NORTH CAROLINA STATE UNIVERSITY
College of Engineering
Department of Computer Science

Lecture 4: Data Mining Data (III)

Dr. Min Chi
Department of Computer Science
North Carolina State University
mchi@ncsu.edu

Outline

- Types of Data
  - Attributes
  - Types of Datasets
- Data Quality
- Measure of Similarity and Dissimilarity
- **Data Preprocessing**

Data Preprocessing

- Aggregation
- Sampling
- Discretization and Binarization
- Attribute Transformation
- Feature creation
- Feature subset selection
- Dimensionality Reduction

Iris Data

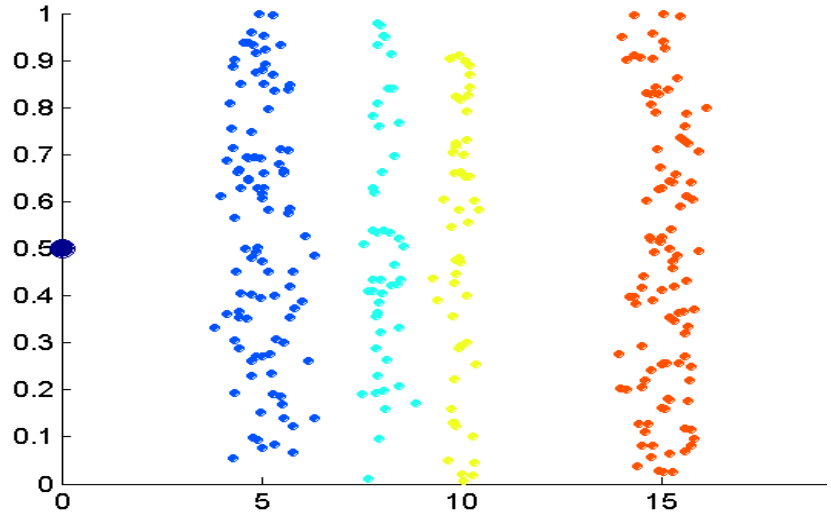
ID	sepal length	sepal width	petal length	petal width	class
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
…	…	…	…	…	Iris-setosa
51	7	3.2	4.7	1.4	Iris-versicolor
52	6.4	3.2	4.5	1.5	Iris-versicolor
53	6.9	3.1	4.9	1.5	Iris-versicolor
…	…	…	…	…	Iris-versicolor
101	6.3	3.3	6	2.5	Iris-virginica
102	5.8	2.7	5.1	1.9	Iris-virginica
103	7.1	3	5.9	2.1	Iris-virginica
104	6.3	2.9	5.6	1.8	Iris-virginica
…	…	…	…	…	Iris-virginica

Discretization

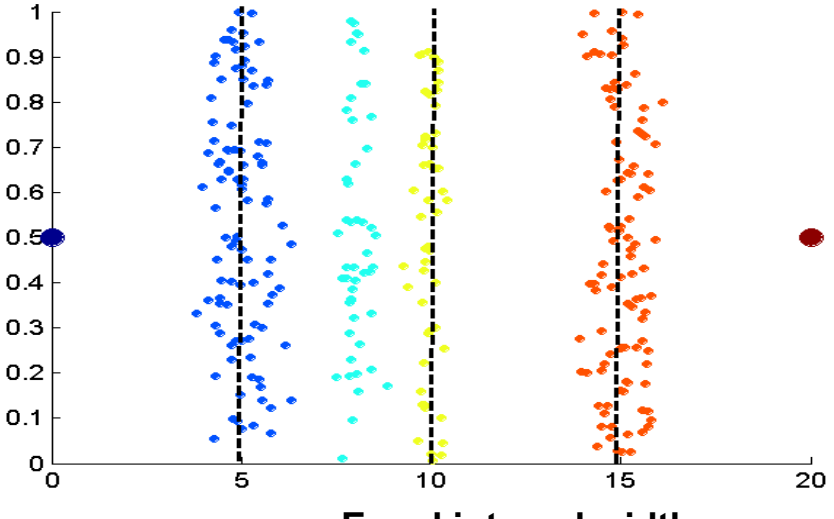
- We discretized the **petal width and length** to have categorical values:

  low, medium, and high

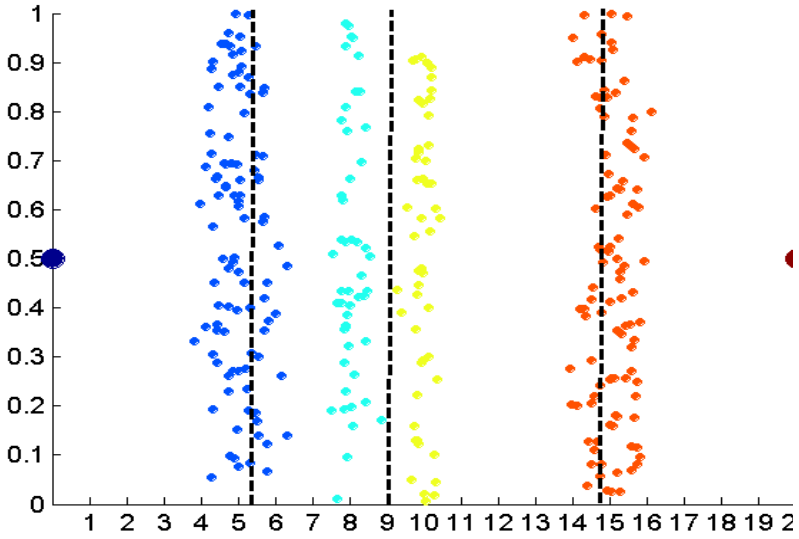
Discretization Without Using Class Labels



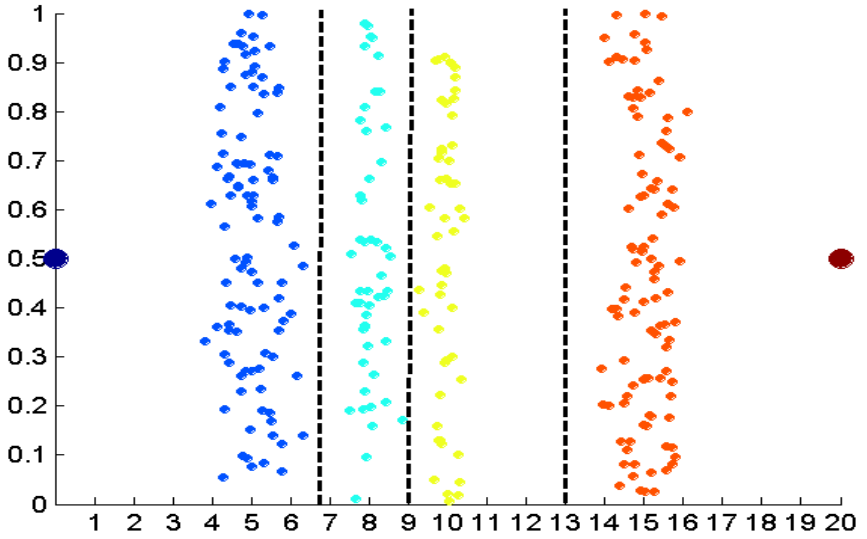
Data



Equal interval width



Equal frequency

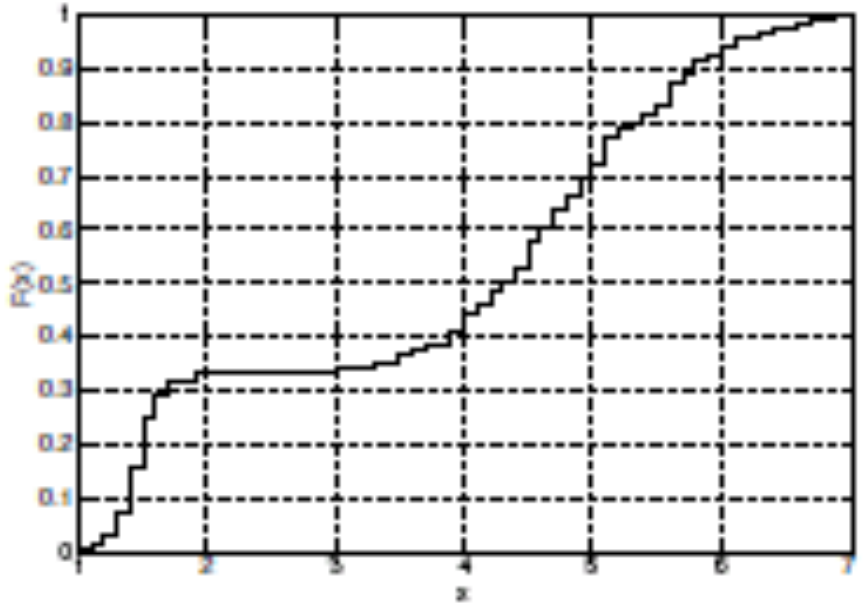


K-means

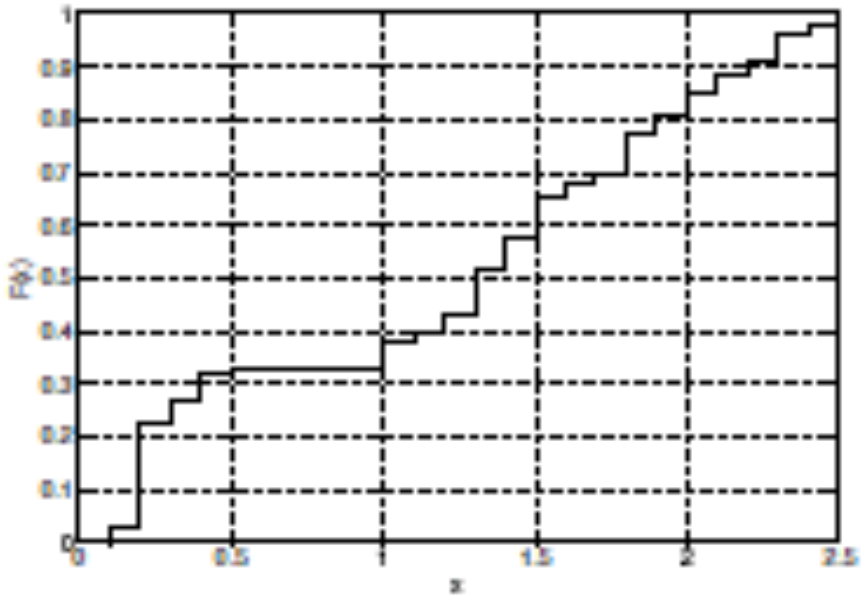
Discretization

- We discretized the **petal width and length** to have categorical values:

  low, medium, and high



(c) Petal Length.

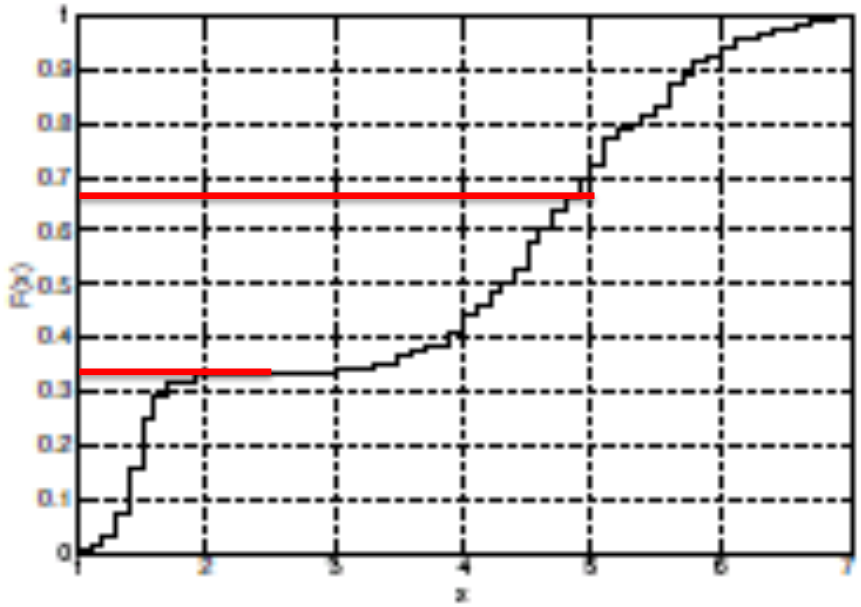


(d) Petal Width.

