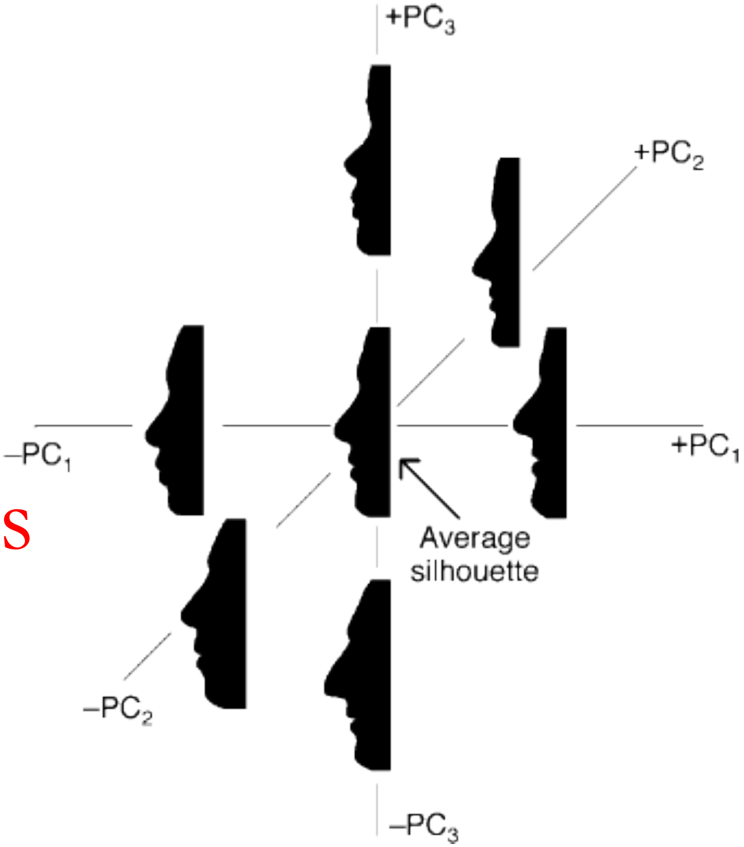
A Tutorial on Data Reduction

Principal Component Analysis Theoretical Discussion



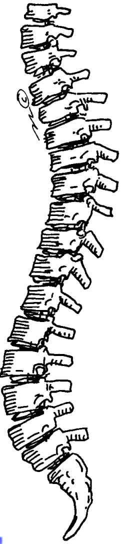
By

Shireen Elhabian and Aly Farag

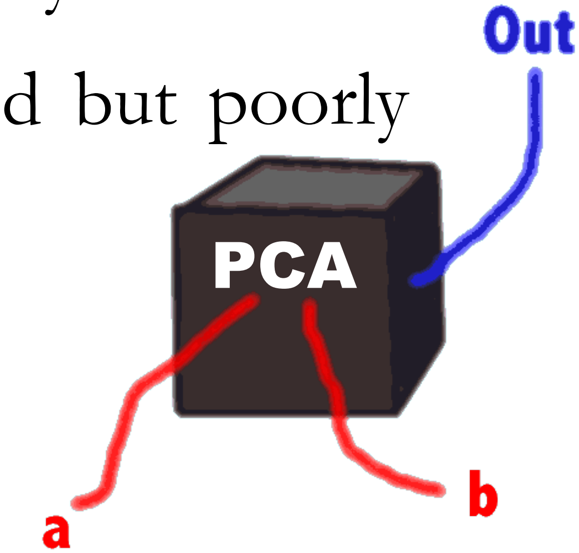
University of Louisville, CVIP Lab

November 2008

PCA is …



- A backbone of modern data analysis.
- A black box that is widely used but poorly understood.



OK … let's dispel the magic behind this black box

PCA - Overview

- It is a mathematical tool from applied linear algebra.
- It is a simple, non-parametric method of extracting relevant information from confusing data sets.
- It provides a roadmap for how to reduce a complex data set to a lower dimension.

Background

- Linear Algebra
- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Linear Discriminant Analysis (LDA)
- Examples
- Face Recognition - Application

Variance

- A measure of the spread of the data in a data set with mean $\overline{X}$

$$\sigma^2 = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{(n-1)}$$

- Variance is claimed to be the original statistical measure of spread of data.

Covariance

- Variance – measure of the deviation from the mean for points in one dimension, e.g., heights

- Covariance – a measure of how much each of the dimensions varies from the mean with respect to each other.

- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions, e.g., number of hours studied and grade obtained.

- The covariance between one dimension and itself is the variance

$$\operatorname{var}(X) = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)}{(n-1)}$$

$$\operatorname{cov}(X, Y) = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{(n-1)}$$

Covariance

- What is the interpretation of covariance calculations?
- Say you have a 2-dimensional data set
  - *X*: number of hours studied for a subject
  - *Y*: marks obtained in that subject
- And assume the covariance value (between X and Y) is: 104.53
- *What does this value mean*?

Covariance

- Exact value is not as important as its sign.

- A positive value of covariance indicates that both dimensions increase or decrease together, e.g., as the number of hours studied increases, the grades in that subject also increase.

- A negative value indicates while one increases the other decreases, or vice-versa, e.g., active social life vs. performance in ECE Dept.

- If covariance is zero: the two dimensions are independent of each other, e.g., heights of students vs. grades obtained in a subject.

Covariance

- Why bother with calculating (expensive) covariance when we could just plot the 2 values to see their relationship?

Covariance calculations are used to find relationships between dimensions in high dimensional data sets (usually greater than 3) where visualization is difficult.

Covariance Matrix

- Representing covariance among dimensions as a matrix, e.g., for 3 dimensions:

$$C = \begin{bmatrix} \operatorname{cov}(X,X) & \operatorname{cov}(X,Y) & \operatorname{cov}(X,Z) \\ \operatorname{cov}(Y,X) & \operatorname{cov}(Y,Y) & \operatorname{cov}(Y,Z) \\ \operatorname{cov}(Z,X) & \operatorname{cov}(Z,Y) & \operatorname{cov}(Z,Z) \end{bmatrix}$$

- Properties:
  - Diagonal: variances of the variables
  - cov(X,Y)=cov(Y,X), hence matrix is symmetrical about the diagonal (upper triangular)
  - m-dimensional data will result in *mxm* covariance matrix

Linear algebra

- Matrix A:

$$A = [a_{ij}]_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$$

- Matrix Transpose

$$B = [b_{ij}]_{n\times m} = A^T \Leftrightarrow b_{ij} = a_{ji}; \quad 1 \le i \le n, 1 \le j \le m$$

