Time-frequency analysis

Time-Frequency Analysis

- The Fourier transform of a musical passage would tell us what notes are played, but it is extremely difficult to figure out when they are played.
- Previous spectral methods stationarity: that waveforms do not change their statistical properties.
- Biological signals are often non-stationary (e.g., the EEG changes depending on the internal mental state).
- Techniques to extract both time and frequency information can be divided into two groups:
    1. Time-frequency methods covered in this chapter.
    2. Time-scale methods better known as *wavelet* analyses described in the next chapter.

The Short-Term Fourier Transform or Spectogram

In the continuous domain, the traditional Fourier Transform is multiplied by a sliding window, $w(t-\tau)$:

$$X(t,f) = \int_{-\infty}^{\infty} x(t)\, w(t - \tau)\, e^{-j 2\pi m f t}\, dt$$

The discrete representation is similar:

$$x[m,k] \square \sum_{n \square 1}^{N} x[n] \square W[n-k] e^{- j\omega m / N} \square$$

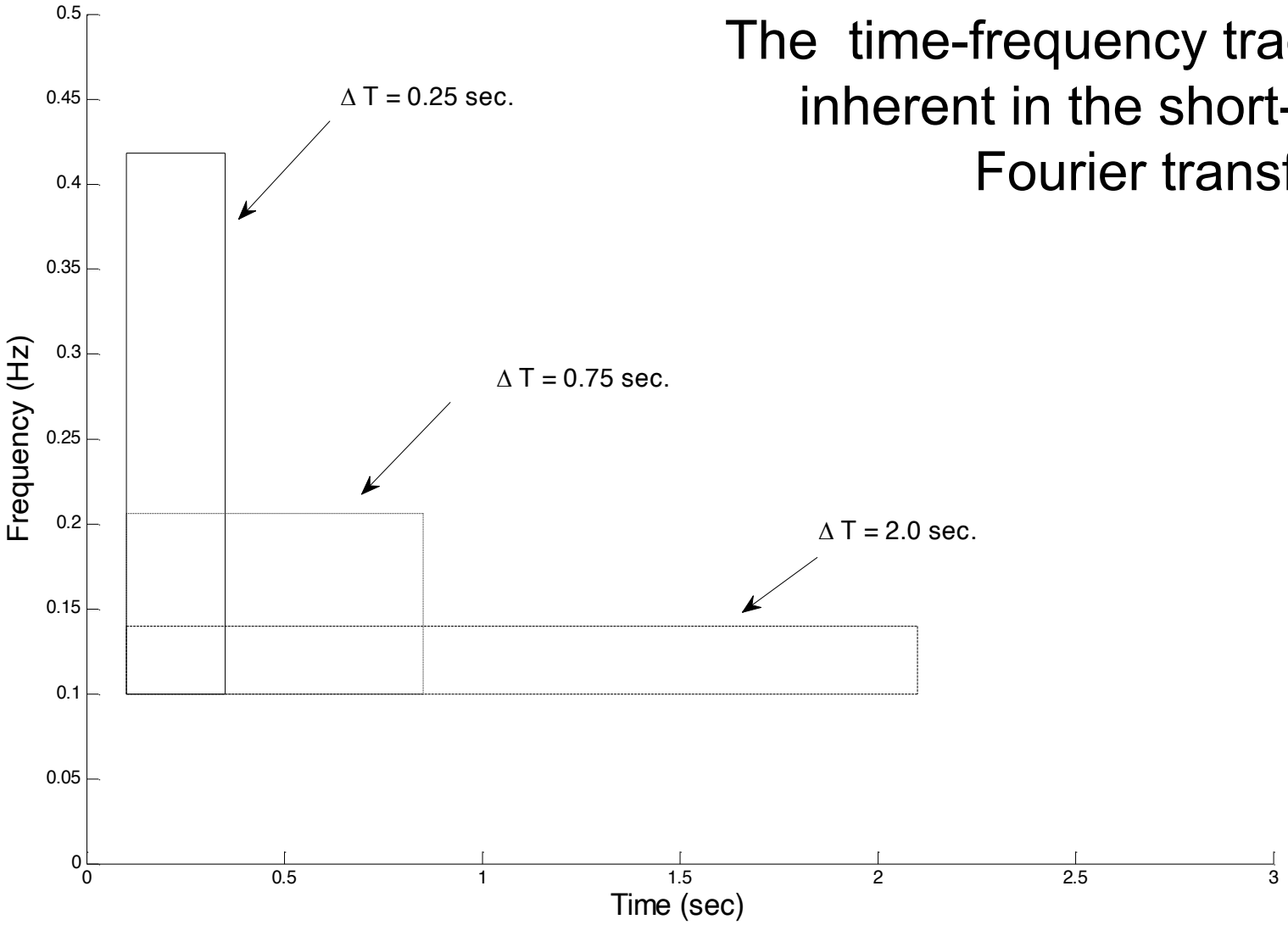
By changing window size it is possible to trade-off between time resolution and frequency resolution.

The STFT has a time-frequency uncertainty (limit) given by:

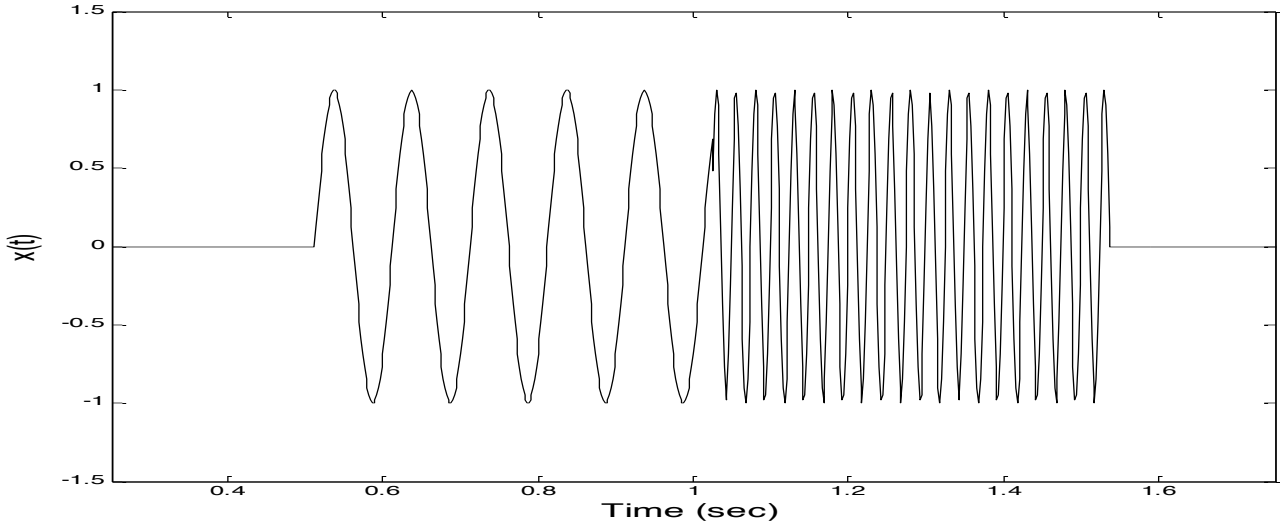
$$B\ T \geq \frac{1}{4\pi}$$

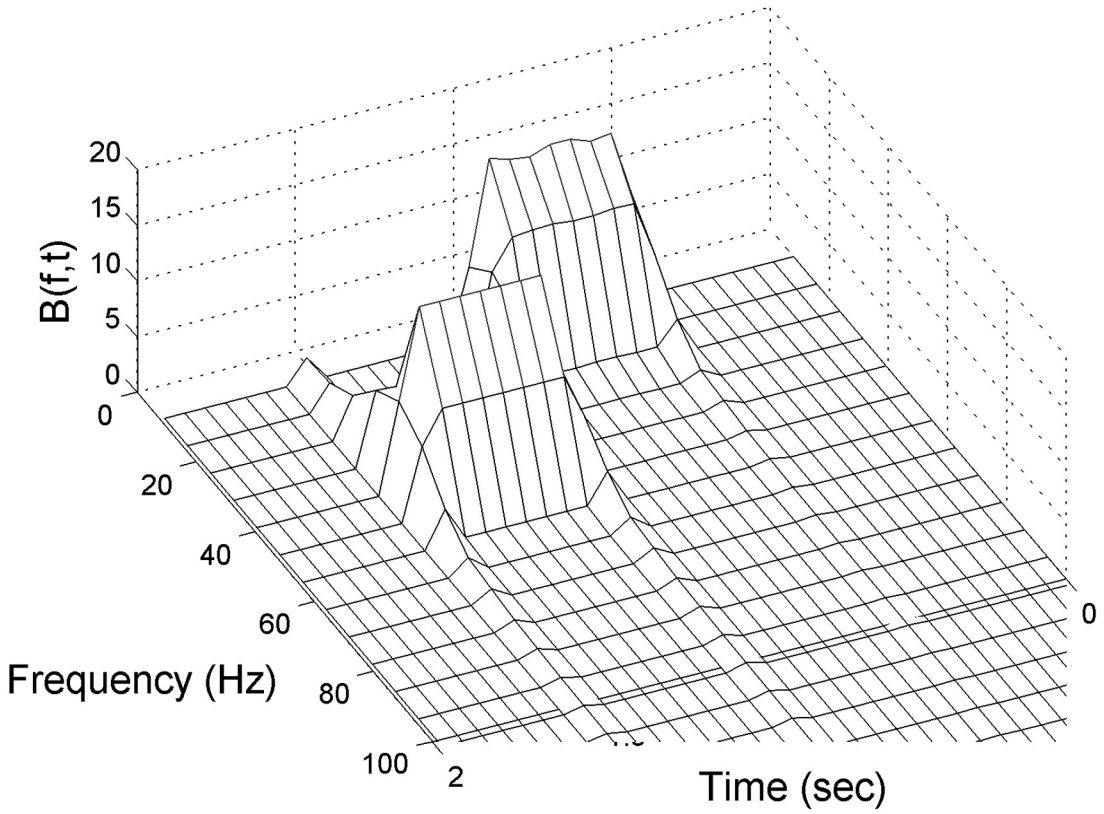
where B is bandwidth resolution and T is time resolution.

The time-frequency tradeoff inherent in the short-term Fourier transform.

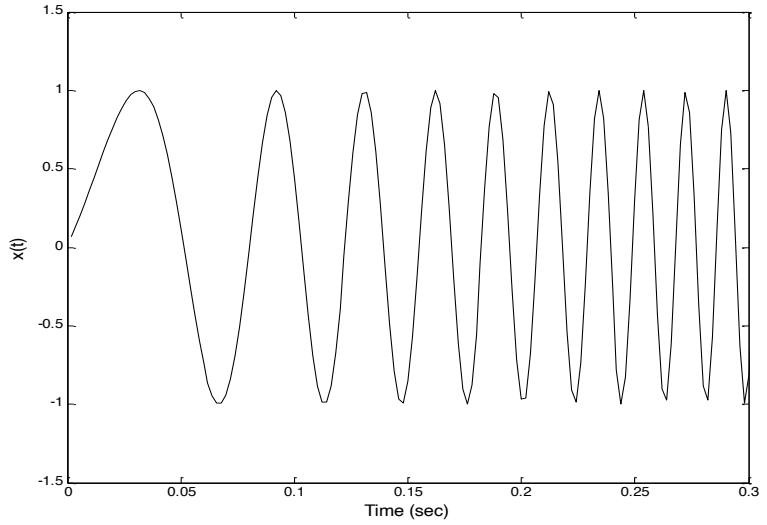


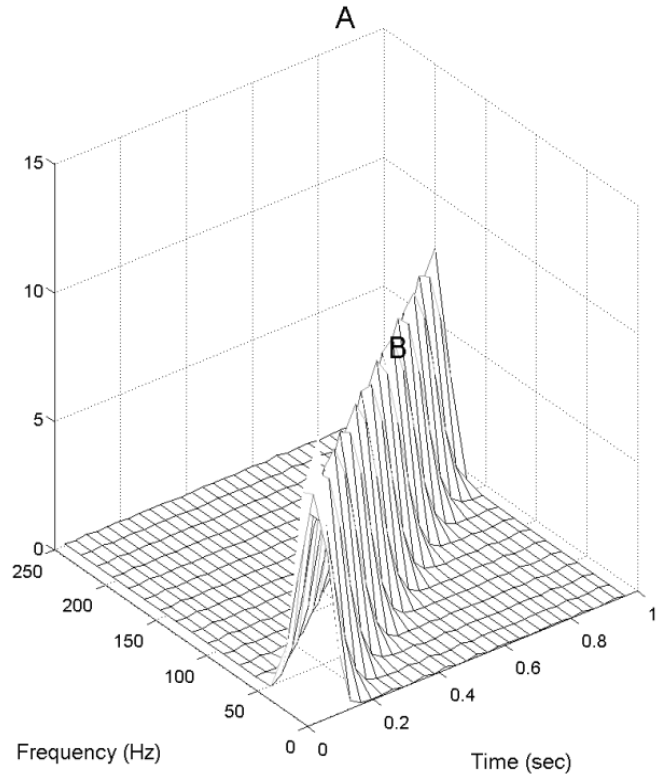
The smallest time resolution gives the poorest frequency resolution (black rectangle) and vice versa (dashed rectangle).

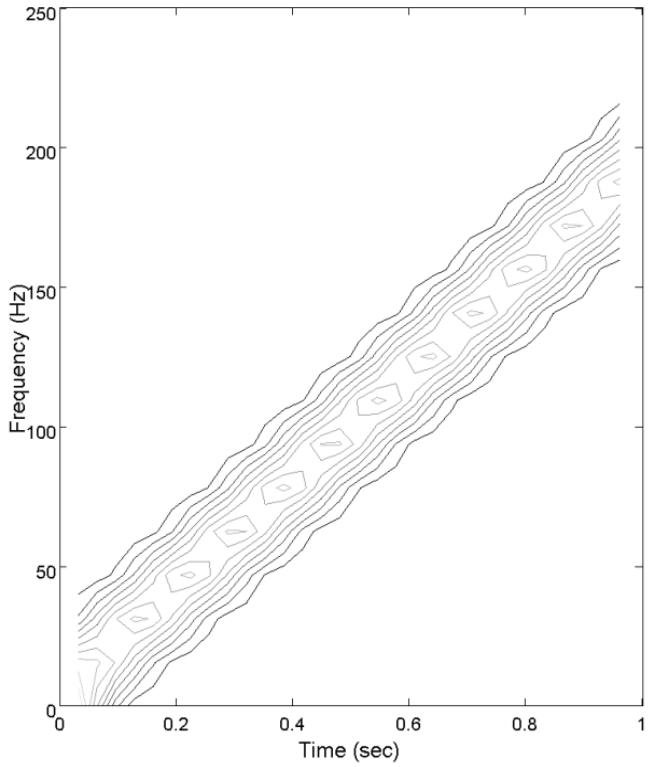

x(t)
Time (sec)


B(f,t)
Frequency (Hz)
Time (sec)

A chirp signal increases in frequency with time.







Wavelet analysis

Wavelet Analysis

- None of the previous approaches provides accurate time-frequency analysis applicable to all signals.
- The wavelet transform is yet another way to describe a waveform that changes over time where the waveform is divided into segments of scale.
- Wavelet analysis uses probing functions that are compact (i.e., have a short time duration); the probing family (i.e., basis) consists of the same function at different scales.

The Continuous Wavelet Transform

In the Continuous Wavelet Transform (CWT) the probing function has a finite time duration and its family members vary in scale:

$$W(a,b) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{|a|}} \Psi^* \left(\frac{t-b}{a} \right) dt$$

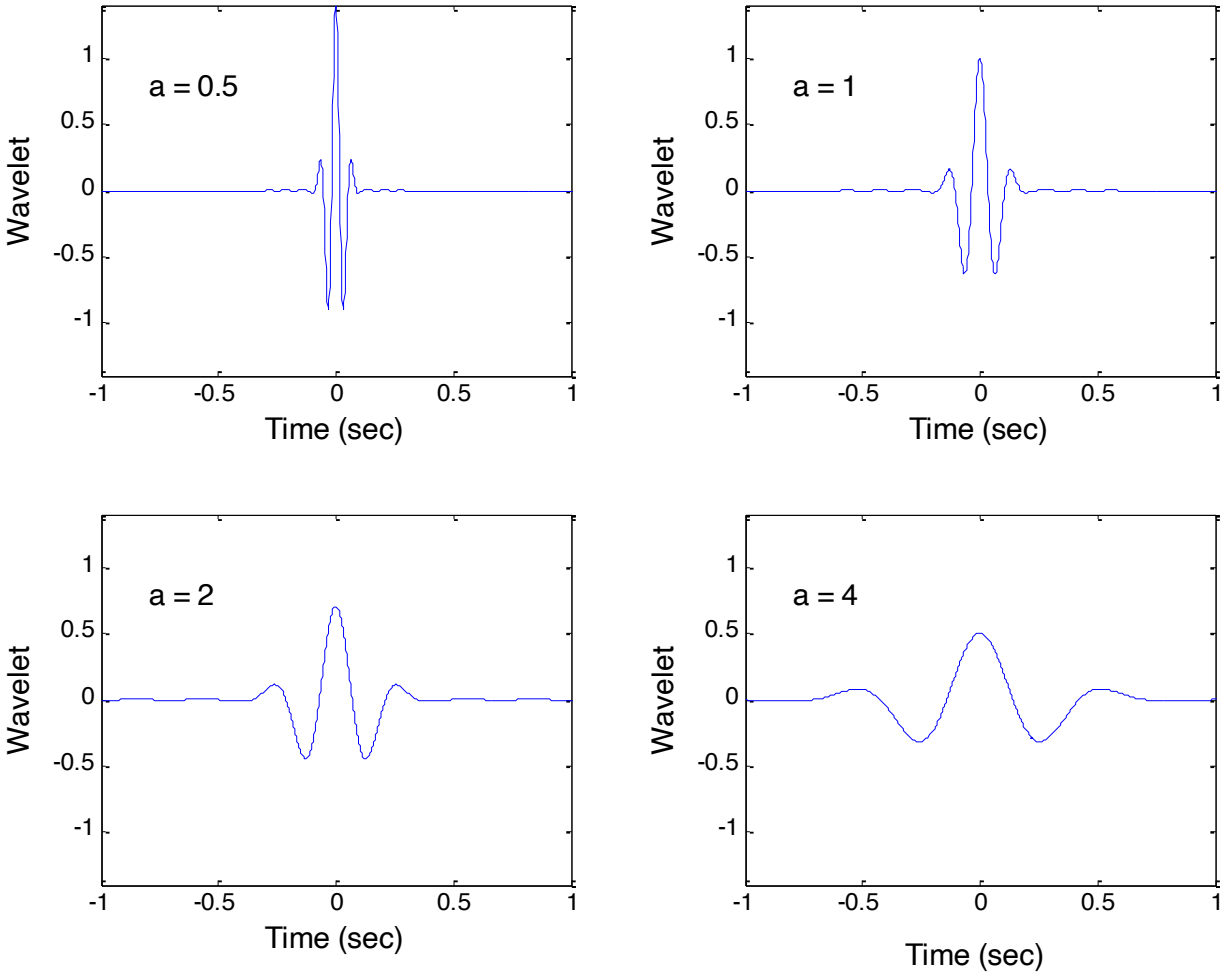
where *b* acts to translate the function across *x*(*t*) just as *t* does in the short-term Fourier transform, and the variable *a* acts to vary the time scale of the probing function, ψ.

