake of simplicity that $\lambda_1 \geq \lambda_2 \cdot \Delta$ for some $\Delta > 1$*
  - This is sometimes called an *eigengap* or even *leading eigengap*
  - Easier to see the algorithms at work with an eigengap – will handle $\Delta = 1$ later
  - **Caution**: textbooks might write eigengap *additively* as $\lambda_1 \geq \lambda_2 + \epsilon$ for $\epsilon > 0$ – same thing

Note: $\mathrm{A}^2 \triangleq AA = V\Lambda V^\top V\Lambda V^\top = V\Lambda^2 V^\top$ where $\Lambda^2 = \mathrm{diag}(\lambda_1^2, \lambda_2^2, \ldots)$

- *Similarly, convince yourself that $A^s = V\Lambda^s V^\top$ for any $s \geq 0$*
- *Will use this curious fact very efficiently to find the leading eigenpair*

The Power Method

Note that if $\lambda_1 \geq \lambda_2 \cdot \Delta$, then $\lambda_1^s \geq \lambda_2^s \cdot \Delta^s$ for $s \geq 1$

Note that this means $\lambda_1^s \geq \lambda_j^s \cdot \Delta^s$ for all $j \geq 2$ as well

Even if $\Delta = 1.1$, with large enough s, gap blows up e.g. $1.1^{100} > 10000$

Thus, with large s, the leading eigenvalue really stands out!

Let us take a vector $\mathbf{x} \in \mathbb{R}^d$ and let $\boldsymbol{\alpha} = V^\top \mathbf{x}$

The vector $\boldsymbol{\alpha}$ represents $\mathbf{x}$ in terms of columns of V i.e. $\mathbf{x} = \sum_{j=1}^d \alpha_j \cdot \mathbf{v}^j$

Notice that since V is orthonormal, we have $VV^\top \mathbf{x} = \mathbf{x}$

This means $A^s\mathbf{x} = V\Lambda^s V^\top \mathbf{x} = V\Lambda^s \boldsymbol{\alpha} = \alpha_1 \lambda_1^s \cdot \mathbf{v}^1 + \sum_{j>1} \alpha_j \lambda_j^s \cdot \mathbf{v}^j$

However, we just saw that $\lambda_1^s \gg \lambda_j^s$ for all $j \geq 2$

This means that $A^s\mathbf{x} \approx \alpha_1 \lambda_1^s \cdot \mathbf{v}^1$ which means that $\hat{\mathbf{v}}^1 \triangleq \frac{A^s\mathbf{x}}{\|A^s\mathbf{x}\|_2} \approx \mathbf{v}^1$

The Power Method

Note that if $\lambda_1 \geq \lambda_2 \cdot \Delta$, then $\lambda_1^s \geq \lambda_2^s \cdot \Delta^s$ for $s \geq 1$

Note that this means $\lambda_1^s \geq \lambda_j^s \cdot \Delta^s$ for all $j \geq 2$ as well

Even if $\Delta = 1.1$, with large enough s, gap blows up e.g. $1.1^{100} > 10000$

Thus, with large s, the leading eigenvalue really stands out!

Let us take a vector $\mathbf{x} \in \mathbb{R}^d$ and let $\boldsymbol{\alpha} = V^\top \mathbf{x}$

The vector $\boldsymbol{\alpha}$ represents $\mathbf{x}$ in terms of columns of V i.e. $\mathbf{x} = \sum_{j=1}^d \alpha_j \cdot \mathbf{v}^j$

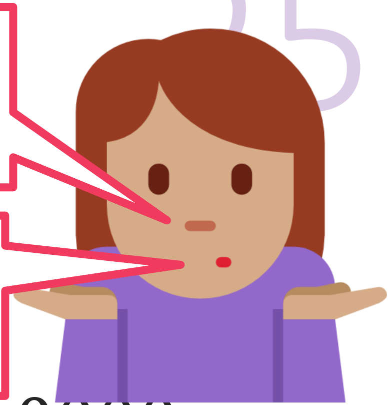
Notice that since V is orthonormal, we have $VV^\top \mathbf{x} = \mathbf{x}$

This means $A^s \mathbf{x} = V \Lambda^s V^\top \mathbf{x} = V \Lambda^s \boldsymbol{\alpha} = \alpha_1 \lambda_1^s \cdot \mathbf{v}^1 +$ $\sum_{j>1} \alpha_j \lambda_j^s \cdot \mathbf{v}^j$

However, we just saw that $\lambda_1^s \gg \lambda_j^s$ for all $j \geq 2$

This means that $A^s \mathbf{x} \approx \alpha_1 \lambda_1^s \cdot \mathbf{v}^1$ which means that $\hat{\mathbf{v}}^1 \triangleq \frac{A^s \mathbf{x}}{\|A^s \mathbf{x}\|_2} \approx \mathbf{v}^1$

The Power Method



How do I find our λ_1?

Note that if $\lambda_1 \geq$

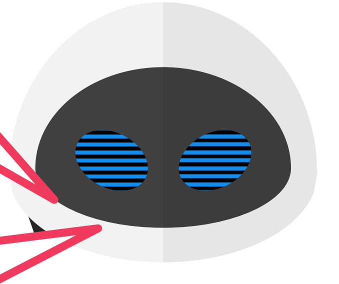
Note that this n

Hmm … this means if our approximation $\hat{\mathbf{v}}^1$ is not good then our approximation $\hat{\lambda}^1$ wont be good either

Even if $\Delta = 1.1$, *with lar* 1.1^{100} 10000

Thus, with large s*, the le*

Find $\hat{\mathbf{v}}^1$ and then use the fact that λ^1 is the eigenvalue of $X^\top X$ corresponding to $\mathbf{v}^1$ (i.e. $X^\top X \mathbf{v}^1 = \lambda^1 \mathbf{v}^1$) to get $\hat{\lambda}^1 \triangleq \|X^\top X \hat{\mathbf{v}}^1\|_2 \approx \lambda^1$



Let us take a vector $\mathbf{x} \in$

$\alpha_i \cdot$

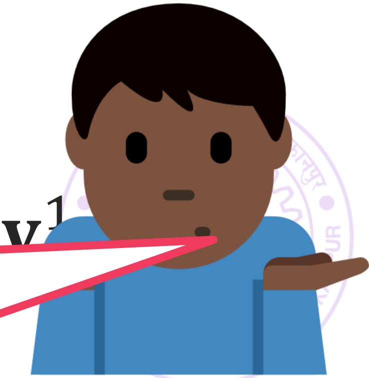
$\mathbf{x}$ must be a vector so that $\alpha_1 \neq 0$ i.e. $\mathbf{x}^\top \mathbf{v}^1 \neq 0$. If $\alpha_1 = 0$, then $\alpha_1 \cdot \lambda_1^s = 0$ as well which means we will never recover the vector $\mathbf{v}^1$. The longer you run i.e. larger the s, the better the approximation you will get.

T

However, we j

This means th

$\mathbf{v}^1$



We obtained our approximation as $\hat{\mathbf{v}}^1 \triangleq \frac{A^s \mathbf{x}}{\|A^s \mathbf{x}\|_2}$. How should we choose $\mathbf{x}$? Will any $\mathbf{x}$ work? How should we choose s?

