Principal Components Analysis (PCA)

Dimensionality Reduction Using Principal Components

CS 6316 – Machine Learning
Fall 2017

OUTLINE

- High-dimensional data sets
- The curse of Dimensionality
- Principal Components Analysis (PCA)
- PCA Method (Example)
- Python Examples (ipynb format)
- Mathematical Background (Appendix)

High-dimensional Data Sets

- The amount of data generated each day from sources such as scientific experiments, cell phones, and smartwatches has been growing exponentially over the last several years
- Not only are the number data sources increasing, but the data itself is also growing **richer** as the *number of features in the data increases*
- Datasets with a large number of features are called *high-dimensional datasets*

High-dimensional Data Sets

- Examples:
  - high-resolution image data
    - where the features are pixels, and which increase in dimensionality as sensor technology improves
  - user movie ratings
    - where the features are movies rated, and where the number of dimensions increases as the user rates more of them.

The Curse of Dimensionality

- A term coined by Bellman in 1961
- Refers to the problems associated with multivariate data analysis as the **dimensionality increases**
- It relates to the fact that the *convergence of any estimator to the true value of a smooth function defined on a space of high dimension is very slow*
- This means that, *a priori*, we need an "enormous" amount of observations to obtain a "*good*" estimate of a function
- Often also results in increased computational burden

An simple example

Let's say you are using a model to predict the location of a large bacteria in a 25cm^2 petri dish
The model might be fairly accurate at pinning the particle down to the nearest square cm

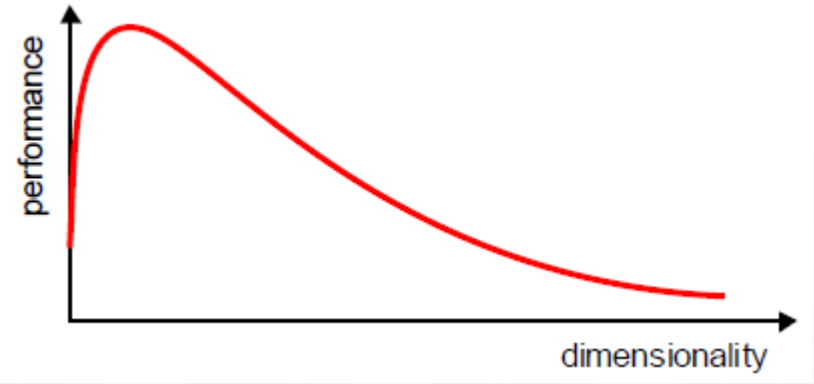
What happens if you add just one more dimension?

An simple example

However, if you add just one more dimension / variable — say, instead of a 2D petri dish you have a 3D beaker — the predictive space increases exponentially, to 125 cm^3. This increases **the *computational burden*.** The task of pinpointing where bacteria might be in a 3D model, has turned into a more challenging task!

What does the curse of dimensionality mean in practice?

- For a given sample size, there is a maximum number of features above which the performance of the classifier will degrade rather than improve
- In most cases, the additional information that is lost by discarding some features is (more than) compensated by a more accurate mapping in the lower-dimensional space