- Vector **a**

$$a = \begin{bmatrix} a_1 \\ \ldots \\ a_n \end{bmatrix}; \quad a^T = [a_1, \ldots, a_n]$$

Matrix and vector multiplication

- Matrix multiplication

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

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

- Outer vector product

$$a = A = [a_{ij}]_{m \times 1};\, b^T = B = [b_{ij}]_{1 \times n};$$

$$c = a \times b = AB, an\ m \times n\ matrix$$

- Vector-matrix product

$$A = [a_{ij}]_{m \times n};\, b = B = [b_{ij}]_{n \times 1};$$

$$C = Ab = an\ m \times 1\ matrix = vector\ of\ length\ m$$

Inner Product

- Inner (dot) product:

$$a^T \cdot b = \sum_{i=1}^{n} a_i b_i$$

- Length (Eucledian norm) of a vector
- **a** is normalized iff ||a|| = 1

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

- The angle between two n-dimesional vectors

$$\cos\theta = \frac{a^T \cdot b}{\| a \| \| b \|}$$

- An inner product is a measure of collinearity:
  - **a** and **b** are orthogonal iff

$$a^T \cdot b = 0$$

  - **a** and **b** are collinear iff

$$a^T \cdot b = \| a \| \| b \|$$

- A set of vectors is *linearly independent* if no vector is a linear combination of other vectors.

Determinant and Trace

- Determinant

$$A = [a_{ij}]_{n \times n};$$

$$\det(A) = \sum_{j=1}^{n} a_{ij} A_{ij}; \quad i = 1, \ldots . n;$$

$$A_{ij} = (-1)^{i+j} \det(M_{ij})$$

det(AB)= det(A)det(B)

- Trace $A = [a_{ij}]_{n \times n}; tr[A] = \sum_{j=1}^{n} a_{jj}$

Matrix Inversion

- A (*n x n*) is nonsingular if there exists B such that:

$$AB = BA = I_n ; B = A^{-1}$$

- A=[2 3; 2 2], B=[-1 3/2; 1 -1]

- A is nonsingular iff

$$\| A \| \neq 0$$

- Pseudo-inverse for a non square matrix, provided $A^T A$ is not singular

$$A^{\#} = [A^T A]^{-1} A^T$$

$$A^{\#} A = I$$

Linear Independence

- A set of *n*-dimensional vectors $\mathbf{x_i}$ Є $\mathbf{R^n}$, are said to be linearly independent if none of them can be written as a linear combination of the others.

- In other words,

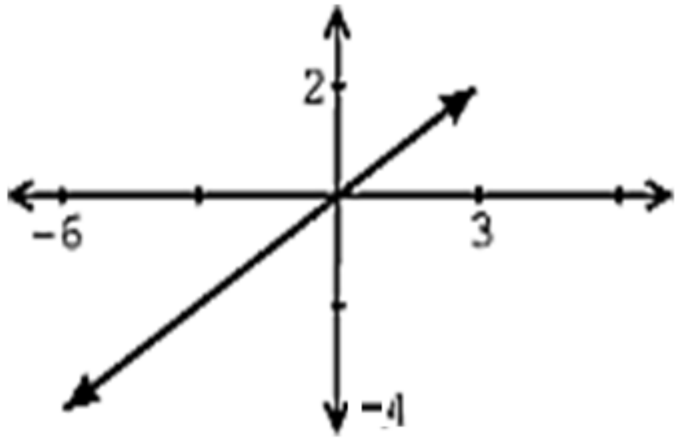
$$c_1x_1 + c_2x_2 + ... + c_kx_k = 0$$

$$iff \qquad c_1 = c_2 = ... = c_k = 0$$

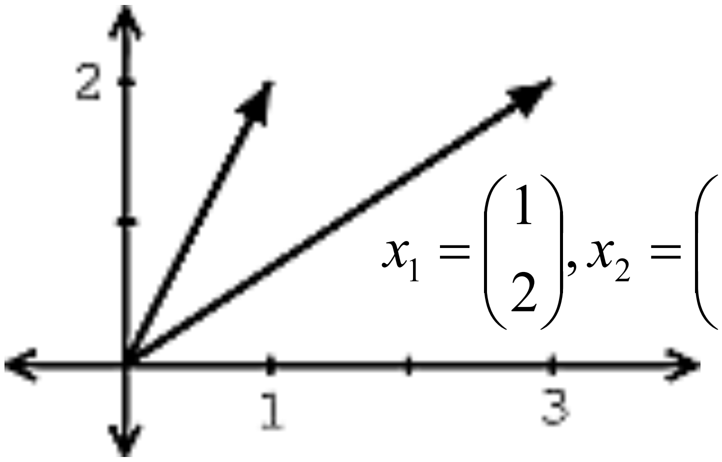
Linear Independence

- Another approach to reveal a vectors independence is by graphing the vectors.

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, x_2 = \begin{pmatrix} -6 \\ -4 \end{pmatrix}$$



Not linearly independent vectors



$$x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, x_2 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Linearly independent vectors

Span

- A span of a set of vectors $\mathbf{x_1}$, $\mathbf{x_2}$, … , $\mathbf{x_k}$ is the set of vectors that can be written as a linear combination of $\mathbf{x_1}$, $\mathbf{x_2}$, … , $\mathbf{x_k}$.

$$span(x_1, x_2, \ldots, x_k) = \{c_1 x_1 + c_2 x_2 + \ldots + c_k x_k \mid c_1, c_2, \ldots, c_k \in \Re\}$$

Basis

- A basis for $\mathbf{R}^n$ is a set of vectors which:
  - Spans $\mathbf{R}^n$, i.e. any vector in this *n*-dimensional space can be written as linear combination of these basis vectors.
  - Are linearly independent
- Clearly, any set of *n*-linearly independent vectors form basis vectors for $\mathbf{R}^n$.

Orthogonal/Orthonormal Basis

- An **orthonormal basis** of an a vector space *V* with an inner product, is a set of basis vectors whose elements are mutually orthogonal and of magnitude 1 (unit vectors).

- Elements in an **orthogonal basis** do not have to be unit vectors, but must be mutually perpendicular. It is easy to change the vectors in an orthogonal basis by scalar multiples to get an orthonormal basis, and indeed this is a typical way that an orthonormal basis is constructed.

- Two vectors are **orthogonal** if they are perpendicular, i.e., they form a right angle, i.e. if their inner product is zero.

$$a^T \cdot b = \sum_{i=1}^{n} a_i b_i = 0 \quad \Rightarrow \quad a \perp b$$

- The standard basis of the *n*-dimensional Euclidean space $\mathbf{R}^n$ is an example of orthonormal (and ordered) basis.

