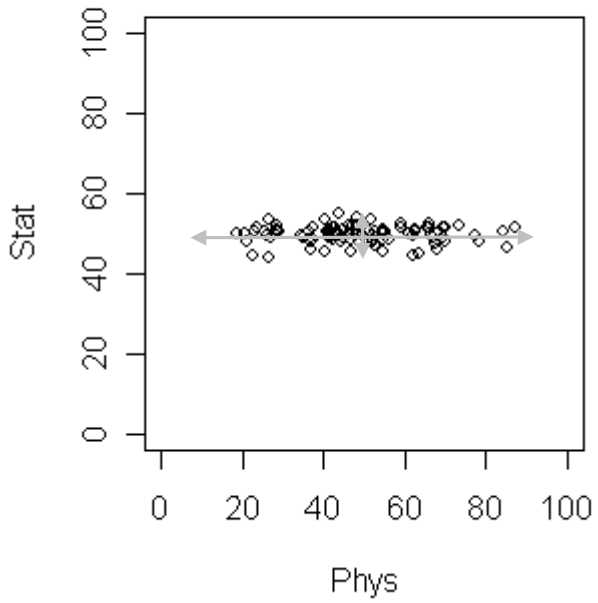
Big Data Analytics

Winter Term 2016/2017

Efficient Principal Compenent Analysis (PCA)

Principal Component Analysis (PCA): A simple example 1/3

- Consider the grades of students in Physics and Statistics.
- If we want to compare among the students, which grade should be more discriminative? Statistics or Physics?

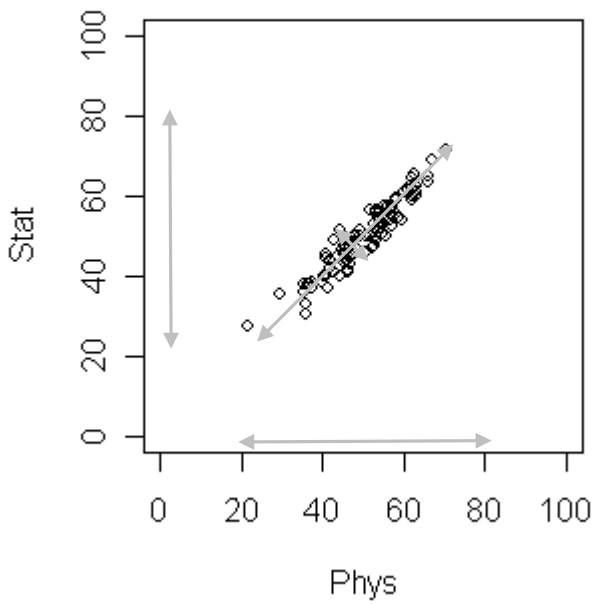


Physics since the variation along that axis is larger.

Based on:
http://astrostatistics.psu.edu/su09/lecturenotes/pca.html

Principal Component Analysis (PCA): A simple example 2/3

- Suppose now the plot looks as below.
- What is the best way to compare students now?



We should take linear combination of the two grades to get the best results.

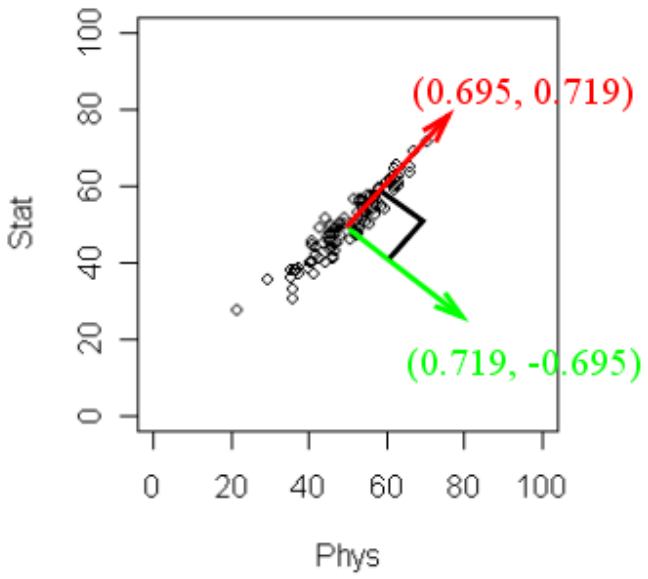
Here the direction of maximum variance is clear.