The * indicates complex conjugation and dividing by $\sqrt{|a|}$ normalizes energy.

Wavelet Families

The wavelet shown is the 'Morlet Wavelet' described by the equation:

$$\psi(t) \square e^{-t^2} \cos\square\pi\sqrt{\frac{2}{\ln 2}}t\square$$



If *b* = 0, and *a* = 1, then the wavelet is in its basic form, the "mother wavelet." If *a* > 1, the wavelet is stretched along the time axis, and if a < 1, the wavelet is contracted.

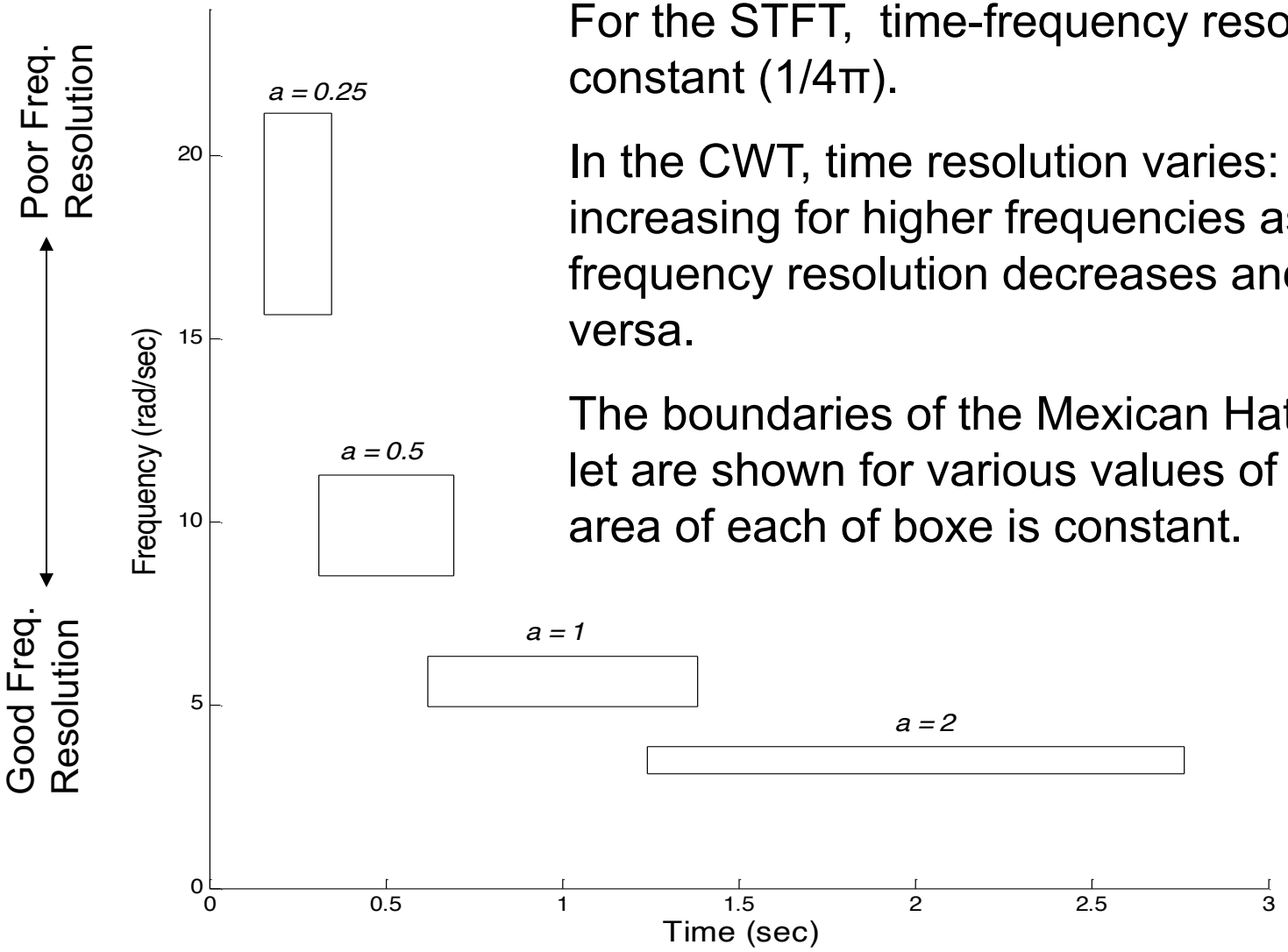
Wavelets: Time-Frequency Trade-off

- There is a built-in tradeoff between time and frequency resolution which is key to the CWT and makes it well suited to analyzing signals with rapidly varying high-frequency components superimposed on slowly varying low-frequency components:

$$\Delta\omega_\psi(a)\,\Delta t_\psi(a) = \Delta\omega_\psi\,\Delta t_\psi = \text{constant} \geq 0.5$$

- This trade-off is illustrated in the next figure which shows the time-frequency boundaries for the Mexican hat wavelet for various values of *a*.

Wavelets: Time-Frequency Trade-off

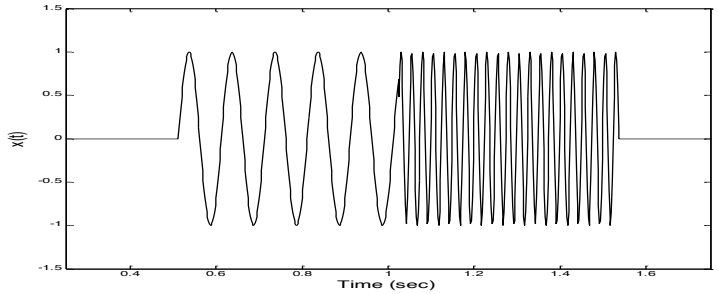


For the STFT, time-frequency resolution is constant (1/4π).

In the CWT, time resolution varies: increasing for higher frequencies as frequency resolution decreases and vise versa.

The boundaries of the Mexican Hat Wavelet are shown for various values of *a.* The area of each of boxe is constant.

CWT of a sample signal



Continuous Wavelet Transform of a signal that changes in step-like manner between 10 and 40 Hz

Discrete Wavelet Transform

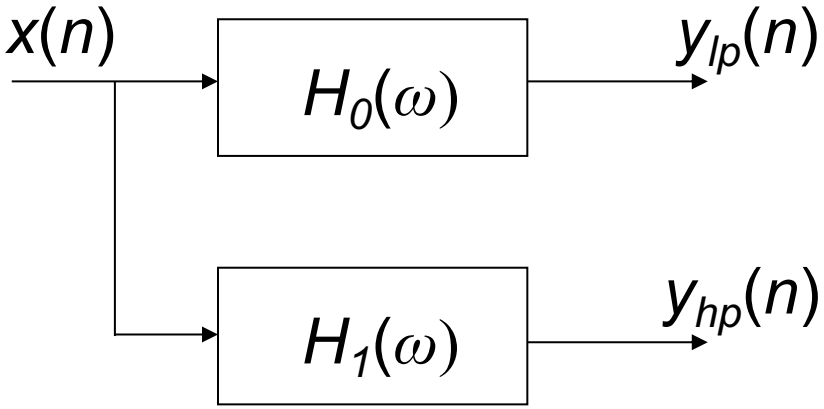
- The CWT is highly redundant: many more coefficients are generated than needed uniquely to specify the signal.
- If the application calls for recovery of the original signal, all of the coefficients will be required and the computational effort could be excessive.
- In applications that require bilateral transformations, a transform that produces the minimum number of coefficients required is preferred.
- The “Discrete Wavelet Transform” (DWT) achieves this parsimony by:

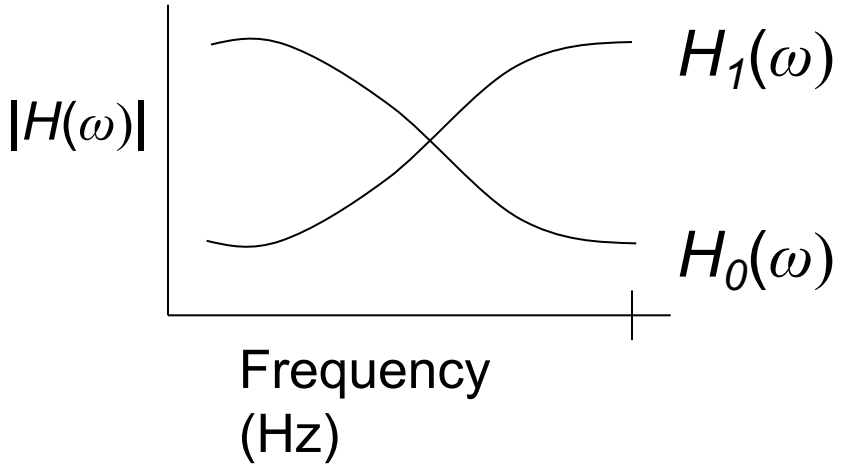
  a) restricting the variation in translation and scale, usually to powers of 2;

  b) downsampling lower scaled data.

Filter Banks

Filter banks are groups of filters that operate on the same signal.





Here a simple filter bank consisting of only two filters applied to the same waveform.

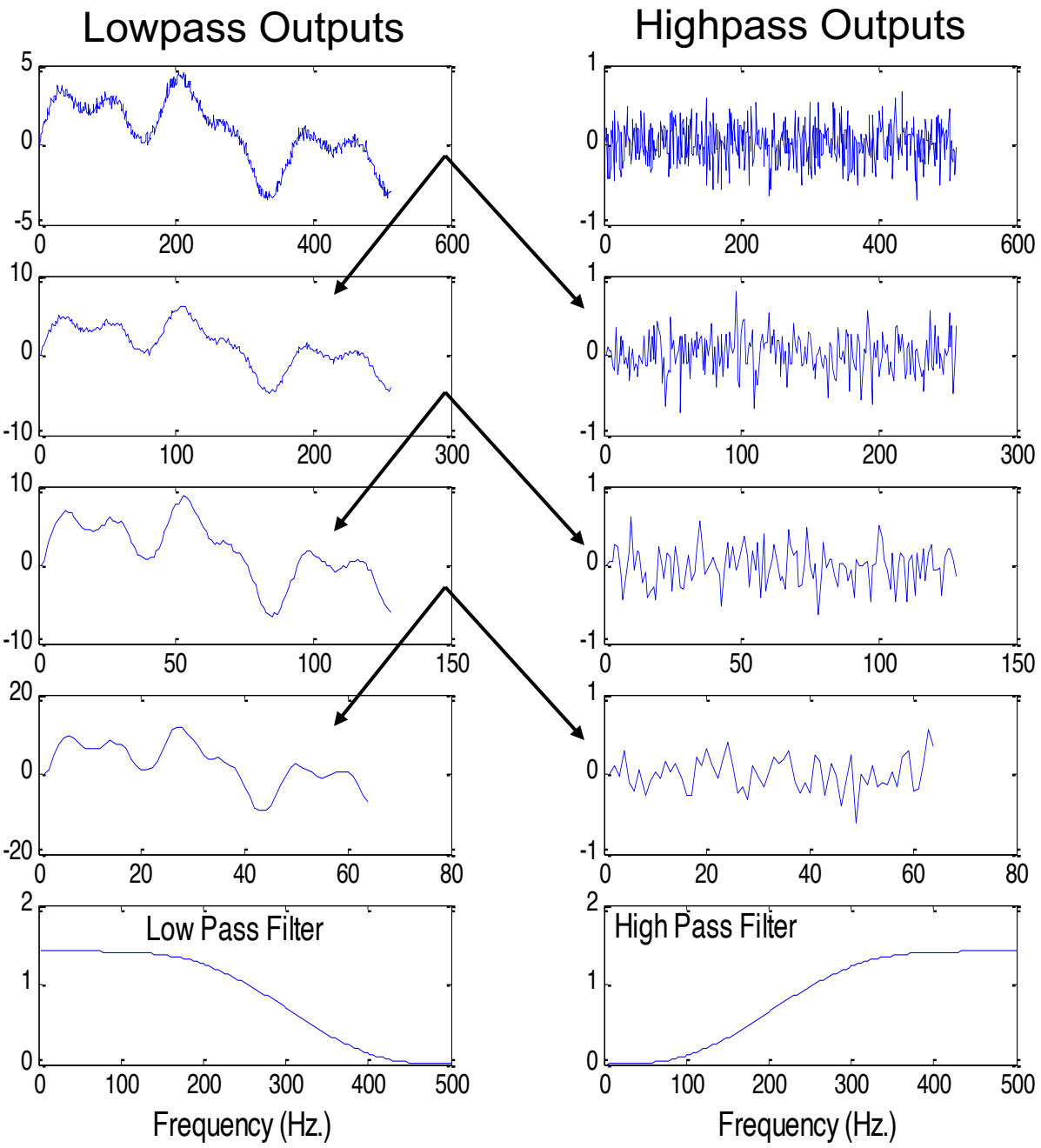
In the DWT, the filters have highpass and lowpass spectral characteristics.

Filter outputs consist of two subbands: a lowpass subband $y_{lp}(n)$ and a highpass subband $y_{hp}(n)$.

These plots show the output of the various filters.

Only the lowest lowpass subband is included in the output and used in the synthesis.

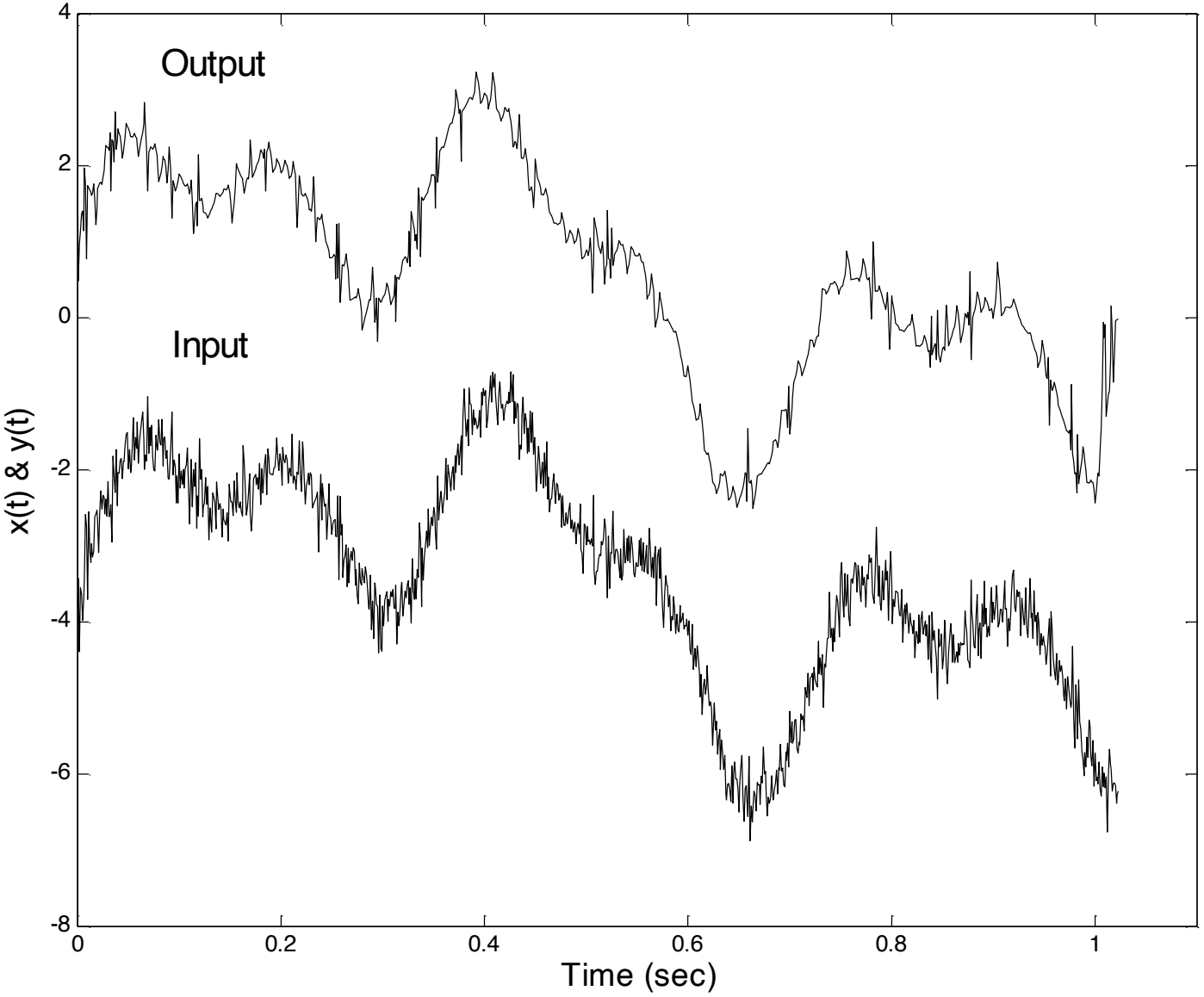
Filter frequency plots are also shown.



Example of Nonlinear Filtering - Denoising

The nonlinear filter reduces some of noise.