Transformation Matrices

- Consider the following:

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

- The square (transformation) matrix scales (3,2)

- Now assume we take a multiple of (3,2)

$$2 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 24 \\ 16 \end{bmatrix} = 4 \times \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Transformation Matrices

- Scale vector (3,2) by a value 2 to get (6,4)
- Multiply by the square transformation matrix
- And we see that the result is still scaled by 4.

WHY?

A vector consists of both length and direction. Scaling a vector only changes its length and not its direction. This is an important observation in the transformation of matrices leading to formation of eigenvectors and eigenvalues.

Irrespective of how much we scale (3,2) by, the solution (under the given transformation matrix) is always a multiple of 4.

Eigenvalue Problem

- The eigenvalue problem is any problem having the following form:

$$\mathbf{A} \,.\, \mathbf{v} = \lambda \,.\, \mathbf{v}$$

A: *m* x *m* matrix

v: *m* x 1 non-zero vector

λ: scalar

- Any value of λ for which this equation has a solution is called the eigenvalue of A and the vector **v** which corresponds to this value is called the eigenvector of **A**.

Eigenvalue Problem

- Going back to our example:

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$\mathbf{A} \;.\; \mathbf{v} \qquad = \lambda \;.\; \mathbf{v}$$

- Therefore, (3,2) is an eigenvector of the square matrix **A** and 4 is an eigenvalue of **A**
- The question is:

Given matrix A, how can we calculate the eigenvector and eigenvalues for A?

Calculating Eigenvectors & Eigenvalues

- Simple matrix algebra shows that:

$$\mathbf{A} \,.\, \mathbf{v} = \lambda \,.\, \mathbf{v}$$

$$\Leftrightarrow \quad \mathbf{A} \,.\, \mathbf{v} - \lambda \,.\, \mathbf{I} \,.\, \mathbf{v} = \mathbf{0}$$

$$\Leftrightarrow \quad (\mathbf{A} - \lambda \,.\, \mathbf{I}\,).\, \mathbf{v} = \mathbf{0}$$

- Finding the roots of $|\mathbf{A} - \lambda \,.\, \mathbf{I}|$ will give the eigenvalues and for each of these eigenvalues there will be an eigenvector

  Example …

Calculating Eigenvectors & Eigenvalues

- Let

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

- Then:

$$|A - \lambda.I| = \left\| \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix} \right\| = (-\lambda \times (-3-\lambda)) - (-2 \times 1) = \lambda^2 + 3\lambda + 2$$

- And setting the determinant to 0, we obtain 2 eigenvalues:

λ_1 = -1 and λ_2 = -2

Calculating Eigenvectors & Eigenvalues

- For λ_1 the eigenvector is:

$$(A - \lambda_1 . I).v_1 = 0$$

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} . \begin{bmatrix} v_{1:1} \\ v_{1:2} \end{bmatrix} = 0$$

$$v_{1:1} + v_{1:2} = 0 \qquad and \qquad -2v_{1:1} - 2v_{1:2} = 0$$

$$v_{1:1} = -v_{1:2}$$

- Therefore the first eigenvector is any column vector in which the two elements have equal magnitude and opposite sign.

Calculating Eigenvectors & Eigenvalues

- Therefore eigenvector v_1 is

$$v_1 = k_1 \begin{bmatrix} +1 \\ -1 \end{bmatrix}$$

where k_1 is some constant.

- Similarly we find that eigenvector v_2

$$v_2 = k_2 \begin{bmatrix} +1 \\ -2 \end{bmatrix}$$

where k_2 is some constant.

Properties of Eigenvectors and Eigenvalues

- Eigenvectors can only be found for square matrices and not every square matrix has eigenvectors.

- Given an *m* x *m* matrix (with eigenvectors), we can find *n* eigenvectors.

- All eigenvectors of a symmetric* matrix are perpendicular to each other, no matter how many dimensions we have.

- In practice eigenvectors are normalized to have unit length.

*Note: covariance matrices are symmetric!

Now … Let's go back

PCA

Nomenclature

- m : number of rows (features/measurement types) in a dataset matrix.
- n : number of columns (data samples) in a dataset matrix.
- $\mathbf{X}$: given dataset matrix (mxn)
- $\mathbf{x}_i$: columns/rows of **X** as indicated.
- $\mathbf{Y}$: re-representation (transformed) matrix of dataset matrix **X** (mxn).
- $\mathbf{y}_i$: columns/rows of **Y** as indicated
- $\mathbf{P}$: transformation matrix
- $\mathbf{p}_i$: rows of **P** (set of new basis vectors).
- $\mathbf{S}_x$: covariance matrix of dataset matrix **X**.
- $\mathbf{S}_y$: covariance matrix of the transformed dataset matrix **Y**.
- $\mathbf{r}$: rank of a matrix
- $\mathbf{D}$: diagonal matrix containing eigen values.
- $\mathbf{V}$: matrix of eigen vectors arranged as columns.

Example of a problem

- We collected *m* parameters about 100 students:
  - Height
  - Weight
  - Hair color
  - Average grade
  - …