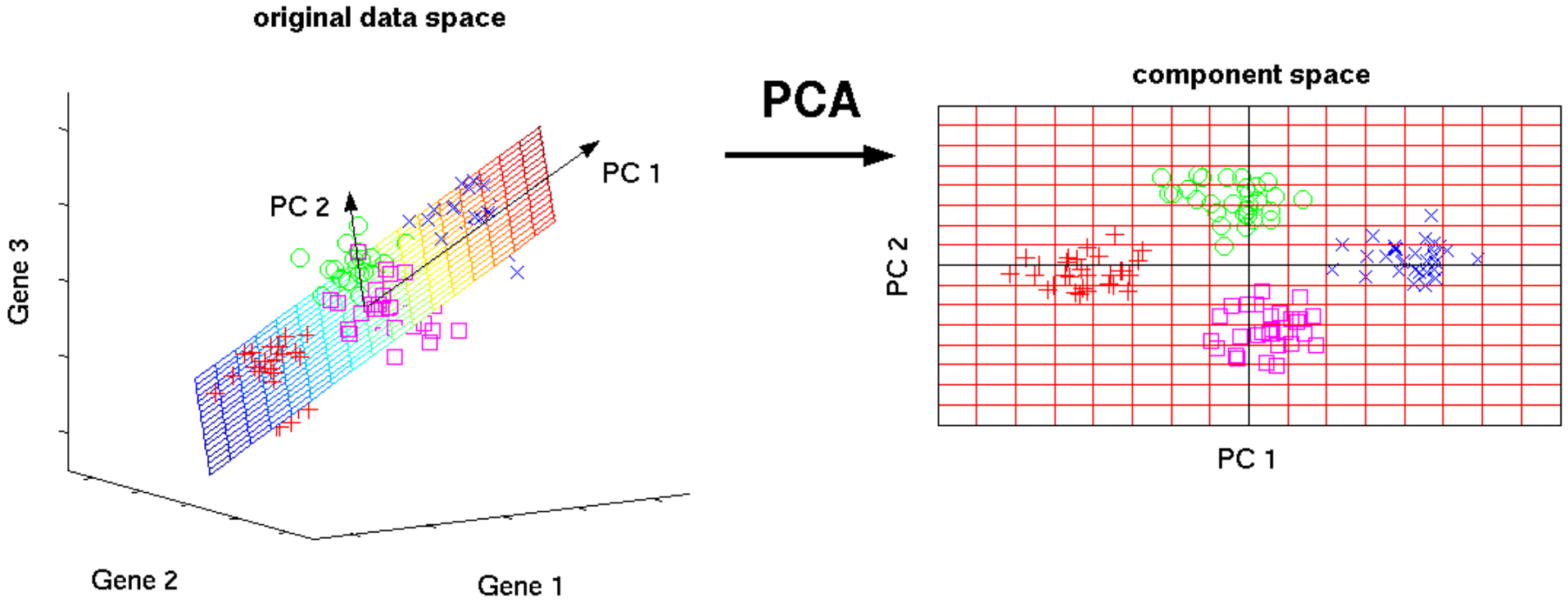


Principal Component Analysis (PCA) ~ In a Nutshell

- *Principal Component Analysis* (PCA) is a dimensionality reduction technique used to transform high-dimensional datasets into a dataset with fewer variables, where the set of resulting variables explains the maximum variance within the dataset
- It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible. With fewer variables, visualization also becomes much more meaningful. PCA is more useful when dealing with 3 or higher dimensional data.

One More Time?

- PCA is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first *principal component*), the second greatest variance on the second coordinate, and so on. In this sense, PCA computes the most meaningful *basis* to express the data



PCA Applications

- PCA is a useful statistical technique that has found application in fields such as
  - face recognition
  - image compression
  - general pattern recognition
    - is a common technique for finding patterns in data of high dimension

When Is PCA Used?

- PCA is used prior to unsupervised and supervised machine learning steps to reduce the number of features used in the analysis, thereby reducing the likelihood of error

Mathematical Background

See last section of this material to review

essential mathematical background

required to understand the process of PCA

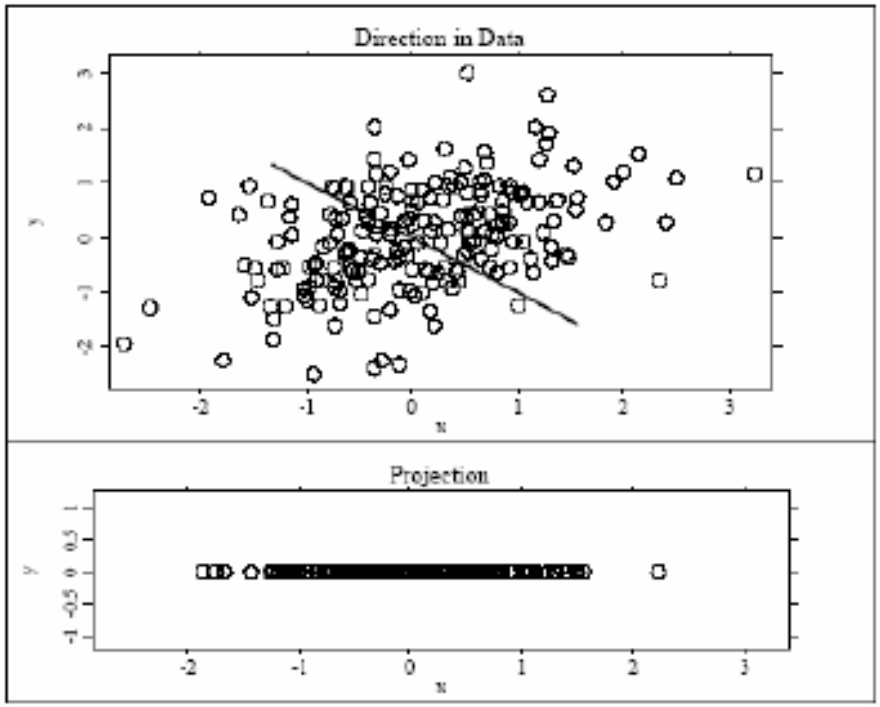
PCA: Method

- Reminder:
  - PCA is a way of identifying patterns in data, and *expressing* the data in such a way as to highlight their similarities and differences
  - Since patterns in data can be hard to find in data of high dimension (i.e. graphical representation no longer available), PCA is a powerful tool for analyzing data
  - Another advantage: once you have found these patterns in the data, you can compress the data (reduce number of dims) without much loss of info