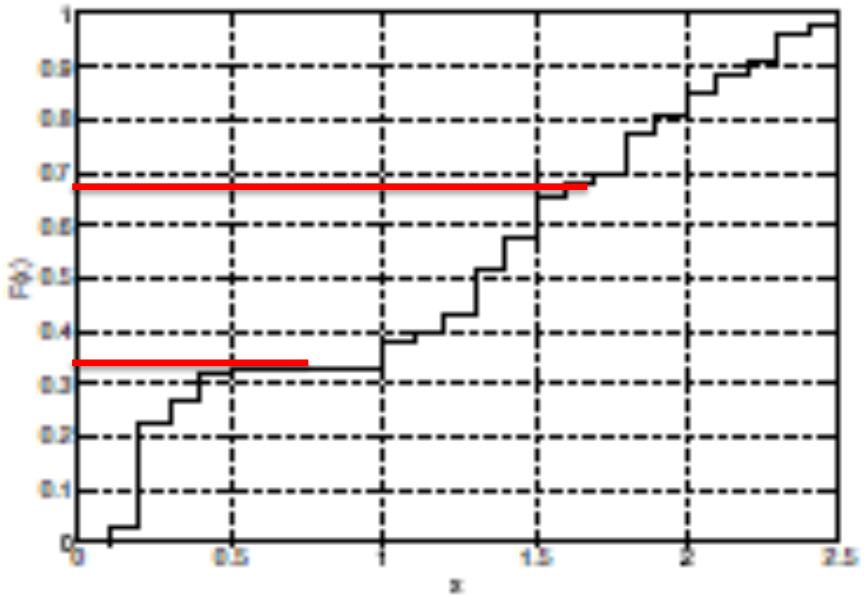
Discretization

- We discretized the petal width and length to have categorical values:

  low, medium, and high



(c) Petal Length.

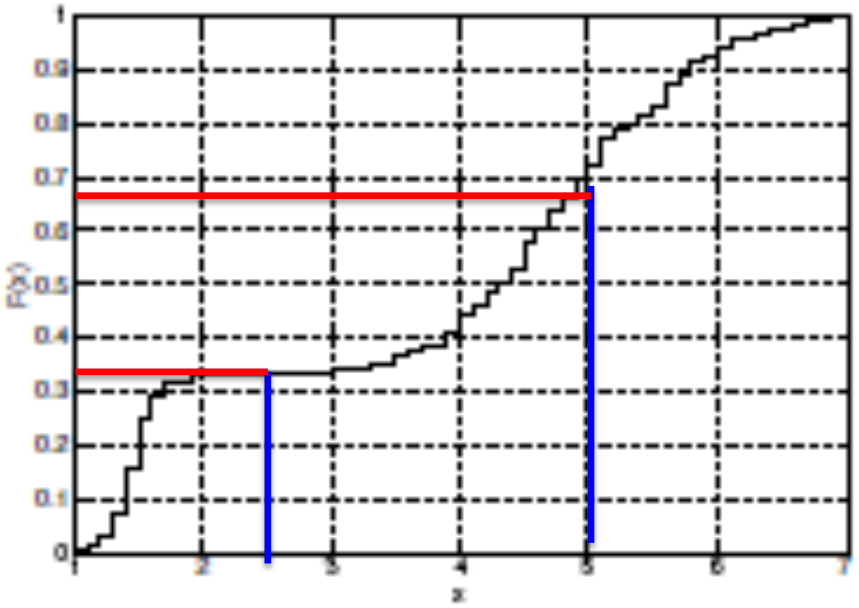


(d) Petal Width.

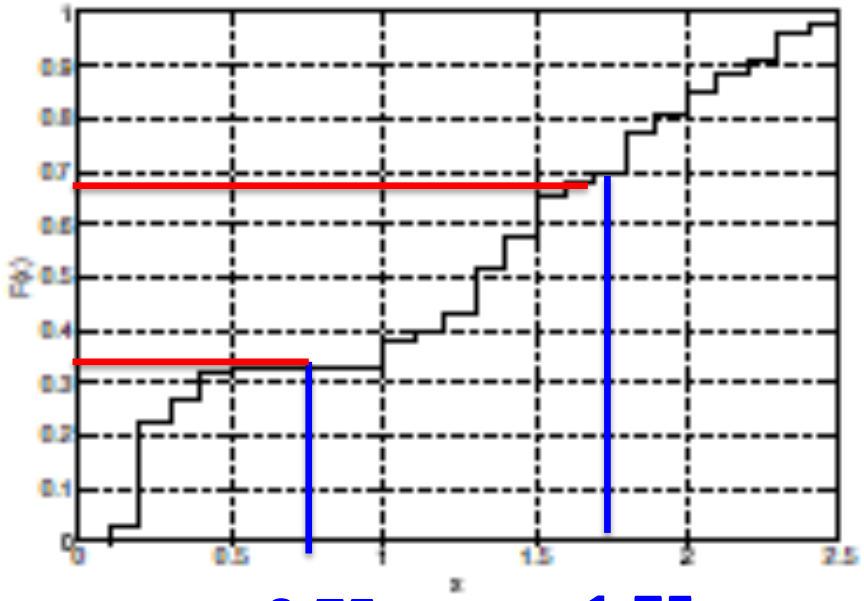
Discretization

- We discretized the petal width and length to have categorical values:

  low, medium, and high



(c) Petal Length.



(d) Petal Width.

Petal length	Petal width	Species
3.5	1	*I. versicolor*
4	1	*I. versicolor*
4.5	1.5	*I. versicolor*
5	1.5	*I. virginica*
1.3	0.3	*I. setosa*
4	1.3	*I. versicolor*
4.4	1.3	*I. versicolor*
3.3	1	*I. versicolor*

Petal length	Petal width	Species
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*
High	Medium	*I. virginica*
Low	Low	*I. setosa*
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*

Petal length	Petal width	Species
3.5	1	*I. versicolor*
4	1	*I. versicolor*
4.5	1.5	*I. versicolor*
5	1.5	*I. virginica*
1.3	0.3	*I. setosa*
4	1.3	*I. versicolor*
4.4	1.3	*I. versicolor*
3.3	1	*I. versicolor*

Petal length	Petal width	Species
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*
High	Medium	*I. virginica*
Low	Low	*I. setosa*
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*
Medium	Medium	*I. versicolor*

Each unique tuple of petal width, petal length, and species type identifies one element of the array.
This element is assigned the corresponding count value.

Resulted a multidimensional array

Note the count attribute!

Petal Length	Petal Width	Species Type	Count
low	low	Setosa	46
low	medium	Setosa	2
medium	low	Setosa	2
medium	medium	Versicolour	43
medium	high	Versicolour	3
medium	high	Virginica	3
high	medium	Versicolour	2
high	medium	Virginica	3
high	high	Versicolour	2
high	high	Virginica	44

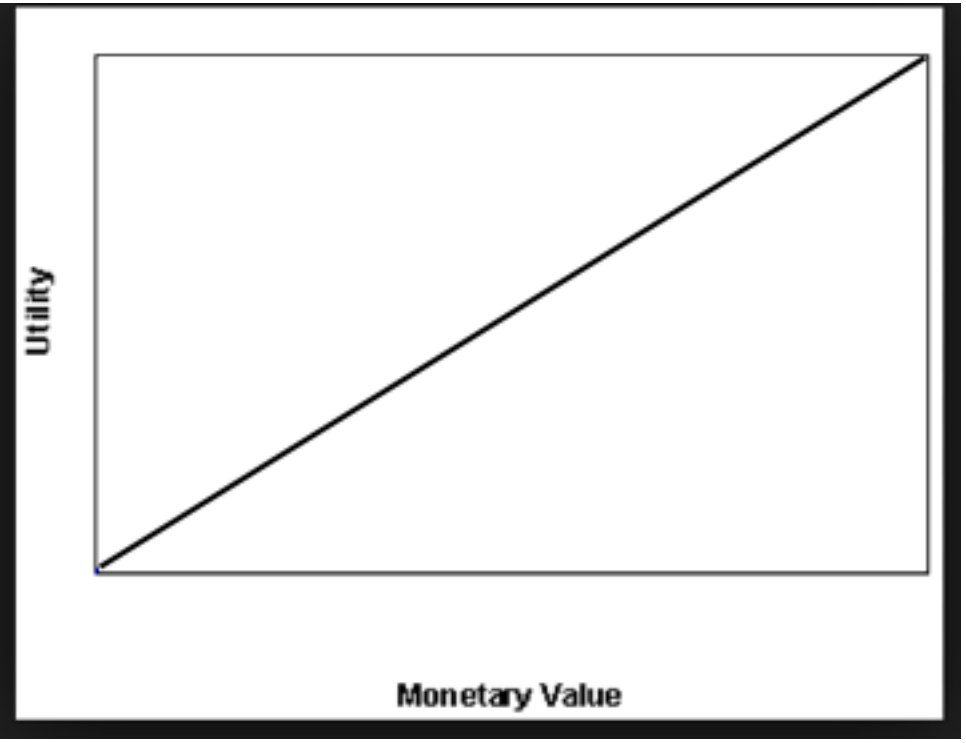
Data Preprocessing

- Aggregation
- Sampling
- Discretization and Binarization
- **Attribute Transformation**
- Feature creation
- Feature subset selection
- Dimensionality Reduction