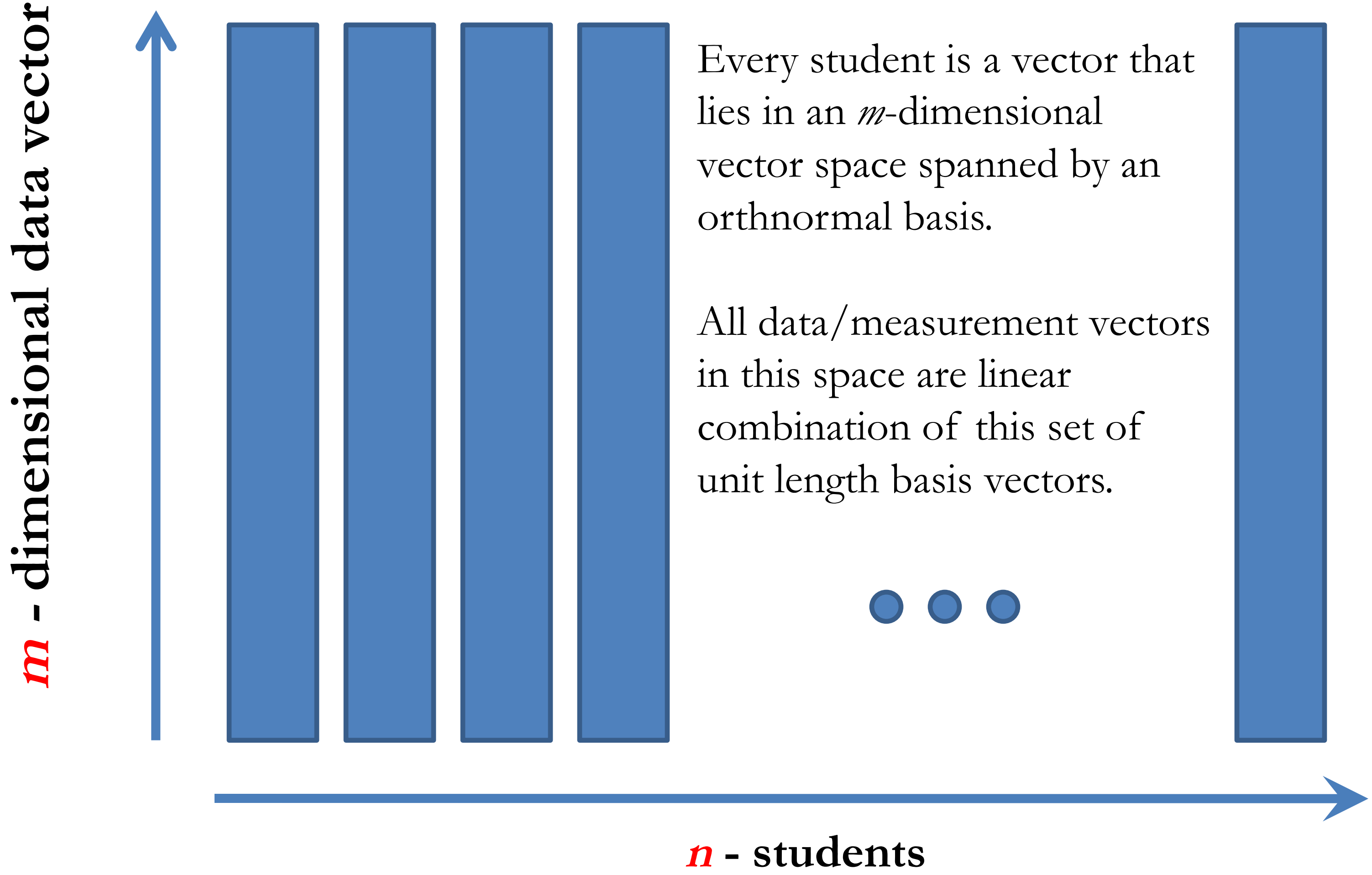
- We want to find the most important parameters that best describe a student.

Example of a problem

- Each student has a vector of data which describes him of length *m*:
  - (180,70,'purple',84,…)

- We have *n* = 100 such vectors. Let's put them in one matrix, where each column is one student vector.
- So we have a *m*x*n* matrix. This will be the input of our problem.

Example of a problem



m - dimensional data vector

Every student is a vector that lies in an m-dimensional vector space spanned by an orthnormal basis.

All data/measurement vectors in this space are linear combination of this set of unit length basis vectors.

n - students

Which parameters can we ignore?

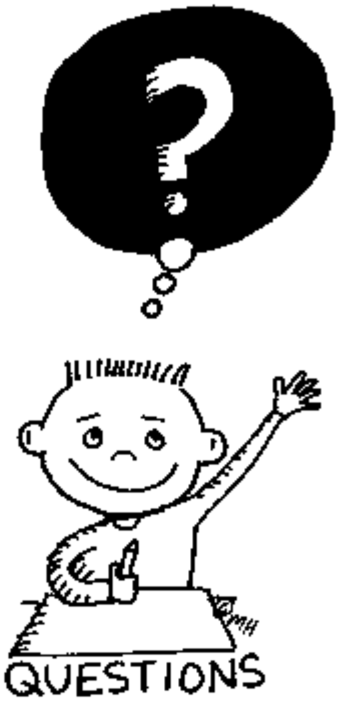
- Constant parameter (number of heads)
  - 1,1,…,1.
- Constant parameter with some noise - (thickness of hair)
  - 0.003, 0.005,0.002,….,0.0008 → low variance
- Parameter that is linearly dependent on other parameters (head size and height)
  - $Z= aX + bY$

Which parameters do we want to keep?

- Parameter that doesn’t depend on others (e.g. eye color), i.e. uncorrelated ➔ low covariance.
- Parameter that changes a lot (grades)
  - The opposite of noise
  - High variance

Questions

- How we describe 'most important' features using math?
  - Variance
- How do we represent our data so that the most important features can be extracted easily?
  - Change of basis



Change of Basis !!!

- Let **X** and **Y** be $m \times n$ matrices related by a linear transformation **P**.
- **X** is the original recorded data set and **Y** is a re-representation of that data set.

$$\mathbf{PX} = \mathbf{Y}$$

- Let's define;
  - $\mathbf{p}_i$ are the rows of **P.**
  - $\mathbf{x}_i$ are the columns of **X.**
  - $\mathbf{y}_i$ are the columns of **Y.**

Change of Basis !!!

- Let **X** and **Y** be $m \times n$ matrices related by a linear transformation **P**.
- **X** is the original recorded data set and **Y** is a re-representation of that data set.

$$\mathbf{PX} = \mathbf{Y}$$

- What does this mean?
  - **P** is a matrix that transforms **X** into **Y**.
  - Geometrically, **P** is a rotation and a stretch (scaling) which again transforms **X** into **Y**.
  - The rows of **P**, $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m\}$ are a set of new basis vectors for expressing the columns of **X.**

Change of Basis !!!

- Lets write out the explicit dot products of **PX.**

$$\mathbf{PX} = \begin{bmatrix} \mathbf{p_1} \\ \vdots \\ \mathbf{p_m} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} & \cdots & \mathbf{x_n} \end{bmatrix}$$

- We can note the form of each column of **Y**.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{p_1} \cdot \mathbf{x_1} & \cdots & \mathbf{p_1} \cdot \mathbf{x_n} \\ \vdots & \ddots & \vdots \\ \mathbf{p_m} \cdot \mathbf{x_1} & \cdots & \mathbf{p_m} \cdot \mathbf{x_n} \end{bmatrix}$$

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{p_1} \cdot \mathbf{x_i} \\ \vdots \\ \mathbf{p_m} \cdot \mathbf{x_i} \end{bmatrix}$$

- We can see that each coefficient of $\mathbf{y_i}$ is a dot-product of $\mathbf{x_i}$ with the corresponding row in **P**.

In other words, the j^{th} coefficient of y_i is a projection onto the j^{th} row of P.

Therefore, the *rows* of P are a new set of basis vectors for representing the *columns* of X.

Change of Basis !!!

- Changing the basis doesn't change the data – only its representation.
- Changing the basis is actually projecting the data vectors on the basis vectors.
- Geometrically, **P** is a rotation and a stretch of **X**.
    - If **P** basis is orthonormal (length = 1) then the transformation **P** is only a rotation

Questions Remaining !!!