The Power Method

THE POWER METHOD

```
1. **Input**: square symmetric matrix $A \in \mathbb{R}^{d \times d}$
2. Initialize $\mathbf{x}^0$ randomly e.g. $\sim \mathcal{N}(\mathbf{0}, I)$
3. For $t = 1,2, \ldots, s$
    1. Let $\mathbf{z}^t = A\mathbf{x}^{t-1}$
    2. Let $\mathbf{x}^t = \mathbf{z}^t$
4. Return leading eigenvector estimate as $\hat{\mathbf{v}}^1 = \mathbf{x}^s / \|\mathbf{x}^s\|_2$
5. Return leading eigenvalue estimate as $\hat{\lambda}_1 = \|A\hat{\mathbf{v}}^1\|_2$
```

In settings with no eigengap, it turns out that there is an entire subspace (i.e. infinitely many eigenvectors) corresponding to the largest eigenvalue

Power Method will return some vector in that subspace but not sure which

The Power Method

THE POWER METHOD

```
1. Input: square symmetric matrix $A \in \mathbb{R}^{d \times d}$
2. Initialize $\mathbf{x}^0$ randomly e.g. $\sim \mathcal{N}(\mathbf{0}, I)$
3. For $t = 1,2, \ldots, s$
    1. Let $\mathbf{z}^t = A\mathbf{x}^{t-1}$
    2. Let $\mathbf{x}^t = \mathbf{z}^t / \|\mathbf{z}^t\|_2$
4. Return leading eigenvector estimate as $\hat{\mathbf{v}}^1 = \mathbf{x}^s$
5. Return leading eigenvalue estimate as $\hat{\lambda}_1 = \|A\hat{\mathbf{v}}^1\|_2$
```

In settings with no eigengap, it turns out that there is an entire subspace (i.e. infinitely many eigenvectors) corresponding to the largest eigenvalue

Power Method will return some vector in that subspace but not sure which

The Power Method

```
THE POWER METHOD
1.  ... ∈ ℝ^d
2.  Initialize $\mathbf{x}^0$ randomly e.g. $\sim \mathcal{N}(\mathbf{0}, I)$
3.  For $t = 1,2,\ldots,s$
    1.  Let $\mathbf{z}^t = A\mathbf{x}^{t-1}$
    2.  Let $\mathbf{x}^t = \mathbf{z}^t/\|\mathbf{z}^t\|_2$
4.  Return leading eigenvector estimate as $\hat{\mathbf{v}}^1 = \mathbf{x}^s/\|\mathbf{x}^s\|_2$
5.  Return leading eigenvalue estimate as $\hat{\lambda}_1 = \|A\hat{\mathbf{v}}^1\|_2$
```

Calculate $A^s\mathbf{x}$ in time $\mathcal{O}(sd^2)$ using s mat-vec mults instead of first calculating A^s which takes $\mathcal{O}(sd^3)$ time

Ensures with high probability that $\langle \mathbf{x}^0, \mathbf{v}^1 \rangle \neq 0$

Good to periodically normalize to prevent overflow errors

Can show that doesn't affect the working of the algo in any way

The Power Method is fast – guaranteed to return an estimate $\|\hat{\mathbf{v}}^1 - \mathbf{v}^1\|_2 \leq \epsilon$ in at most $s = \mathcal{O}\left(d \log \frac{1}{\epsilon}\right)$ iterations (proof beyond CS771). To find smaller eigenpairs, we "peel" off largest eigenpair we have found and repeat process

The Peeling Method

```
THE PEELING METHOD
1. Input: square symmetric matrix $A \in \mathbb{R}^{d \times d}$
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \dots, k$
    1. Let $\left(\hat{\lambda}_j, \hat{\mathbf{v}}_j\right) \leftarrow \text{POWER–METHOD}\left(A^{j-1}\right)$
    2. Let $A^j \leftarrow A^{j-1} - \hat{\lambda}_j \cdot \hat{\mathbf{v}}_j \left(\hat{\mathbf{v}}_j\right)^\top$
4. Return $\left\{\hat{\lambda}_j, \hat{\mathbf{v}}_j\right\}_{j=1}^{k}$
```

The Peeling Method

THE PEELING METHOD

```
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \dots, k$
    1. Let $\left(\hat{\lambda}_j, \hat{\mathbf{v}}_j\right) \leftarrow$ POWER–METHOD$\left(A^{j-1}\right)$
    2. Let $A^j \leftarrow A^{j-1} - \hat{\lambda}_j \cdot \hat{\mathbf{v}}_j\left(\hat{\mathbf{v}}_j\right)^\top$
4. Return $\left\{\hat{\lambda}_j, \hat{\mathbf{v}}_j\right\}_{j=1}^k$
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The Peeling Method

```
THE PEELING METHOD
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \dots, k$
    1. Let $(\hat{\lambda}_j, \hat{\mathbf{v}}_j) \leftarrow$ POWER−METHOD
    2. Let $A^j \leftarrow A^{j-1} - \hat{\lambda}_j \cdot \hat{\mathbf{v}}_j (\hat{\mathbf{v}}_j)^\top$
4. Return $\{\hat{\lambda}_j, \hat{\mathbf{v}}_j\}_{j=1}^k$
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The “peeling” step

The Peeling Method

```
THE PEELING METHOD
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \ldots, k$
    1. Let $\left(\hat{\lambda}_j, \hat{\mathbf{v}}_j\right) \leftarrow$ POWER−METHOD
    2. Let $A^j \leftarrow A^{j-1} - \hat{\lambda}_j \cdot \hat{\mathbf{v}}_j \left(\hat{\mathbf{v}}_j\right)^\top$
4. Return $\left\{\hat{\lambda}_j, \hat{\mathbf{v}}_j\right\}_{j=1}^{k}$
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The “peeling” step

$$A = \lambda_1 \mathbf{v}^1 (\mathbf{v}^1)^\top + \lambda_2 \mathbf{v}^2 (\mathbf{v}^2)^\top + \lambda_3 \mathbf{v}^3 (\mathbf{v}^3)^\top + \lambda_4 \mathbf{v}^4 (\mathbf{v}^4)^\top + \cdots$$

The Peeling Method

THE PEELING METHOD

```
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \ldots, k$
    1. Let $(\hat{\lambda}_j, \hat{\mathbf{v}}_j) \leftarrow$ POWER−METHOD
    2. Let $A^j \leftarrow A^{j-1} - \hat{\lambda}_j \cdot \hat{\mathbf{v}}_j (\hat{\mathbf{v}}_j)^\top$
4. Return $\{\hat{\lambda}_j, \hat{\mathbf{v}}_j\}_{j=1}^k$
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The “peeling” step

$$A = \lambda_1 \mathbf{v}^1 (\mathbf{v}^1)^\top + \lambda_2 \mathbf{v}^2 (\mathbf{v}^2)^\top + \lambda_3 \mathbf{v}^3 (\mathbf{v}^3)^\top + \lambda_4 \mathbf{v}^4 (\mathbf{v}^4)^\top + \cdots$$

The Peeling Method

```
THE PEELING METHOD
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \dots, k$
    1. Let $(\hat{\lambda}_j, \hat{\mathbf{v}}_j) \leftarrow$ POWER−METHOD
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The “peeling” step

After leading eigenpair is peeled off, the eigenpair with the second largest eigenvalue becomes the new leading pair and Power Method can now recover this

$$A = \lambda_1 \mathbf{v}^1 (\mathbf{v}^1)^\top + \lambda_2 \mathbf{v}^2 (\mathbf{v}^2)^\top + \lambda_3 \mathbf{v}^3 (\mathbf{v}^3)^\top + \lambda_4 \mathbf{v}^4 (\mathbf{v}^4)^\top + \cdots$$

The Peeling Method

THE PEELING METHOD

```
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \dots, k$
    1. Let $(\hat{\lambda}_j, \hat{\mathbf{v}}_j) \leftarrow$ POWER−METHOD
```

Takes overall $\mathcal{O}(kd^2 s)$ time to return the top k leading eigenpairs of A

The “peeling” step

After leading eigenpair is peeled off, the eigenpair with the second largest eigenvalue becomes the new leading pair and Power Method can now recover this

$$A = \lambda_1 \mathbf{v}^1 (\mathbf{v}^1)^\top + \lambda_2 \mathbf{v}^2 (\mathbf{v}^2)^\top + \lambda_3 \mathbf{v}^3 (\mathbf{v}^3)^\top + \lambda_4 \mathbf{v}^4 (\mathbf{v}^4)^\top + \cdots$$

The Peeling Method

THE PEELING METHOD