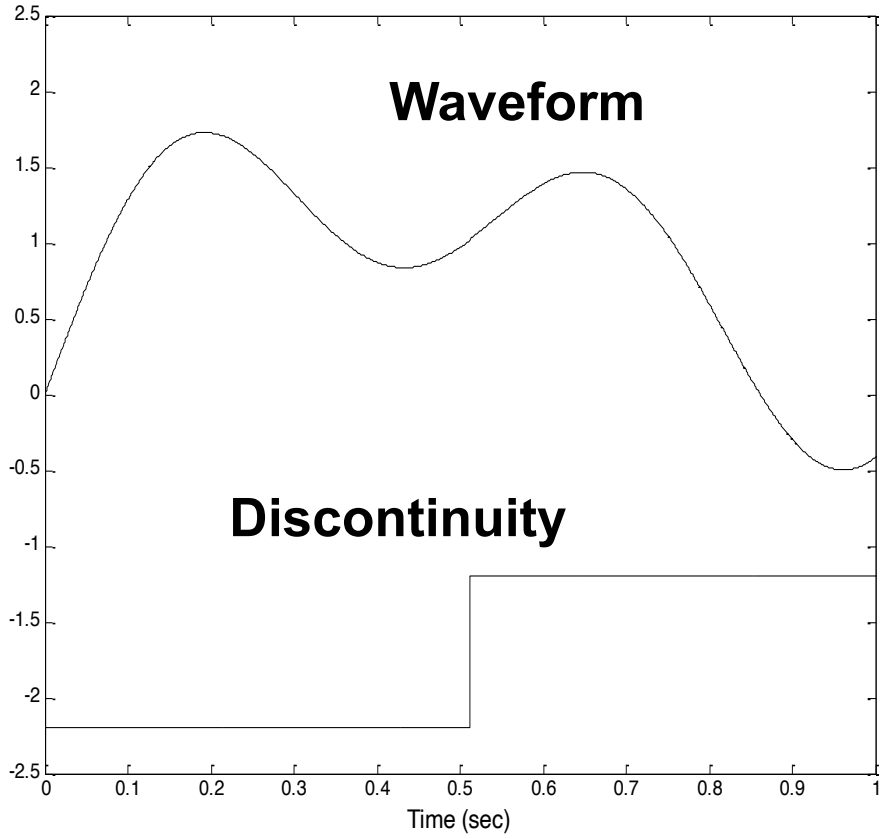
Increasing the threshold reduces the noise further.



Discontinuity Detection

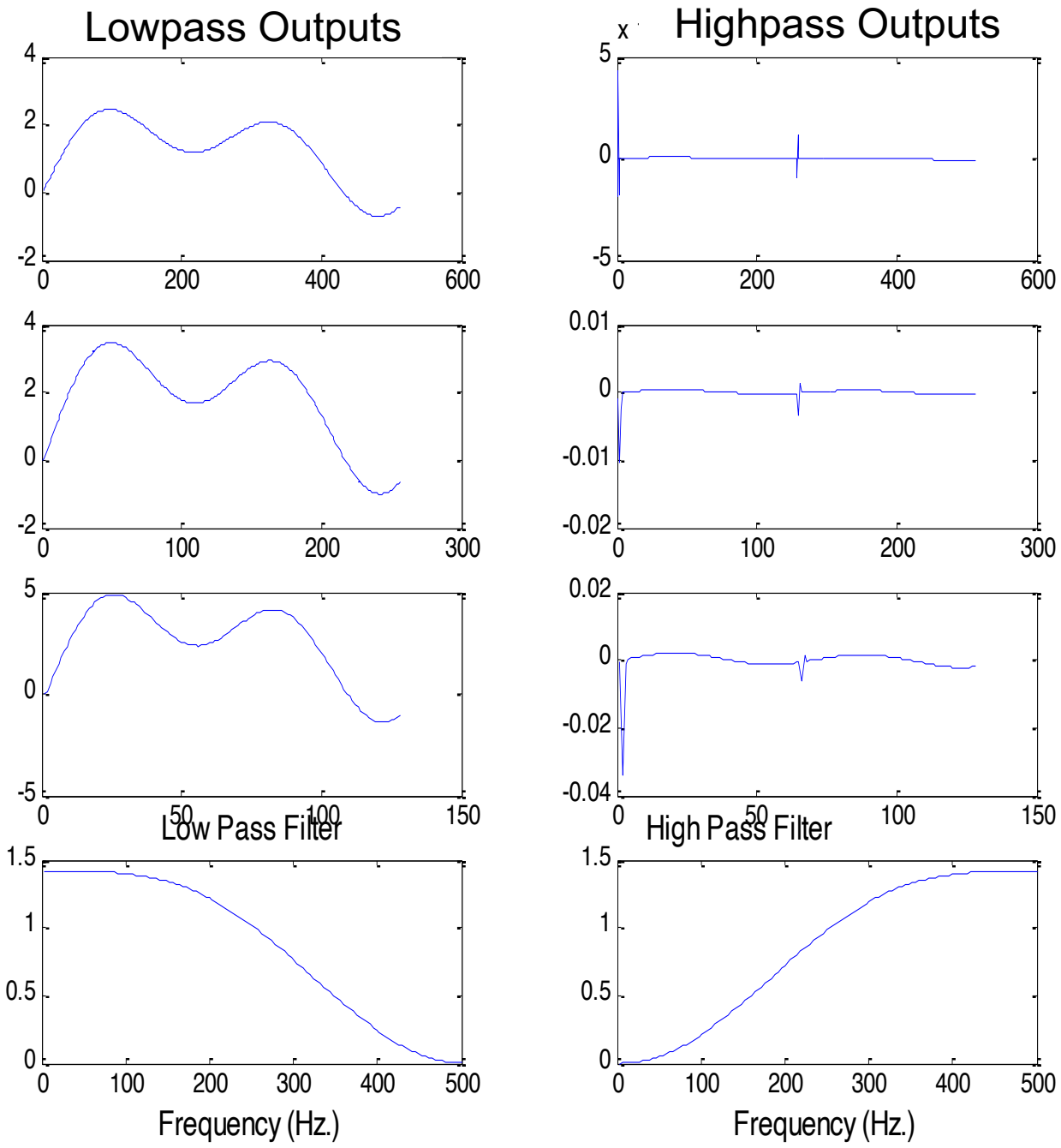
Add a small step change to the second derivative of a waveform. Apply a three-level analysis filter bank and examine the high-frequency subbands for evidence of this subtle discontinuity.

The discontinuity at 0.5 sec. is not visible in the waveform.



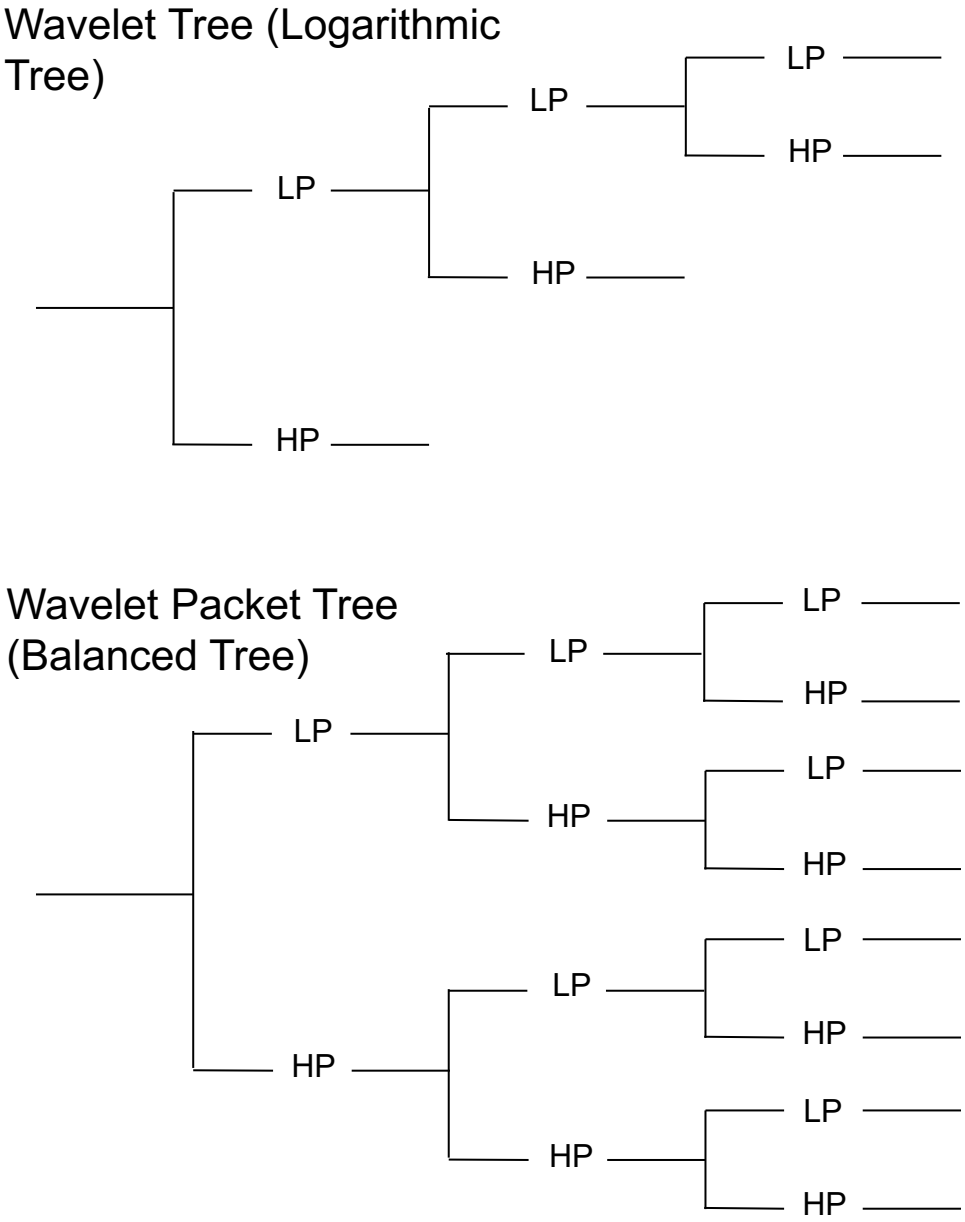
Example 7.5 Results

The discontinuity is clearly identified in the highpass subbands after application of a three-level analysis filter bank.



Feature Detection - Wavelet Packets

A wavelet packet tree (lower tree) does a complete decomposition, generating both lowpass and highpass subbands at every level. The outputs of this decomposition can be used in feature detection.



Optimal and adaptive filters

Optimal and Adaptive Filters

- Optimal Filters use knowledge about the signal and noise characteristics to get maximum separation. They are fixed and do not change over time.

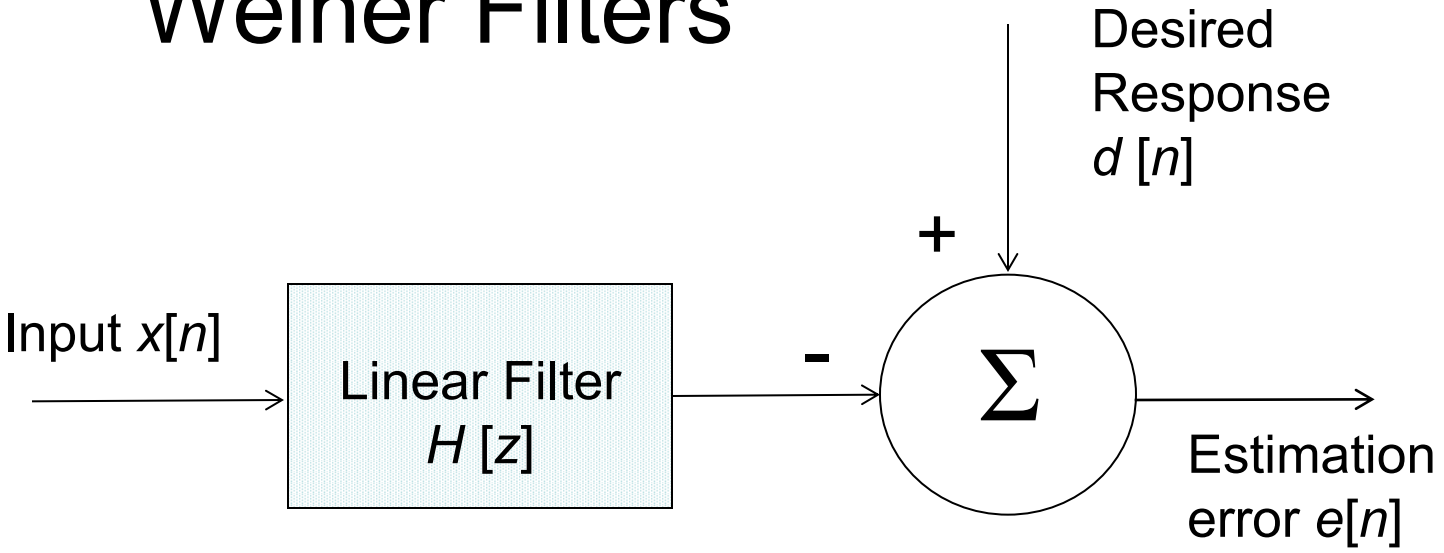
- Adaptive Filters adjust to provide good separation for signals that change their properties (statistical properties) over time. They are not fixed and might not be optimal.

- Both filters are described in terms of FIR filters. They have only numerator coefficients; i.e., ($b[n]$).

Optimal Signal Processing
Wiener Filters

- Optimal filter theory was developed to provide structure to the process of selecting the best filter frequency characteristics.
- A wide range of different approaches can be used to develop an optimal filter depending on the nature of the problem: specifically what, and how much, is known about signal and noise features.
- If a representation of the desired signal is available, then a well-developed and popular class of filters known as Wiener Filters can be applied.

Weiner Filters



- The basic concept behind the Wiener Filter theory is to minimize the difference between the filtered output and some desired output, $d[n]$.
- This minimization is based on the "least mean square" approach which adjusts the filter coefficients to reduce the square of the difference between the desired and actual waveform after filtering, $e^2[n]$.

Weiner Filters

- While a number of different filter types could be used, FIR filter are popular as they are inherently stable. FIR filters are used here.
- FIR filters have only numerator terms in the transfer function (i.e., only zeros) and can be implemented using convolution:

$$y[k] = \sum_{n=1}^{L} b[n]x[k-n]$$

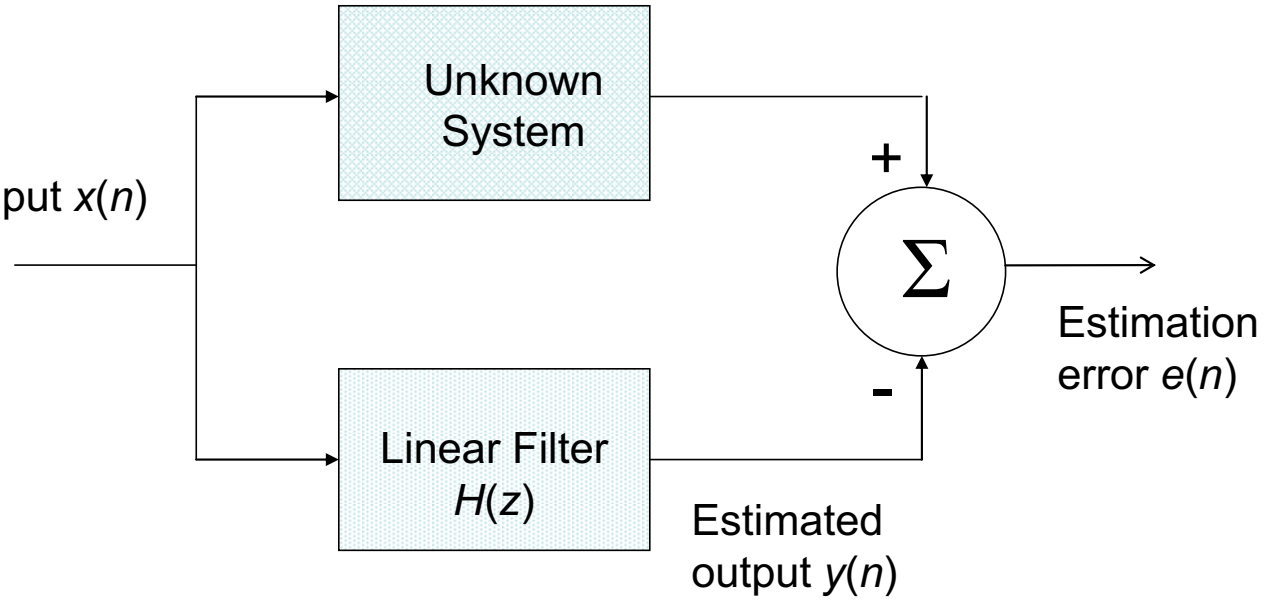
where $b[n]$ is the impulse response of the linear filter and $x[n]$ is the input signal.

The output of the filter, $y[k]$, is an estimate of the desired signal, $d[k]$, and the error signal found by subtraction: $e[k] = d[k] - y[k]$.

Wiener Filter ~ System Identification

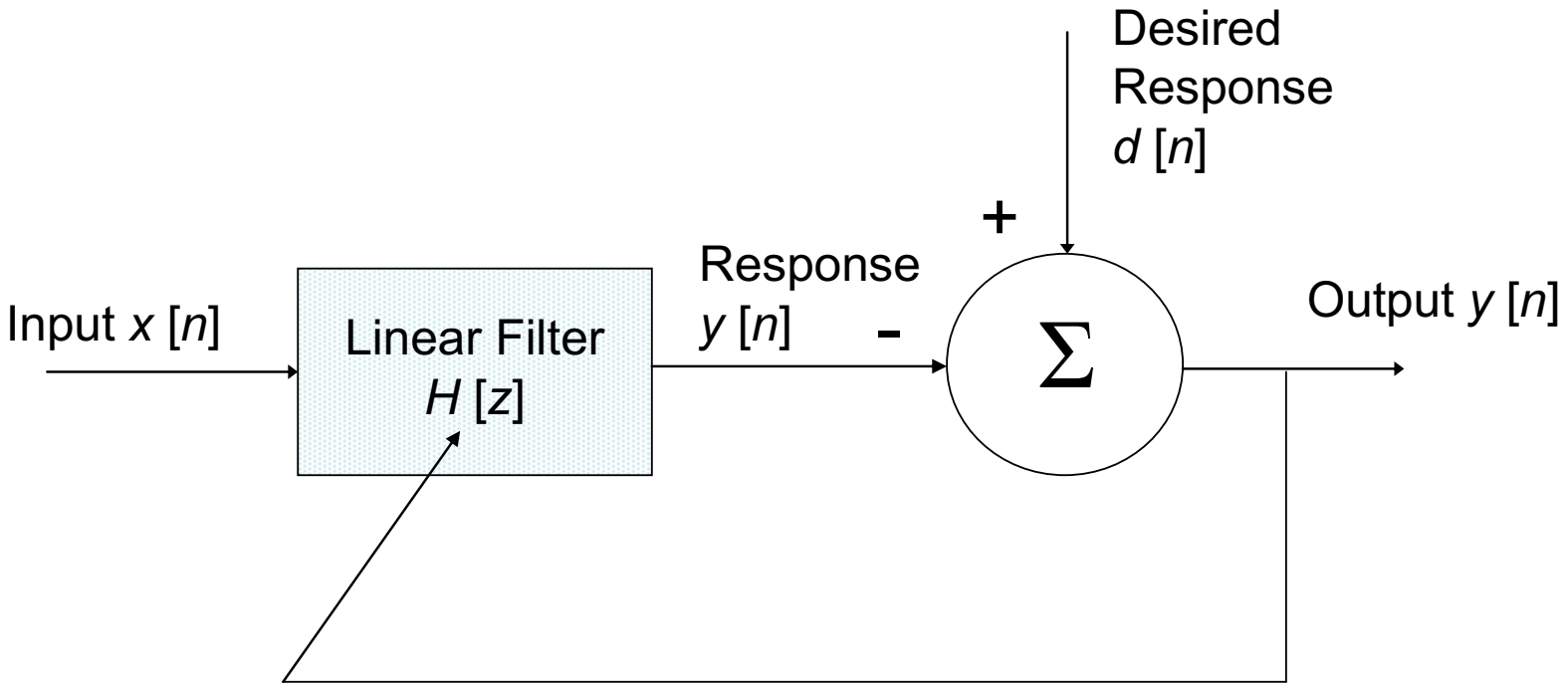
The Wiener-Hopf approach has a number of other applications in addition to standard filtering including systems identification, interference canceling, and inverse modeling or deconvolution. For system identification, the filter is placed in parallel with the unknown system.