- Assuming linearity, the problem now is to find the appropriate *change of basis*.
- The *row* vectors {$\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_m}$} in this transformation will become the *principal components* of **X**.
- Now,
  - What is the best way to *re-express* **X**?
  - What is the good choice of basis **P**?
- Moreover,
  - *what features we would like* ***Y*** *to exhibit?*

What does "best express" the data mean ?!!!

- As we've said before, we want to filter out noise and extract the relevant information from the given data set.
- Hence, the representation we are looking for will decrease both noise and redundancy in the data set at hand.

Noise …

- Noise in any data must be low or – no matter the analysis technique – no information about a system can be extracted.

- There exists no absolute scale for the noise but rather it is measured relative to the measurement, e.g. recorded ball positions.

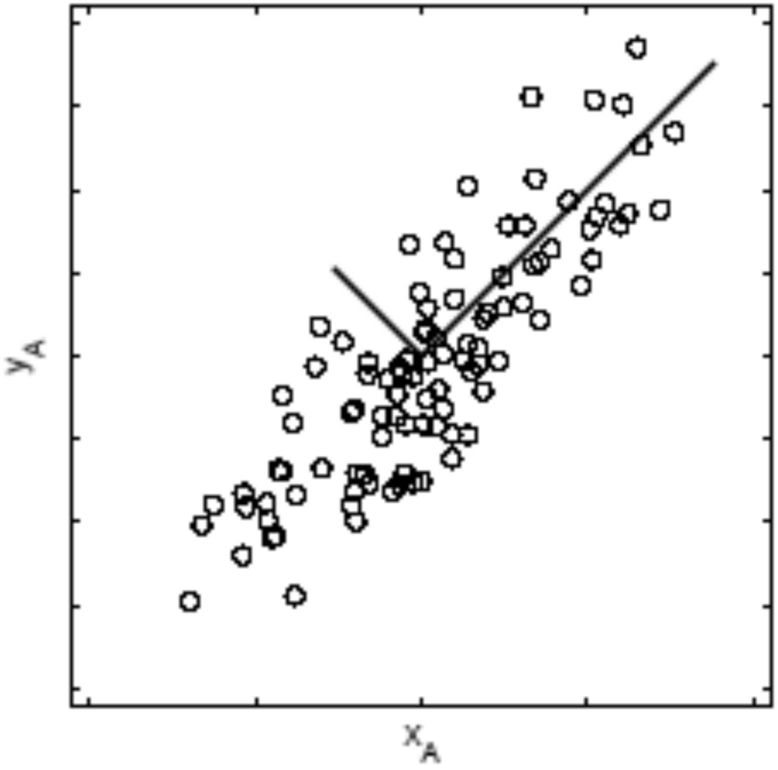
- A common measure is the *signal-to-noise ration (SNR)*, or a ratio of variances.

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}$$

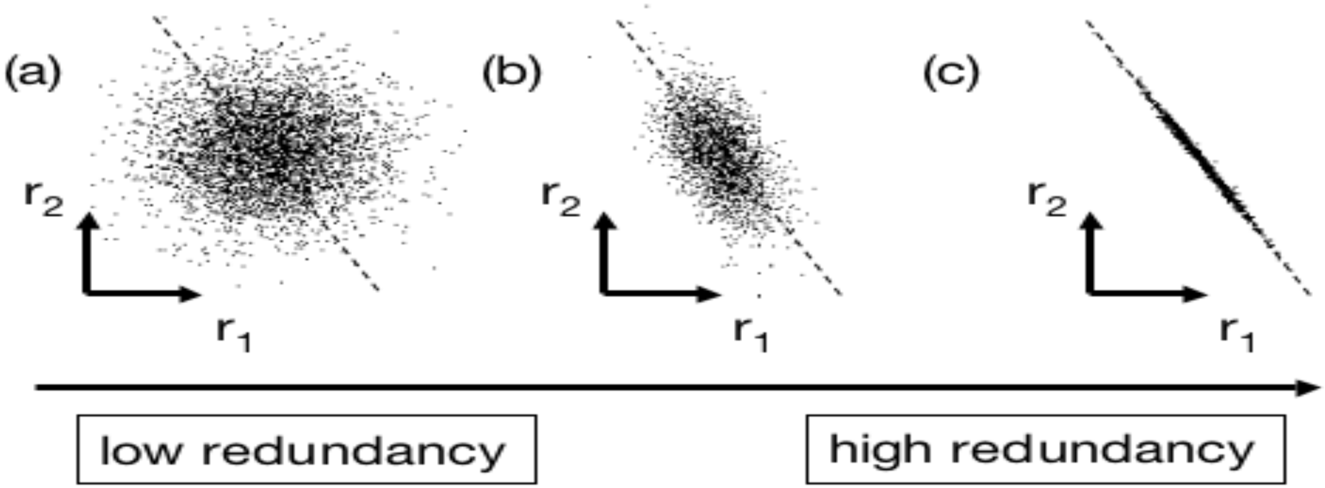
- A high SNR (>>1) indicates high precision data, while a low SNR indicates noise contaminated data.

Why Variance ?!!!

- Find the axis rotation that maximizes SNR = maximizes the variance between axis.



Redundancy …



- Multiple sensors record the same dynamic information.

- Consider a range of possible plots between two arbitrary measurement types r_1 and r_2.

- Panel(a) depicts two recordings with no redundancy, i.e. they are *un-correlated*, e.g. person's height and his GPA.

- However, in panel(c) both recordings appear to be strongly related, i.e. one can be expressed in terms of the other.

Covariance Matrix

- Assuming zero mean data (subtract the mean), consider the indexed vectors {$\mathbf{x}_{1,}$ $\mathbf{x}_{2,}$ ...,$\mathbf{x}_m$} which are the *rows* of an *mxn* matrix **X**.

- Each row corresponds to all measurements of a particular measurement type ($\mathbf{x}_i$).

- Each column of **X** corresponds to a set of measurements from particular time instant.