In general → PCA

Based on:
http://astrostatistics.psu.edu/su09/lecturenotes/pca.html

Principal Component Analysis (PCA): A simple example 3/3

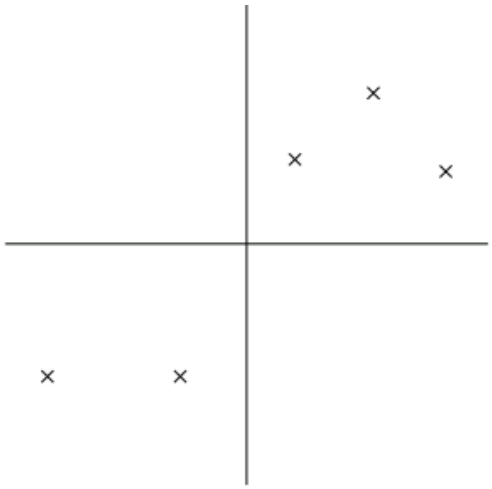
- PCA returns two principal components
  - The first gives the direction of the maximum spread of the data.
  - The second gives the direction of maximum spread perpendicular to the first



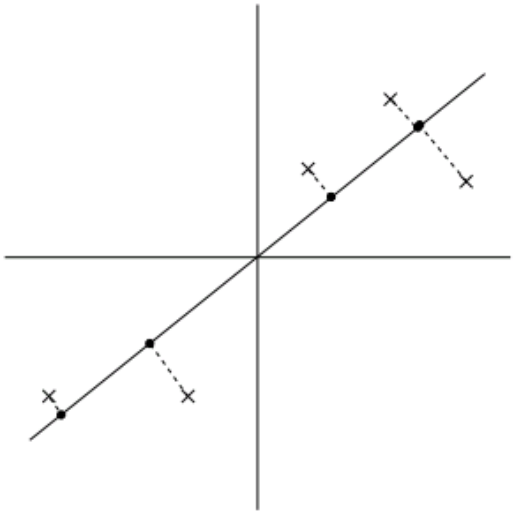
Based on:
http://astrostatistics.psu.edu/su09/lecturenotes/pca.html

Intuition

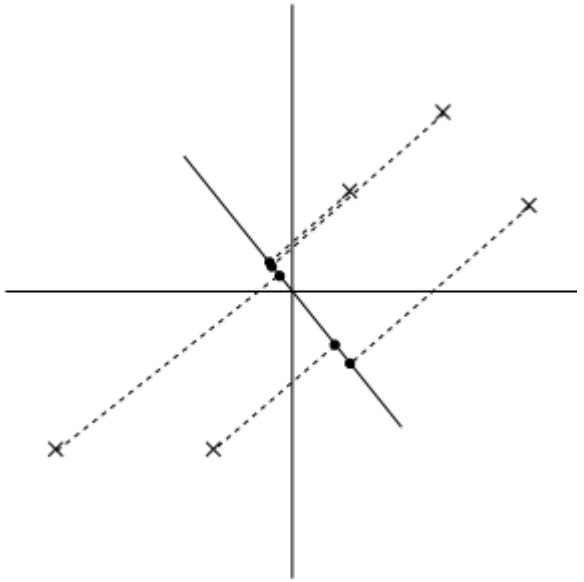
- The data starts off with some amount of variance/information in it. We would like to choose a direction *u* so that if we were to approximate the data as lying in the direction/subspace corresponding to *u*, as much as possible of this variance is still retained.



Initial data



Direction 1



Direction 2

Idea: Choose the direction that maximizes the variance of the projected data

Principal Component Analysis (PCA)

- PCA computes the most meaningful basis to re-express a noisy, garbled data set.
- Think of PCA as choosing a new coordinate system for the data, the principal components being the unit vectors along the axes
- PCA asks: *Is there another basis, which is a linear combination of the original basis, that best expresses our dataset?*
- General form: $PX=Y$