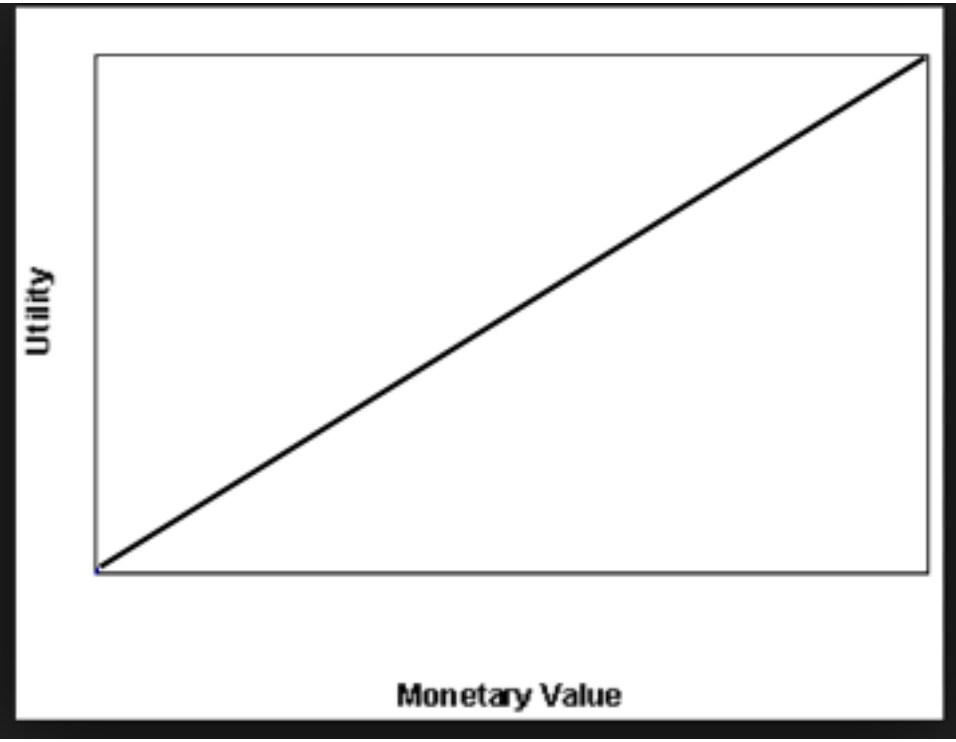
Attribute Transformation

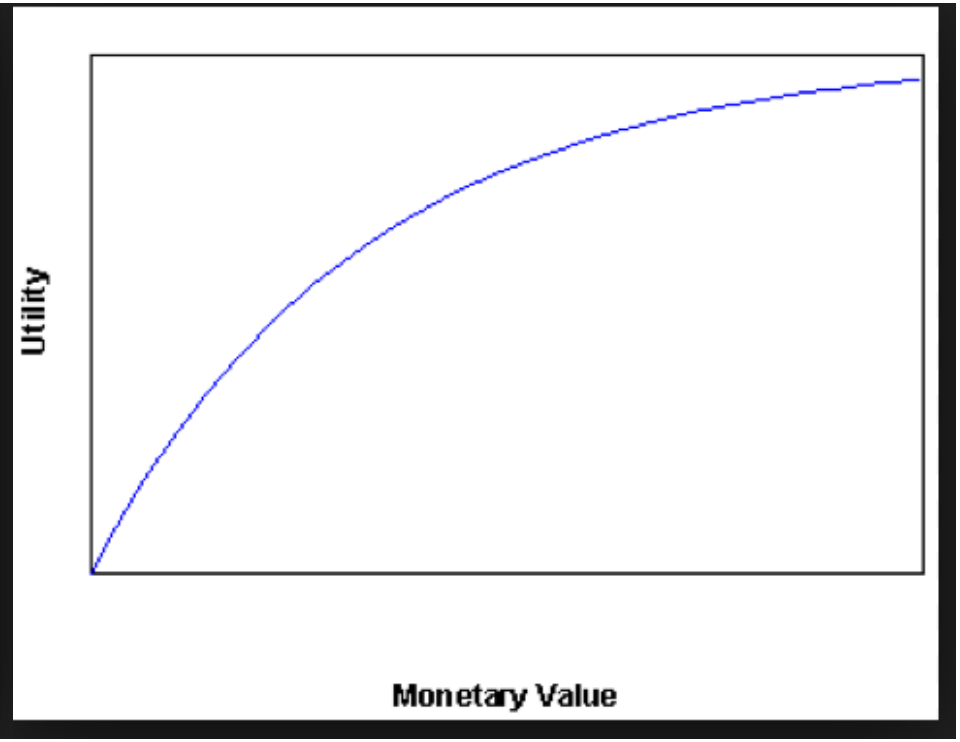
- A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
  - Simple functions: x^k, log(x), e^x, |x|
  - Standardization and Normalization

Attribute Transformation in Action

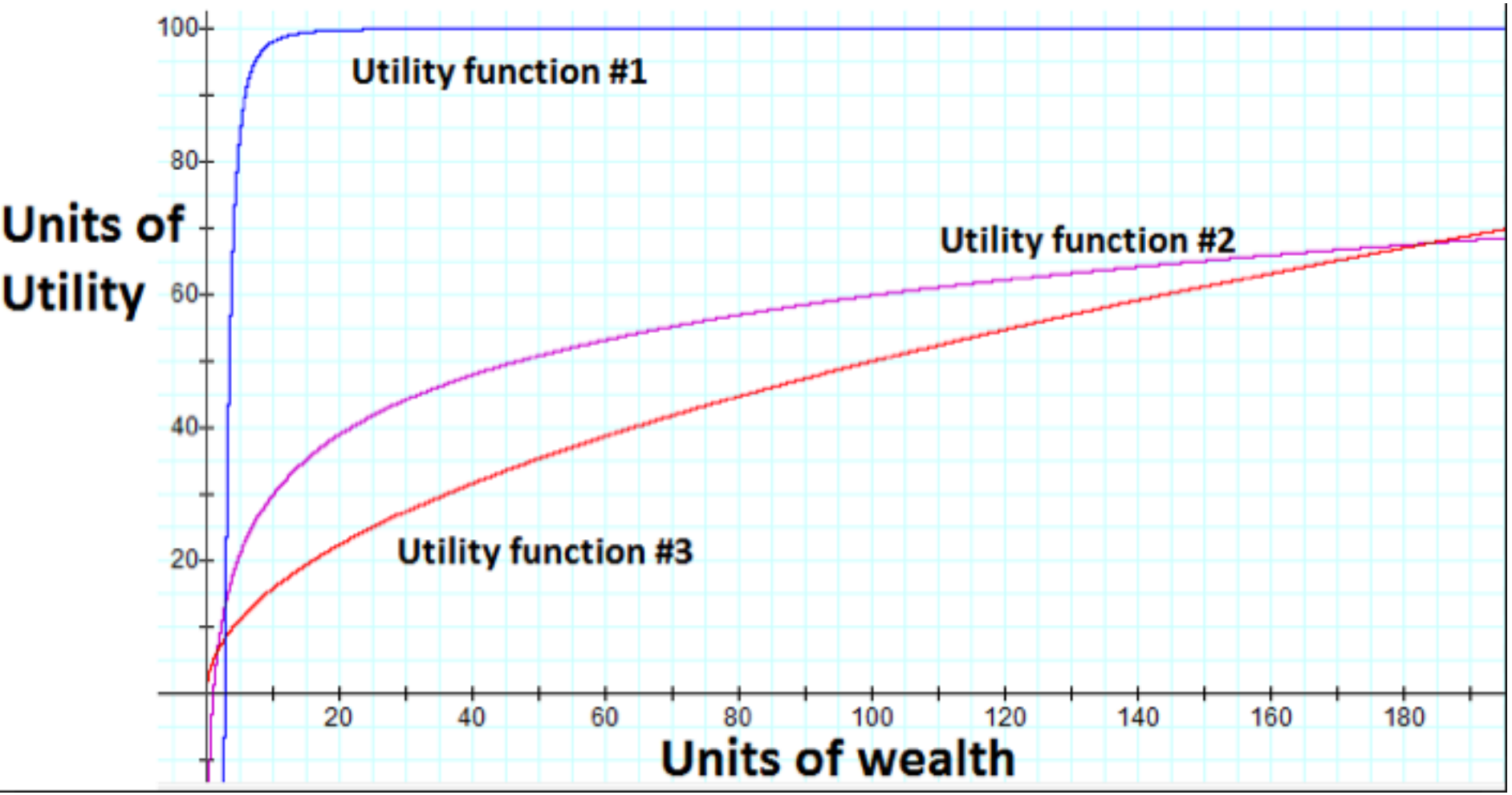


Attribute Transformation in Action





Try different ones!



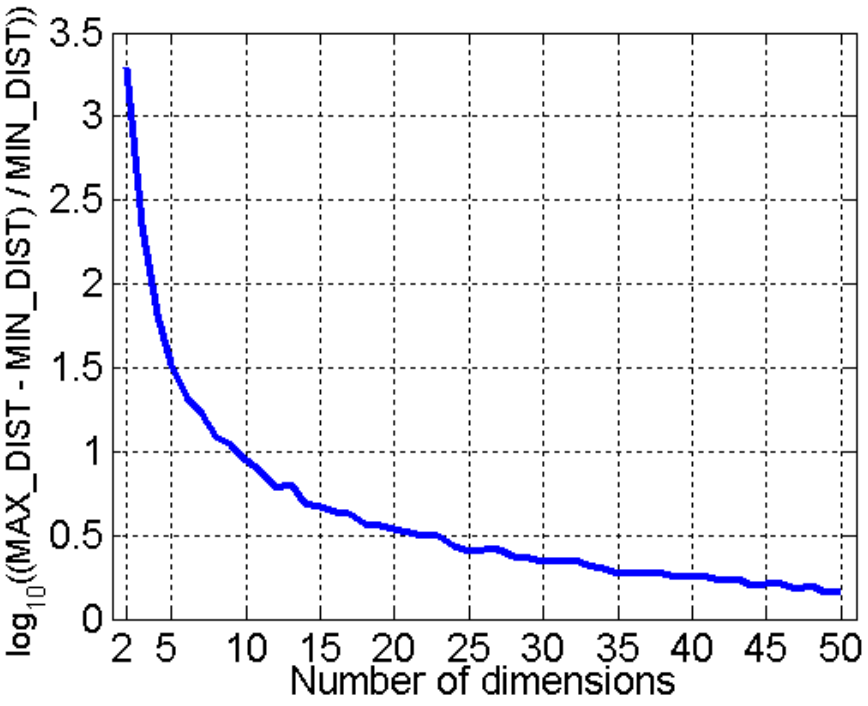
Data Preprocessing

- Aggregation
- Sampling
- Discretization and Binarization
- Attribute Transformation
- **Feature subset selection**
- Dimensionality Reduction

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies

- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



- **Randomly generate 500 points**
- **Compute difference between max and min distance between any pair of points**

Feature Subset Selection

- One important way to reduce dimensionality of data
- Redundant features
  - duplicate much or all of the information contained in one or more other attributes
  - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
  - contain no information that is useful for the data mining task at hand
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

Feature Subset Selection

- Techniques:
  - Brute-force approach:
    - Try all possible feature subsets as input to data mining algorithm
  - Embedded approaches:
    - Feature selection occurs naturally as part of the data mining algorithm
  - Filter approaches:
    - Features are selected before data mining algorithm is run
  - Wrapper approaches:
    - Use the data mining algorithm as a black box to find best subset of attributes

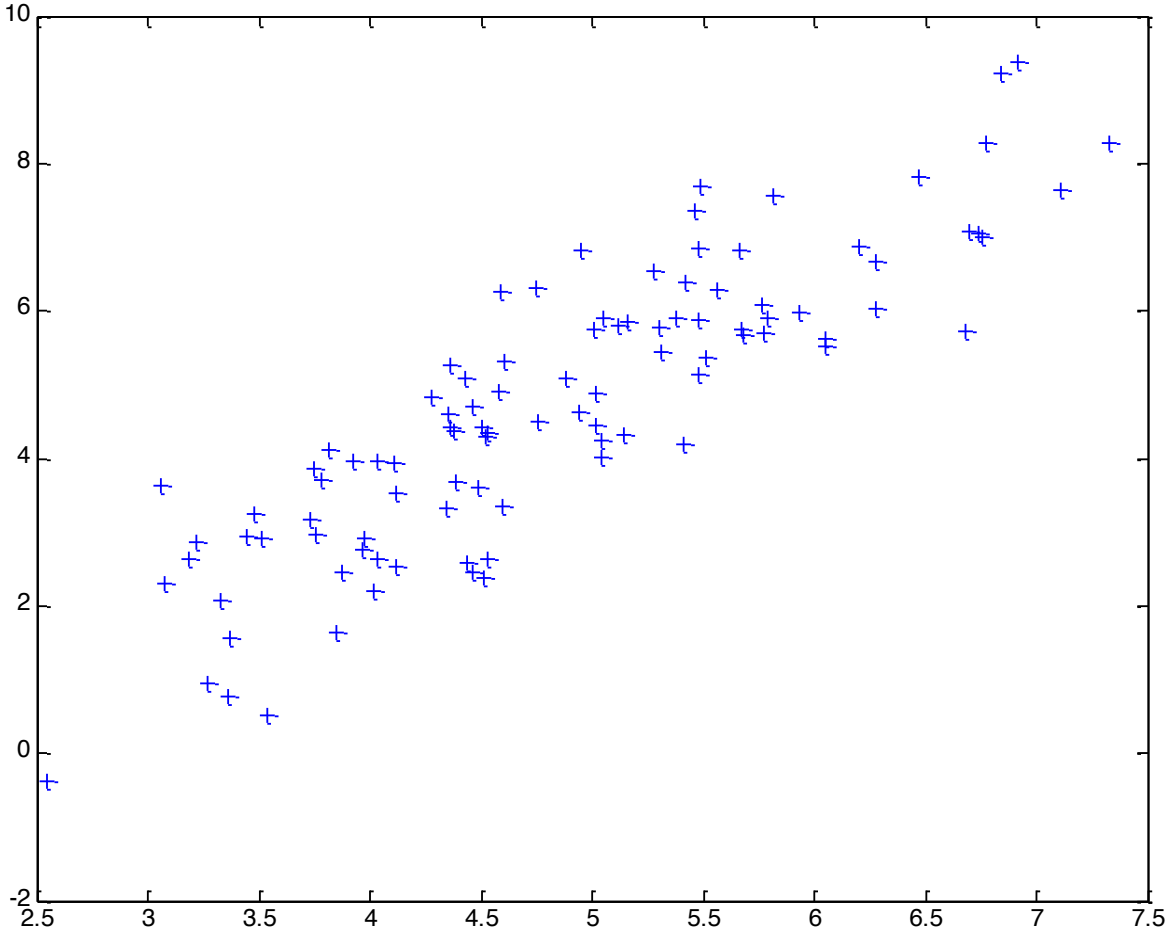
Data Preprocessing

- Aggregation
- Sampling
- Discretization and Binarization
- Attribute Transformation
- Feature subset selection
- **Dimensionality Reduction**