- We now arrive at a definition for the covariance matrix $\mathbf{S}_\mathbf{X}$.

$$\mathbf{S_X} \equiv \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \quad \text{where} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_m} \end{bmatrix}$$

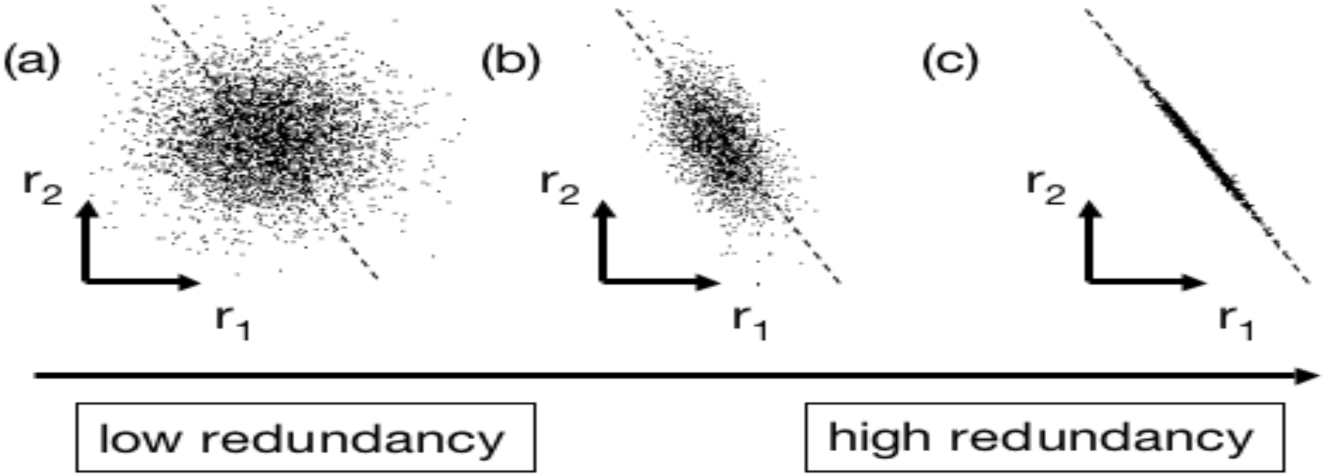
Covariance Matrix

$$\mathbf{S_X} \equiv \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \quad \text{where} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_m} \end{bmatrix}$$

- The ij^th element of the variance is the dot product between the vector of the i^th measurement type with the vector of the j^th measurement type.
  - $\mathbf{S_X}$ is a square symmetric *m*×*m* matrix.
  - The diagonal terms of $\mathbf{S_X}$ are the variance of particular measurement types.
  - The off-diagonal terms of $\mathbf{S_X}$ are the covariance between measurement types.

Covariance Matrix

$$\mathbf{S}_{\mathbf{X}} \equiv \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \quad \text{where} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_m} \end{bmatrix}$$



- Computing $\mathbf{S}_{\mathbf{X}}$ quantifies the correlations between all possible pairs of measurements. Between one pair of measurements, a large covariance corresponds to a situation like panel (c), while zero covariance corresponds to entirely uncorrelated data as in panel (a).

Covariance Matrix

- Suppose, we have the option of manipulating $\mathbf{S_X}$. We will suggestively define our manipulated covariance matrix $\mathbf{S_Y}$.

What features do we want to optimize in S_Y?

Diagonalize the Covariance Matrix

Our goals are to find the covariance matrix that:

1. Minimizes redundancy, measured by covariance. (off-diagonal), i.e. we would like each variable to co-vary as little as possible with other variables.

2. Maximizes the signal, measured by variance. (the diagonal)

Since covariance is non-negative, the optimized covariance matrix will be a diagonal matrix.

Diagonalize the Covariance Matrix

PCA Assumptions

- PCA assumes that all basis vectors $\{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ are orthonormal (i.e. $\mathbf{p}_i \cdot \mathbf{p}_j = \delta_{ij}$).
- Hence, in the language of linear algebra, PCA assumes **P** is an orthonormal matrix.
- Secondly, PCA assumes the directions with the largest variances are the most "important" or in other words, most principal.
- *Why are these assumptions easiest?*

Diagonalize the Covariance Matrix

PCA Assumptions

- By the variance assumption PCA first selects a normalized direction in *m*-dimensional space along which the variance in **X** is maximized - it saves this as $\mathbf{p}_1$.
- Again it finds another direction along which variance is maximized, however, because of the orthonormality condition, it restricts its search to all directions perpendicular to all previous selected directions.
- This could continue until *m* directions are selected. The resulting ordered set of **p**'s are the *principal components*.
- The variances associated with each direction $\mathbf{p}_\mathrm{i}$ quantify how principal each direction is. We could thus rank-order each basis vector $\mathbf{p}_\mathrm{i}$ according to the corresponding variances.
- This pseudo-algorithm works, however we can solve it using linear algebra.

Solving PCA: Eigen Vectors of Covariance Matrix

- We will derive our first algebraic solution to PCA using linear algebra. This solution is based on an important property of *eigenvector decomposition*.

- Once again, the data set is **X**, an *m*×*n* matrix, where *m* is the number of measurement types and *n* is the number of data trials.

- The goal is summarized as follows:
  - Find some orthonormal matrix **P** where **Y = PX** such that is $\mathbf{S_Y} \equiv \frac{1}{n-1}\mathbf{Y}\mathbf{Y}^T$ diagonalized. The rows of **P** are the *principal components* of **X**.

Solving PCA: Eigen Vectors of Covariance Matrix

- We begin by rewriting $\mathbf{S_Y}$ in terms of our variable of choice **P**.

$$\begin{aligned}
\mathbf{S_Y} &= \frac{1}{n-1}\mathbf{Y}\mathbf{Y}^T \\
&= \frac{1}{n-1}(\mathbf{PX})(\mathbf{PX})^T \\
&= \frac{1}{n-1}\mathbf{P}\mathbf{X}\mathbf{X}^T\mathbf{P}^T \\
&= \frac{1}{n-1}\mathbf{P}(\mathbf{X}\mathbf{X}^T)\mathbf{P}^T \\
\mathbf{S_Y} &= \frac{1}{n-1}\mathbf{P}\mathbf{A}\mathbf{P}^T
\end{aligned}$$

- Note that we defined a new matrix $\mathbf{A} = \mathbf{XX}^\mathbf{T}$, where **A** is *symmetric*

- Our roadmap is to recognize that a symmetric matrix (**A**) is diagonalized by an orthogonal matrix of its eigenvectors

- A symmetric matrix **A** can be written as $\mathbf{VDV}^\mathbf{T}$ where **D** is a diagonal matrix and **V** is a matrix of eigenvectors of **A** arranged as columns.

- The matrix **A** has $r \leq m$ orthonormal eigenvectors where r is the rank of the matrix.