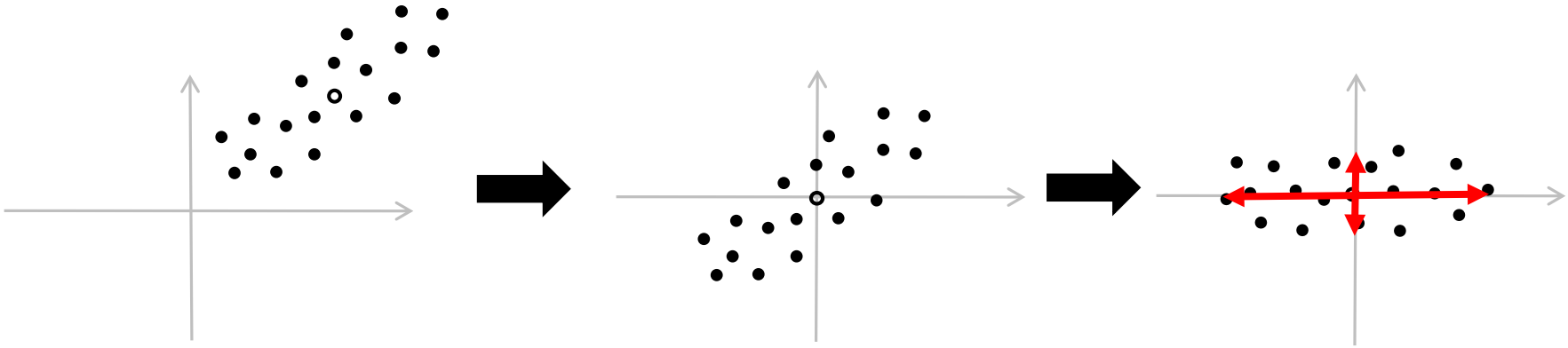
where P is a linear transformation, X is the original dataset and Y the re-representation of this dataset.

  - P is a matrix that transforms X into Y
  - Geometrically, P is a *rotation* and a *stretch* which again transforms X into Y
  - The eigenvectors are the rotations to the new axes
  - The eigenvalues are the amount of stretching that needs to be done
- The *p*'s are the principal components
  - Directions with the largest variance … those are the most important, most *principal*.

Principal Component Analysis (PCA)

Idea: Rotate the data space in a way that the principal components are placed along the main axis of the data space

=> Variance analysis based on principal components



- Rotate the data space in a way that the direction with the largest variance is placed on an axis of the data space
- Rotation is equivalent to a basis transformation by an orthonormal basis
  - Mapping is equal of angle and preserves distances:

$$x \cdot B = x\left(b_{*,1}, \ldots, b_{*,d}\right) = \left(\left\langle x, b_{*,1}\right\rangle, \ldots, \left\langle x, b_{*,d}\right\rangle\right) \quad mit \quad \forall_{i \neq j} \left\langle b_i, b_j \right\rangle = 0 \wedge \forall_{1 \leq i \leq d} \left\| b_i \right\| = 1$$

- B is built from the largest variant direction which is orthogonal to all previously selected vectors in B.

What do we need to know for PCA

- Basics of statistical measures:
  - variance
  - covariance
- Basics of linear algebra:
  - Matrices
  - Vector space
  - Basis
  - Eigenvectors, eigenvalues

Variance

- A measure of the spread of the data

$$VAR(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

- Variance refers to a single dimension, e.g., height

Covariance

- A measure of how much two random variables vary together

$$COV(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$$

- What the values mean
  - Positive values: both dimensions move together (increase or decrease)
  - Negative values: while one dimension increases the other decreases
  - Zero value: the dimensions are independent of each other.

Covariance matrix

- Describes the variance of all features and the pairwise correlations between them

$$\Sigma_D = \begin{pmatrix} VAR(X_1) & \cdots & COV(X_1, X_d) \\ \vdots & \ddots & \vdots \\ COV(X_d, X_1) & \cdots & VAR(X_d) \end{pmatrix}$$

$$VAR(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \qquad COV(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$$

- Properties:
  - For *d*-dimensional data, *d*x*d* covariance matrix
  - symmetric matrix as COV(X,Y)=COV(Y,X)

Data matrix

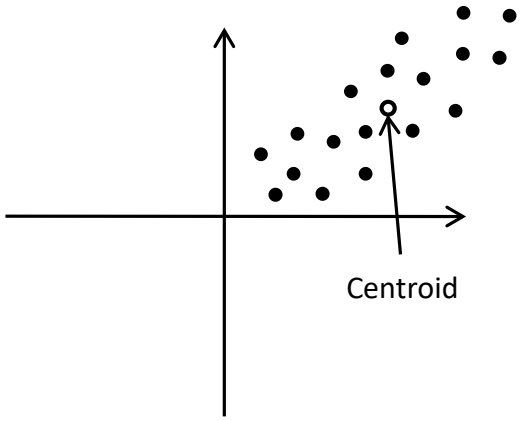
- Given n vectors $v_i \in IR^d$, $n \times d$ matrix

$$D = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} v_{1,1} & \cdots & v_{1,d} \\ \vdots & \ddots & \vdots \\ v_{n,1} & \cdots & v_{n,d} \end{pmatrix}$$