```
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \ldots, k$
    1. Let $(\hat{\lambda}_j, \hat{\mathbf{v}}_j) \leftarrow$ POWER−METHOD
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The "peeling" step

After leading eigenpair is peeled off, the eigenpair with the second largest eigenvalue becomes the new leading pair and Power Method can now recover this

$$A = \lambda_1 \mathbf{v}^1 (\mathbf{v}^1)^\top + \lambda_2 \mathbf{v}^2 (\mathbf{v}^2)^\top + \lambda_3 \mathbf{v}^3 (\mathbf{v}^3)^\top + \lambda_4 \mathbf{v}^4 (\mathbf{v}^4)^\top + \cdots$$

The Peeling Method

THE PEELING METHOD

```
1. Input: square symme
2. Initialize $A^0 \leftarrow A$
3. For $j = 1, \ldots, k$
    1. Let $(\hat{\lambda}_j, \hat{\mathbf{v}}_j) \leftarrow$ POWER−METHOD$(A^{j-1})$
    2. $A^j \leftarrow A^{j-1} - \hat{\lambda}_j \cdot \hat{\mathbf{v}}_j (\hat{\mathbf{v}}_j)^\top$
```

Takes overall $\mathcal{O}(kd^2s)$ time to return the top k leading eigenpairs of A

The "peeling" step

After leading eigenpair is peeled off, the eigenpair with the second largest eigenvalue becomes the new leading pair and Power Method can now recover this

Some residue might still be left due to inaccurate estimation of $\lambda_j, \mathbf{v}^j$ but usually small if s sufficiently large

$$A = \lambda_1 \mathbf{v}^1 (\mathbf{v}^1)^\top + \lambda_2 \mathbf{v}^2 (\mathbf{v}^2)^\top + \lambda_3 \mathbf{v}^3 (\mathbf{v}^3)^\top + \lambda_4 \mathbf{v}^4 (\mathbf{v}^4)^\top + \cdots$$

PCA – the inside story

Recap: given a matrix $X \in \mathbb{R}^{n \times d}$ with SVD $X = U\Sigma V^\top$, find top $k \leq d$ singular triplets of the SVD

In other words, find $\hat{U} \in \mathbb{R}^{n \times k}, \hat{V} \in \mathbb{R}^{d \times k}, \hat{\Sigma} \in \mathbb{R}^{k \times k}$ such that $\hat{U} = [\mathbf{u}^1, \dots, \mathbf{u}^k], \hat{V} = [\mathbf{v}^1, \dots, \mathbf{v}^k]$ have orthonormal cols and contain the k largest singular vectors and $\hat{\Sigma}$ is diagonal and contains the largest k singular values

***Note**: this gives $\hat{X} = \hat{U}\hat{\Sigma}\hat{V}^\top$ which has rank k ($\hat{\Sigma}$ has only k non-zero entries)*

Turns out that this matrix $\hat{X}$ has many other nice properties too

Storing $\hat{X}$ requires only $(n + d + 1) \cdot k$ space (X requires nd space)

$\hat{X}$ is the best approximation to X among all rank-k matrices i.e. it is the global optimum to the following problem: $\arg \min_{Z \in \mathbb{R}^{n \times d}, \text{rank}(Z) \leq k} \|X - Z\|_F^2$

For a matrix X, the Frobenius norm $\|X\|_F$ is obtained by either stretching X into a long vector and taking its L2 norm or else taking the L2 norm of the vector formed out of the singular values of X i.e. $\|X\|_F^2 = \sum_{ij} X_{ij}^2 = \sum_i \sigma_i^2$

PCA – the inside story

Recap: given a matrix $X \in \mathbb{R}^{n \times d}$ with SVD $X = U\Sigma V^\top$, find top $k \leq d$ singular triplets of t

Rows of $\hat{U}, \hat{V}$ not necessarily orthonormal

In other words, find $\in \mathbb{R}^{k \times k}$ *such that* $\hat{U} = [\mathbf{u}^1, \ldots, \mathbf{u}^k], \hat{V} = [\mathbf{v}^1, \ldots, \mathbf{v}^k]$ *have orthonormal cols and contain the* k *largest singular vectors and* $\hat{\Sigma}$ *is diagonal and contains the largest* k *singular values*

Note*: this gives* $\hat{X} = \hat{U}\hat{\Sigma}\hat{V}^\top$ *which has rank* k *(*$\hat{\Sigma}$ *has only* k *non-zero entries)*

Turns out that this matrix $\hat{X}$ *has many other nice properties too*

Storing $\hat{X}$ *requires only* $(n + d + 1) \cdot k$ *space (*X *requires* nd *space)*

$\hat{X}$ *is the best approximation to* X *among all rank-*k *matrices i.e. it is the global optimum to the following problem:* $\arg \min_{Z \in \mathbb{R}^{n \times d}, \text{rank}(Z) \leq k} \|X - Z\|_F^2$

For a matrix X*, the Frobenius norm* $\|X\|_F$ *is obtained by either stretching* X *into a long vector and taking its L2 norm or else taking the L2 norm of the vector formed out of the singular values of* X *i.e.* $\|X\|_F^2 = \sum_{ij} X_{ij}^2 = \sum_i \sigma_i^2$

Low dimensional Modelling

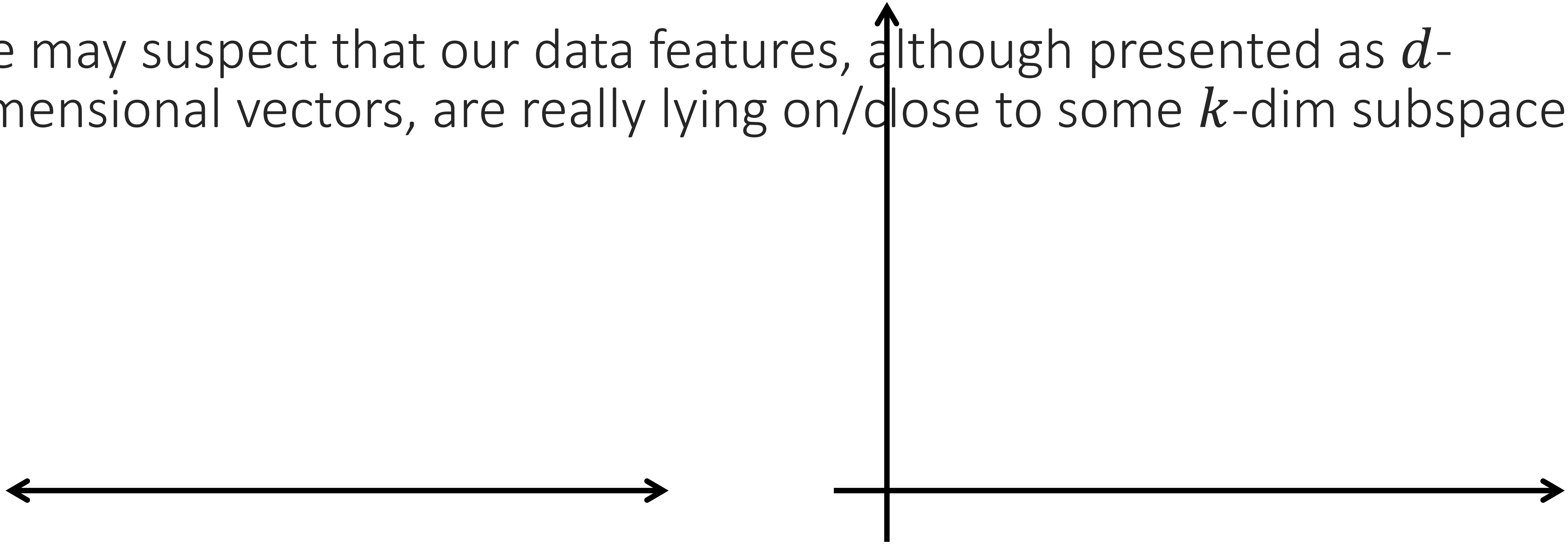
We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