Solving PCA: Eigen Vectors of Covariance Matrix

- Now comes the trick. *We select the matrix* **P** *to be a matrix where each row* $\mathbf{p}_i$ *is an eigenvector of* $\boldsymbol{XX^T}$.
- By this selection, $\mathbf{P} = \mathbf{V}^{\mathbf{T}}$. Hence $\mathbf{A} = \mathbf{P}^{\mathbf{T}}\mathbf{D}\mathbf{P}$.
- With this relation and the fact that $\mathbf{P}^{-1} = \mathbf{P}^{\mathbf{T}}$ since the inverse of orthonormal matrix is its transpose, we can finish evaluating $\mathbf{S}_{\mathbf{Y}}$ as follows;

$$
\begin{aligned}
\mathbf{S}_{\mathbf{Y}} &= \frac{1}{n-1}\mathbf{P}\mathbf{A}\mathbf{P}^T \\
&= \frac{1}{n-1}\mathbf{P}(\mathbf{P}^T\mathbf{D}\mathbf{P})\mathbf{P}^T \\
&= \frac{1}{n-1}(\mathbf{P}\mathbf{P}^T)\mathbf{D}(\mathbf{P}\mathbf{P}^T) \\
&= \frac{1}{n-1}(\mathbf{P}\mathbf{P}^{-1})\mathbf{D}(\mathbf{P}\mathbf{P}^{-1}) \\
\mathbf{S}_{\mathbf{Y}} &= \frac{1}{n-1}\mathbf{D}
\end{aligned}
$$

- It is evident that the choice of **P** diagonalizes $\mathbf{S}_{\mathbf{Y}}$. This was the goal for PCA.
- Now lets do this in steps by counter example.

PCA Process – STEP 1

- Subtract the mean from each of the dimensions
- This produces a data set whose mean is zero.
- Subtracting the mean makes variance and covariance calculation easier by simplifying their equations.
- The variance and co-variance values are not affected by the mean value.
- Suppose we have two measurement types $\mathbf{X}_1$ and $\mathbf{X}_2$, hence $m = 2$, and ten samples each, hence $n = 10$.

PCA Process – STEP 1

X_1	X_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.2	0.9

$$\Rightarrow \quad \begin{aligned} \overline{X_1} &= 1.81 \\ \overline{X_2} &= 1.91 \end{aligned} \quad \Rightarrow$$

X_1'	X'_2
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01

http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf

PCA Process – STEP 2

- Calculate the covariance matrix

$$S_X = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$$

- Since the non-diagonal elements in this covariance matrix are positive, we should expect that both the $\mathbf{X}_1$ and $\mathbf{X}_2$ variables increase together.
- Since it is symmetric, we expect the eigenvectors to be orthogonal.

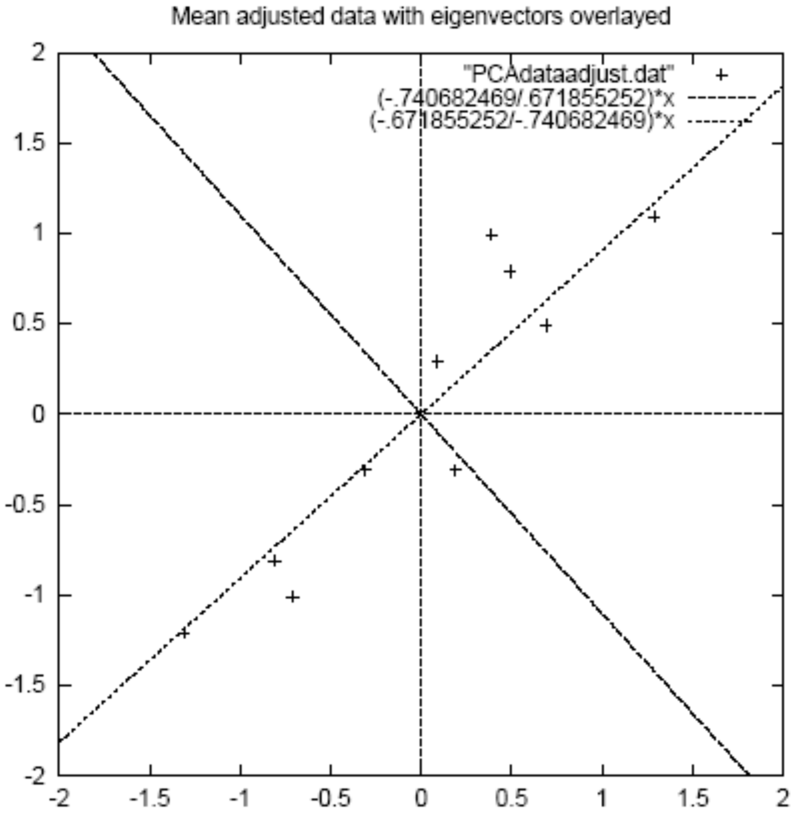
PCA Process – STEP 3

- Calculate the eigen vectors **V** and eigen values **D** of the covariance matrix

$$D = \begin{bmatrix} 0.490833989 \\ 1.28402771 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

PCA Process – STEP 3



A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

Eigenvectors are plotted as diagonal dotted lines on the plot. (note: they are perpendicular to each other).

One of the eigenvectors goes through the middle of the points, like drawing a line of best fit.

The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

PCA Process – STEP 4

- Reduce dimensionality and form *feature vector*

The eigenvector with the *highest* eigenvalue is the *principal component* of the data set.

In our example, the eigenvector with the largest eigenvalue is the one that points down the middle of the data.

Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance.

PCA Process – STEP 4

Now, if you'd like, you can decide to *ignore* the components of lesser significance.

You do lose some information, but if the eigenvalues are small, you don't lose much

- *m* dimensions in your data
- calculate *m* eigenvectors and eigenvalues
- choose only the first *r* eigenvectors
- final data set has only *r* dimensions.

PCA Process – STEP 4

- When the λ_i's are sorted in descending order, the proportion of variance explained by the *r* principal components is:

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{m} \lambda_i} = \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_r}{\lambda_1 + \lambda_2 + \ldots + \lambda_p + \ldots + \lambda_m}$$

- If the dimensions are highly correlated, there will be a small number of eigenvectors with large eigenvalues and *r* will be much smaller than *m*.

- If the dimensions are not correlated, *r* will be as large as *m* and PCA does not help.

PCA Process – STEP 4

- Feature Vector

$$\text{FeatureVector} = (\lambda_1\ \lambda_2\ \lambda_3\ \ldots\ \lambda_r)$$

(take the eigenvectors to keep from the ordered list of eigenvectors, and form a matrix with these eigenvectors in the columns)

We can either form a feature vector with both of the eigenvectors:

$$\begin{bmatrix} -0.677873399 & -0.735178656 \\ -0.735178656 & 0.677873399 \end{bmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{bmatrix} -0.677873399 \\ -0.735178656 \end{bmatrix}$$

PCA Process – STEP 5

- Derive the new data

FinalData = RowFeatureVector x RowZeroMeanData

RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top.