Dimensionality Reduction

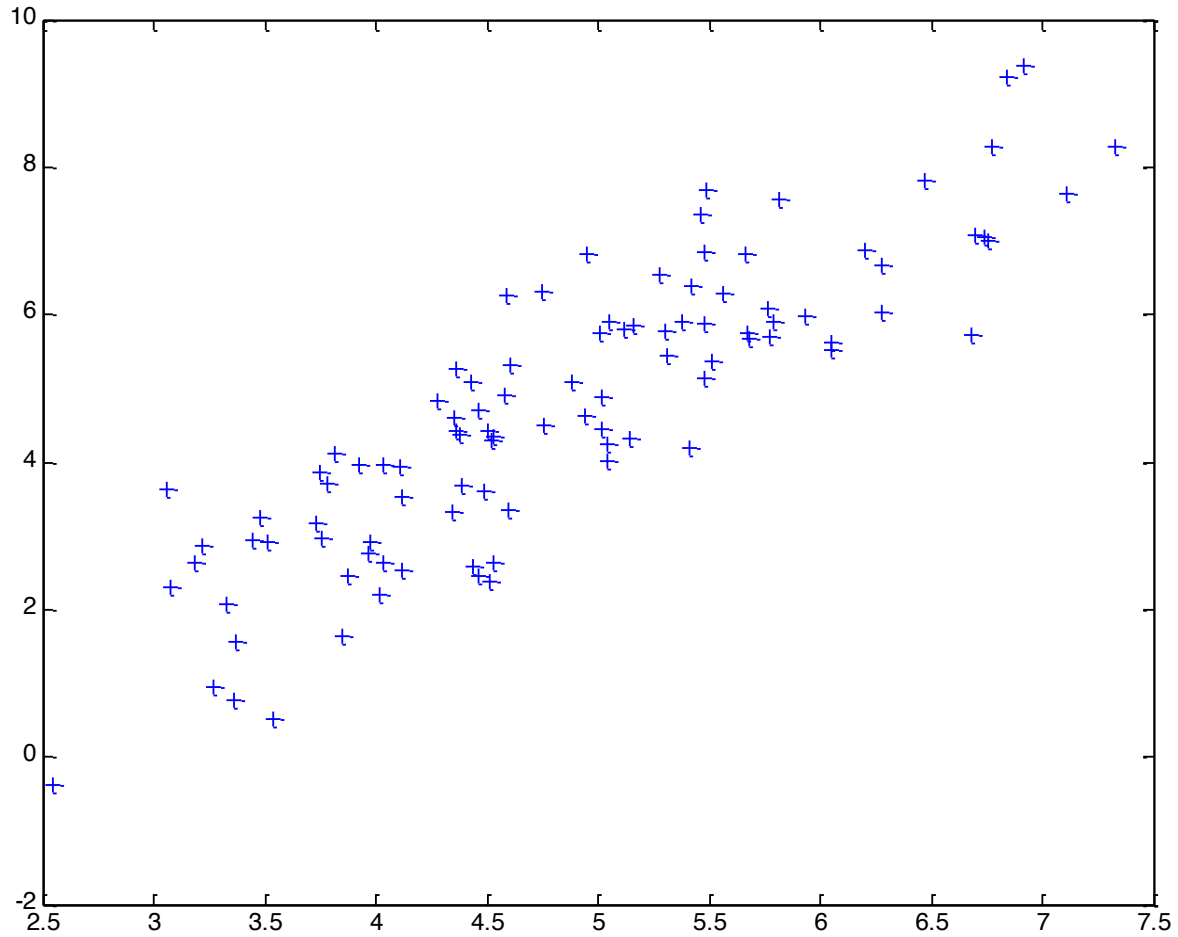
- Purposes:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise
- Techniques
  - Principal Component Analysis
  - Singular Value Decomposition
  - Others: supervised and non-linear techniques

2d Data



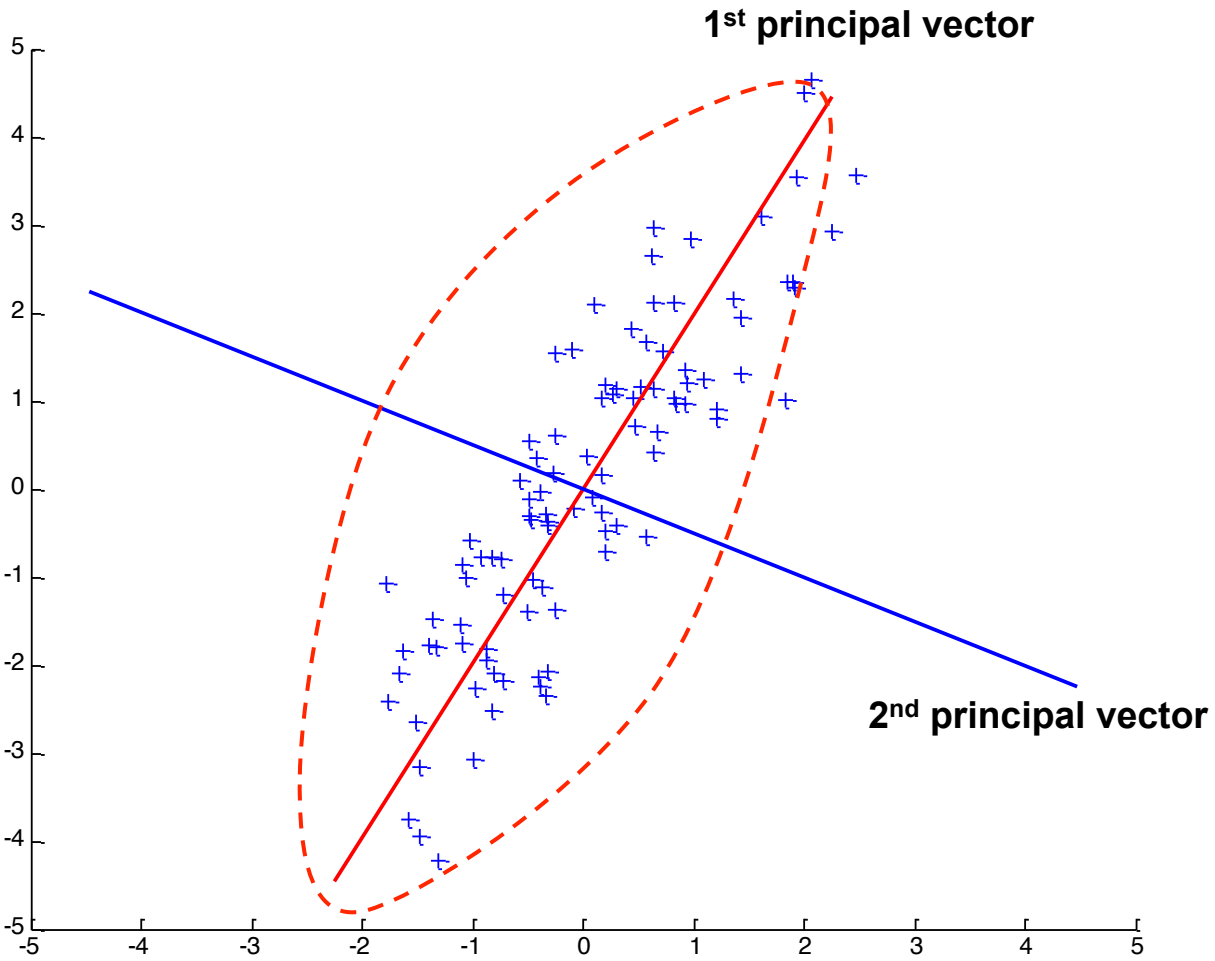
Dimensionality Reduction: PCA

- Goal is to find a projection that captures the largest amount of variation in data


10
8
6
4
2
0
-2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5

Principal Components

- Principal vectors are orthogonal
- Gives best axis to project

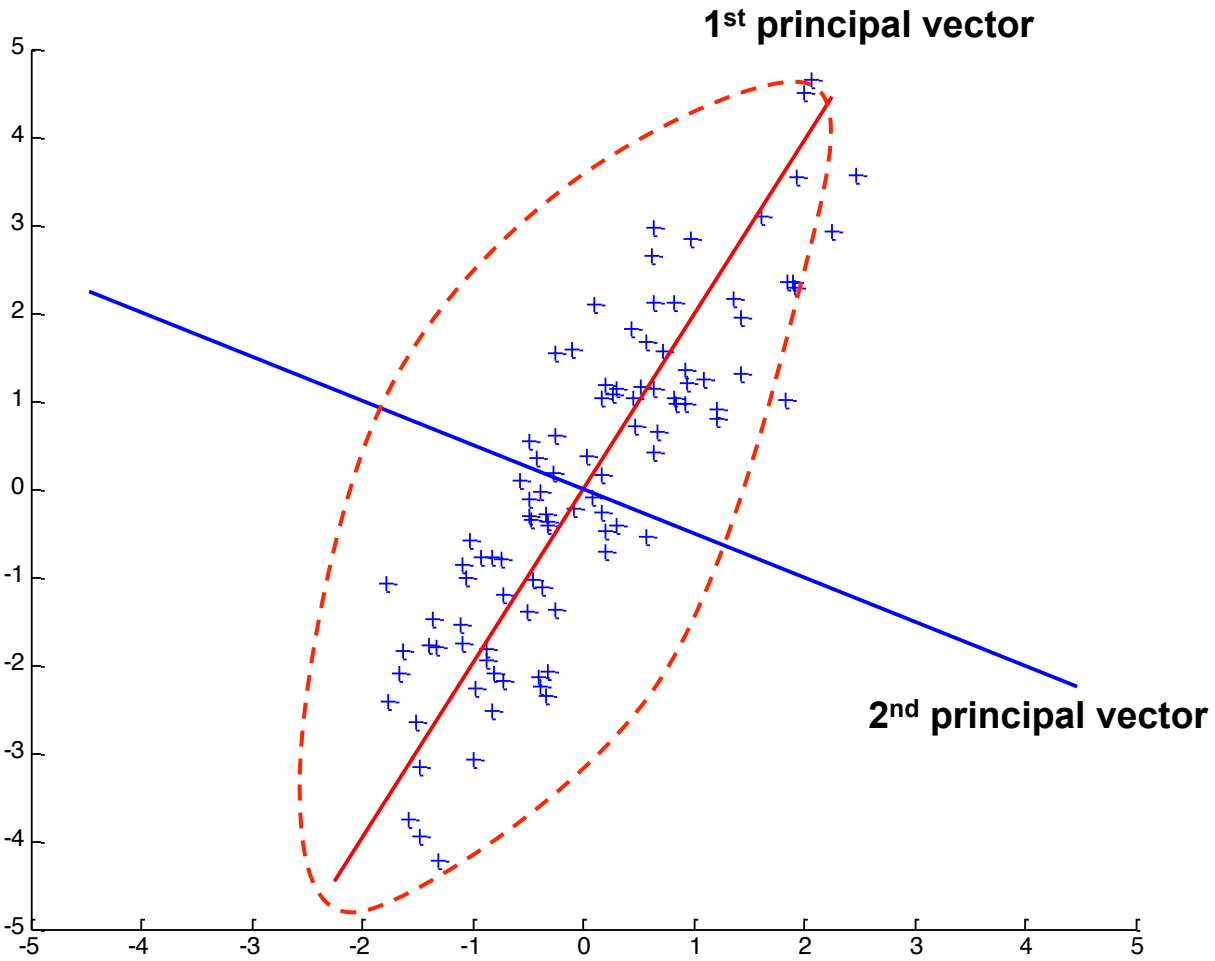


Dimensionality Reduction: PCA

- Find the eigenvectors of the covariance matrix
- The eigenvectors define the new space