is called data matrix

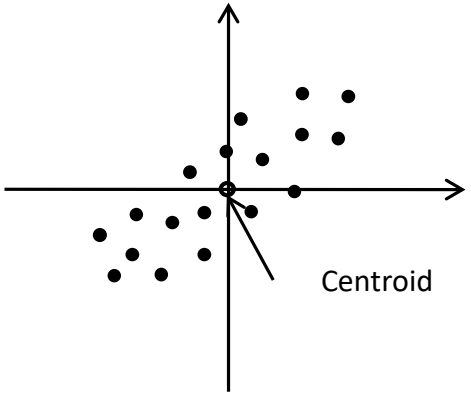
- Centroid/mean vector of D:

$$\vec{\mu} = \frac{1}{n} \cdot \sum_{i=1}^{n} v_i$$



- Centered data matrix:

$$D_{cent} = \begin{pmatrix} v_1 - \vec{\mu} \\ \vdots \\ v_d - \vec{\mu} \end{pmatrix}$$



Covariance matrix and centered data matrix

- The covariance matrix can be expressed in terms of the centered data matrix as follows:

$$\Sigma_D = \begin{pmatrix} VAR(X_1) & \cdots & COV(X_1, X_d) \\ \vdots & \ddots & \vdots \\ COV(X_d, X_1) & \cdots & VAR(X_d) \end{pmatrix} = \frac{1}{n} D_{cent}^T D_{cent}$$

Vector/ Matrix basics

- Inner (dot) product of vectors x, y:

$$x \cdot y = x^T \cdot y = \begin{pmatrix} x_1 & \cdots & x_d \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_d \end{pmatrix} = \langle x, y \rangle = \sum_{i=1}^{d} x_i \cdot y_i$$

- Outer product of vectors x, y:

$$x \otimes y = x \cdot y^T = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} \cdot \begin{pmatrix} y_1 & \cdots & y_d \end{pmatrix} = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_d \\ \vdots & \ddots & \vdots \\ x_d y_1 & \cdots & x_d y_d \end{pmatrix}$$

- Matrix multiplication:

$$A = [a_{ij}]_{m \times p};\ B = [b_{ij}]_{p \times n};$$

$$AB = C = [c_{ij}]_{m \times n}, where\ c_{ij} = row_i(A) \cdot col_j(B)$$

- Length of a vector
  - Unit vector: if $||a||=1$

$$\|a\| = \sqrt{a^T \cdot a} = \sqrt{\sum_{i=1}^{n} a_i^2}$$

Eigenvectors and eigenvalues

- Let D be d×d square matrix.
- A non zero vector v_i is called an *eigenvector* of *D* if and only if there exists a scalar λ_i such that: $Dv_i=\lambda_i v_i$.
  - λ_i is called an *eigenvalue* of *D*.

- How to find the eigenvalues/eigenvectors of D?
  - By solving the equation: $\det(D-\lambda I_{dxd})=0$ we get the eigenvalues
    - I_{dxd} is the identity matrix
  - For each eigenvalue λ_i, we find its eigenvector by solving $(D-\lambda_i)v_i=0$

Eigenvectors decomposition

- Let D be *dxd* square matrix.
- Eigenvalue decomposition of the data matrix

$$D = V \Lambda V^T$$

$$V = (v_1, \cdots, v_d) \quad mit \underset{i \neq j}{\forall} \langle v_i, v_j \rangle = 0 \quad und \overset{d}{\underset{i=1}{\forall}} \|v_i\| = 1$$

$$\Lambda = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{pmatrix}$$

Every eigenvector is a unit vector

The eigenvectors are linearly independent

The corresponding eigenvalues

- The columns of V are the eigenvectors of D
- The diagonal elements of Λ are the eigenvalues of D

PCA steps

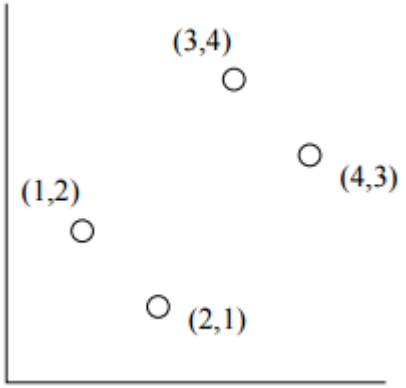
Feature reduction using PCA

1. Compute the covariance matrix Σ
2. Compute the eigenvalues and the corresponding eigenvectors of Σ
3. Select the *k* biggest eigenvalues and their eigenvectors (V‘)
4. The *k* selected eigenvectors represent an orthogonal basis
5. Transform the original $n \times d$ data matrix D with the $d \times k$ basis V‘:

$$D \cdot \mathrm{V}' = \begin{pmatrix} \mathrm{x}_1 \\ \vdots \\ \mathrm{x}_n \end{pmatrix} \left(v'_1, \cdots, v'_k\right) = \begin{pmatrix} \langle \mathrm{x}_1, v'_1 \rangle & \cdots & \langle \mathrm{x}_1, v'_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathrm{x}_n, v'_1 \rangle & \cdots & \langle \mathrm{x}_n, v'_k \rangle \end{pmatrix}$$

Example of transformation

- Original



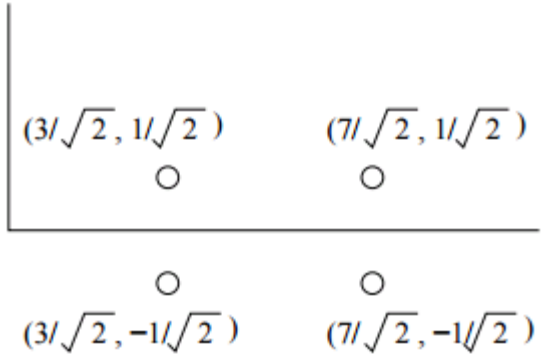
Eigenvectors

$$\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} \qquad \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

- Transformed data

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & -1/\sqrt{2} \\ 7/\sqrt{2} & 1/\sqrt{2} \\ 7/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

In the rotated coordinate system



Source: http://infolab.stanford.edu/~ullman/mmds/ch11.pdf

Percentage of variance explained by PCA

- Let *k* be the number of top eigenvalues out of *d* (*d* is the number of dimensions in our dataset)
- The percentage of variance in the dataset explained by the *k* selected eigenvalues is:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

- Similarly, you can find the variance explained by each principal component
- Rule of thumb: keep enough to explain **85%** of the variation

PCA results interpretation

- Example: iris dataset (d=4), results from R
- 4 principal components

```
                    PC1         PC2        PC3         PC4
Sepal.Length  0.5038236 -0.45499872  0.7088547  0.19147575
Sepal.Width  -0.3023682 -0.88914419 -0.3311628 -0.09125405
Petal.Length  0.5767881 -0.03378802 -0.2192793 -0.78618732
Petal.Width   0.5674952 -0.03545628 -0.5829003  0.58044745
```

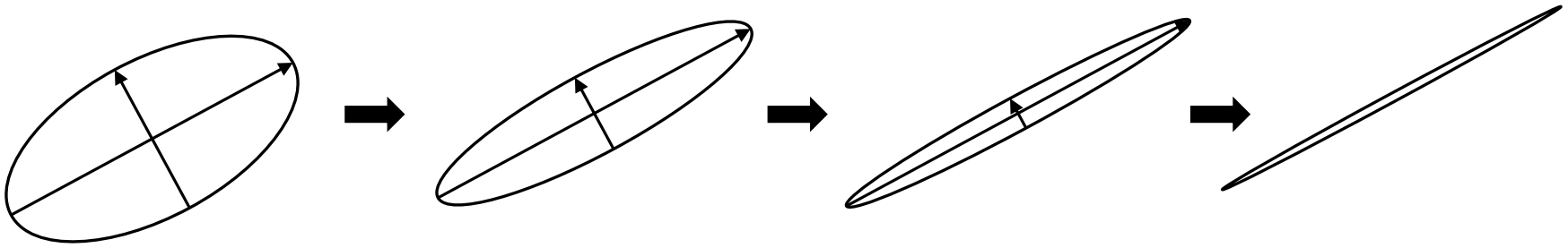
```
Importance of components:
                          PC1    PC2     PC3     PC4
Proportion of Variance 0.7331 0.2268 0.03325 0.00686
Cumulative Proportion  0.7331 0.9599 0.99314 1.00000
```

Computing PCA via Power Iteration

Problem:

- Computing the eigenvalues with standard algorithms is often expensive (many algorithm are well-known)
- Standard methods often in involve matric inversions ($O(n^3)$)
- For large matrices more efficient methods are required:
- Most prominent is the power iterations method ($O(n^2)$)

Intuition: Multiplying a matrix with itself increases the strongest direction relative to the other direction.