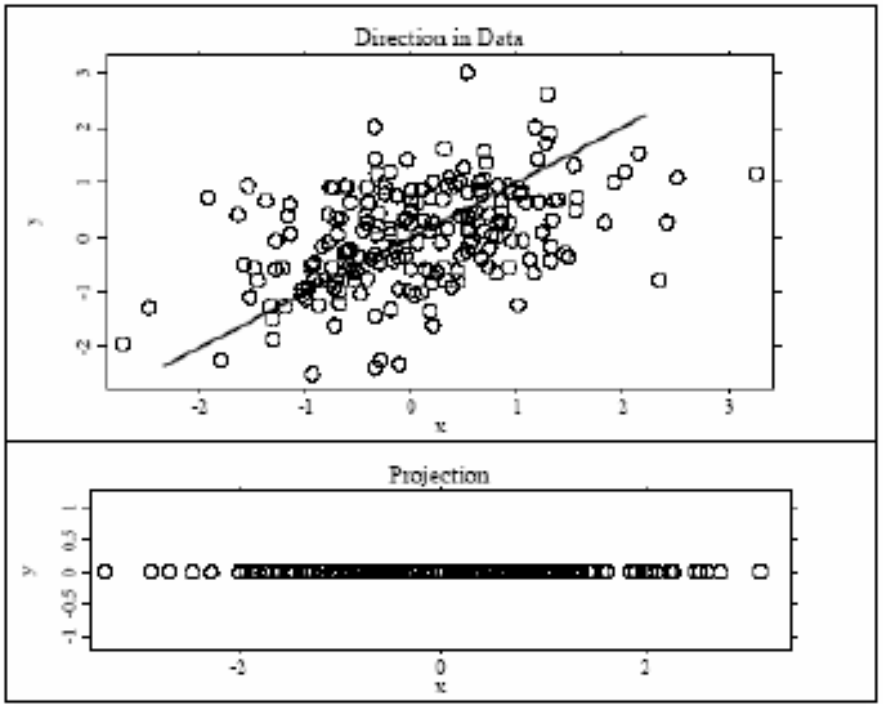
Good vs Better Projection

(remember eigenvectors are principal components!)

Good



Better



PCA Method

- Normalize variables [good idea; not always done]
  - Center (subtract data mean from each data point)
  - Scale
- Computer the covariance matrix of the transformed data
- Computer the eigenvectors and eigenvalues in decreasing order
- Choose the number of new dimensions and select the first 'd' eigenvectors
- Project the transformed data points on the first 'd' eigenvectors

PCA: Method

- Step 1: Get some data!
  - Toy example ~ small number of data points
  - 2 dimensions
  - … *why?* Ability to provide **plots** (easier to visualize)

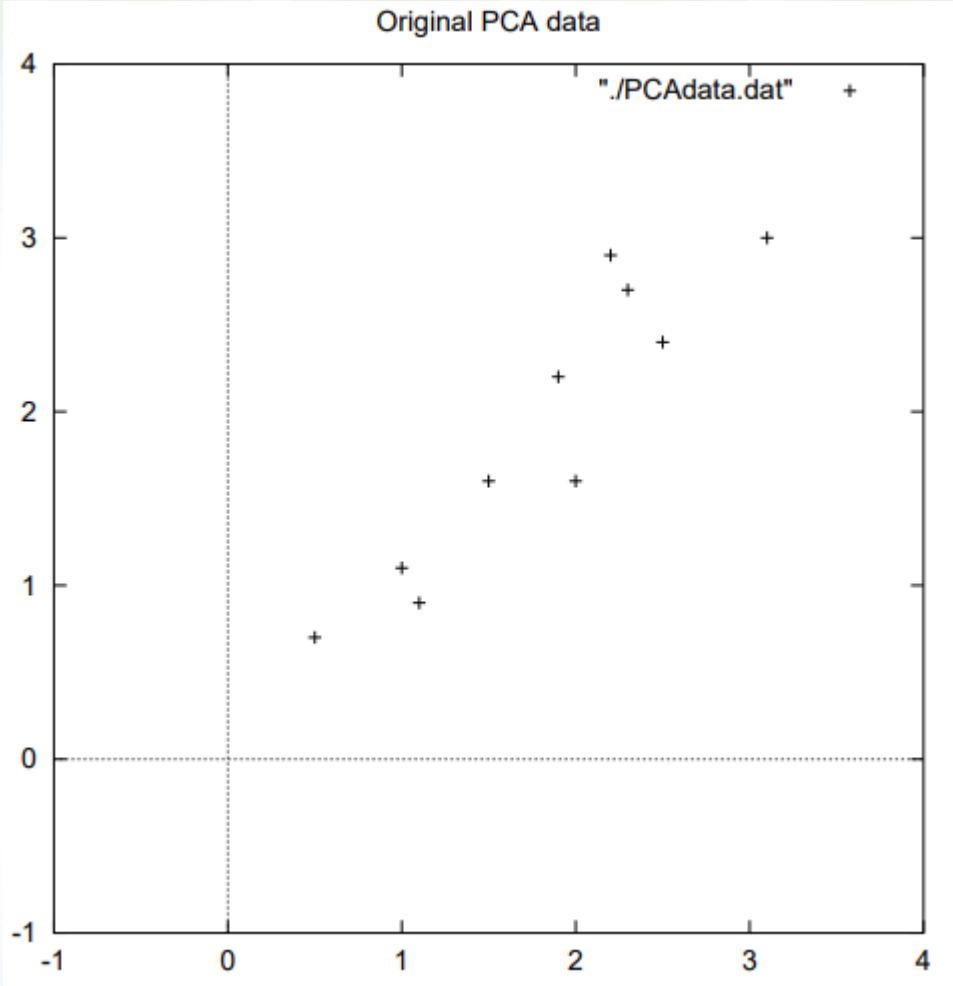
- Step 2: Subtract the mean
  - For PCA to work properly, you have to subtract the mean from each of the data dimensions
  - The mean subtracted is the average across each dim
  - E.g. all x values have $\bar{x}$ (the mean of the x values of all the data points) subtracted