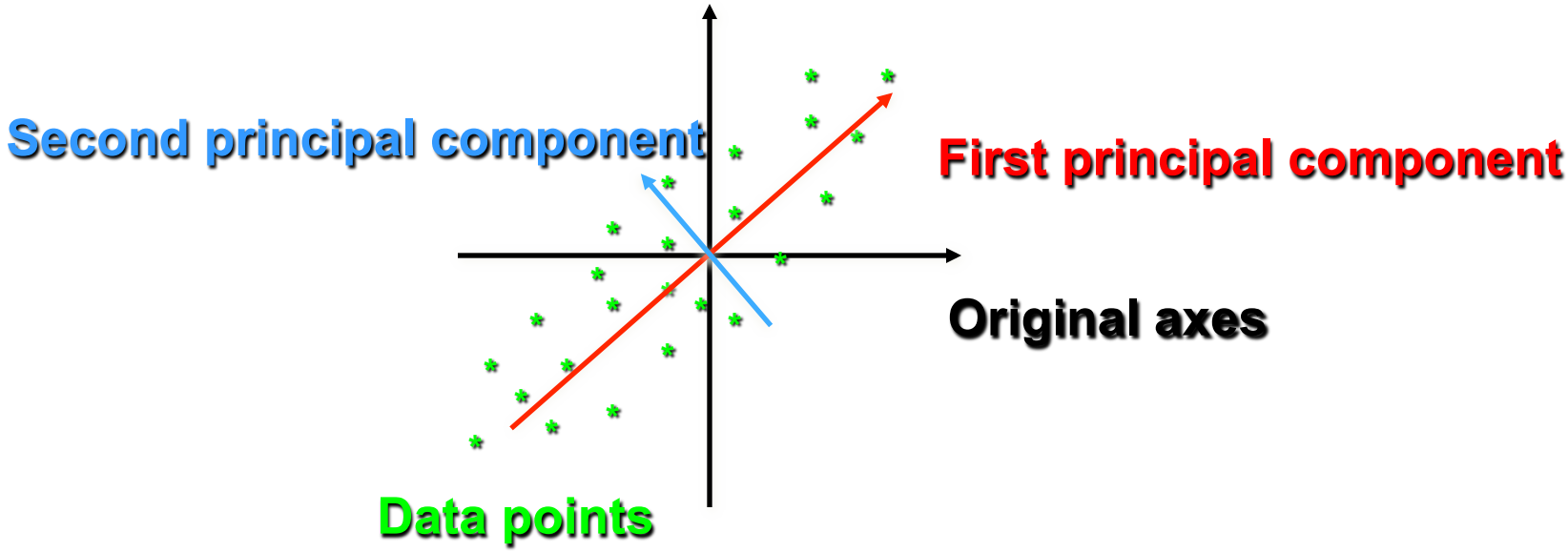
Principal Components

- Gives best axis to project
- Minimum RMS error
- Principal vectors are orthogonal

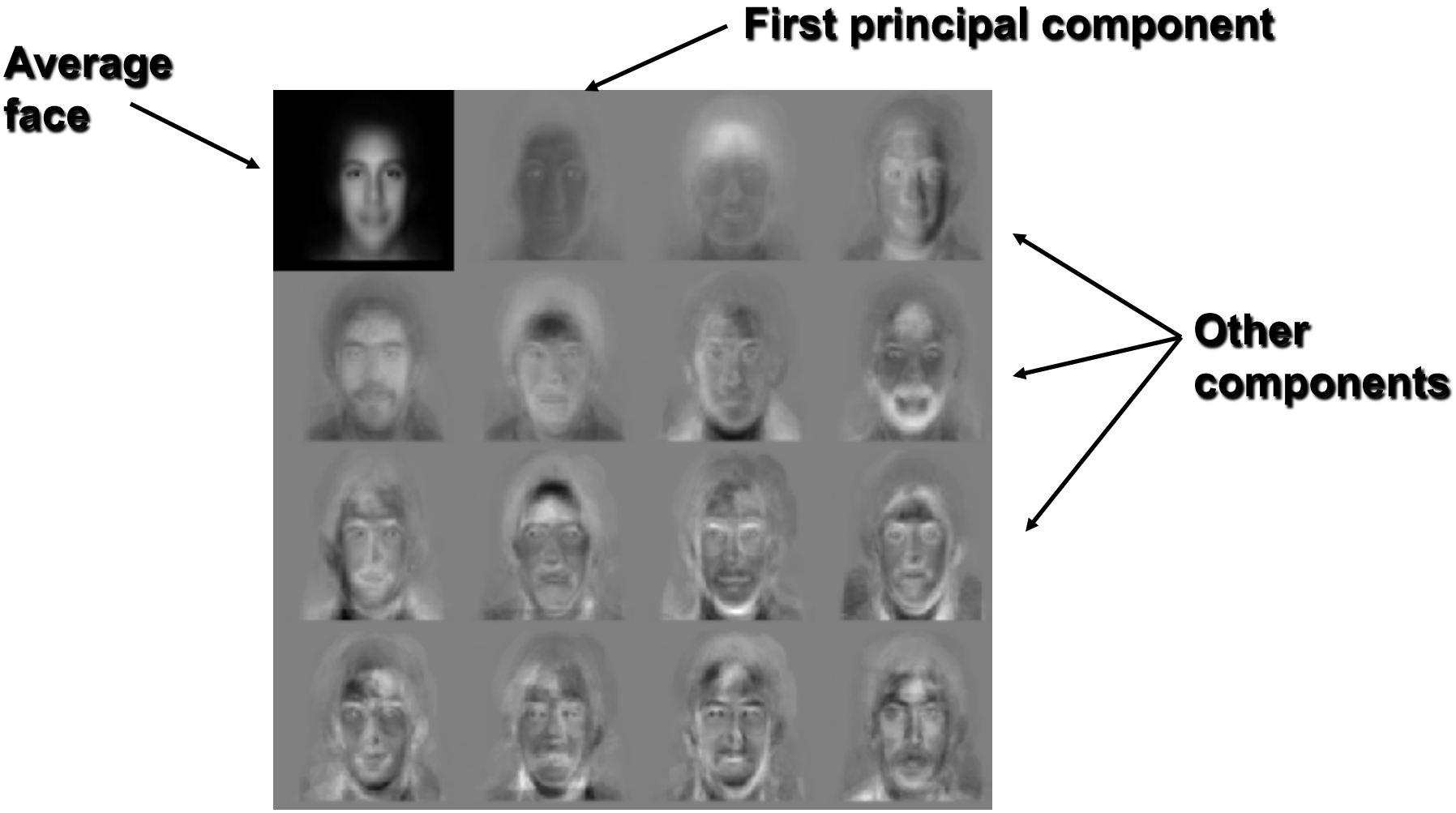


PCA

- Principal Components Analysis (PCA): approximating a **high-dimensional** data set with a lower-dimensional linear subspace



PCA on Faces: “Eigenfaces”



PCA for Relighting

- Images under different illumination

[Matusik & McMillan

PCA for Relighting

- Images under different illumination
- Most variation captured by first 5 principal components – can re-illuminate by combining only a few images

[Matusik & McMillan

PCA Example

Attributes	Type	Explain
manufact	Nominal	Manufacturer
model	Nominal	Model
sales	Ratio	Sales in thousands
resale	Ratio	4-year resale value
type	Ordinal	{0, Automobile} {1, truck} Vehicle type
price	Ratio	Price in thousands
engine_s	Ratio	Engine size
horsepow	Ratio	Horsepower
wheelbas	Ratio	Wheelbase
width	Ratio	Width
length	Ratio	Length
curb_wgt	Ratio	Curb weight
fuel_cap	Ratio	Fuel capacity
mpg	Ratio	Fuel efficiency

PCA Example

manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	length	curb_...	fuel_cap	mpg
Acura	RL	8.588	29.725	0	42.000	3.5	210	114.6	71.4	197	3.850	18.0	22
Audi	A4	20.397	22.255	0	23.990	1.8	150	102.6	68.2	178	2.998	16.4	27
Audi	A6	18.780	23.555	0	33.950	2.8	200	108.7	76.1	192	3.561	18.5	22
Audi	A8	1.380	39.000	0	62.000	4.2	310	113.0	74.0	198	3.902	23.7	21
BMW	323i	19.747	.	0	26.990	2.5	170	107.3	68.4	176	3.179	16.6	26
BMW	328i	9.231	28.675	0	33.400	2.8	193	107.3	68.5	176	3.197	16.6	24
BMW	528i	17.527	36.125	0	38.900	2.8	193	111.4	70.9	188	3.472	18.5	25
Buick	Century	91.561	12.475	0	21.975	3.1	175	109.0	72.7	195	3.368	17.5	25
Buick	Regal	39.350	13.740	0	25.300	3.8	240	109.0	72.7	196	3.543	17.5	23
Buick	Park A...	27.851	20.190	0	31.965	3.8	205	113.8	74.7	207	3.778	18.5	24
Buick	LeSabre	83.257	13.360	0	27.885	3.8	205	112.2	73.5	200	3.591	17.5	25
Cadillac	DeVille	63.729	22.525	0	39.895	4.6	275	115.3	74.5	207	3.978	18.5	22

PCA Example

Attributes	Type	Explain
manufact	Nominal	Manufacturer
model	Nominal	Model
sales	Ratio	Sales in thousands
resale	Ratio	4-year resale value
type	Ordinal	{0, Automobile} {1, truck} Vehicle type
price	Ratio	Price in thousands
engine_s	Ratio	Engine size
horsepow	Ratio	Horsepower
wheelbas	Ratio	Wheelbase
width	Ratio	Width
length	Ratio	Length
curb_wgt	Ratio	Curb weight
fuel_cap	Ratio	Fuel capacity
mpg	Ratio	Fuel efficiency

10 variables – 10 Eigenvalues

Component	Initial Eigenvalues		
	Total	% of Variance	Cumulative %
1	5.994	59.938	59.938
2	1.654	16.545	76.482
3	1.123	11.227	87.709
4	.339	3.389	91.098
5	.254	2.541	93.640
6	.199	1.994	95.633
7	.155	1.547	97.181
8	.130	1.299	98.480
9	.091	.905	99.385
10	.061	.615	100.000

Total Variance

Component	Initial Eigenvalues		
	Total	% of Variance	Cumulative %
1	5.994	59.938	59.938
2	1.654	16.545	76.482
3	1.123	11.227	87.709
4	.339	3.389	91.098
5	.254	2.541	93.640
6	.199	1.994	95.633
7	.155	1.547	97.181
8	.130	1.299	98.480
9	.091	.905	99.385
10	[illegible]	.615	100.000

The **Total** column gives the eigenvalue, or amount of variance in the original variables accounted for by each component.