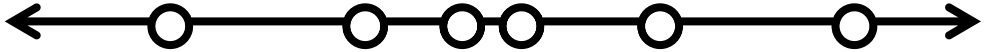
Low dimensional Modelling

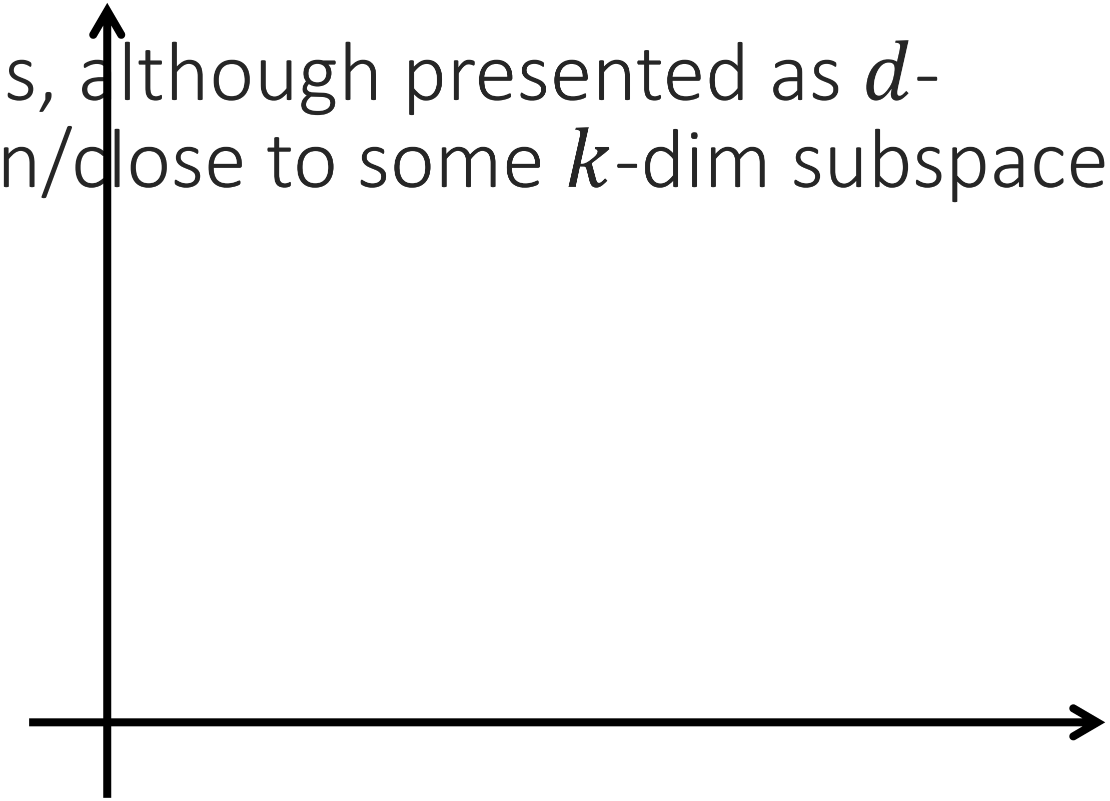
We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace



Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

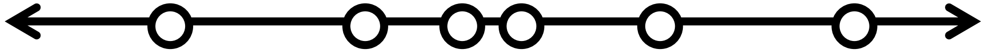


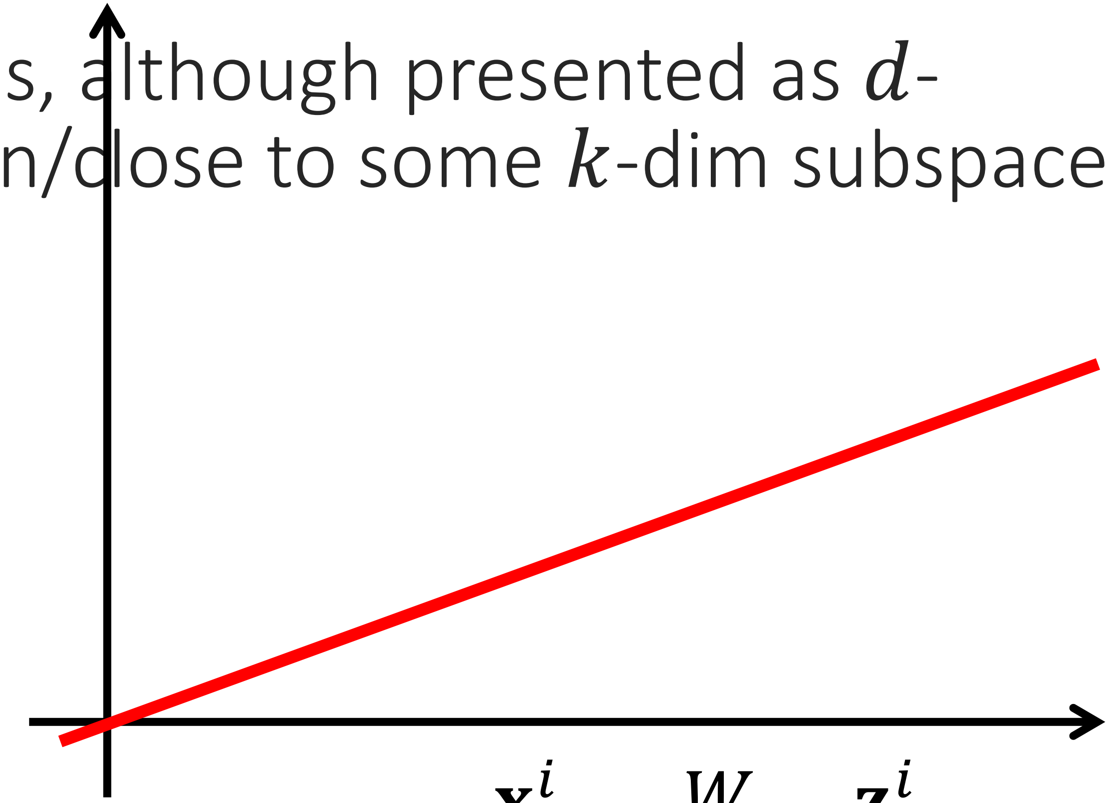


$\mathbf{z}^i \in \mathbb{R}^k$

Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

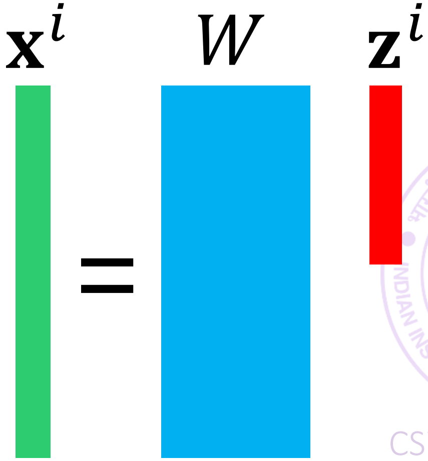




$$\mathbf{z}^i \in \mathbb{R}^k$$

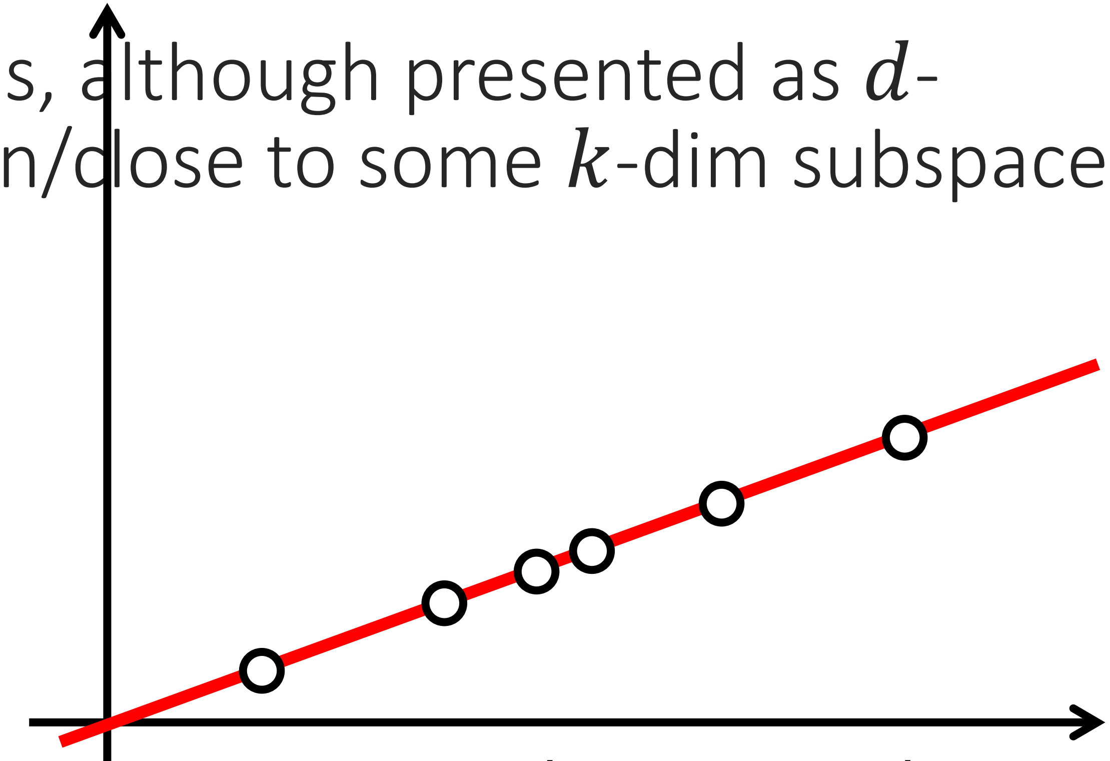
$$\mathbf{x}^i = W\mathbf{z}^i$$

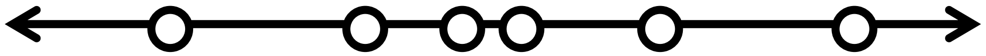
$$W \in \mathbb{R}^{d \times k}$$

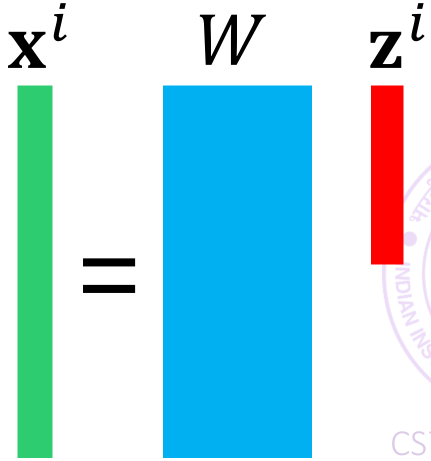


Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace







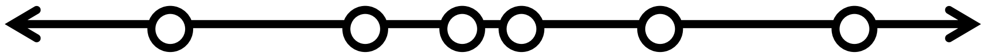
$\mathbf{z}^i \in \mathbb{R}^k$

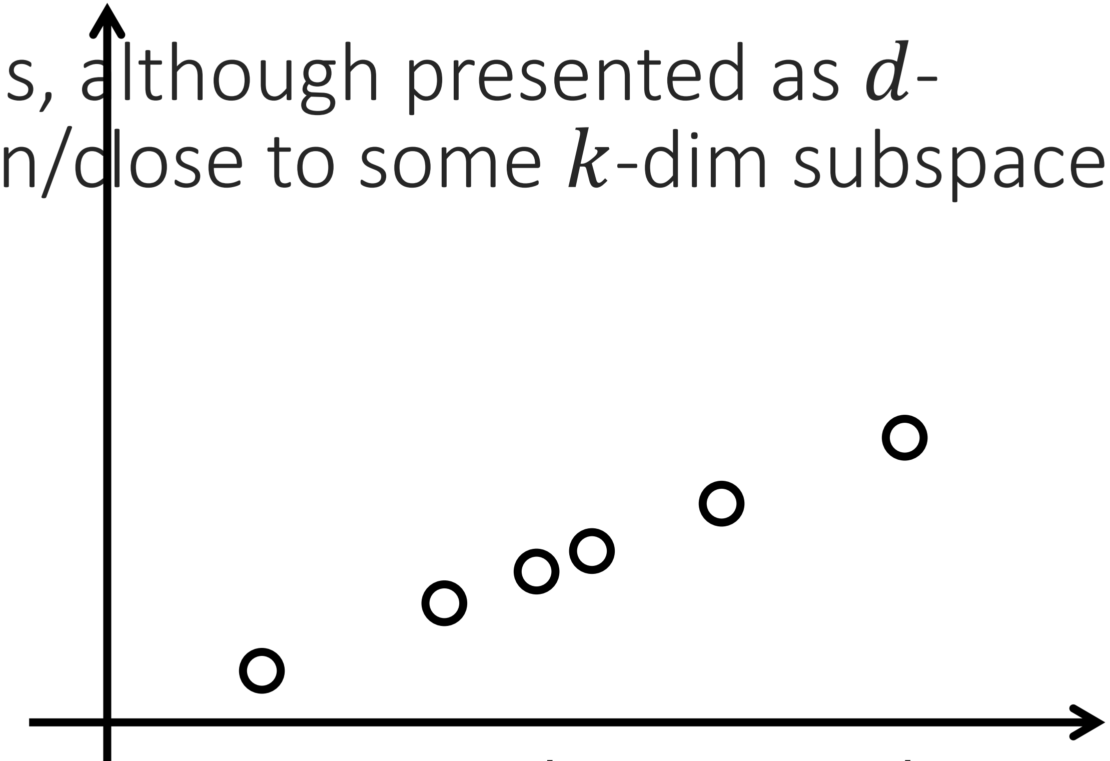
$$\mathbf{x}^i = W\mathbf{z}^i$$

$$W \in \mathbb{R}^{d\times k}$$

Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

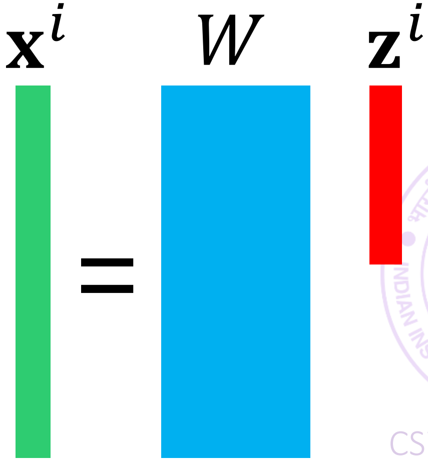




$$\mathbf{z}^i \in \mathbb{R}^k$$

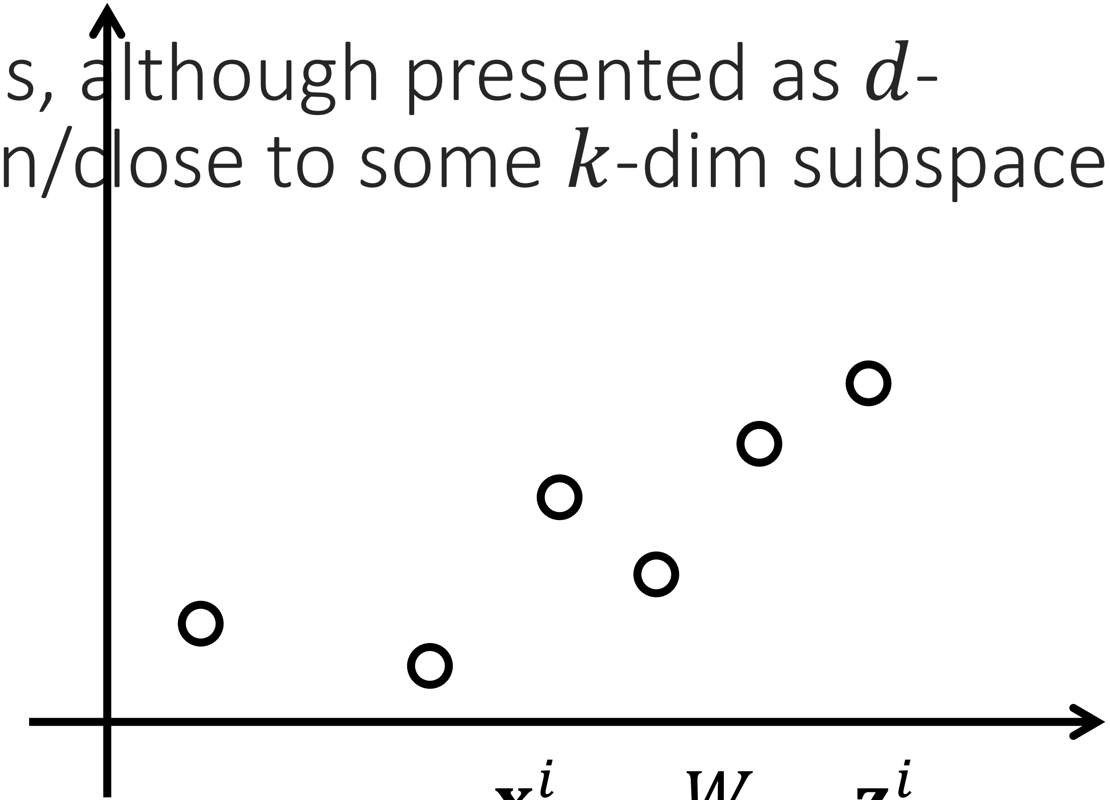
$$\mathbf{x}^i = W\mathbf{z}^i$$

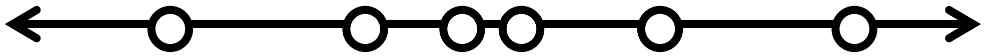
$$W \in \mathbb{R}^{d\times k}$$



Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

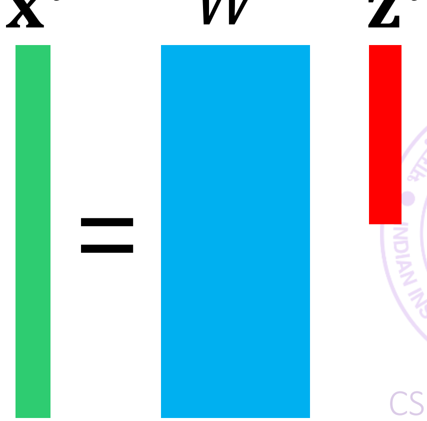




$$\mathbf{z}^i \in \mathbb{R}^k$$

$$\mathbf{x}^i = W\mathbf{z}^i + \boldsymbol{\epsilon}^i$$

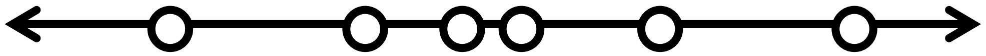
$$W \in \mathbb{R}^{d \times k}$$

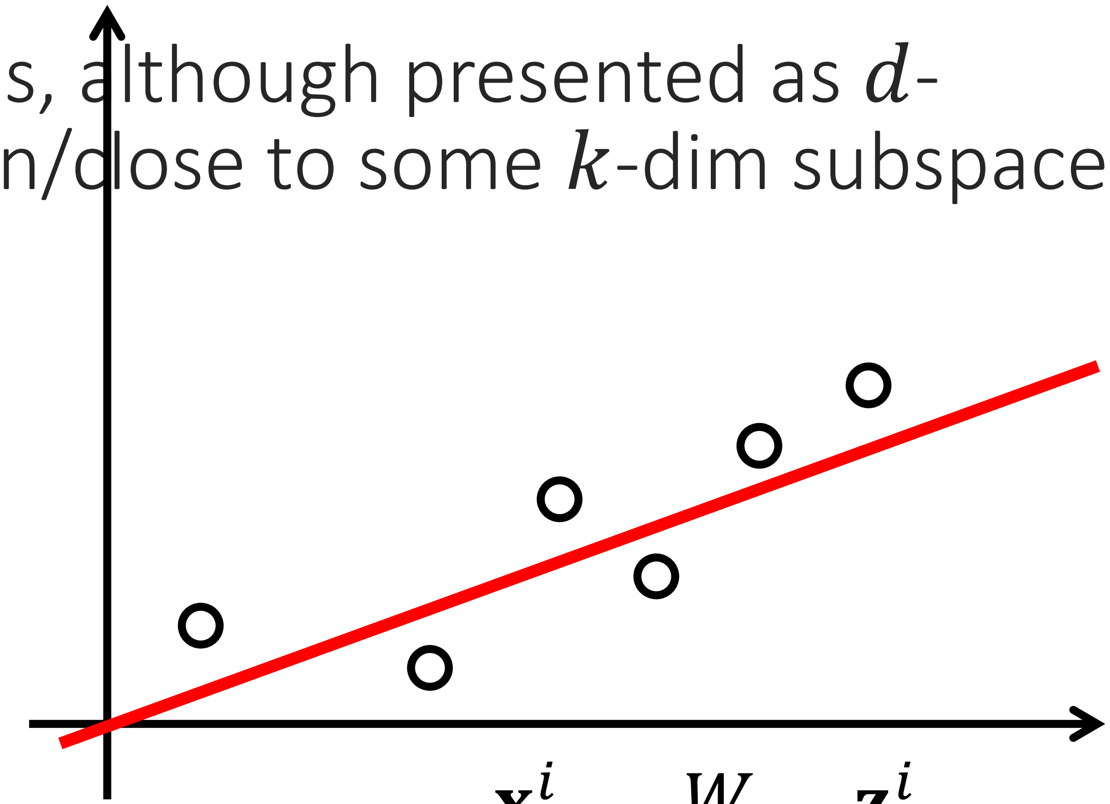


$\mathbf{x}^i$ = W $\mathbf{z}^i$

Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vectors, are really lying on/close to some k-dim subspace