Data =

x	y
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	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01



PCA: Method

- Step 3: Calculate the covariance matrix
  - (See background information for reminder on how to do this) Given this is a 2-D data set, the covariance matrix will be 2 x 2

$$cov = \begin{pmatrix} 0.61656 & 0.61544 \\ 0.61544 & 0.71656 \end{pmatrix}$$

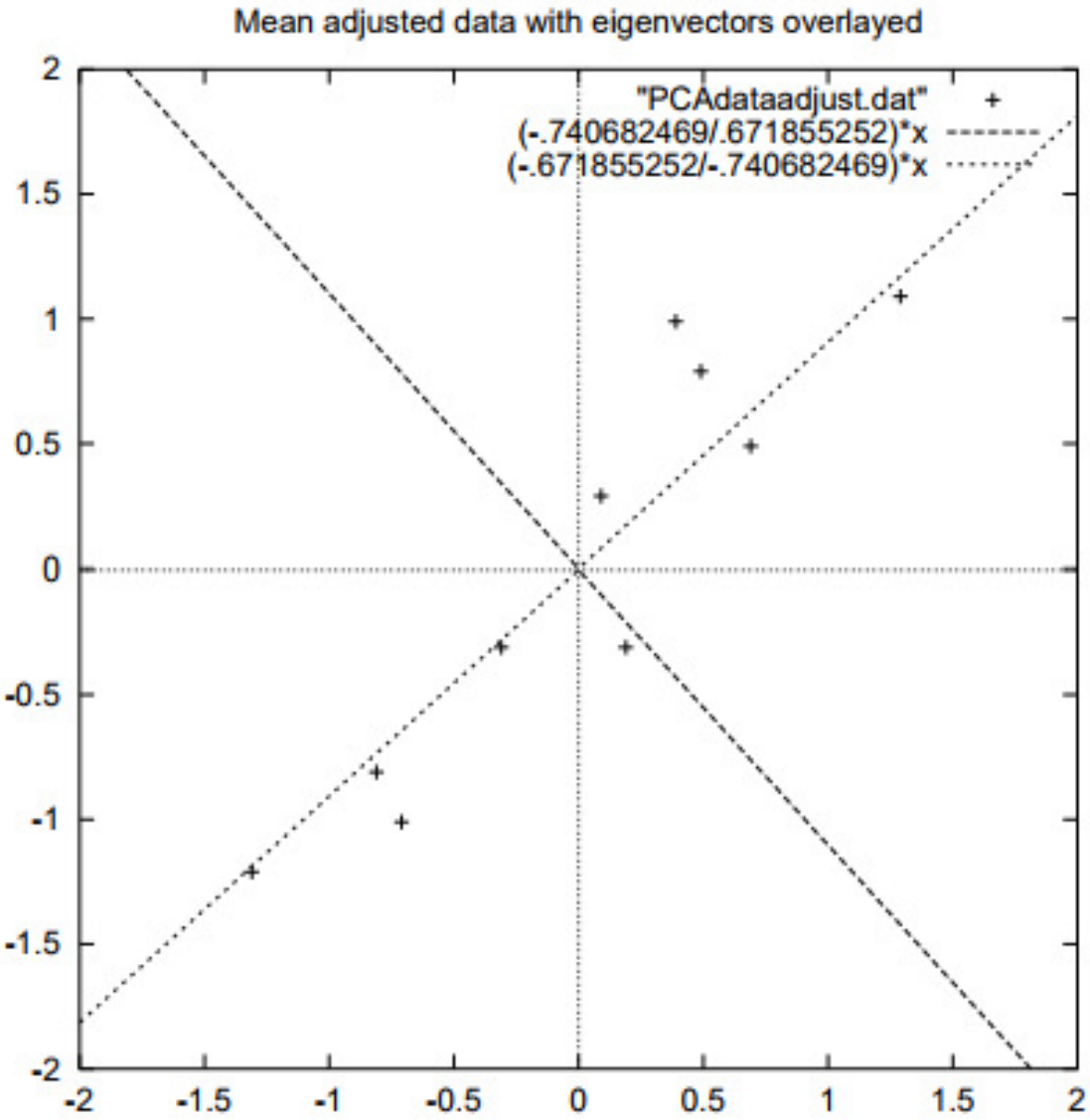
  - Since the non-diagonal elements are positive, we conclude that both the x and y variables increase together