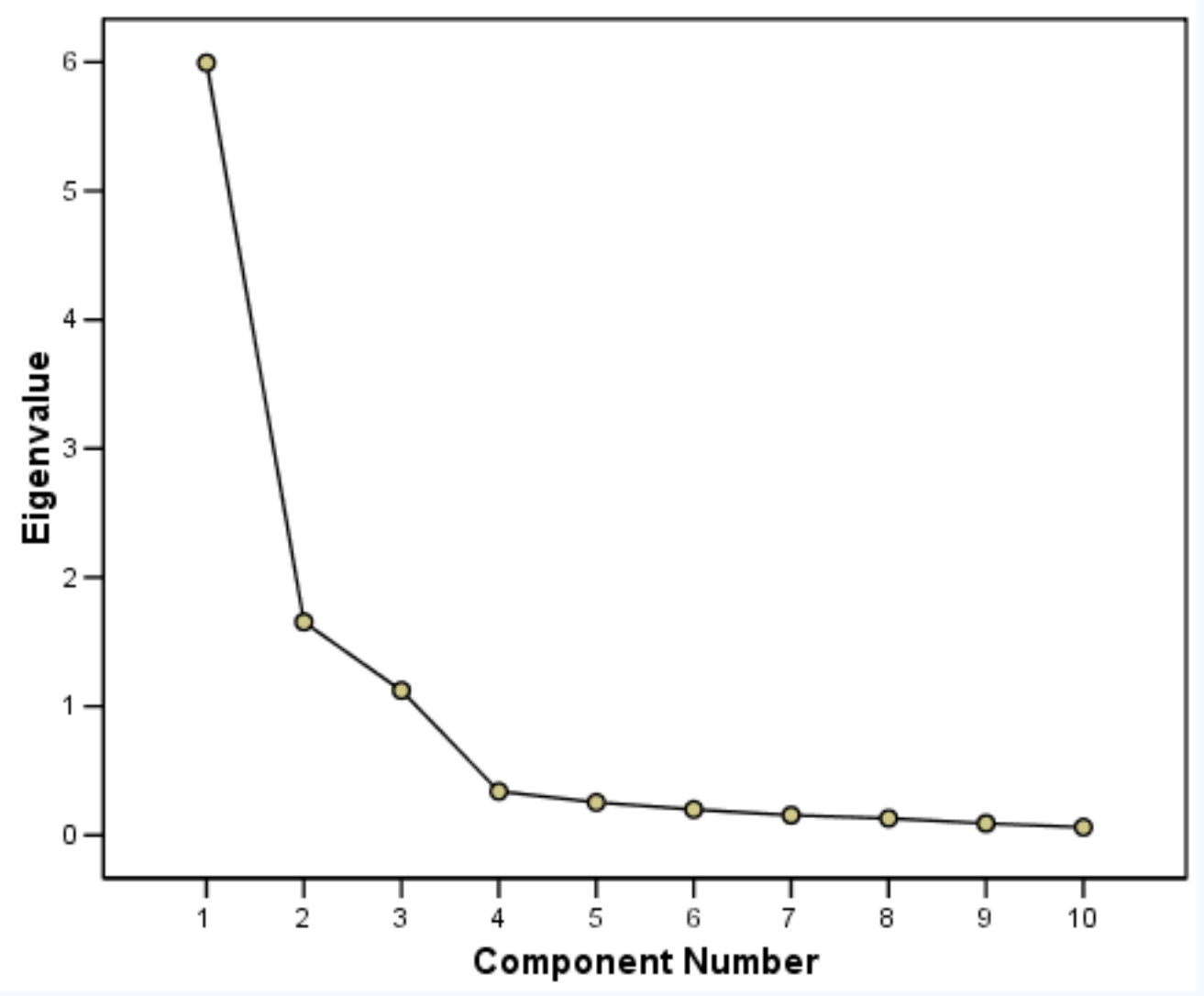
Component	Initial Eigenvalues		
	Total	% of Variance	Cumulative %
1	5.994	59.938	59.938
2	1.654	16.545	76.482
3	1.123	11.227	87.709
4	.339	3.389	91.098
5	.254	2.541	93.640
6	.199	1.994	95.633
7	.155	1.547	97.181
8	.130	1.299	98.480
9	.091	.905	99.385
10	.061	.615	100.000

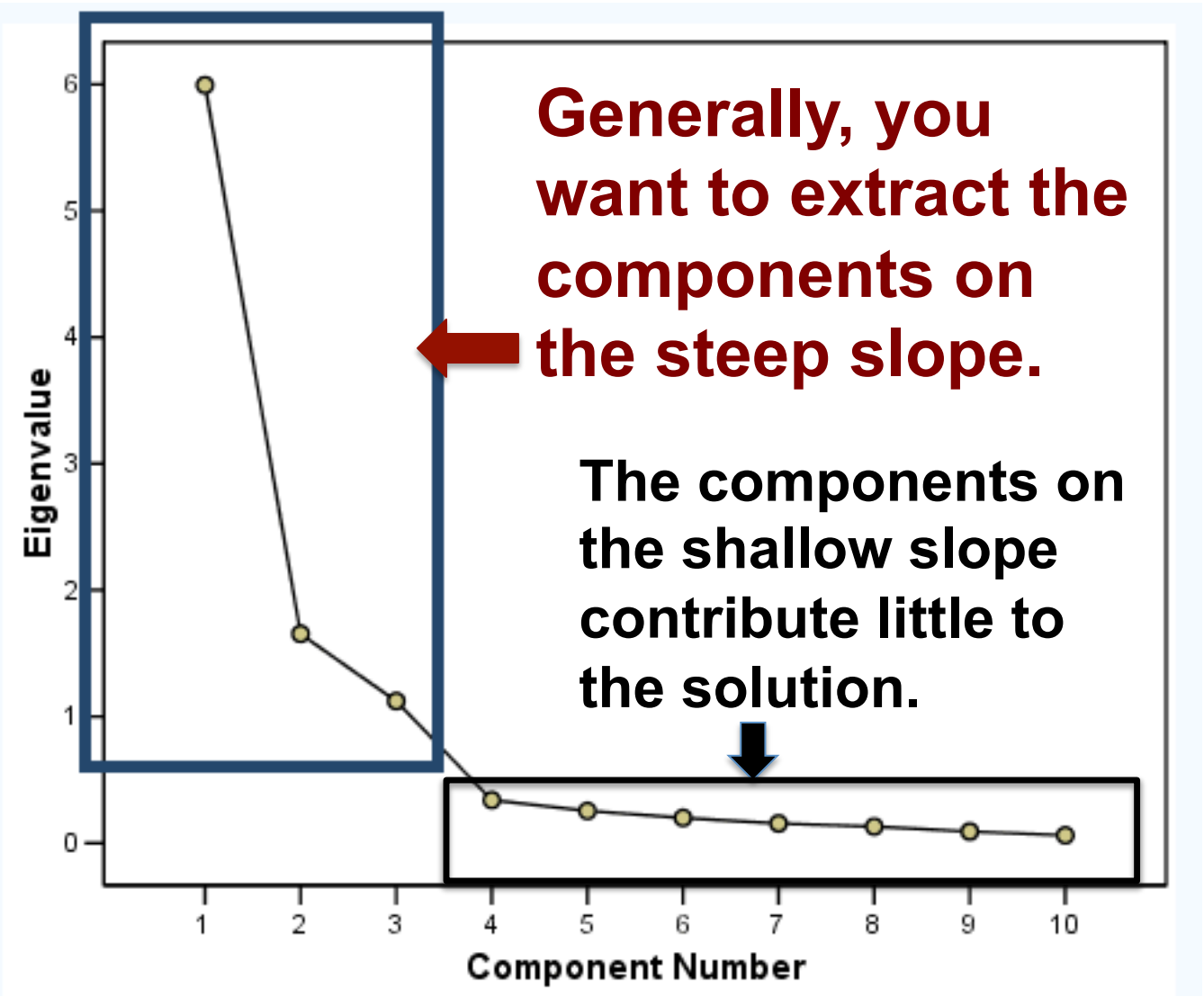
10 variables – 10 Eigenvalues

Component	Initial Eigenvalues		
	Total	% of Variance	Cumulative %
1	5.994	59.938	59.938
2	1.654	16.545	76.482
3	1.123	11.227	87.709
4	.339	3.389	91.098
5	.254	2.541	93.640
6	.199	1.994	95.633
7	.155	1.547	97.181
8	.130	1.299	98.480
9	.091	.905	99.385
10	.061	.615	100.000

A rule of thumb, pick Eigenvalue > 1

Eigenvalue Scree Plot




Generally, you want to extract the components on the steep slope.
The components on the shallow slope contribute little to the solution.
Eigenvalue
0
1
2
3
4
5
6
1
2
3
4
5
6
7
8
9
10
Component Number

What are the Principle Components?

	Component		
	1	2	3
Vehicle type	-.101	.095	.954
Price in thousands	.935	-.003	.041
Engine size	.753	.436	.292
Horsepower	.933	.242	.056
Wheelbase	.036	.884	.314
Width	.384	.759	.231
Length	.155	.943	.069
Curb weight	.519	.533	.581
Fuel capacity	.398	.495	.676
Fuel efficiency	-.543	-.318	-.681

What are the Principle Components?

	Component
	1
Vehicle type	-.101
Price in thousands	.935
Engine size	.753
Horsepower	.933
Wheelbase	.036
Width	.384
Length	.155
Curb weight	.519
Fuel capacity	.398
Fuel efficiency	-.543

What are the Principle Components?

	Component		
	1	2	3
Vehicle type		.095	
Price in thousands		-.003	
Engine size		.436	
Horsepower		.242	
Wheelbase		.884	
Width		.759	
Length		.943	
Curb weight		.533	
Fuel capacity		.495	
Fuel efficiency		-.318	

What are the Principle Components?

	Component		
	1	2	3
Vehicle type			.954
Price in thousands			.041
Engine size			.292
Horsepower			.056
Wheelbase			.314
Width			.231
Length			.069
Curb weight			.581
Fuel capacity			.676
Fuel efficiency			-.681

What is the magic of PCA?

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:

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

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:

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

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

- 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:

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

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

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

- Finding the roots of $|\mathbf{A} - \lambda \cdot \mathbf{I}|$ 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:

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

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

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

$$\lambda_1 = -1 \text{ and } \lambda_2 = -2$$

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.

What properties do these eigenvectors have?

1. **The sum of the eigenvalues of a matrix is equal to the sum of its diagonal elements, which is called the trace of a matrix**
2. **Eigenvectors can only be found for square matrices.**
3. **Not every square matrix has eigenvectors.**
4. **Given an *d* x *d* matrix that does have eigenvectors, there are *d* of them.**

PCA, what is our A?

- Covariance Matrix: Representing covariance among dimensionsas a matrix, e.g., for 3 dimensions:

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

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

What properties do these eigenvectors have?

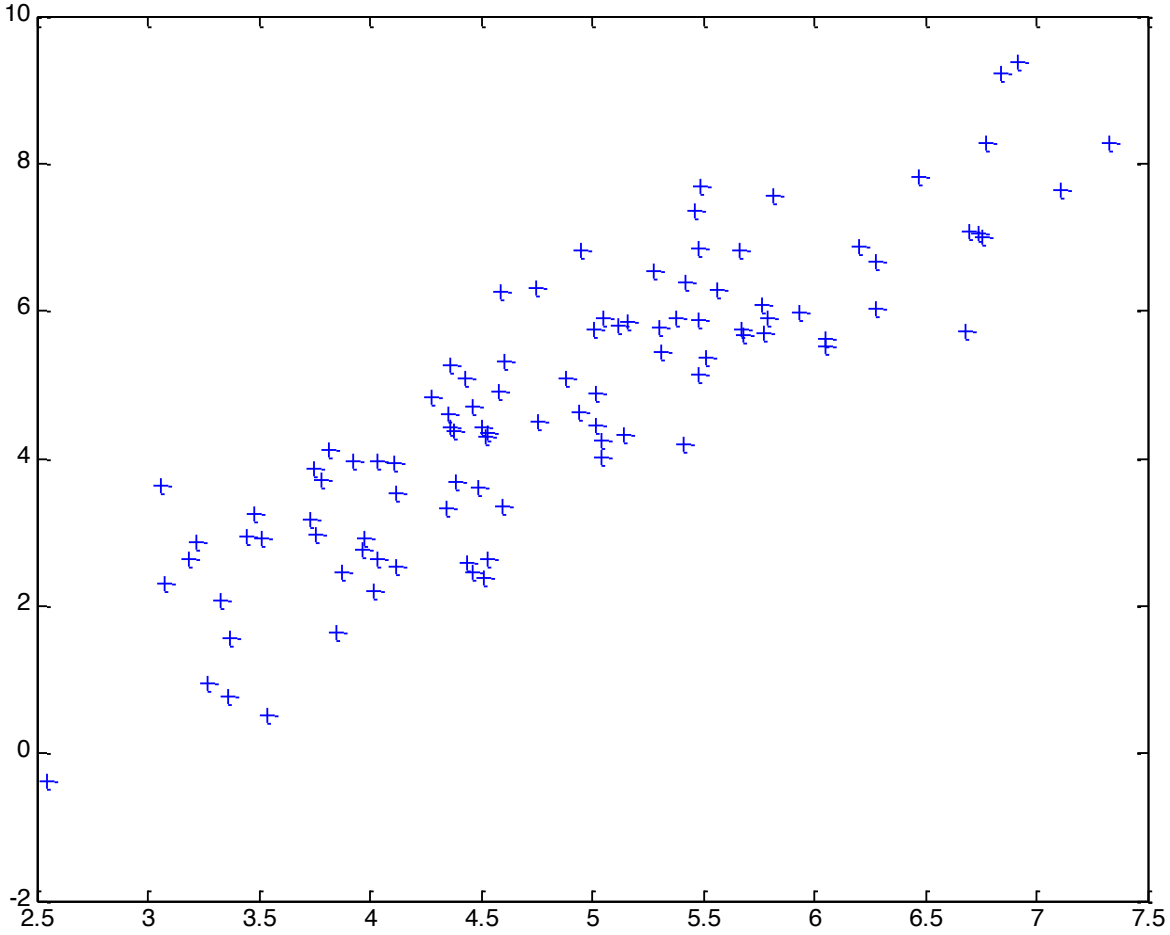
An dxd symmetric matrix A:

- **has *d* real eigenvalues $\lambda_1, \ldots \lambda_d$**

- **There are d orthogonal eigenvectors (i.e. xi . xj =0 for i <> j).**

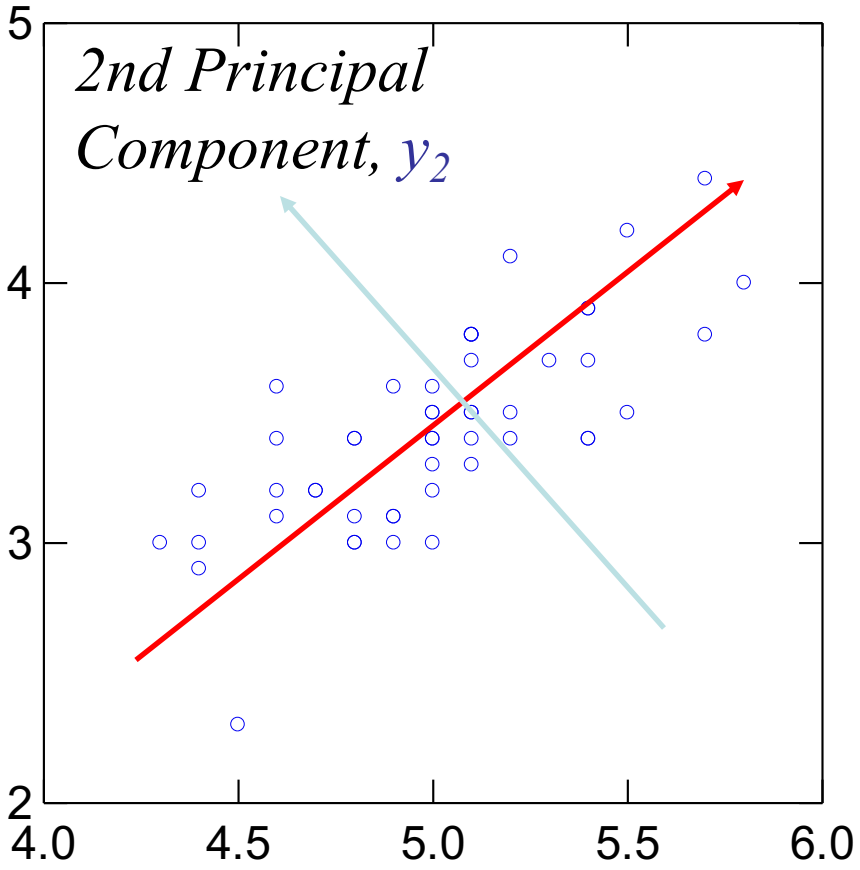
- **There exists a matrix P , for which the columns are eigenvectors of A,such that $P^{-1}AP = D$ and D is diagonal matrix.**

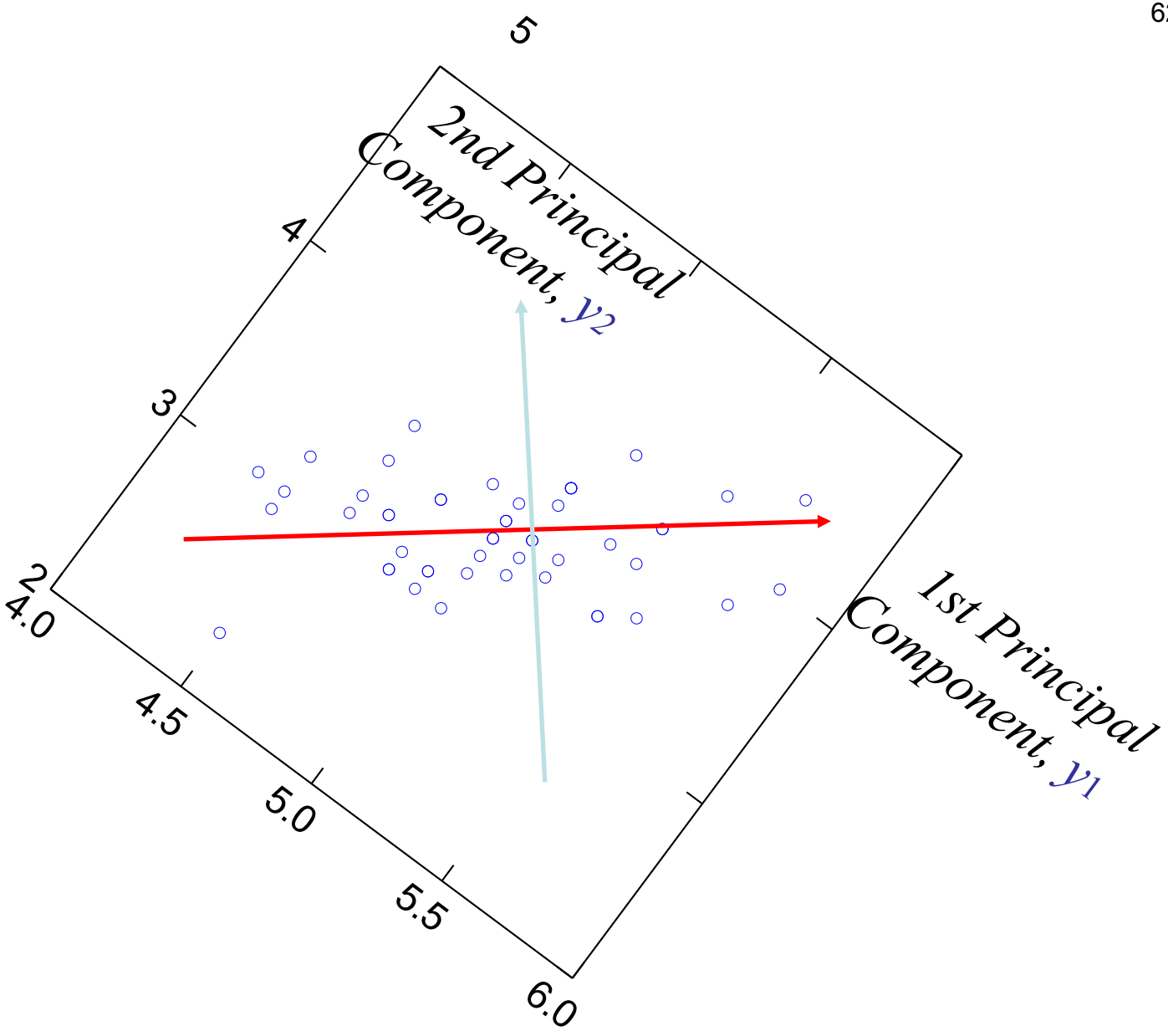
2d Data



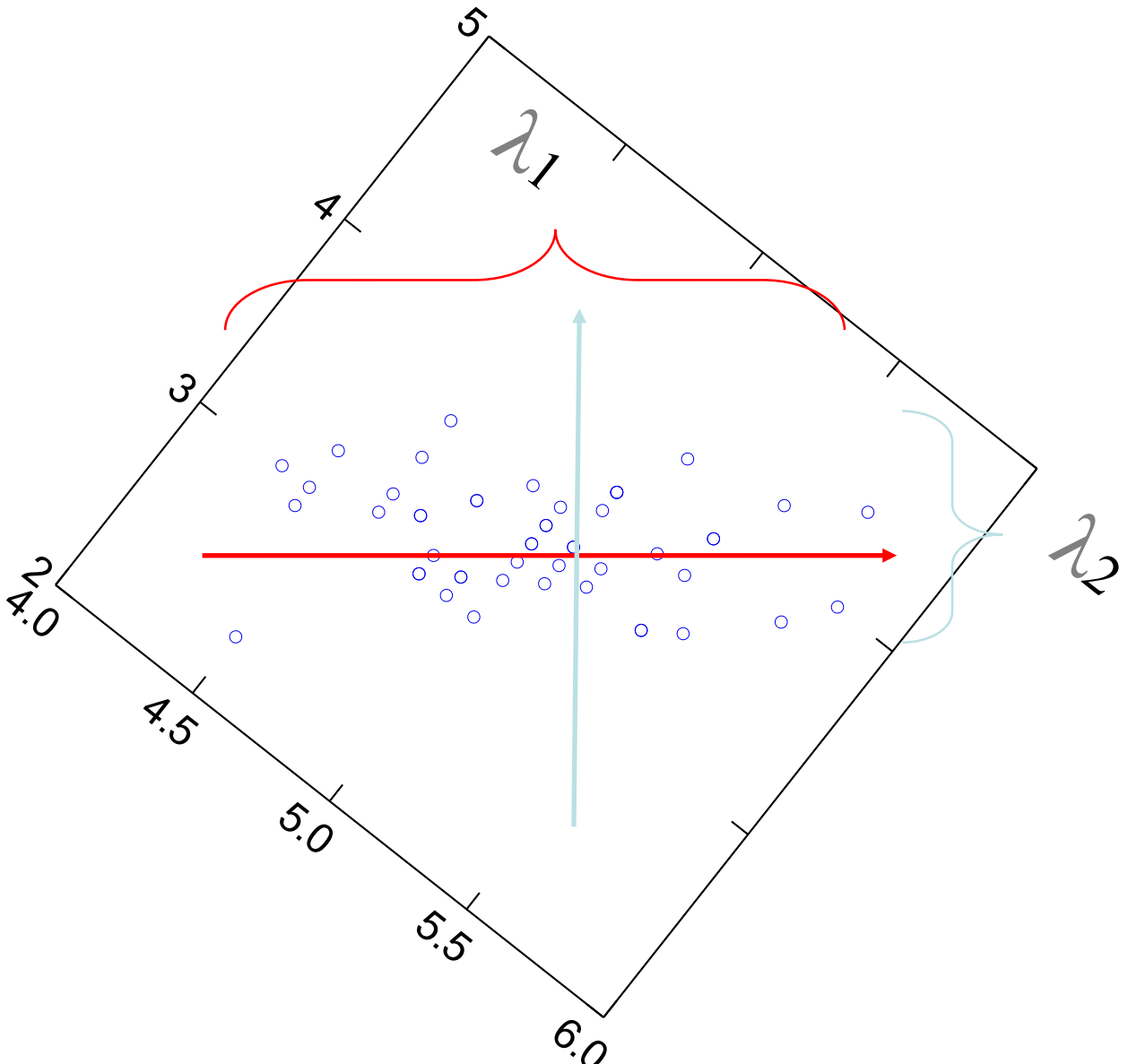
Dimensionality Reduction: PCA

- Goal is to find a projection that captures the largest amount of variation in data


2nd Principal
Component, y_2
1st Principal
Component, y_1
5
4
3
2
4.0
4.5
5.0
5.5
6.0