The desired output is the output of the unknown system, and the filter coefficients are adjusted so that the filter's output best matches that of the unknown system.



Adaptive Filters

Classical filters and optimal Wiener filters have fixed frequency characteristics that cannot respond to changes that might occur during the course of the signal.



Adaptive filters modify their properties based on selected features of the signal being analyzed.

Adaptive Filtering

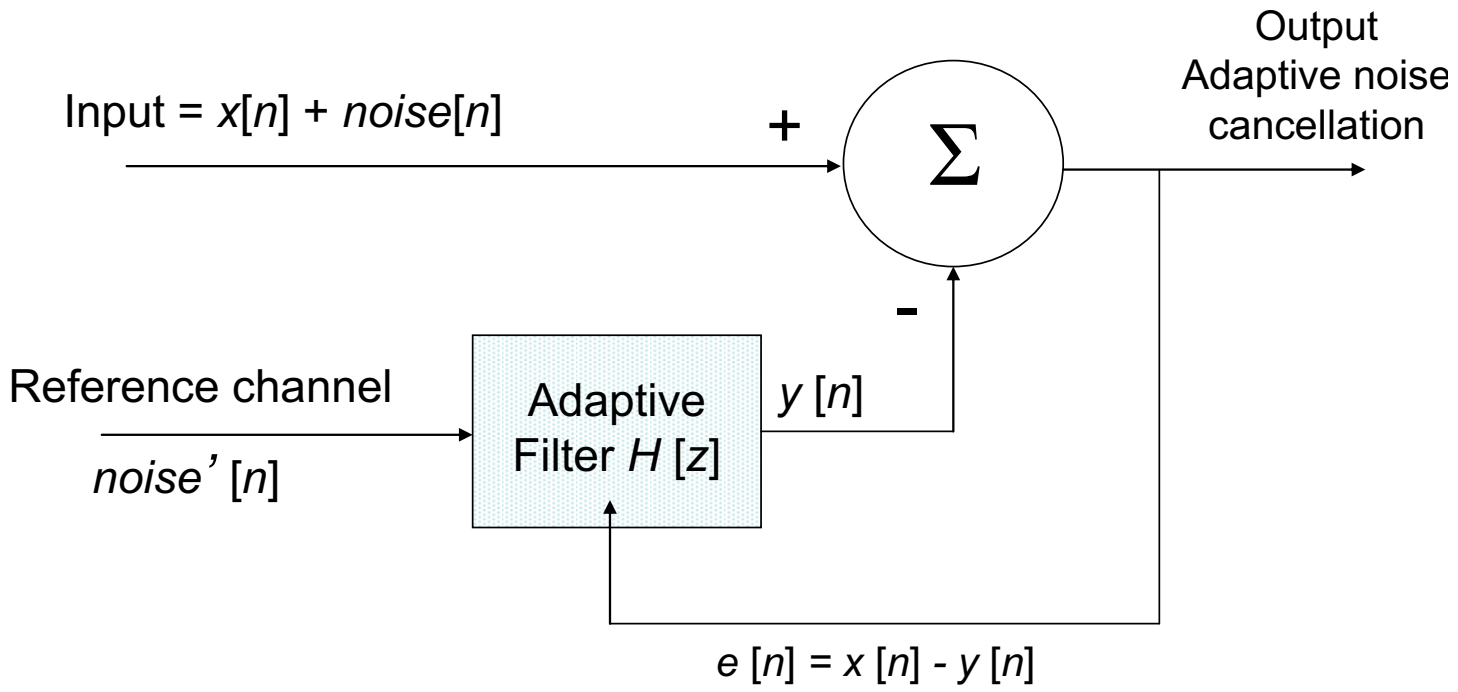
- Adaptive filters are often implemented as FIR filters, but with coefficients $b_n[k]$ that change over time. Again:

$$y[k] = \sum_{n=1}^{L} b_n[n]x[k-n]$$

- The adaptive filter operates by modifying the filter coefficients, $b_n[k]$, based on some signal property.
- In the Wiener approach, the analysis is applied to the entire waveform, and the resultant optimal filter coefficients are similarly applied to the entire waveform: a so-called block approach.
- In adaptive filtering, the filter coefficients are adjusted and applied in an ongoing basis as indicated next.

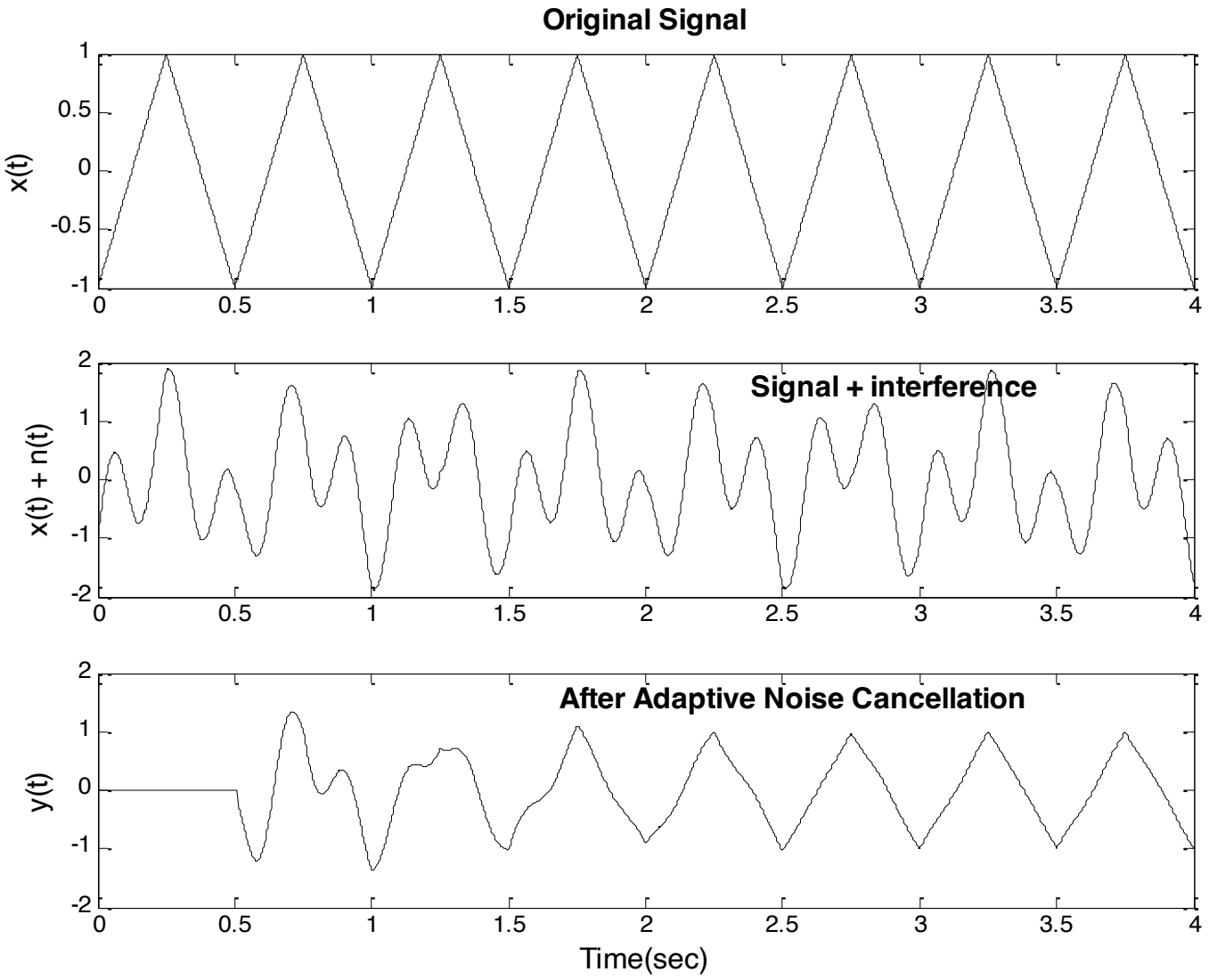
Adaptive Noise Cancellation

Adaptive noise cancellation requires a reference signal that contains components of the noise, but not the signal.



Adaptive Noise Cancellation (ANC).

In this application, approximately 1000 samples (2.0 sec) are required for the filter to adapt.



Multivariate Analyses

Multivariate Analysis

In multivariate analysis, multiple variables are analyzed together and often represented as a single vector variable that includes the different variables :

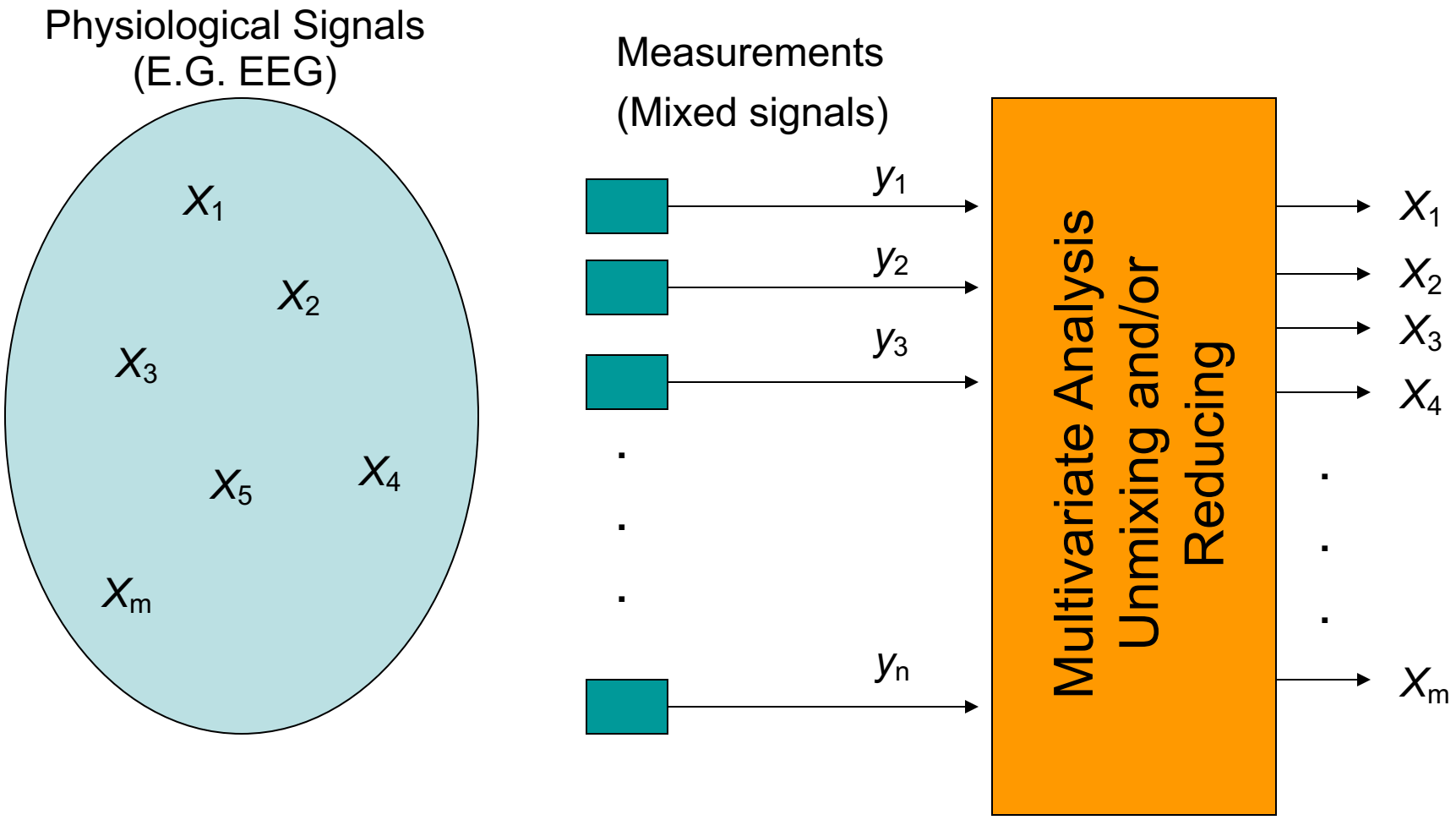
$$\mathbf{x} = [x_m(1), x_m(2), \ldots x_m(N)]^T \quad \text{for } 1 \le m \le M$$

The '*T*' stands for transposed.

The variable $\boldsymbol{X}$ is composed of *M* variables (rows) each containing *N* observations (columns). In signal analysis, the observations are time samples, (*t* = 1,...N)

$$X = \begin{bmatrix} x_1[1] & x_1[2] & x_1[3] & \cdots & x_1[N] \\ x_2[1] & x_2[2] & x_2[3] & & x_2[N] \\ x_3[1] & x_3[2] & \ddots & & x_3[N] \\ \vdots & \vdots & & \ddots & \vdots \\ x_M[1] & x_M[2] & x_M[3] & \cdots & x_M[N] \end{bmatrix}$$

Multivariate Applications in Behavioral Physiology



The *y*'s (measured) are mixtures of the *x*'s.

You have the y's, but you want the *x*'s.

Usually $n > m$.

Multivariate Data Transformations

- In transformations that reduce the dimensionality of a multi-variable data set, the idea is to transform one set of variables into a new set where some of the new variables have values that are much smaller that the rest.
- The data transformation used to produce the new set of variables is often a linear function.
- A linear transformation can be represented mathematically as:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_M(t) \end{bmatrix} = W \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{bmatrix}$$

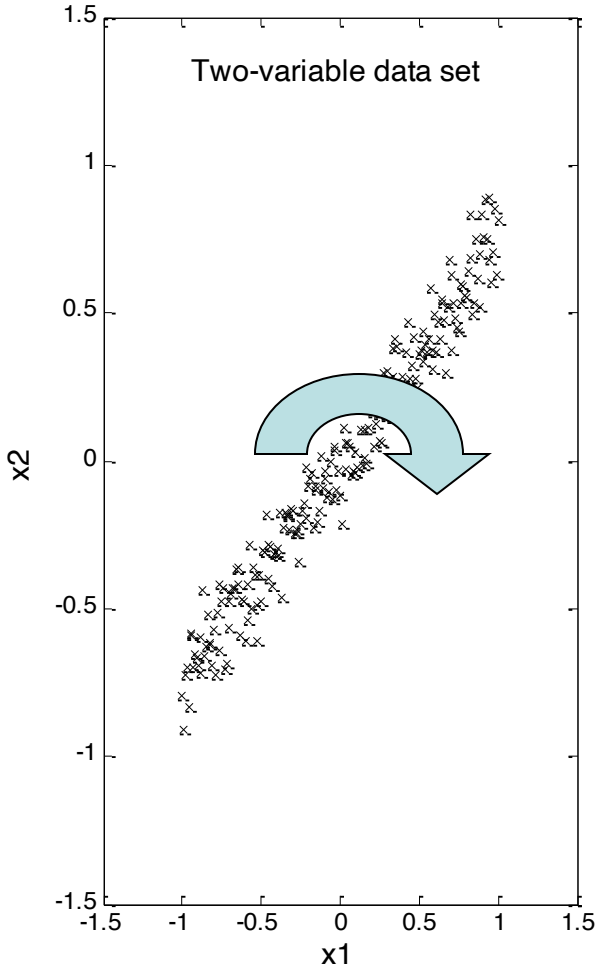
or: $y_i(t) = \sum_{j=1}^{M} w_{ij} x_j(t) \qquad i = 1, \ldots N$

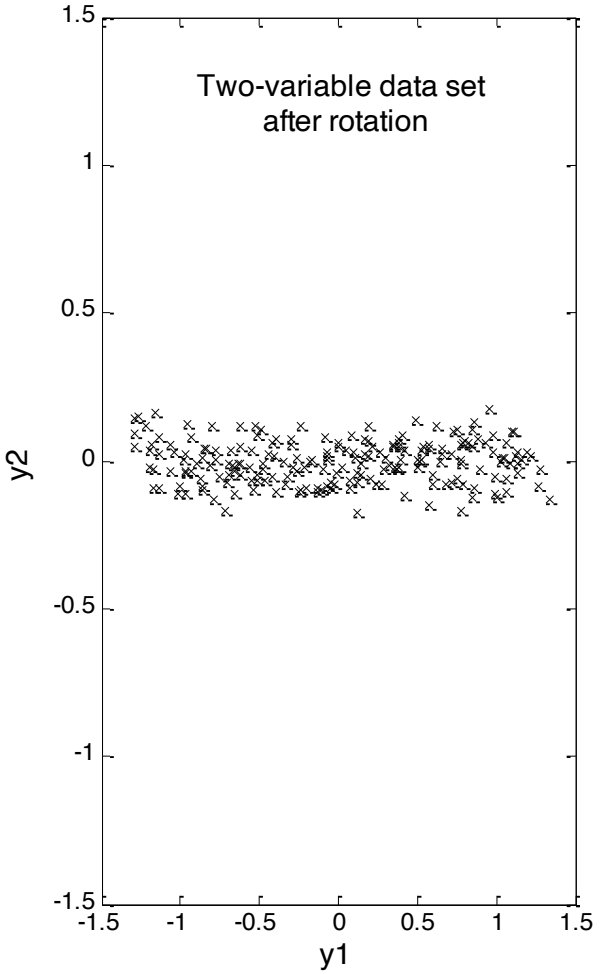
where W (= w_{ij}) is a matrix that defines the transformation.