PCA: Method

- Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix
  - Covariance matrix is square –the eigenvectors and eigenvalues can be calculated
  - It is important to notice that the eigenvectors are both unit eigenvectors (i.e. length = 1) This is important for PCA. Luckily programming packages will give you unit eigenvectors

$$eigenvalues = \begin{pmatrix} 0.04908 \\ 1.2840 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -0.73518 & -0.67787 \\ 0.67787 & -0.73518 \end{pmatrix}$$



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

- As expected from the covariance matrix, the two variables do indeed increase together
- Eigenvectors plotted (diagonal dotted lines on the plot)
- The eigenvectors are perpendicular to each other
- Provide us with information about the patterns in the data
- By this process we've been able to extract lines that characterize the data

PCA: Method

- Step 5: Choose components and forming a feature vector
  - This is where the notion of data compression and reduced dimensionality comes into the picture
  - Turns out, the eigenvector with the highest eigenvalue is the *principle component* of the data set
  - Once eigenvectors are found from the covariance matrix, the next stop is to order them by eigenvalue, *highest to lowest* → components in order of significance
  - If you leave out some components, the final data will have less dimensions, and you will lose some info (if eigenvalues are small, you won't lose much)

PCA: Method

- Step 5: (cont'd)
  - Next, form a feature vector
    - A matrix of vectors
    - Matrix of eigenvectors you want to keep (columns)

    $FeatureVector = (\text{eig}_1 \ \text{eig}_2 \ \text{eig}_3 \ \dots \ \text{eig}_n)$
  - Given this example: there are 2 eigenvectors. ***Either*** we can form a feature vector with both of the eigenvectors:

$$FeatureVector = \begin{pmatrix} -0.67787 & -0.73518 \\ -0.73518 & 0.67787 \end{pmatrix}$$

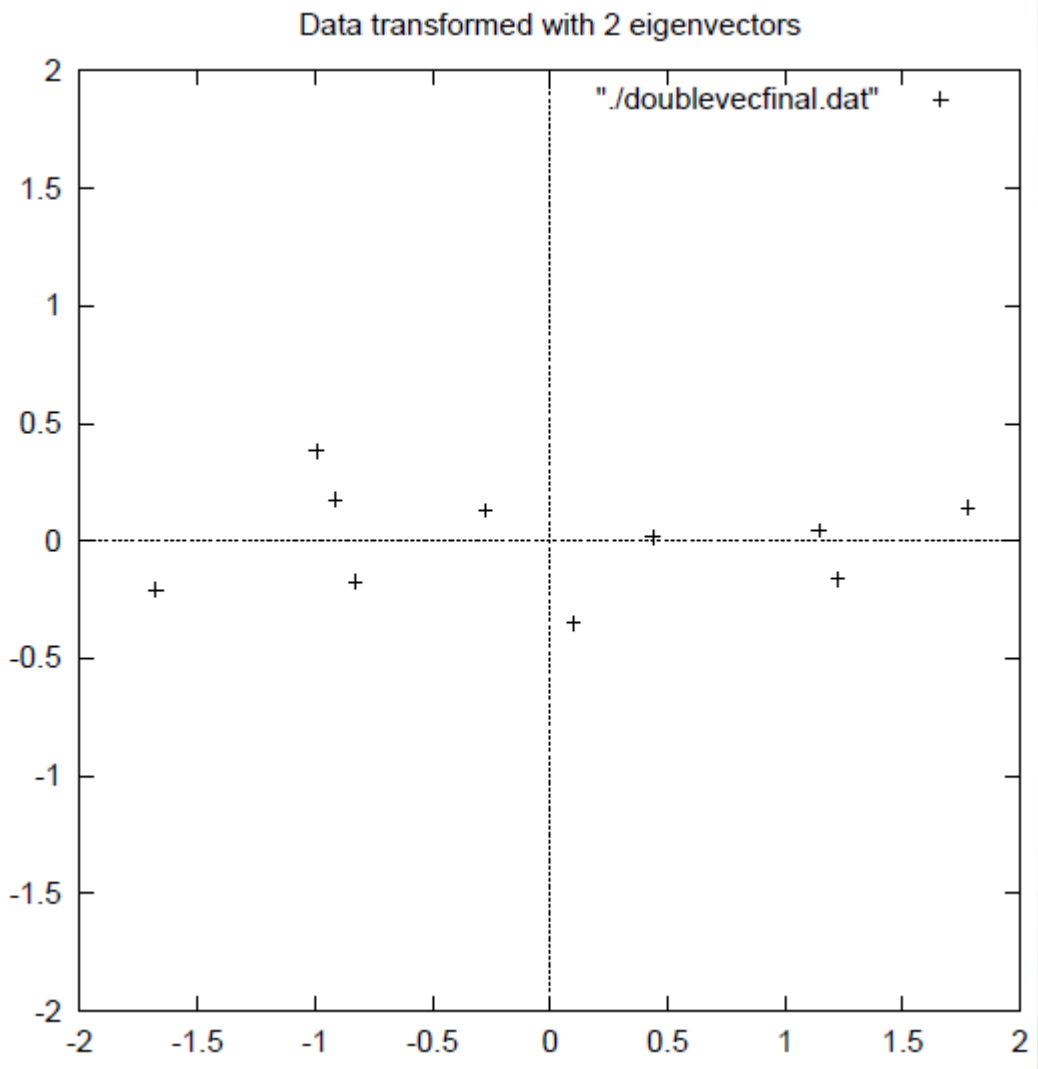
  - ***Or*** leave out the smaller, less significant component and only have a single column