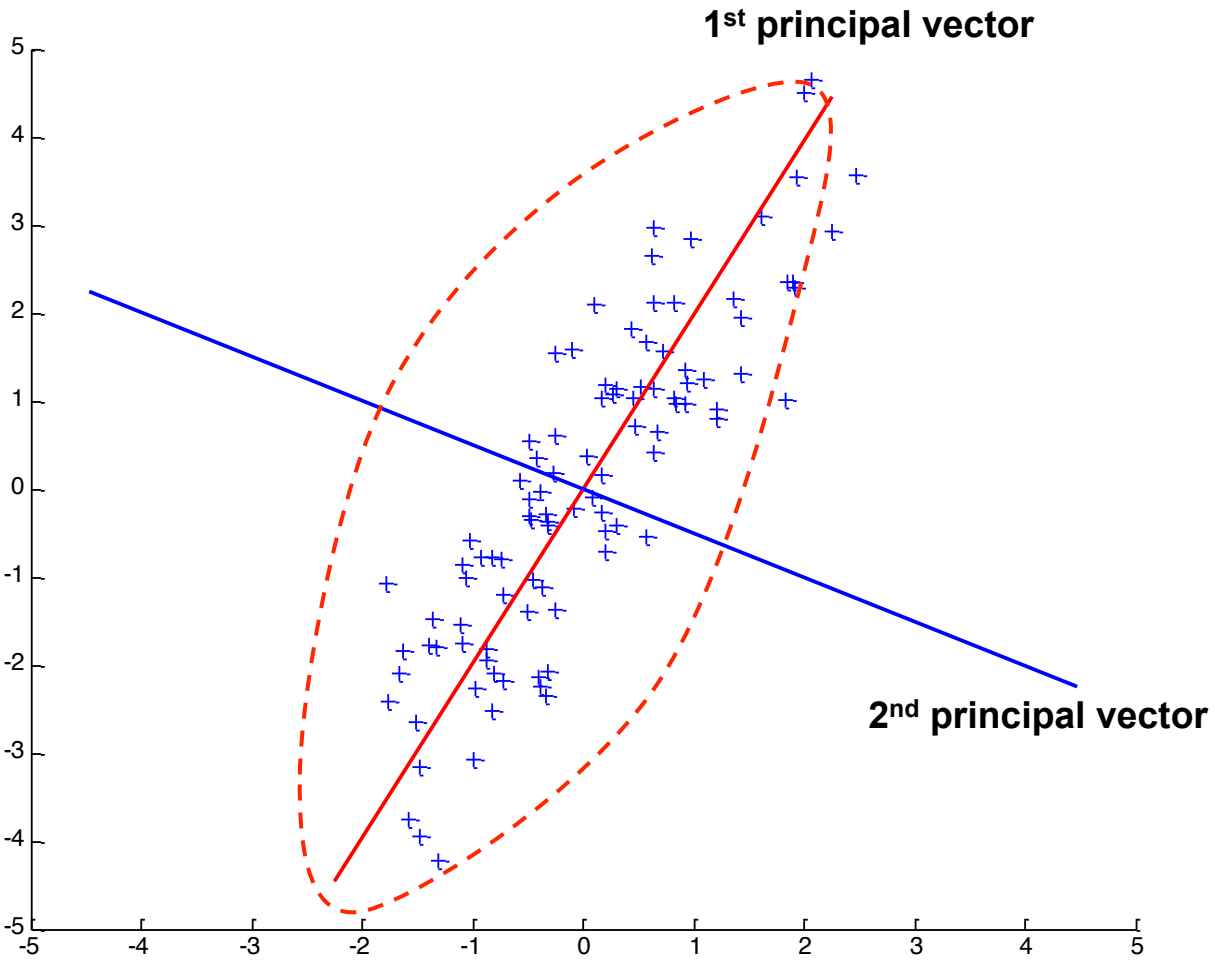

5
4
3
2
4.0
4.5
5.0
5.5
6.0
2nd Principal Component, y_2
1st Principal Component, y_1

PCA Eigenvalues



Principal Components

- Gives best axis to project
- Minimum RMS error
- Principal vectors are orthogonal



PCA: *General*

From d original variables: $x_1, x_2, \ldots, x_d$:

Produce d new variables: $y_1, y_2, \ldots, y_d$:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1d}x_d$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2d}x_d$$

$\ldots$

$$y_d = a_{d1}x_1 + a_{d2}x_2 + \ldots + a_{dd}x_d$$

PCA: *General*

From d original variables: $x_1,x_2,\ldots,x_d$:

Produce d new variables: $y_1,y_2,\ldots,y_d$:

$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1d}x_d$

$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2d}x_d$

...

$y_d = a_{d1}x_1 + a_{d2}x_2 + \ldots + a_{dd}x_d$

such that:

y_d's are uncorrelated (orthogonal)

y_1 explains as much as possible of original variance in data set

y_2 explains as much as possible of remaining variance

etc.

Dimensionality Reduction: PCA

- Find the eigenvectors of the covariance matrix
- The eigenvectors define the new space

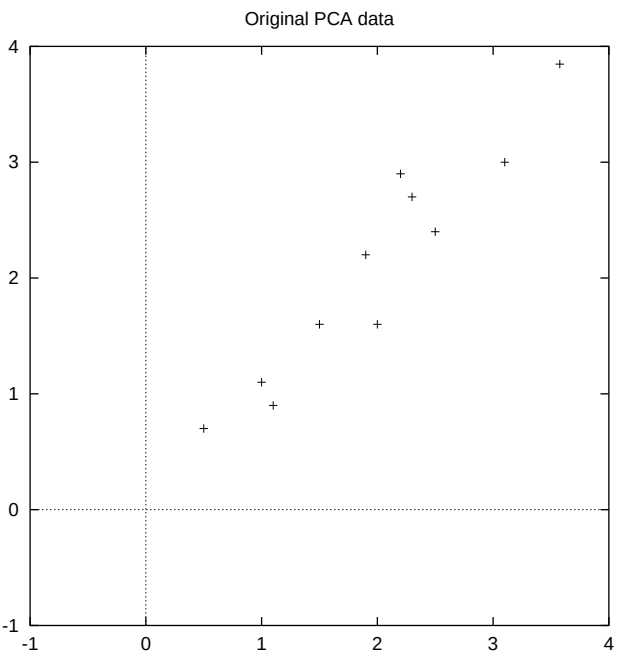
An Example

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 example data, original data o a plot of the data

Calculate the covariance matrix

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

70

So, since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variable increase together.

Calculate the eigenvectors and eigenvalues of the covariance matrix

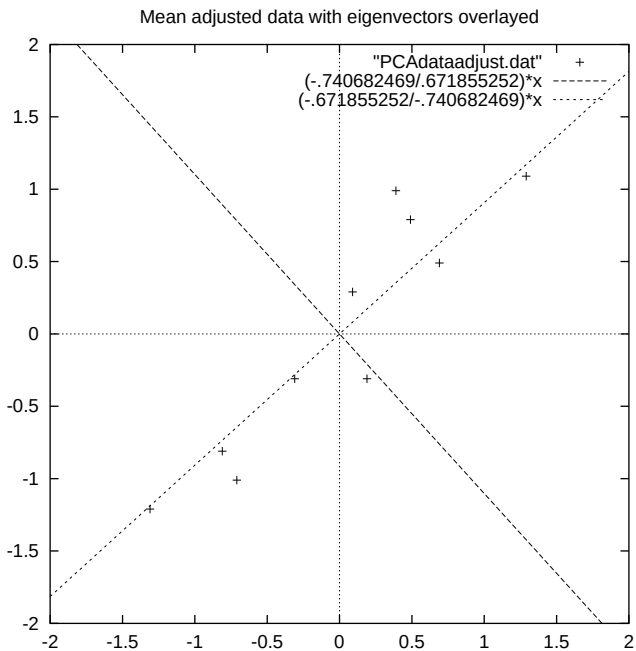
$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

The eigenvectors and eigenvalues of the covariance matrix

$$eigenvalues \quad \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

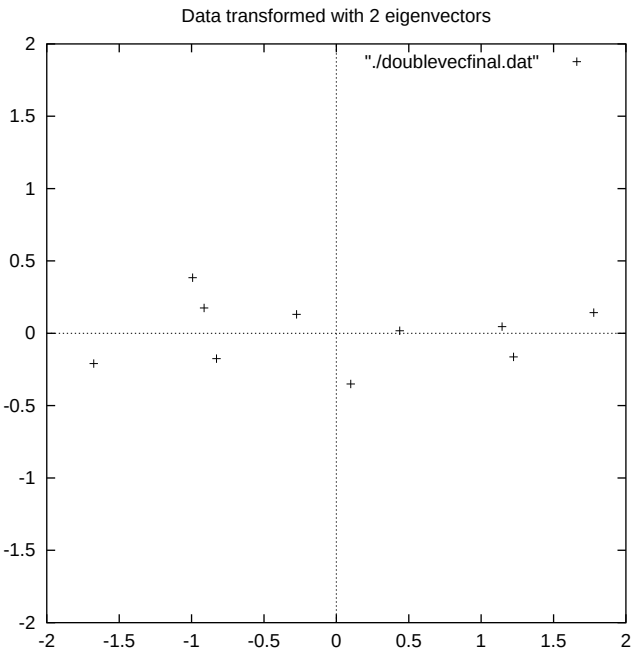
$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ 677873399 & -.735178656 \end{pmatrix}$$



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

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

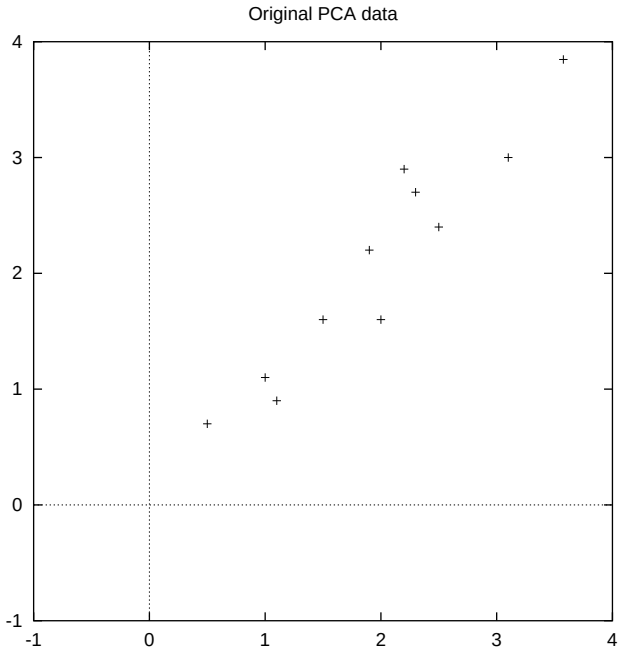


The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

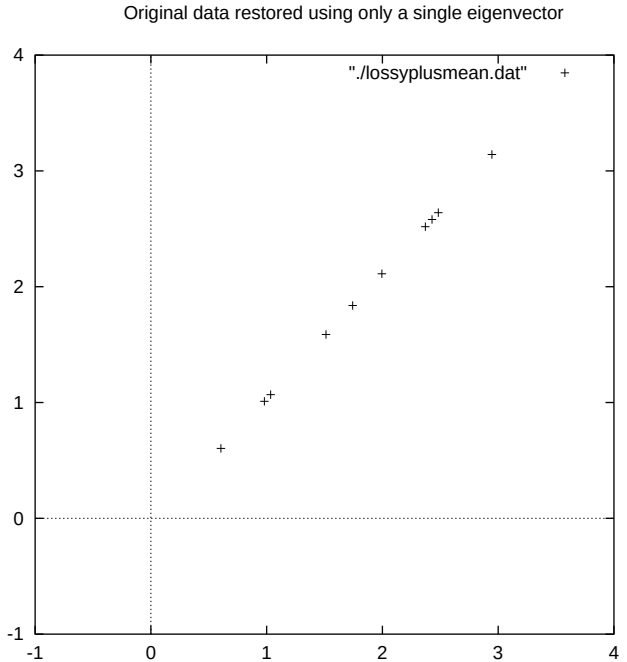
Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

The data after transforming using only the most significant eigenvector



PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data



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

Paper.

Exercise

PCA: feature selection and feature extraction

- This suggests that you can focus on Price in thousands, Length, and Vehicle type in further analyses.
- but you can do even better by saving component scores.