Power Iterations general idea

- given: data n×d matrix X and the corresponding covariance matrix Σ=(X-μ(X)^T(X-μ(X)) where μ(X) is the mean vector of X.
- consider the eigenvalue decomposition of $\Sigma = V^T \Lambda V$ where
  $V = (v_1, .., v_d)$: is the columnwise orthonormal eigenvector basis
  $\Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{bmatrix}$: is the diagonal eigenvalue matrix

Note: $\Sigma^t = (V^T \Lambda V)^t = V^T \Lambda V \cdot V^T \Lambda V \cdot \ldots \cdot V^T \Lambda V = V^T \Lambda^t V$

$$= V^T \begin{bmatrix} \lambda_1^t & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d^t \end{bmatrix} V$$

What is the i^{th} power of a diagonal matrix ?

- if PCA is well-defined all $\lambda >= 0$
- taking the i^{th} power: All values $\lambda > 1$ increase with the power and all λ values < 0 decrease exponentially fast.
- When normalizing the λ by $\sum_{i=1}^{d} \lambda_i$, we observe the following:

  for $\lambda_i \neq \lambda_j$ and $t \to \infty : \exists \lambda_{i*}: \frac{\lambda_{i*}^t}{\sum_{i=1}^{d} \lambda_i^t} \to 1 \; and \; \forall j \neq i*: \frac{\lambda_j^t}{\sum_{i=1}^{d} \lambda_i^t} \to 0$
- under normalization over all diagonal entries, only one component remains.
- Thus: the rank of Σ^t converges to 1 and the only component remaining is the strongest eigenvector.

Determining the Strongest Eigenvalue

The following algorithm computes the strongest eigenvalue of matrix M:

```
Input: d×d data matrix M
x_0 = random unit vector
while x_i/||x_i||- x_{i-1}/||x_{i-1}||> ε do
        x_i = M^i x_0
        i=i+1
return x_i/||x_i||
```

Why does this work?

$$M^t x = [v_1, .., v_d] \begin{bmatrix} 0 & .. & 0 \\ .. & \lambda_j^t & .. \\ 0 & .. & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ : \\ v_d \end{bmatrix} x = [v_1, .., v_d] \begin{bmatrix} 0 & .. & 0 \\ .. & \lambda_j^t & .. \\ 0 & .. & 0 \end{bmatrix} \begin{bmatrix} \langle v_1, x \rangle \\ : \\ \langle v_d, x \rangle \end{bmatrix}$$

$$= [v_1, .., v_d] \begin{bmatrix} 0 \\ \lambda_j^t \langle v_j, x \rangle \\ 0 \end{bmatrix} = \begin{bmatrix} v_{1,1} \cdot 0 + .. + v_{1,j} \cdot \lambda_j^t \langle v_j, x \rangle + v_{1,d} \cdot 0 \\ : \\ v_{d,1} \cdot 0 + .. + v_{d,j} \cdot \lambda_j^t \langle v_j, x \rangle + v_{d,d} \cdot 0 \end{bmatrix} = v_j \cdot \lambda_j^t \langle v_j, x \rangle$$

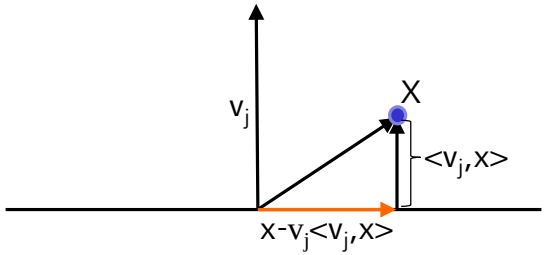
in other words the $M^T x$ has the same direction as the strongest eigenvector v_j.

Power Iterations: the complete method

- we now have a method to determine the strongest eigenvalue
- to compute the k-strongest eigenvalues we proceed as follows:

```
For i=1 to k:
  determine the strongest eigenvalue v_i
  reproject data X to the space being orthogonal to v_i: x'
  = x-v_i<v_i,x>
output the v_i
```

- explanation for the reprojection:



- if there are two equally strong eigenvalues $\lambda_i = \lambda_j$ then the algorithm return an arbitrary vector from *span*(v_i, v_j)
- for $\lambda_i \approx \lambda_j$: the algorithm converges slower

Conclusion

- PCA is an important method for feature reduction
- general and complete for eigenvalue decomposition are often inefficient(compute the characteristic polynomial, using matrix inversion etc.)
- Power iterations are linear in the size of the matrix, i.e. quadratic in the dimension d.
- Power iterations compute only the k strongest eigenvalues but not all (stop when k strongest v are found)
- rely only on matrix multiplications