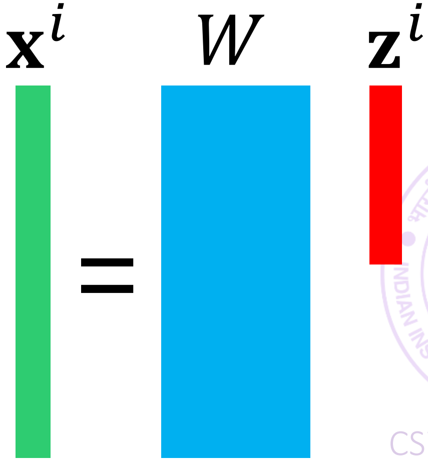




$\mathbf{z}^i \in \mathbb{R}^k$

$$\mathbf{x}^i = W\mathbf{z}^i + \boldsymbol{\epsilon}^i$$

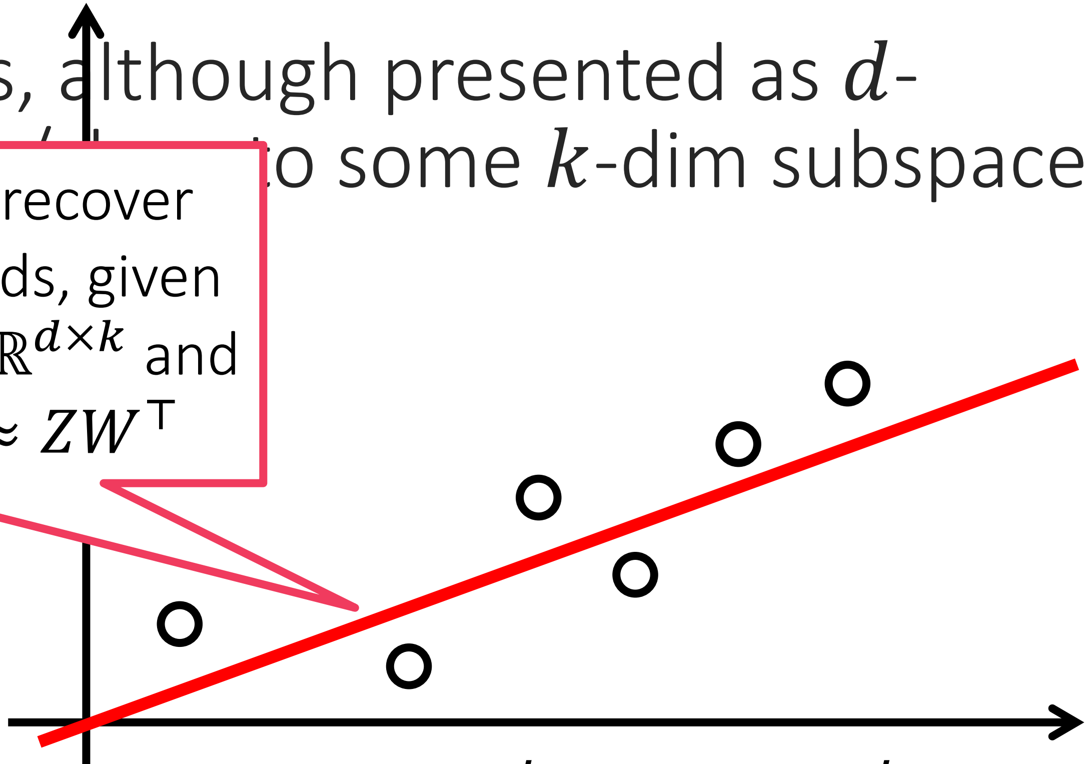
$$W \in \mathbb{R}^{d \times k}$$

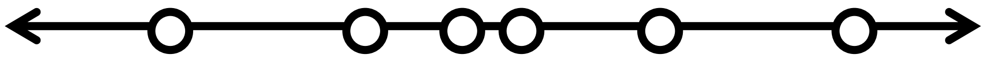


Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vecto o some k-dim subspace

Given $\mathbf{x}^1, \ldots, \mathbf{x}^n$, can we recover $W, \mathbf{z}^1, \ldots \mathbf{z}^n$? In other words, given $X \in \mathbb{R}^{n\times d}$, recover $W \in \mathbb{R}^{d\times k}$ and $Z \in \mathbb{R}^{n\times k}$ such that $X \approx ZW^\top$

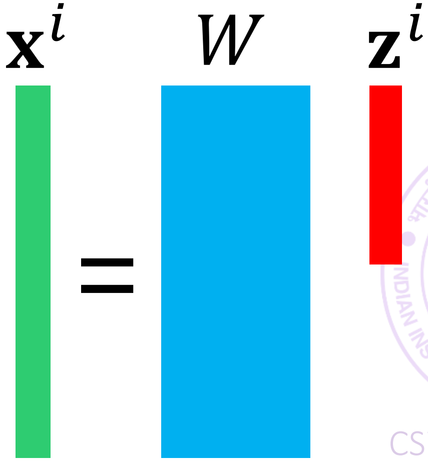




$$\mathbf{z}^i \in \mathbb{R}^k$$

$$\mathbf{x}^i = W\mathbf{z}^i + \boldsymbol{\epsilon}^i$$

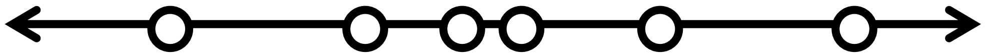
$$W \in \mathbb{R}^{d\times k}$$

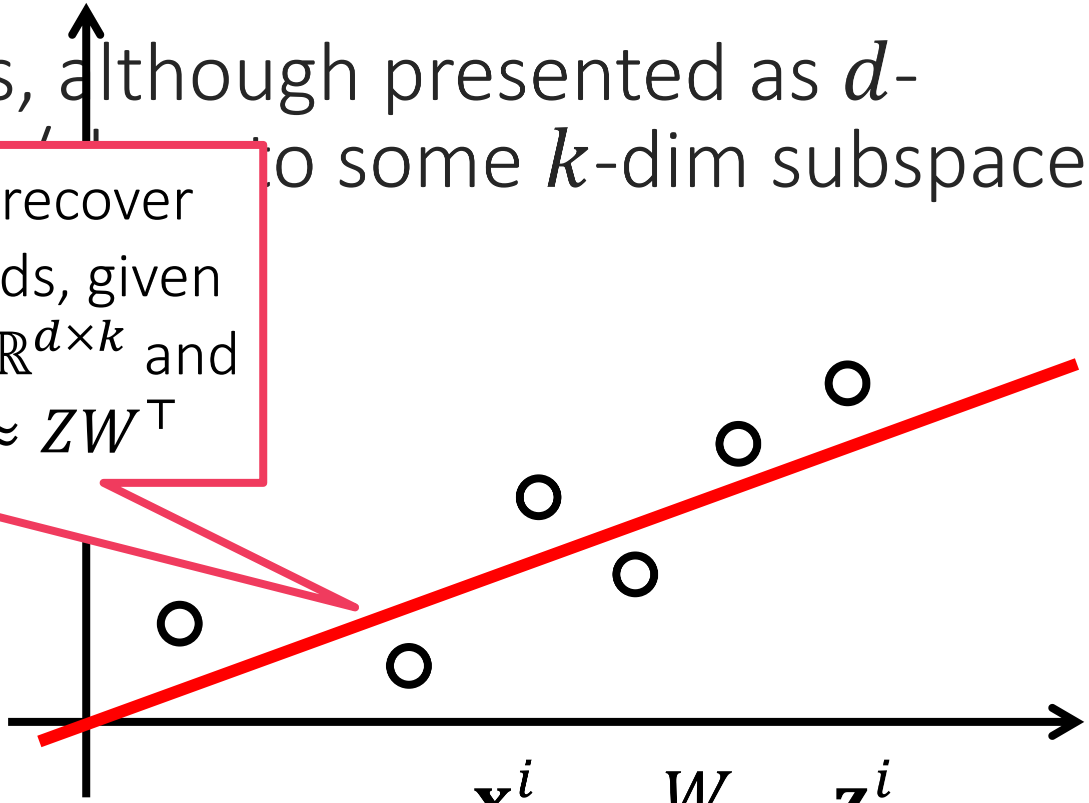


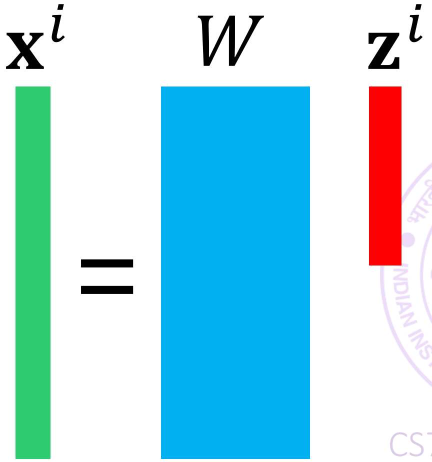
Low dimensional Modelling

We may suspect that our data features, although presented as d-dimensional vecto … o some k-dim subspace

Given $\mathbf{x}^1, \dots, \mathbf{x}^n$, can we recover $W, \mathbf{z}^1, \dots \mathbf{z}^n$? In other words, given $X \in \mathbb{R}^{n \times d}$, recover $W \in \mathbb{R}^{d \times k}$ and $Z \in \mathbb{R}^{n \times k}$ such that $X \approx ZW^\top$