Linear Transformations

A linear transformation can be interpreted as a rotation of the data set.

- These are called "Scatter Plots," plots of one data variable against another (point-by-point).
- The rotated data set still contains two variables, but the variance of one of the variables is quite small compared to the other.
- Perhaps this new variable (y_2) just represents noise and could be eliminated





Principal Component Analysis

- PCA is a linear transformation used to transform one set of variables into another set of variables.
- PCA is used to provide information on the true dimensionality of a data set.
- PCA tells you if the data set can be transformed into a fewer number of variables that still contain most of the essential information.
- PCA also implements that transformation.

Principal Components Analysis (continued)

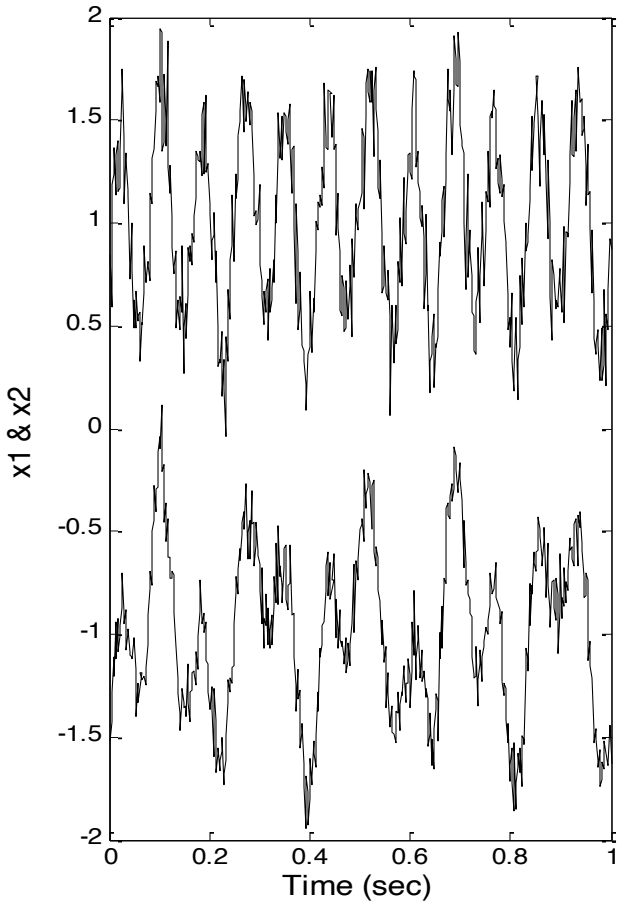
- PCA rotates (transforms) a data set until the all the variables are uncorrelated. (This is the geometrical interpretation of PCA)

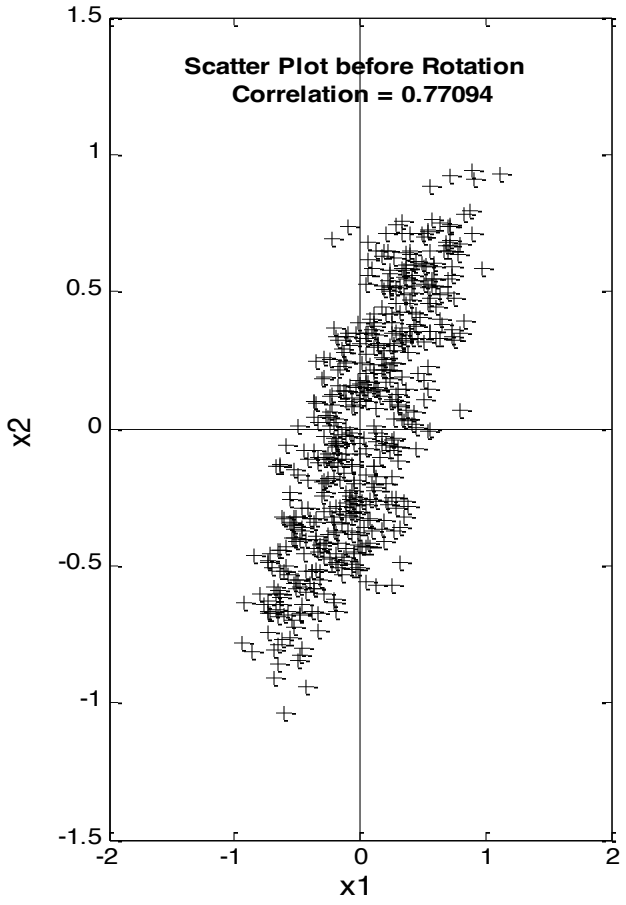
- Uncorrelated variables provide no information on one another.
  They have no information in common, but they are not necessarily independent (unless the variables have Gaussian distributions).

- Examples will use only two variables as two-variable data sets are easier to plot and to visualize, but approach generalized to any number of variables.

This figure shows a two-variable data set made from the mixtures of two sine waves. Each mixture contains different amplitudes of the two sinusoids and noise.

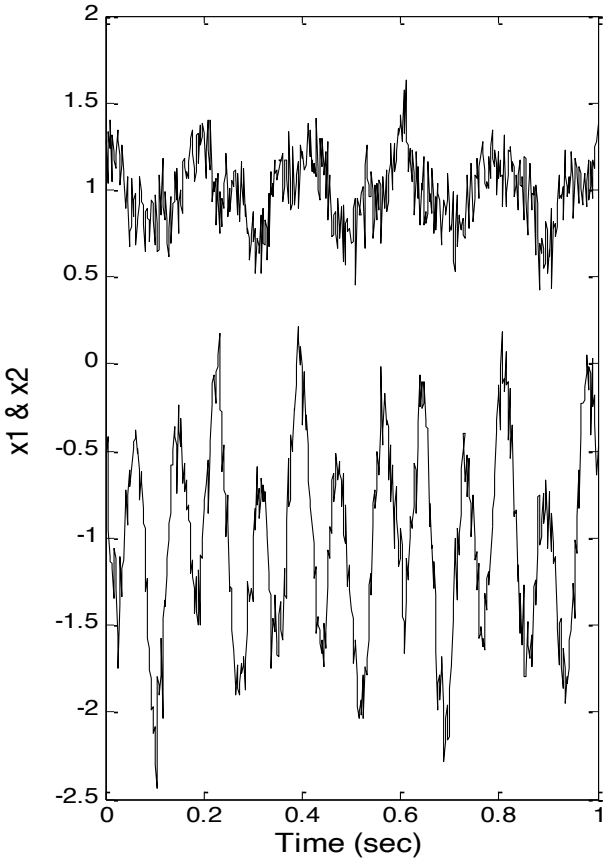
- There is a high degree of correlation between x_1 and x_2.
- Knowing the value of x_1 gives you a range of possible x_2 values.

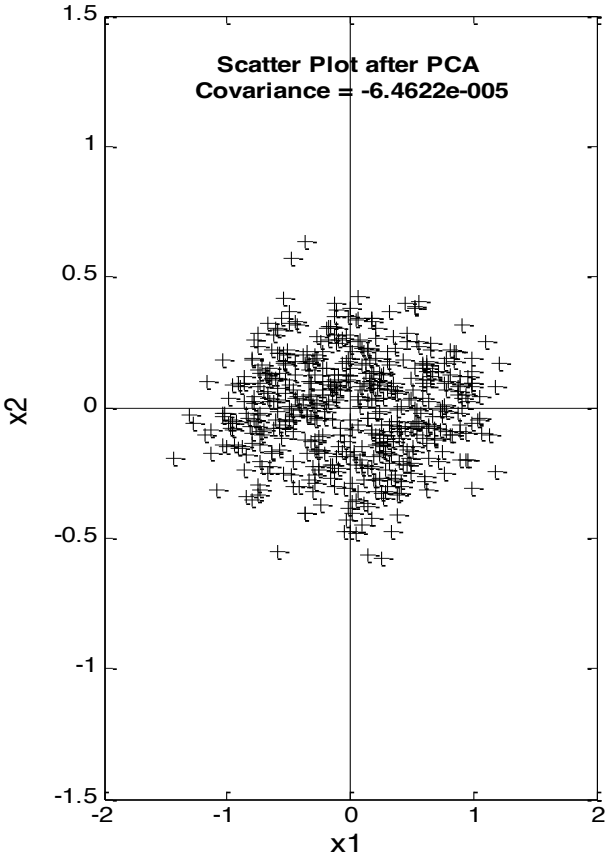




After rotation the two variables are no longer correlated. (The means are also removed: the data are "centered" in the scatter plot.)

- However the new variables are still mixtures of the two sources (just a different mixing).
- The new variables are not more meaningful than the original variables.





- Moreover this data set can not be reduced. In this case, you really do need two variables to represent the two sinusoids.

Data Reduction using PCA

- The eigenvalues describe how much of the variance is accounted for by the associated principal component and if a component is really necessary.
- Eigenvalues are ordered by size; that is:

$$\lambda_1 > \lambda_2 > \lambda_3 \ldots > \lambda_M.$$

- If an eigenvalue is zero or 'close to' zero, then its associated principal component contributes little to the data and can be eliminated. This component accounts for only a small amount of the variance in the data.
- This tells us the effective dimension of the data set.
- How do you decide if an eigenvalue is small enough so that its associated component can be removed from the data set?

Data Reduction (cont)

There are two popular methods for determining eigenvalue thresholds.

1) Take the sum of all eigenvectors (which must account for all the variance), then delete those eigenvalues that fall below some percentage of that sum.

2) Plot the eigenvalues in order and look for breakpoints in the slope of this curve. Eigenvalues representing noise should not change much in value and will plot as a flatter slope when plotted sequentially.

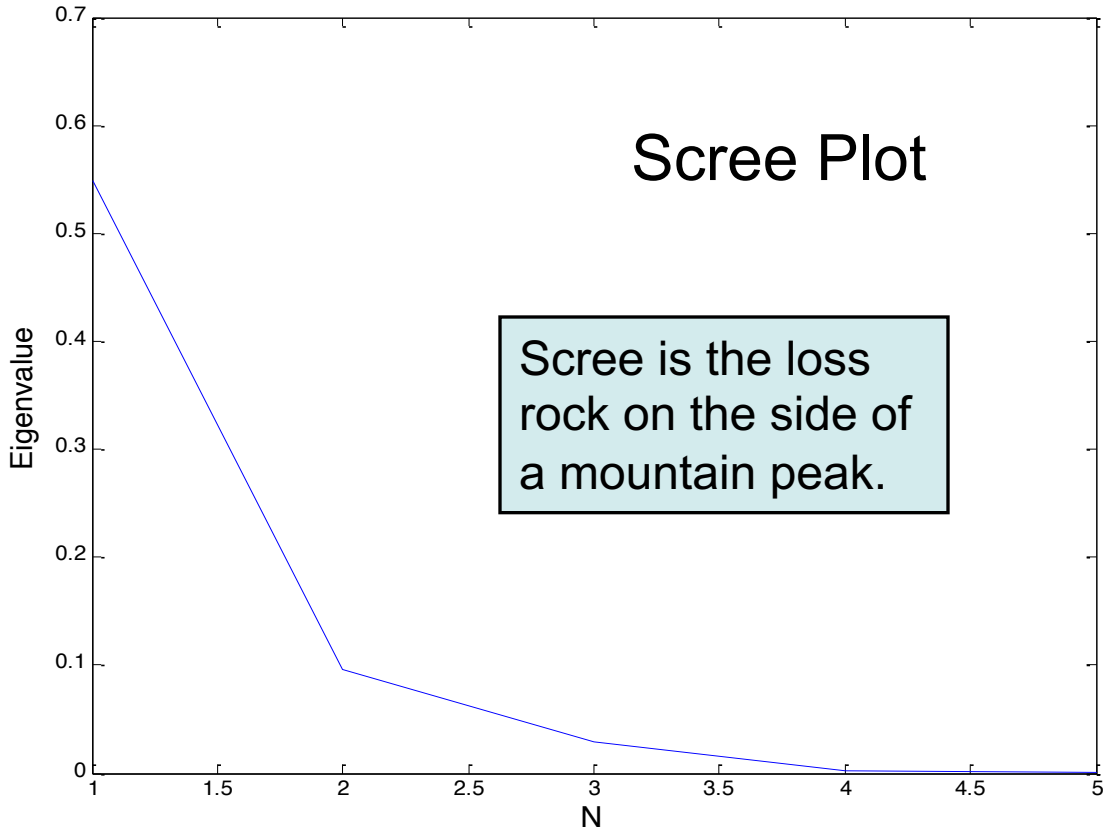
   Such a plot is called the Scree Plot.

The Scree Plot

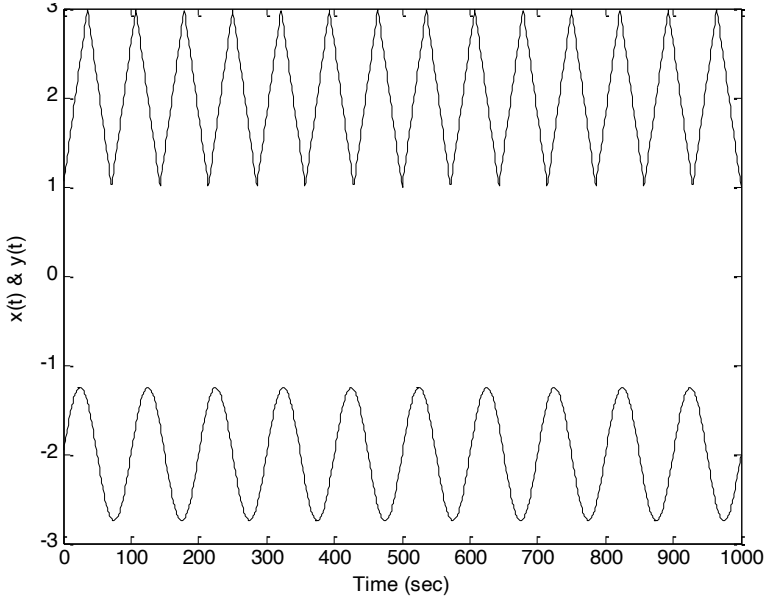
The eigenvalues are in order of large to small. Plotting them in order will show their relative value.

Such a plot is called the Scree plot

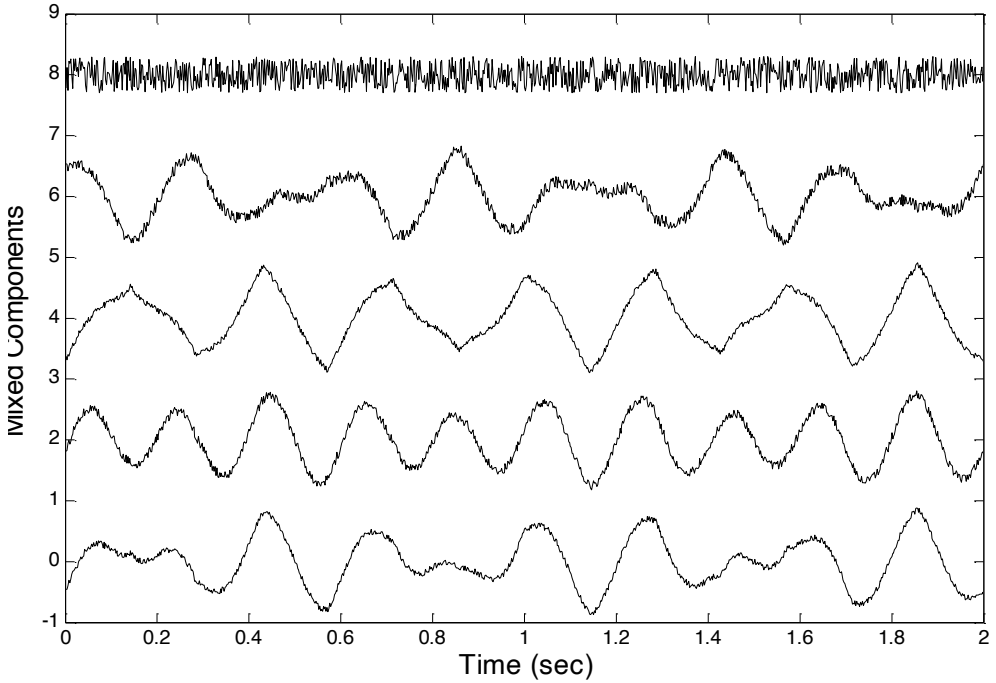
The actual dimension of the data set is taken where the Scree plot becomes more-or-less flat.



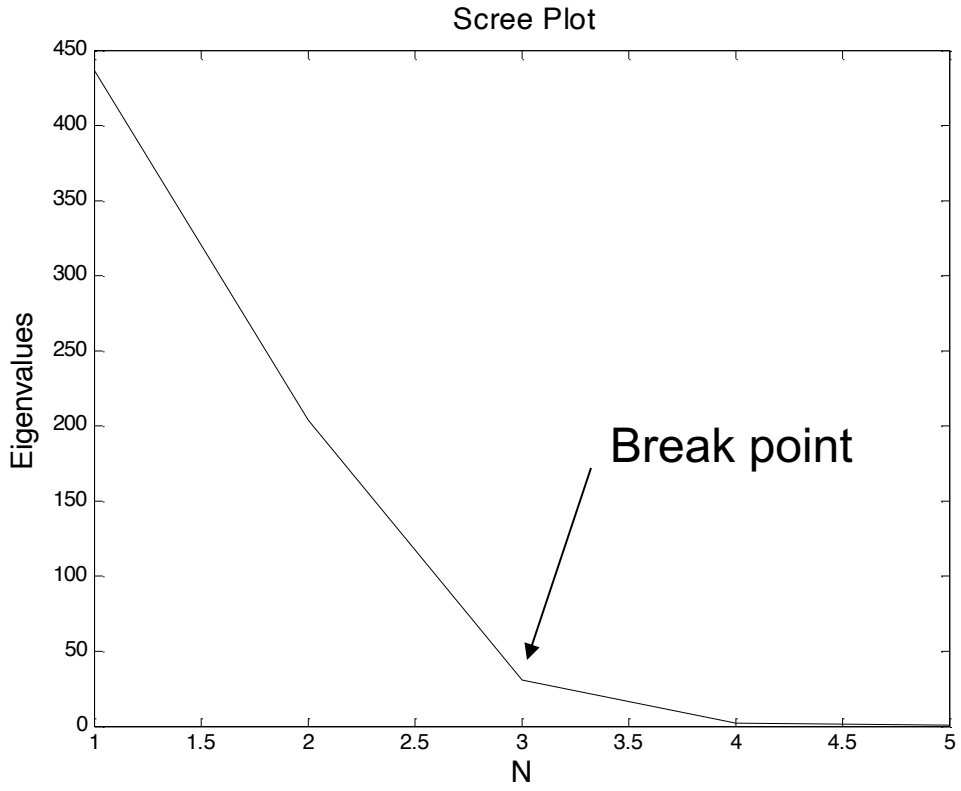
Original Data



Mixtures of Original Data used for PCA analysis



The scree plot of eigenvalue against component number shows a break at 3, suggesting the last 3 components are negligible and only 2 principal components are required to describe the data.