RowZeroMeanData is the mean-adjusted data *transposed*, i.e., the data items are in each column, with each row holding a separate dimension.

PCA Process – STEP 5

- FinalData is the final data set, with data items in columns, and dimensions along rows.

- What does this give us?

  The original data *solely in terms of the vectors we chose.*

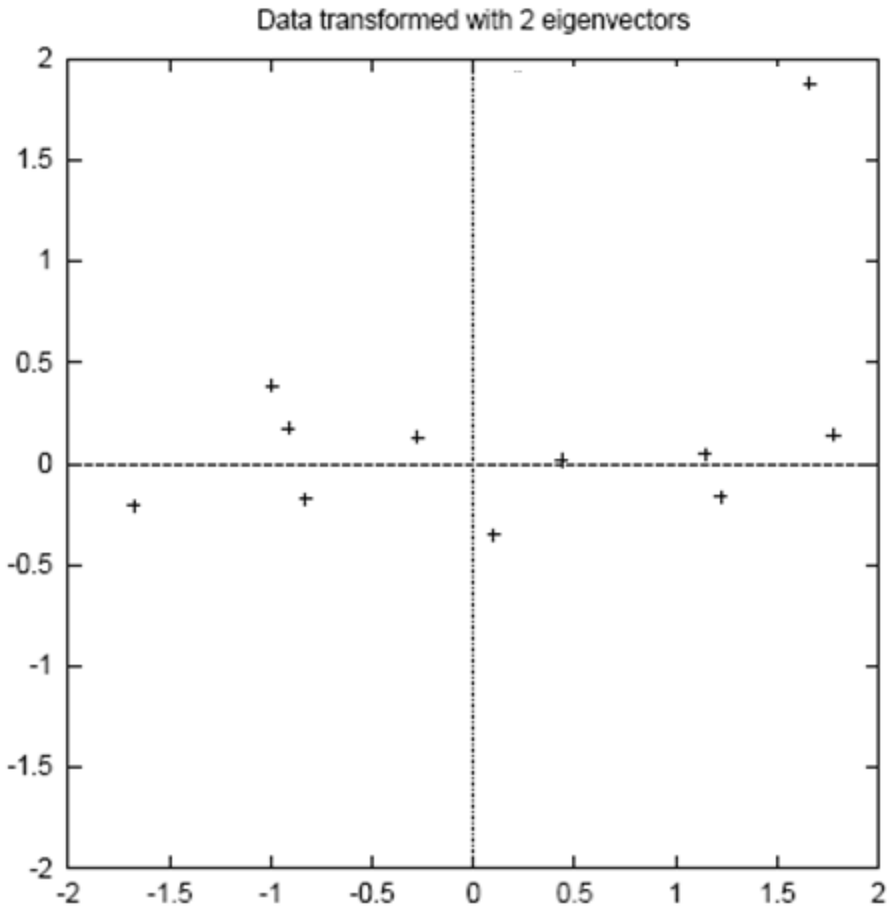
- We have changed our data from being in terms of the axes $\mathbf{X}_1$ and $\mathbf{X}_2$, to now be in terms of our 2 eigenvectors.

PCA Process – STEP 5

FinalData (transpose: dimensions along columns)

Y_1	Y_2
-0.827870186	-0.175115307
1.77758033	0.142857227
-0.992197494	0.384374989
-0.274210416	0.130417207
-1.67580142	-0.209498461
-0.912949103	0.175282444
0.0991094375	-0.349824698
1.14457216	0.0464172582
0.438046137	0.0177646297
1.22382956	-0.162675287

PCA Process – STEP 5



The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

Reconstruction of Original Data

- Recall that:

  FinalData = RowFeatureVector x RowZeroMeanData

- Then:

  RowZeroMeanData = RowFeatureVector^{-1} x FinalData

- And thus:

  RowOriginalData = (RowFeatureVector^{-1} x FinalData) + OriginalMean

- If we use unit eigenvectors, the inverse is the same as the transpose (hence, easier).

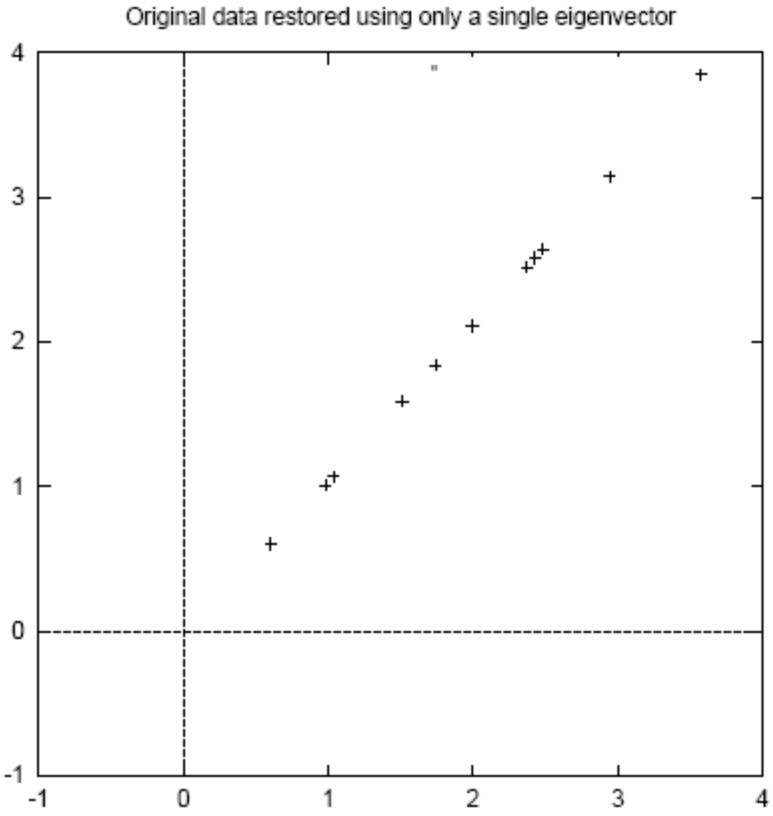
Reconstruction of Original Data

- If we reduce the dimensionality (i.e., $r < m$), obviously, when reconstructing the data we lose those dimensions we chose to discard.

- In our example let us assume that we considered only a single eigenvector.

- The final data is Y_1 only and the reconstruction yields…

Reconstruction of Original Data



The reconstruction from the data that was derived using only a single eigen vector

The variation along the principal component is preserved.

The variation along the other component has been lost.

References

- J. SHLENS, "TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS," 2005 http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf

- Introduction to PCA, http://dml.cs.byu.edu/~cgc/docs/dm/Slides/