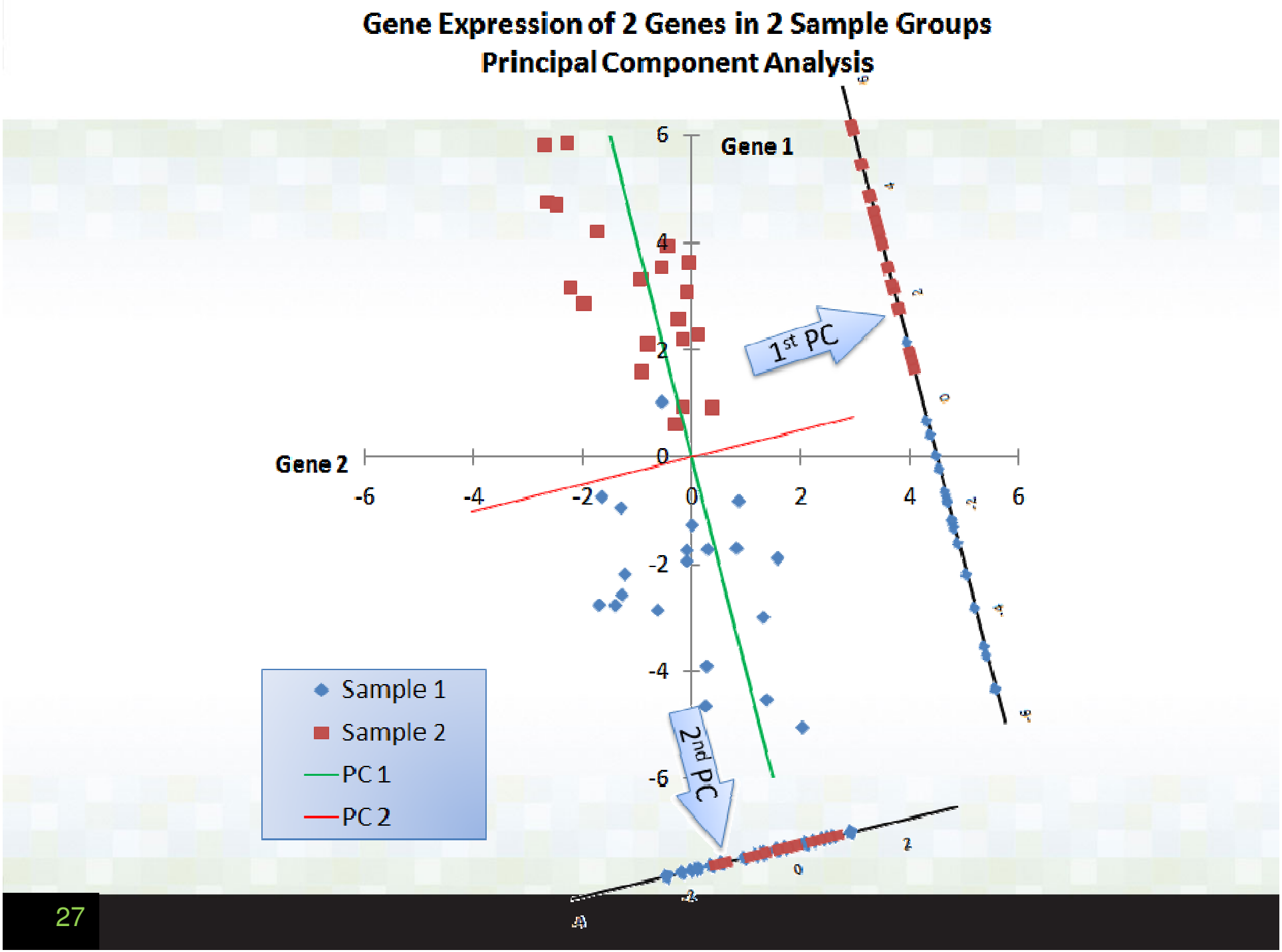
$$FeatureVector = \begin{pmatrix} -0.67787 \\ -0.73518 \end{pmatrix}$$

PCA: Method

- Step 6: Deriving the new data set
  - Once we have chosen the eigenvectors we wish to keep in our data and formed a feature vector, we transform the data
  - We simply take the transpose of the vector and multiply it on the left of the original data set, transposed
  - Transformed data will give us the original data *solely in terms of the vectors we chose*
  - Data no longer in terms of x- and y- axis but *in terms of the eigenvectors* (only in terms of vectors we decided to keep)