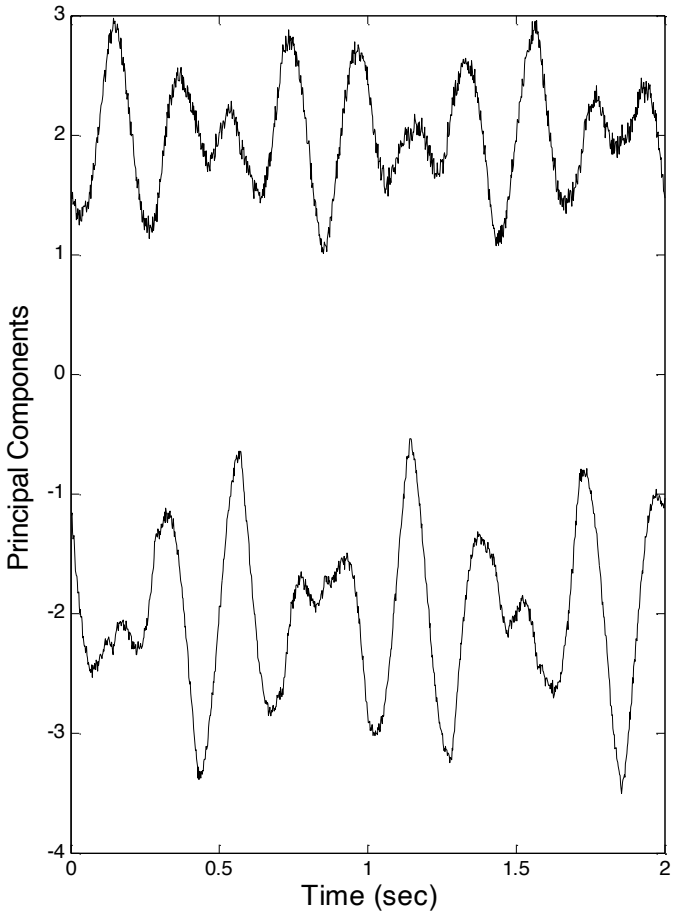


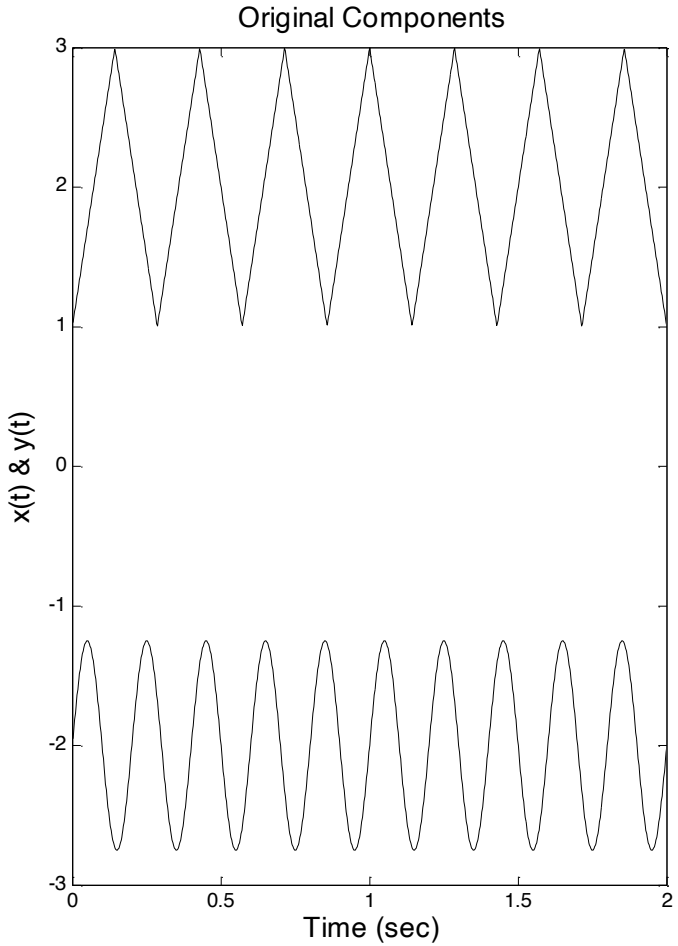
The last three components account for only 4.9% of the variance of the data, suggesting that the actual dimension of the data is two.

The percentage of variance accounted by the sums of the various eigenvalues is given as:

CP 1-5	CP 2-5	CP 3-5	CP 4-5	CP 5
100 %	35 %	4.9 %	0.44 %	0.14 %

Example ~ Results





Plot of the first two principal components and the original two sources.

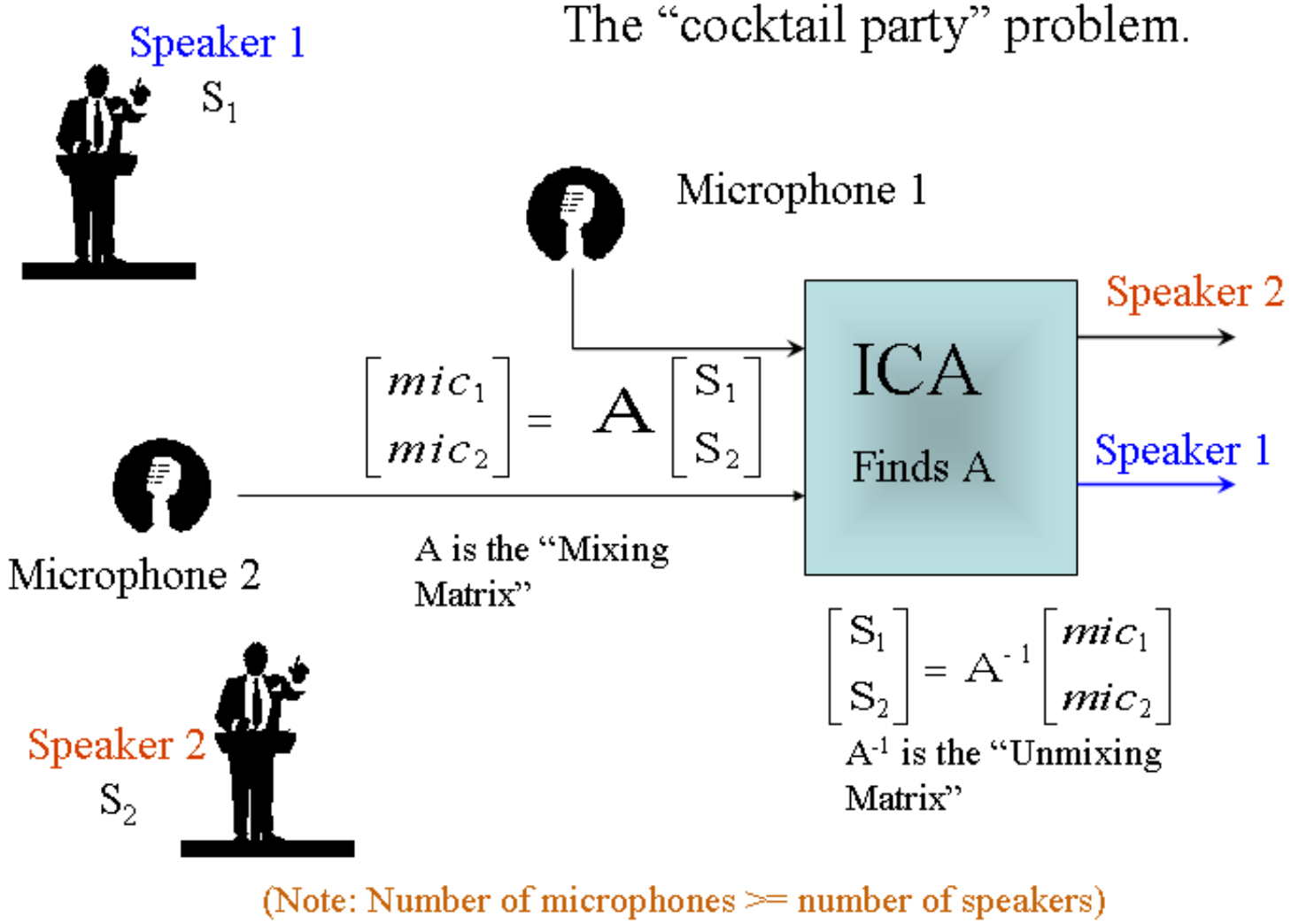
Even though the two signals are uncorrelated they are not independent since they are clearly still mixtures of the two sources.

Independent Component Analysis

- Independent Component Analysis seeks to transform the original data set into a number of independent variables. The motivation is primarily to uncover more meaningful variables, not to reduce the dimensions of the data set.

- When data set reduction is also desired it is usually accomplished by preprocessing the data set using PCA.

- One of the more dramatic applications of Independent Component Analysis (ICA) is found in the 'cocktail party problem.' In this situation, multiple people are speaking simultaneously in the same room.



ICA "unmixes" the signals in each microphone to recover the speech of each speaker. No information is needed about either the placement of speakers or microphones nor the content of the speeches.

Independent Component Analysis Equations

In matrix form, this equation becomes:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} = A \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix}$$

Measured Hidden

or $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$

where $\boldsymbol{s}$ is a vector composed of all the source signals, $\boldsymbol{A}$ is the mixing matrix composed of the constant elements ai,j, and $\boldsymbol{x}$ is a vector of the measured signals.

The model described by these equations is also known as a *latent variables* model since the source variables, $\boldsymbol{s}$, cannot be observed directly: they are hidden, or “latent” in $\boldsymbol{x}$.

- ICA techniques are used to solve for the unmixing matrix, $\boldsymbol{A}^{-1}$, from which the independent components, $\boldsymbol{s}$, can be obtained:

$$\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{x}$$ where $\boldsymbol{A}^{-1}$ is the unmixing matrix.

- If you know the mixing matrix, $\boldsymbol{A}$ finding the unmixing matrix is easy. But usually you do not know $\boldsymbol{A}$.

- ICA can find $\boldsymbol{A}^{-1}$ even when the mixing matrix is not known.

- If the measured signals, $\boldsymbol{x}$, are related to the underlying source signals, $\boldsymbol{s}$, by a linear transformation (i.e., a rotation and scaling operation), then some inverse transformation (rotation/scaling) should recover the original signals.

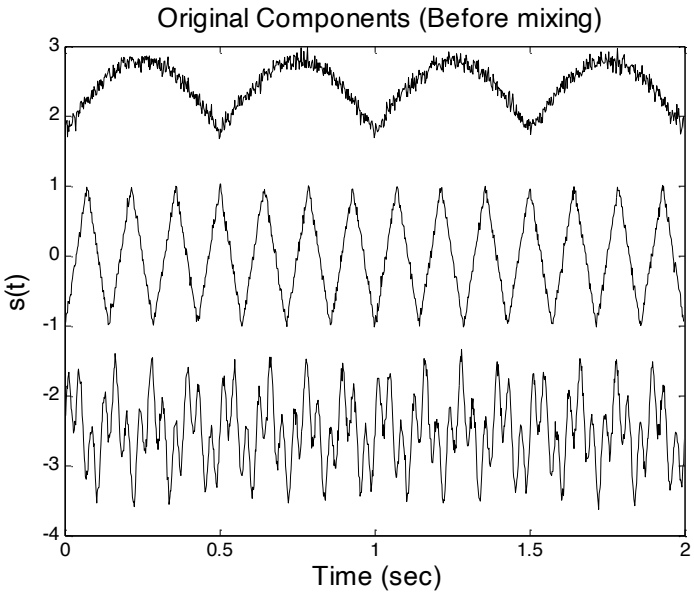
ICA Limitations

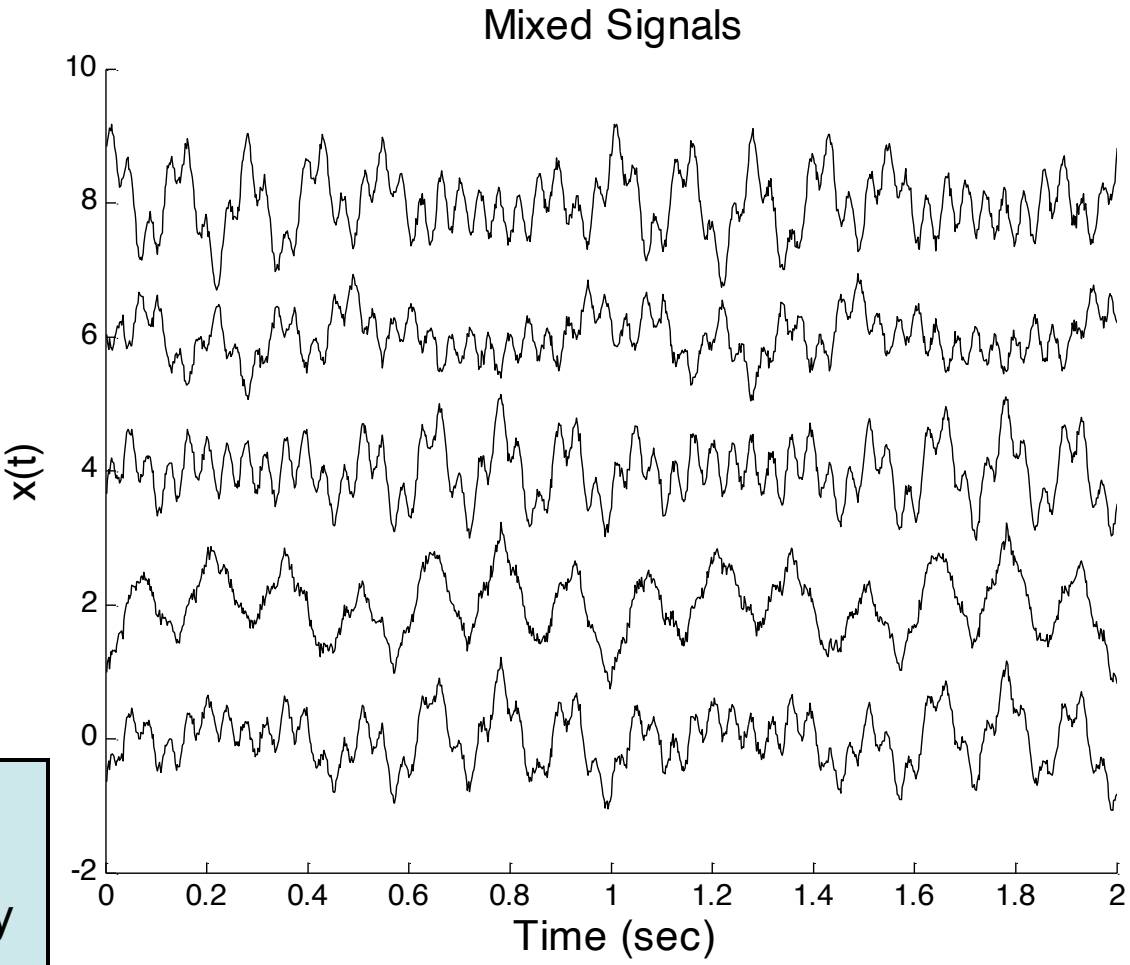
- ICA can only be applied to non-Gaussian signals because it relies on higher-order statistics to separate the variables. Higher-order statistics (i.e. moments and related measures) of Gaussian signals are zero.

- Since ICA has available only the measured variables (it has no information on either the mixing matrix, $\boldsymbol{A}$, or the underlying source variables, $\boldsymbol{s}$.), there are some limits to what ICA can do:

  1. ICA cannot determine the variances, hence the energies or amplitudes, of the actual sources.

  2. Unlike PCA, the order of the components cannot be established. (Logical if amplitudes cannot be determined.)





The original signals are not discernible in the mixture, nor is it obvious that there are only three source signals in the mixture.