$\mathbf{z}^i \in \mathbb{R}^k$

$$\mathbf{x}^i = W\mathbf{z}^i + \boldsymbol{\epsilon}^i$$

Dictionary/Factor Loading matrix

$W \in \mathbb{R}^{d \times k}$

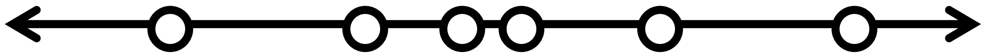
Low

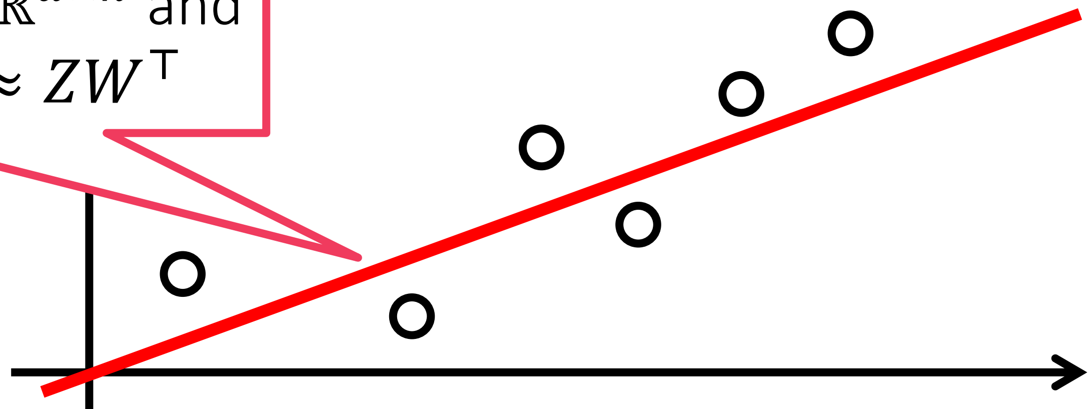
PCA can help you solve this problem. Find $\hat{U}, \hat{V}, \hat{\Sigma}$ and set $Z = \hat{U}\,\hat{\Sigma}$ and $W = \hat{V}$. As noted earlier, it will give us the best possible approximation any W, Z with only k columns could have given

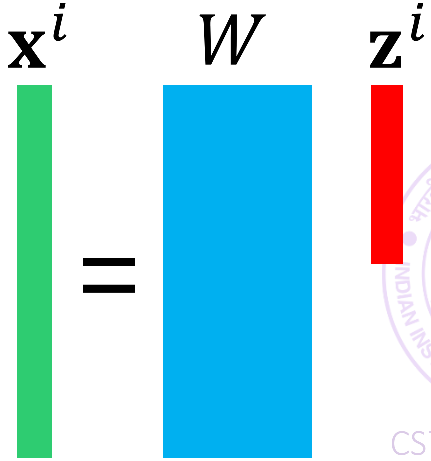
We ma

dimensional vecto … o some k-dim subspace

Given $\mathbf{x}^1, \dots, \mathbf{x}^n$, can we recover $W, \mathbf{z}^1, \dots \mathbf{z}^n$? In other words, given $X \in \mathbb{R}^{n \times d}$, recover $W \in \mathbb{R}^{d \times k}$ and $Z \in \mathbb{R}^{n \times k}$ such that $X \approx ZW^\top$







$$\mathbf{z}^i \in \mathbb{R}^k$$

$$\mathbf{x}^i = W\mathbf{z}^i + \boldsymbol{\epsilon}^i$$

Dictionary/Factor Loading matrix

$$W \in \mathbb{R}^{d \times k}$$

PCA vs Regression

The previous setup may seem like regression where “labels” are vectors and model is a matrix instead of a vector

Linear Regression

- $\mathbf{z}^i \in \mathbb{R}^k$,
- $y^i = \langle \mathbf{w}, \mathbf{z}^i \rangle + \epsilon^i$
- $\mathbf{w} \in \mathbb{R}^k$
- $\epsilon^i \in \mathbb{R}$

- Observed data $(\mathbf{z}^i, y^i) \in \mathbb{R}^k \times \mathbb{R}$

Low-rank Modelling

- $\mathbf{z}^i \in \mathbb{R}^k$,
- $\mathbf{x}^i = W\mathbf{z}^i + \boldsymbol{\epsilon}^i$
- $W \in \mathbb{R}^{d \times k}$
- $\boldsymbol{\epsilon}^i \in \mathbb{R}^d$

- Observed data $\mathbf{x}^i \in \mathbb{R}^d$

The most important difference is that in linear regression, “features” $\mathbf{z}^i$ are visible, in low-rank modelling setting, they are absent (latent)

In fact, latent variable modelling (AltOpt, EM) can indeed be used too

Shortcomings of PCA

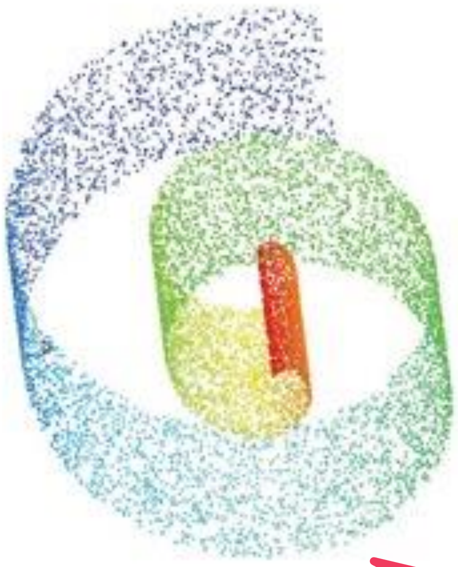
PCA will reveal hidden structure within data if that hidden structure is a linear subspace

PCA fails to reveal hidden structure in data if data is lying on curved (hyper) surfaces

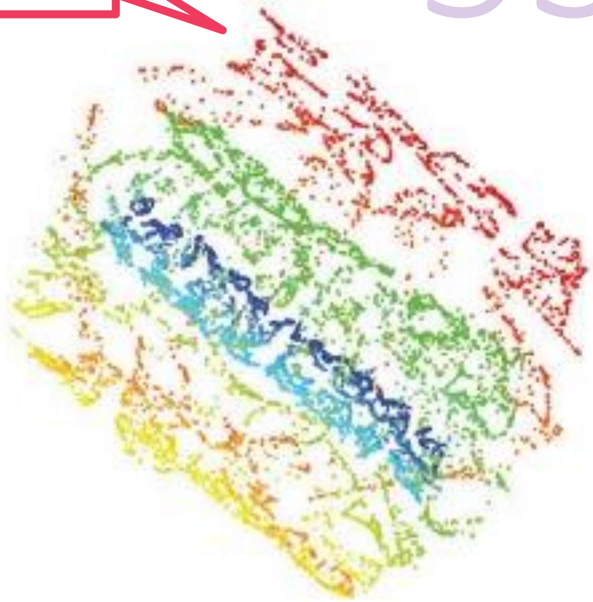
PCA may also fail if data is lying on an affine subspace

However, this can be easily overcome by mean centering data i.e. find $\boldsymbol{\mu} = \frac{1}{n}X^\top\mathbf{1}$ *and do PCA with* $\tilde{X} = X - \mathbf{1}\boldsymbol{\mu}^\top$

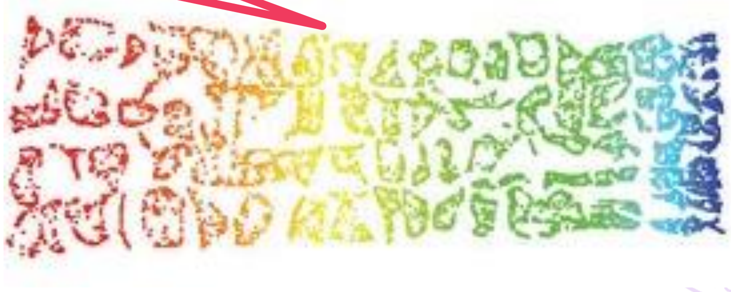
Mean centering removes displacement



"Swiss Roll" data – used to be very popular in ML



What PCA will give us



What we really want