Example showing PCA analysis using both eigenvectors




Gene Expression of 2 Genes in 2 Sample Groups
Principal Component Analysis
Gene 1
Gene 2
1st PC
2nd PC
Sample 1
Sample 2
PC 1
PC 2
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

Main Limitation of PCA

- The main limitation of PCA is that it does not consider class separability since it does not take into account the class label of the feature vector
  - PCA simply performs a coordinate rotation that aligns the transformed axes with the directions of maximum variance
  - *There is no guarantee that the directions of **maximum variance** will contain good features for **discrimination***

Mathematical Background

Essential elementary mathematics required to understand the process of Principal Components Analysis (PCA)

Standard Deviation

- n = the number of elements in the set X
- $\bar{X}$ = is the mean of the set X ("X-bar")
- s = standard deviation

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

- We all have a good idea about what standard deviation is so I won't go into further details. Simply put, the Standard Deviation of a data set is a measure of how spread out the data is

Variance

- Variance is another measure of the spread of the data in the data set.

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

- It is simply the standard deviation squared (s^2)
- Both standard deviation and variance are measure of the spread of the data. Standard deviation is the most common measure, but variance is also used
- Variance is introduced to provide a solid background for the next topic (*covariance*)

Covariance

- ***Standard deviation*** and ***variance*** are measures that are purely 1-dimensional
- Many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions
- Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set *independently* of the other dimensions

Covariance

- It is useful to have a similar measure to find out how much the dimensions vary from the mean *with respect to each other*
- Covariance is always measured between 2 dimensions
- If you calculate the covariance between one dimension and *itself*, you get the variance! (cov(X,X)=variance)
- Here is the formula for covariance:

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

Covariance

- For each data item, multiply the difference between the x value and the mean of x, by the difference between the y value and the mean of y. Add all these up, and divide by (n-1)

- Exact value is not as important as **it's sign** (i.e. positive or negative

Covariance

- Exact value is not as important as **it's sign** (i.e. positive or negative)
  - If it's positive: both dim's *increase together*, as X goes up, Y goes up
  - If it's negative: as one dim goes up the *other dim goes down*, as X goes up, Y goes down
  - If it's zero: the *two dims are independent of each other* (no correlation between the dims)
- Note that $cov(X, Y) = cov(Y, X)$

  (Look at formula… multiplication is commutative!)

A Utility of Covariance

- The luxury of being able to visualize data is only available at 2 and 3 dimensions

- Since the covariance value can be calculated between any 2 dimensions in a data set, this technique is often used to find relationships between dimensions in high-dimensional data sets where visualization is difficult

The Covariance Matrix

- Recall that covariance is always measured between 2 dimensions

- For an n-dimensional data set, you can calculate $\frac{n!}{(n-2)!*2}$ different covariance values

- A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a **matrix**

The Covariance Matrix

- Covariance matrix for a set of data with n dimensions is:

$$C^{n\times n} = \left(c_{i,j}, \qquad c_{i,j} = cov\left(Dim_i, Dim_j\right)\right)$$

- Where $C^{n\times n}$ is a matrix with n rows and n columns, and Dim_x is the xth dimension

The Covariance Matrix

- For a covariance matrix for data with "n" dimensions:
  - n x n matrix (square)
  - Each entry is the cov between 2 dims (e.g. entry on row 2, col 3, is the **cov value** calculated between 2nd dim and 3rd dim
  - Diagonal: cov between one of the dims and itself → variances for that dim i.e. cov(X,X)
  - Reminder: cov(X,Y)=cov(Y,X) so this matrix is SYMMETRICAL about the main diagonal

The Covariance Matrix

Let's make a covariance matrix for an imaginary

***3-dimensional** data set, using the usual dimensions x, y, and z. This covariance matrix has 3 rows and 3 columns, and the values are:*

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Matrix Algebra

- In this section will present a background for the matrix algebra required in PCA

- Specific topics covered here will be centered on eigenvectors and eigenvalues of a given matrix

I am assuming you have some basic knowledge of matrices

Eigenvectors

NOT AN EIGENVECTOR

TRANSFORMATION MATRIX

Figure 1 (a and b):

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \end{pmatrix}$$

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

AN EIGENVECTOR

IS an integer multiple of the original vector

Figure 2 (a and b):

$$2 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix} \text{ (scale the } \begin{pmatrix} 3 \\ 2 \end{pmatrix} \text{ vector first)}$$

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

EIGENVALUE

Eigenvectors

- In Figure 1a, the resulting vector is not an integer multiple of the original vector, whereas in Figure 1b, the resulting vector is exactly 4 times the original vector
- The square matrix can be thought of as a **transformation matrix**
- It is the nature of the transformation that the eigenvectors arise from

  [Transformation Square Matrix][Vector1] = [Vector2]