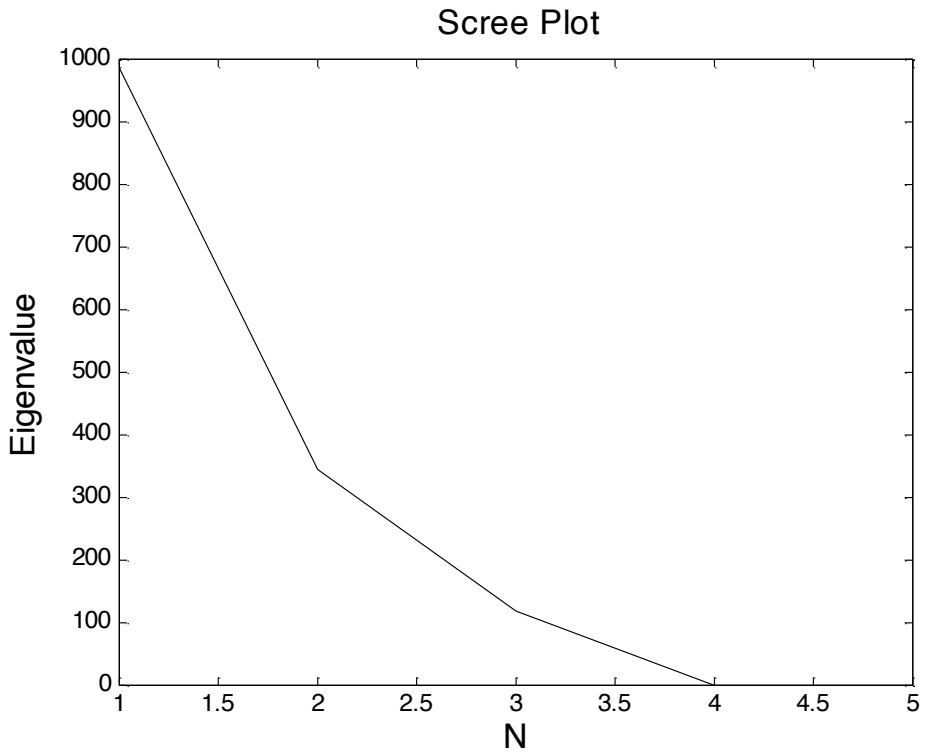
Variance

Table of Variances

CP 1-5	CP 2-5	CP 3-5	CP 4-5	CP 5
100 %	31.3 %	8.25 %	0.0 %	0.0 %

The percent variances indicate that none of the variances is represented by the last two components, but 8.25% is contained in the last three components suggesting that three components are present. (Using a cutoff of 5%)

Scree plot

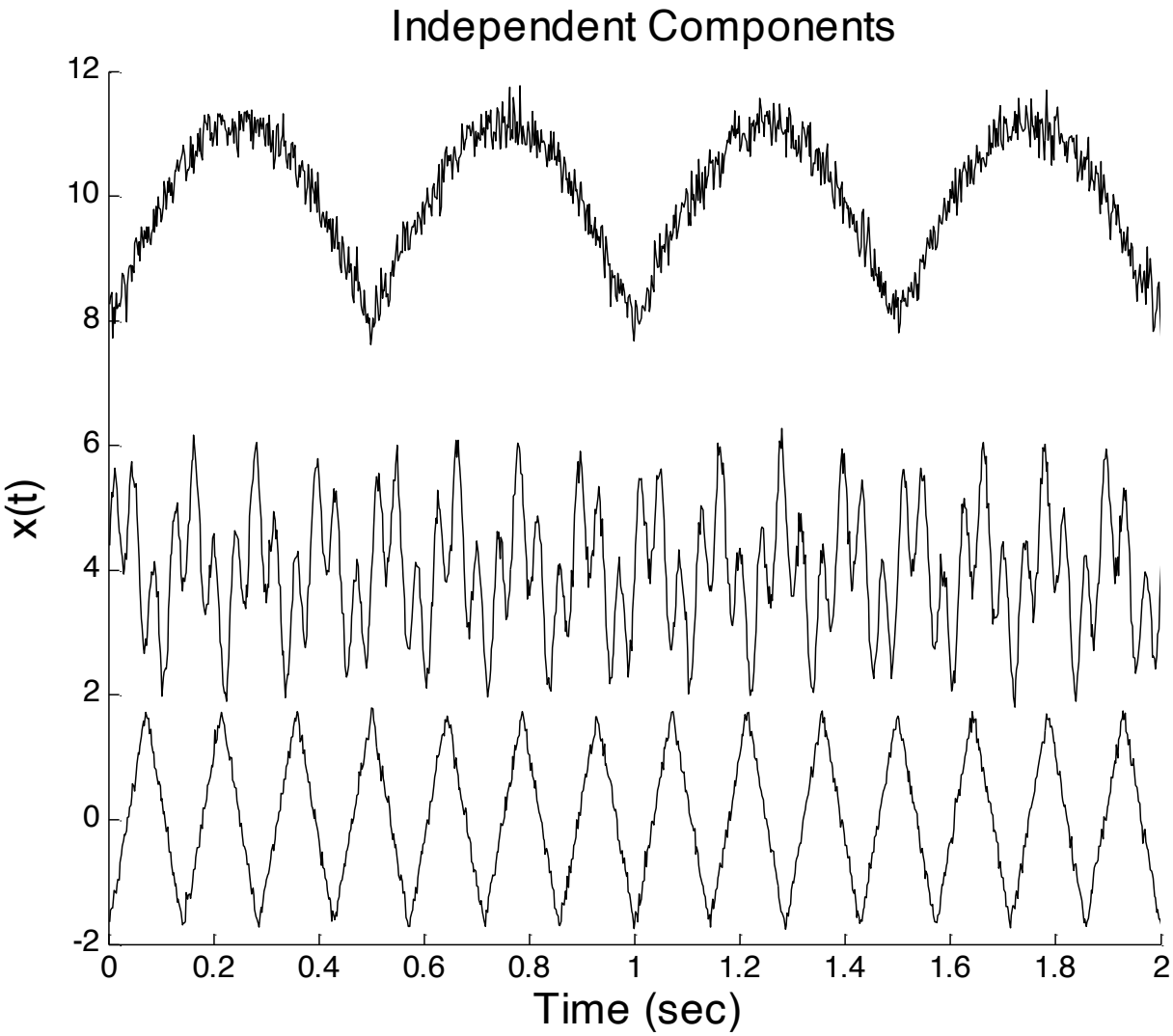


The scree plot of the eigenvalues obtained from the five-variable data set does show a break at three but a sharper break at four where it and goes to zero.

This suggests that there are at least 2 and probably 3 separate components,

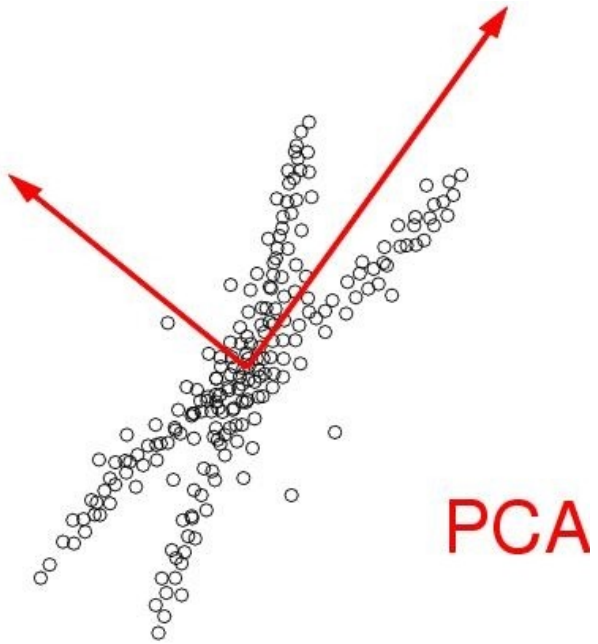
Independent components

Applying ICA to the five-variable mixture recovers the original source signals.

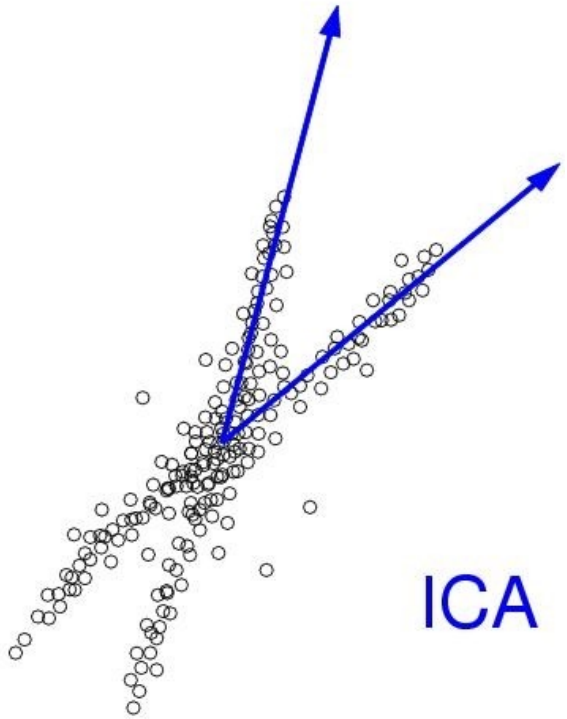


They are the same as the original signals except for a slight change in signal amplitude which is expected.

PCA finds the directions of maximum variance



ICA finds the directions of maximum independence



Chaos and Nonlinear Dynamics

Why nonlinear processing?

- The real world is nonlinear
  - Often we can make linear assumptions, but we lose information that way.
- Nonlinear analysis can provide greater information regarding **dynamics** of the measured system
  - Dynamics refers to the governing rules of the system from which the signal originated

Nonlinear Vs. Linear

- Nonlinear signals come from nonlinear systems
- Nonlinear systems have governing equations with nonlinear terms
  - Sine, exponentials, logarithms, etc…
- The input-output relationships do not follow super-position
- A hallmark of a nonlinear system is that small changes can be amplified, or vice versa

Simple Nonlinear System

- $y = x^2$
  - Here y and x are variables representing the output and input of a signal
- Inputs of a and b to this system result in a^2 and b^2
- But an input of $(a + b)$ results in an output of
  $a^2 + 2ab + b^2$ which is not equal to $a^2 + b^2$
- This system fails the test of super position

A More Complicated Nonlinear System

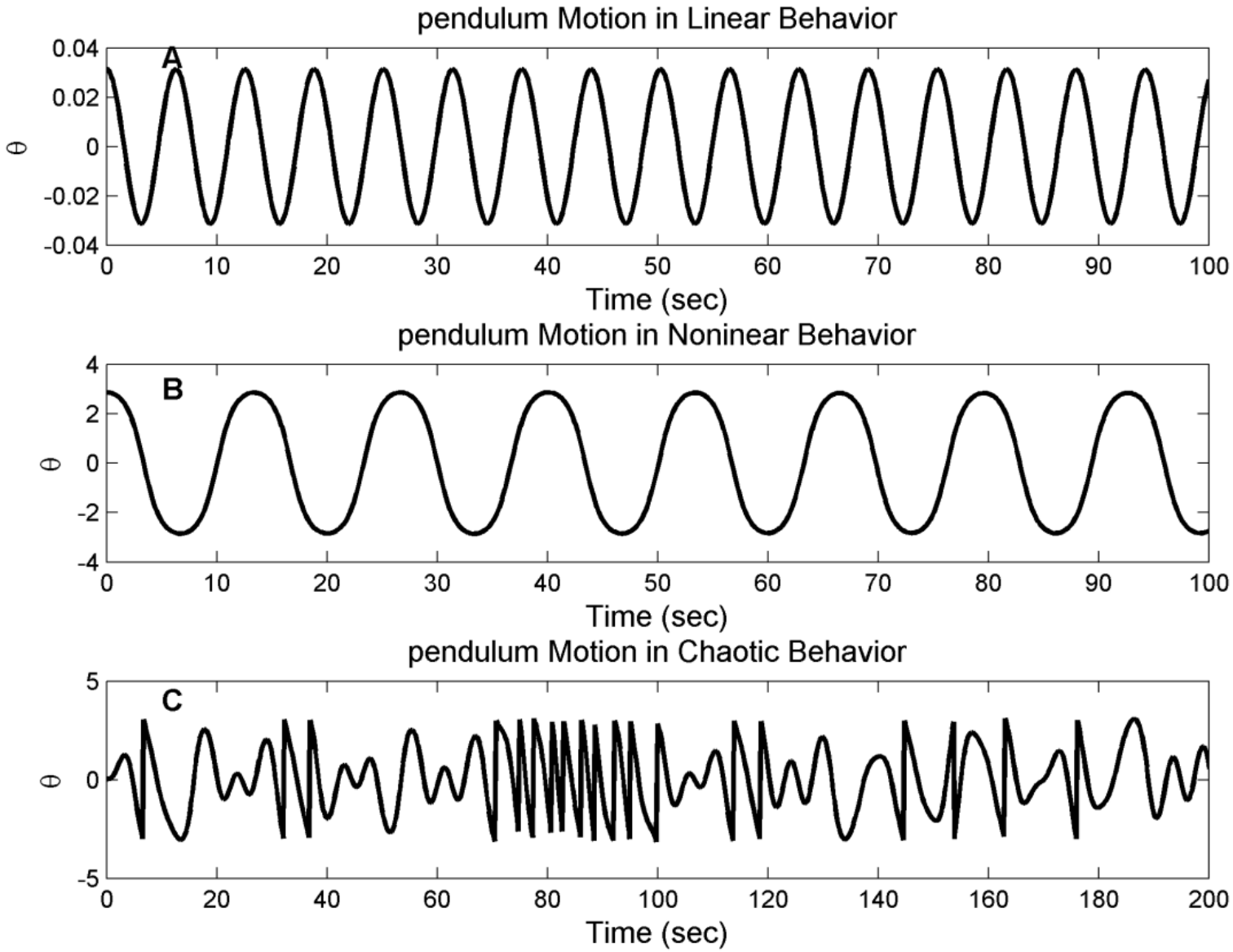
- The damped driven pendulum can be modeled as

$$\dot{\theta} = \omega$$

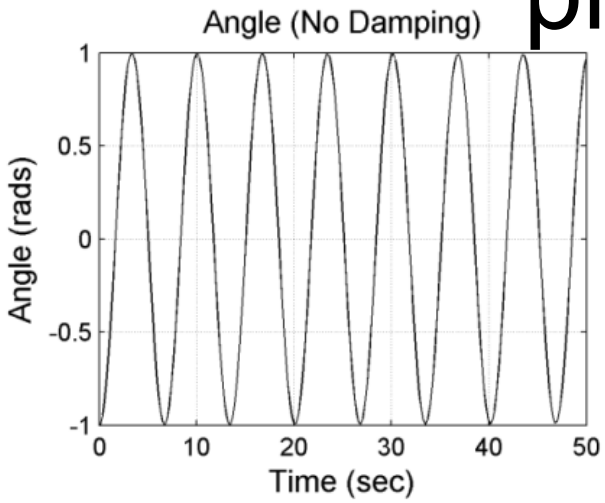
$$\dot{\omega} = -\frac{g}{l}\sin(\theta) + k\sin(t) - b\omega$$

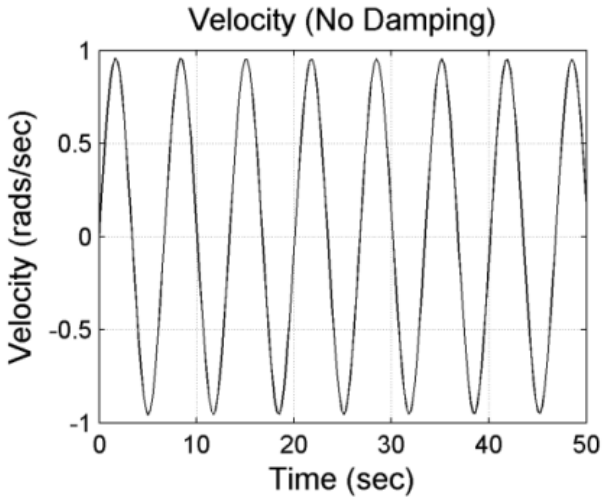
- Where θ and ω are the angular position and velocity of the pendulum, *b* is the damping constant, and *k* is the driving force function, *g* is the force of gravity, and *l* is the length of the string.
- This system has altered modes of behavior depending on the values of *b* and *k* and the initial conditions
- This system is nonlinear because of the sine terms

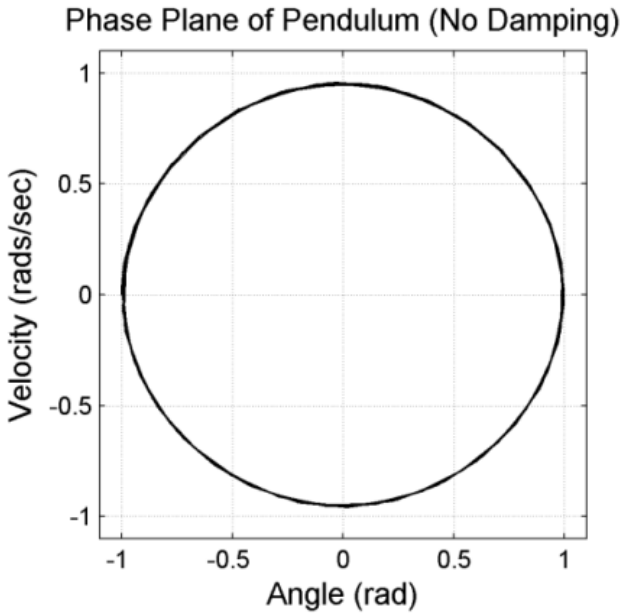
Pendulum in 3 modes of behavior



Visualizing the state-space with a phase-plane plot



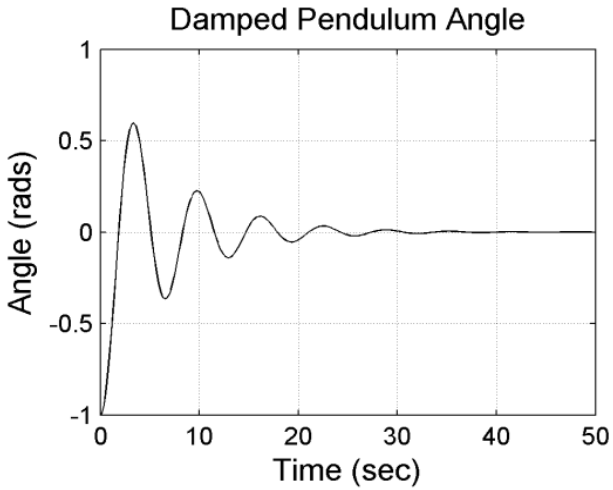


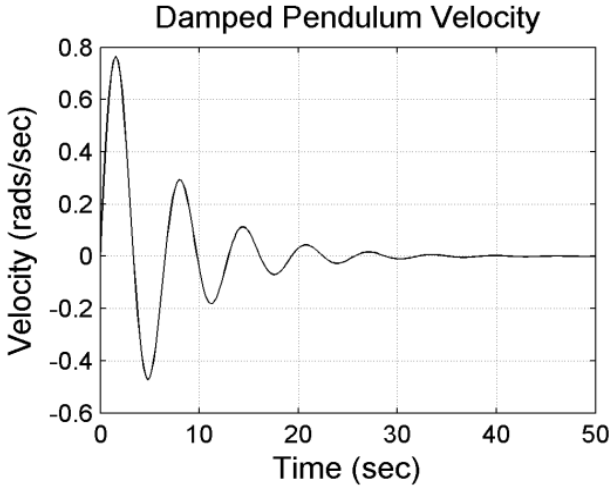


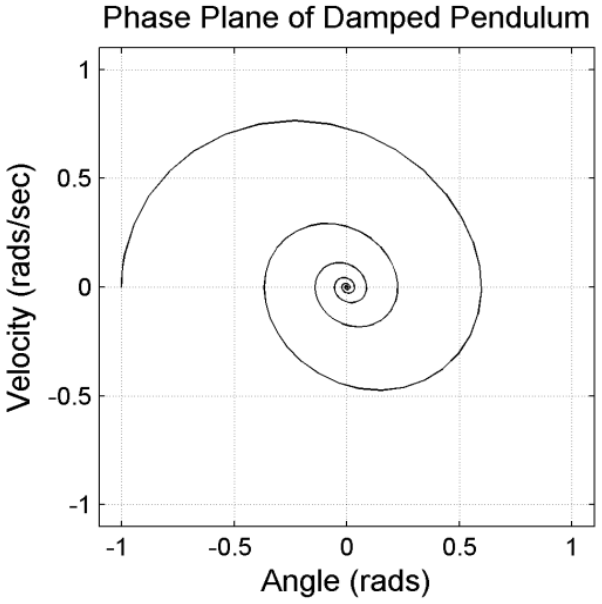
- A plot of the state variables against each other
- If the phase-space has 2 dimensions, a plot is known as the phase-plane
- Depicts the **attractor** of the system

The attractor is the tendency of the system at equilibrium. In this example it is a limit cycle

Visualizing the state-space with a phase-plane plot







- Pendulum with damping
- Here the attractor is a stable node

Phase-space plots of measured data

- The phase-space can give us important information about a system (from its signals)
- Nonlinear systems can have complicated phase-space plots
- However, we often do not have access to all the variables.
  - For example, body temperature is one variable of the inflammation response but others are invasive
- Is there a way to recover, or at least estimate the phase-space of a multidimensional system from a phase-space plot?