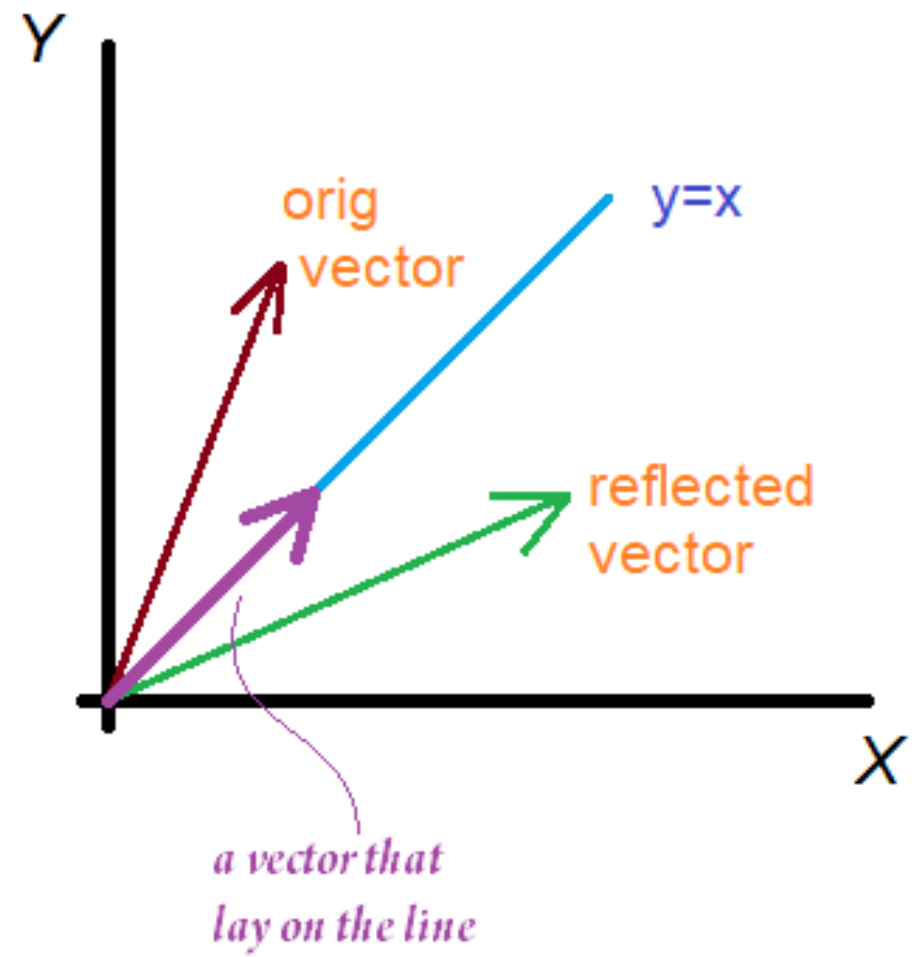
  Vector two has been transformed from its original position

Eigenvectors

- Let's assume a square transformation matrix multiplied on the left to a vector, transforms that vector in the following way: Reflects the vector in the line y=x
  [Sq Matrix][Vector] -> Reflected in y=x

- If a vector lies on this line (y=x), its reflection would be *itself*

- THIS vector (and all multiples of it) would be an **Eigenvector** of that transformation matrix


Y
orig
vector
y=x
reflected
vector
X
a vector that
lay on the line

Eigenvectors

- Can only be found for square matrices
- Not every square matrix has eigenvectors
- Given an nxn that does have eigenvectors, there are "n" of them (a 3x3 matrix will have 3 eigenvectors)
- All eigenvectors of a matrix are ***orthogonal*** / ***perpendicular*** (i.e. at right angles to each other) no matter how many dimensions
  - We can express the data in terms of these perpendicular eigenvectors (instead of in terms of x and y axes)

Eigenvectors

- If a vector is **scaled** before multiplying all you are doing is making it longer, not changing it's direction! (The multiplier will still be the same – see Figure 1b vs 2b)
- In order to keep eigenvectors standard, eigenvectors are found/scaled to have length 1 – as you know length of a vector doesn't affect whether it's eigenvector or not

Eigenvalues

- Eigenvalues are closely related to eigenvectors

- An eigenvalue is the amount by which the original vector was scaled after multiplication by the square (transformation) matrix
- In the previous example, the value was 4
- **4** is the eigenvalue associated with that eigenvector (see Figures 2a and 2b)

- Eigenvectors and eigenvalues always come in pairs

PCA Examples

PCA Example (1) Showing Eigenvectors *(run this FIRST!)*

PCA Example (2) Iris 3 Components

References:

~Principal Components Analysis ~ R. Gutierrez-Osuna ~ Pattern Analysis ~ TAMU

~statisticshowto.com ~ dimensionality

~Finney, D.J. (1977). "Dimensions of Statistics." Journal of the Royal Statistical Society. Series C (Applied Statistics). 26, No.3, p.285-289

~Principal Component Analysis ~ Tutorial ~ Districtdatalabs

~A tutorial on PCA ~ L.I.Smith ~ 2002