Answer: Yes, With Delay Embedding

- Florence Takens showed that the method of delayed embedding sufficiently reconstructs a time series
- Reconstruction is just an estimate, not perfect
- Method
  - Use a delayed version of the signal as a "new" dimensional measurement time series. Each additional series is delayed by a multiple of the chosen delay τ

Delay embedding

- Formally we would say time series $x[n]$ of length N can be reconstructed into multidimensional time series $y[n_d, k]$ of k dimensions from 1 to m, where each delayed vector n_d comes from $x[n]$ delayed by τ

$$y[n_d, k] = x[n \square (k-1)\tau, k] \ldots x[N-(m-1)\tau, m],$$

Getting a good embedding

- Takens showed that
  - *m* should be twice D where D is the "true" dimension of the system
    - True meaning if we had access to each dimension
  - Theoretically τ could be any number
- However in practice
  - *m* just needs to be "sufficiently" large
  - τ can't be too large or too small
    - Each delayed vector should have some correlation to the previous vector, but not too much

Methods of finding embedding dimension

- Trial and error
- False Nearest neighbors
- Principle component analysis

Properties of Chaos

- Exponential Divergence
  - Divergence is the phenomenon of trajectories of a system that begin with similar initial conditions ending up with very different trajectories.
  - This is the opposite of convergence, in which systems tend towards the same value over long periods of time
  - While non, chaotic systems may show divergence, only chaotic systems have trajectories that diverge exponentially.

Information-theoretic approaches

Information and Regularity

- When we measure a signal, if the signal is meaningful, it should teach us something, or communicate something
- Can we quantify what was transmitted?
  - One way to quantify information is with bits (proposed by Shannon in 1948)
  - Bits are handy as they correspond to computer memory
- For example, assume we want to transmit the result of a coin flip, how many bits do we need?
  - 1, since 0 can represent heads and 1 can represent tails (or any convention you want)
- What about the letter A?
  - Depends on the system, but assuming a system that can only contain upper and lower case letters, the number of bits would be $\log_2(26*2) = 5.7$, so it would take at least 6 bits.

Signal entropy

- In general, the number of bits needed to transmit a signal is related to the signal uncertainty
  - Known as the entropy
- Entropy is related to the probability spectrum of the available states of the system.
- A unbiased coin is a specific case of a system where all states have equal probability, so we can take the log of the number of states.
  - But for unbiased, we need to use

$$H_x = -\sum_m p(x) \log_2 p(x),$$

  - Where $p(x)$ is the probability spectrum of the states of the system.

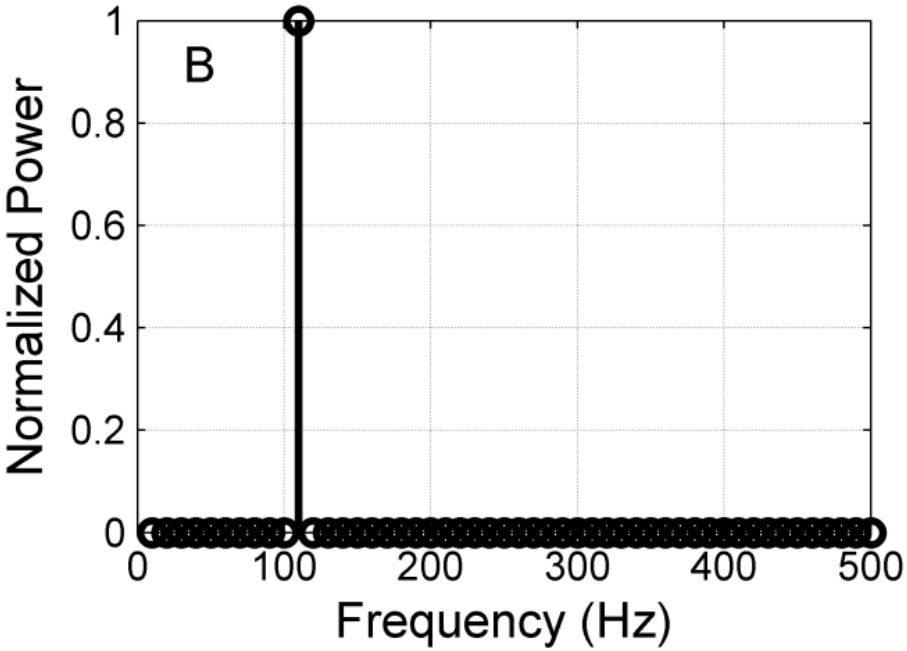
Entropy of a Biased Coin

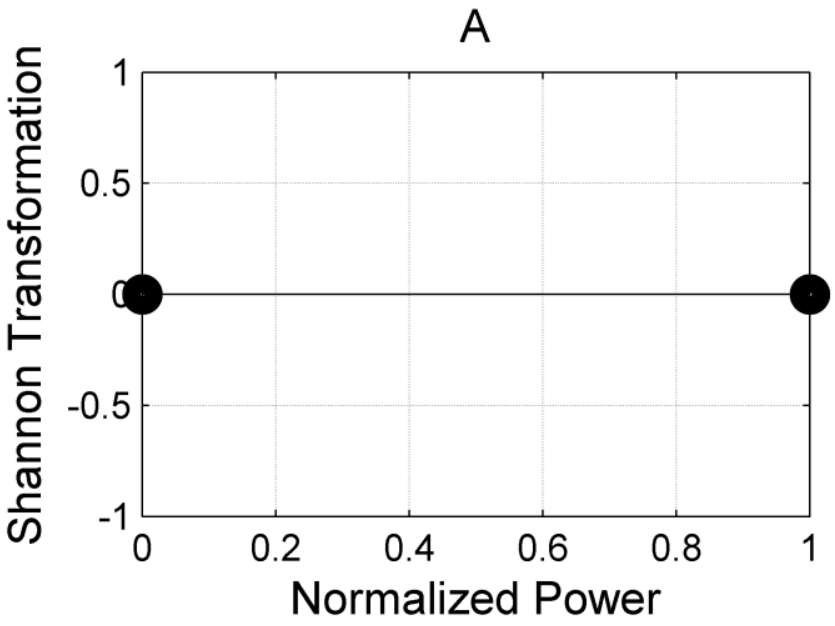
- Say you have a biased coin and after 10,000 flips you record 7,500 heads and 2,500 tails. The entropy would therefore be
  - -p(heads)*log2p(heads)-p(tails)$\log_2$p(tails)
  - –(log(.75)*.75+log(.25)*.25)=.8113
  - This is less than for the unbiased coin because uncertainty has been removed
    - The coin is most likely going to land on heads.
  - The maximum entropy of a system occurs when p(x) = 1/N for all x and is 0 when p(x) = 1

Entropy and regularity

- Entropy for a stochastic signal seems intuitive
  - Stochastic signals are defined by probability spectra
- What about deterministic signals?
  - Entropy can be considered a measurement of signal regularity.
  - Regularity means having periodic properties (not necessarily a sinusoid)
  - Sine waves are very regular, so they are low entropy
  - Deterministic signals may or may not contain high entropy, but are more complicated to analyze than stochastic system

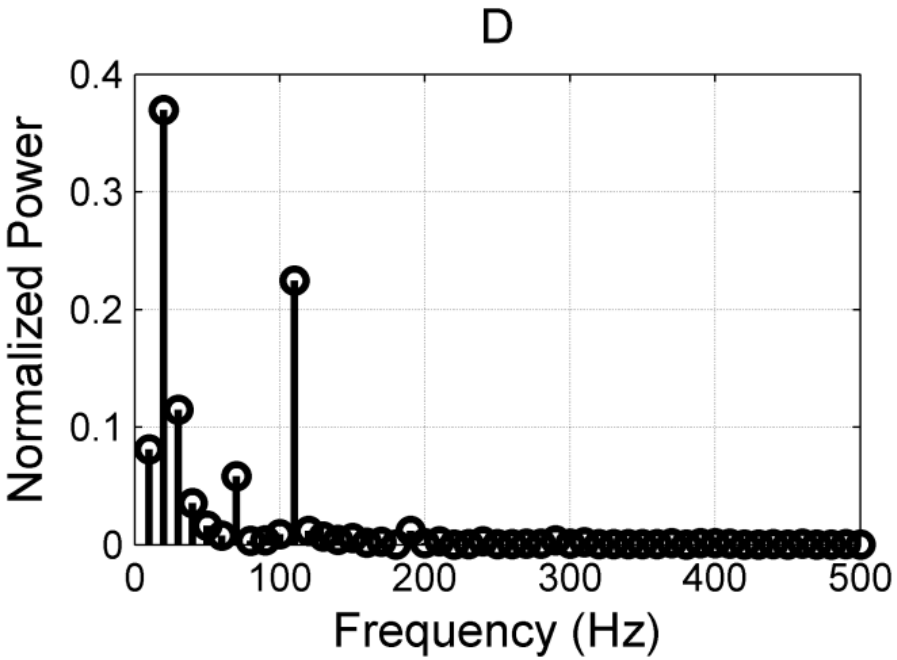
Spectral Entropy in a Sine Wave

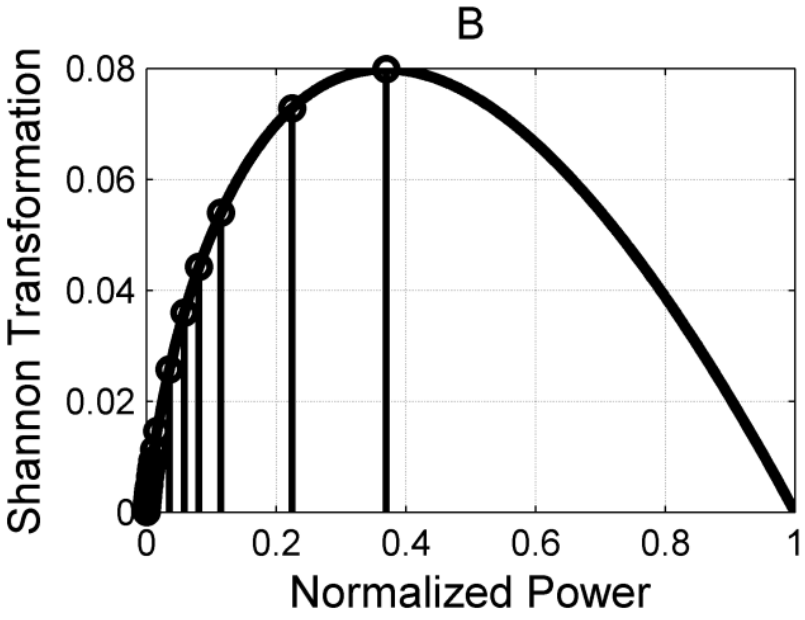




- The normalized magnitude spectrum of the 100 Hz sine wave shows unit power at 100 Hz. (B)
- The log transformed magnitude spectrum shows components at 0 and 1.
- These sum to 0 for zero entropy

Spectral Entropy in a noisy wave





- Noisy sine wave has components between zero and 1
- These are log transformed to non-zero values
- Sums to entropy greater than zero

Need for Advanced Methods

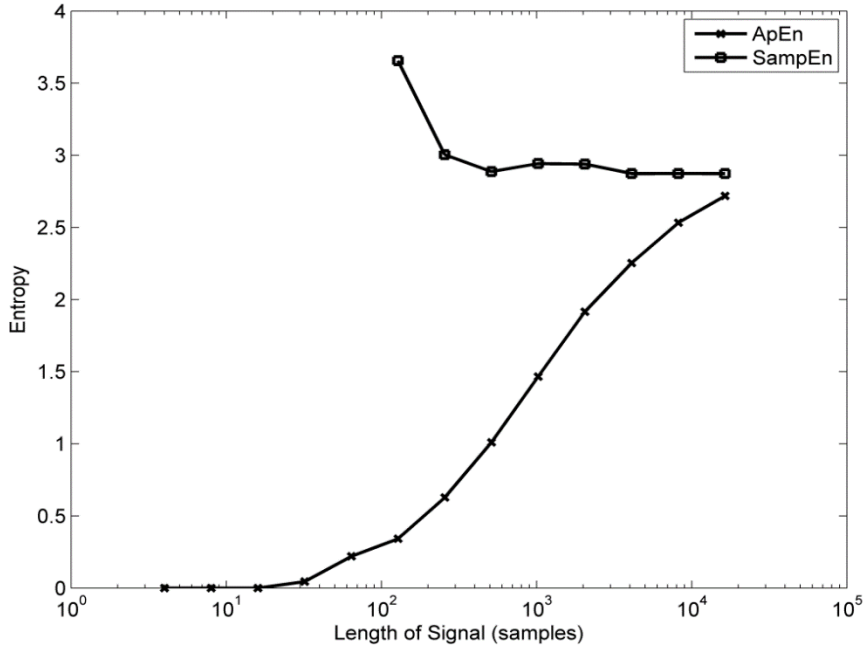
- Spectral entropy
  - Works well for deterministic signals but based on linear properties
- Probability based entropy
  - Works well on stochastic systems (less useful for deterministic systems)
  - Also based on linear (statistical) properties
- We need a method for signal entropy that works well on deterministic signals
- Enter Approximate entropy (and it's refinement, Sample entropy)
  - Can take nonlinear properties into account because they can account for signal dynamics

Approximate Entropy (ApEn)

- Similar to the histogram method, but instead of binning individual samples, we bin short segments
- Since these are now vectors we compare segments using their Euclidean distance
- Segments can take any length, we refer to this as m
- Segments can be separated by a delay τ
- If a sample has many similar repeating segments, it high regularity and low signal entropy

Sample Entropy (SampEn): A refinement of ApEn

- Conceptually similar to ApEn, but reformulated to be more stable and removes bias due to self match



The effects of signal length on ApEn and SampEn. SampEn (squares) gives a reasonable estimate of the entropy for a signal with about 256 samples, but the ApEn (x markers) is not accurate until the signal has more than 2000 samples.

Mutual Information

- We have so far only considered the information in recordings of one event
- What about a system with compound events?
- For example
  - Measure a subjects weight and blood pressure.
- How much information does weight give us about blood pressure?
- What about blood pressure and weight?
- Mutual information (MI) is the quantification of this entity in bits

What do we need to compute mutual information?

- Continuing the previous example, we need…
  - The probability spectrum of blood pressure
    - the probability that a person has a given blood pressure as a function of pressure
  - The probability spectrum of weight
    - the probability that a person has a given weight
  - The joint probability spectra
    - the probability that given a persons weight, they have a given blood pressure, and vice versa.
- What we are really interested in is the information content of the above quantities

More formally

- The MI between two events *x* and *y* is the sum of their individual entropies (H_x, H_y) minus the entropy of one event given the other (H_{xy}).

$$MI_{xy} = H_x \square H_y - H_{xy} = \sum_{mk} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}.$$

$$H_x = -\sum_{m} p(x) \log_2 p(x),$$

$$H_{xy} = -\sum_{mk} p(x,y) \log_2 p(x,y).$$

Mutual information from signals

- We need to estimate $p(x,y)$
- Since this is a compound event we estimate with a 2D histogram
  - Cells, not bins
- Each cell contains the count of pairs of values
- Probability distributions $p(x)$ and $p(y)$ can be obtained by summing across the rows and columns so we don't need to compute them again

Linear discriminant analysis and support vector machines

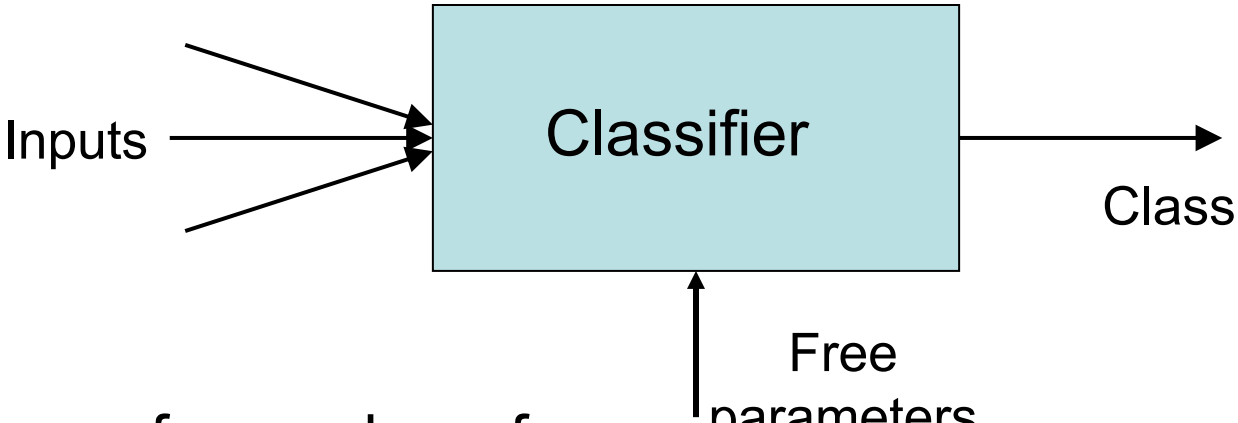
Classification

- Determining a disease or condition from a range of measurements or other diagnostic data is an application of classification.
- Classification is applied to a pattern of descriptors (measurements: the input space). It finds a class which best fits each pattern.
- Most Biomedical Engineering problems involve a small number of classes, often only two classes: diseased or normal, malignant or benign.

Classification

- Classification attempts to associate a pattern of input variables with either a specific class or another variable.
- If the output is a variable, the analysis is referred to as "regression"
- If the output is a discrete number identifying a specific class, the analysis is referred to as "classification."
- Classifiers establish a relationship between an input pattern and a discrete output: they can be viewed as mathematical functions and classifier development can thought of as *function approximation* (or *identification, estimation*, or *pattern recognition).*

- A classifier is an input-output device.



- For the pattern of a number of inputs, it determines the mostly likely condition associated with that pattern.
- The inputs can have any value, but typical output values are ±1 or 0, 1.
- Classification is done using two basic strategies: supervised and unsupervised learning.

Supervised and Unsupervised Learning

- In unsupervised learning, the classifier attempts to find patterns within the input data itself: the classifier has no a priori knowledge of the data patterns and, perhaps, not even the number of classes that exist.

- In supervised learning, the classifier is first “trained” using data for which the correct class is known. The training data set is known as the training set and includes the correct answer(s); i.e., the correct classification.

Both these strategies fall under the general heading of machine learning.

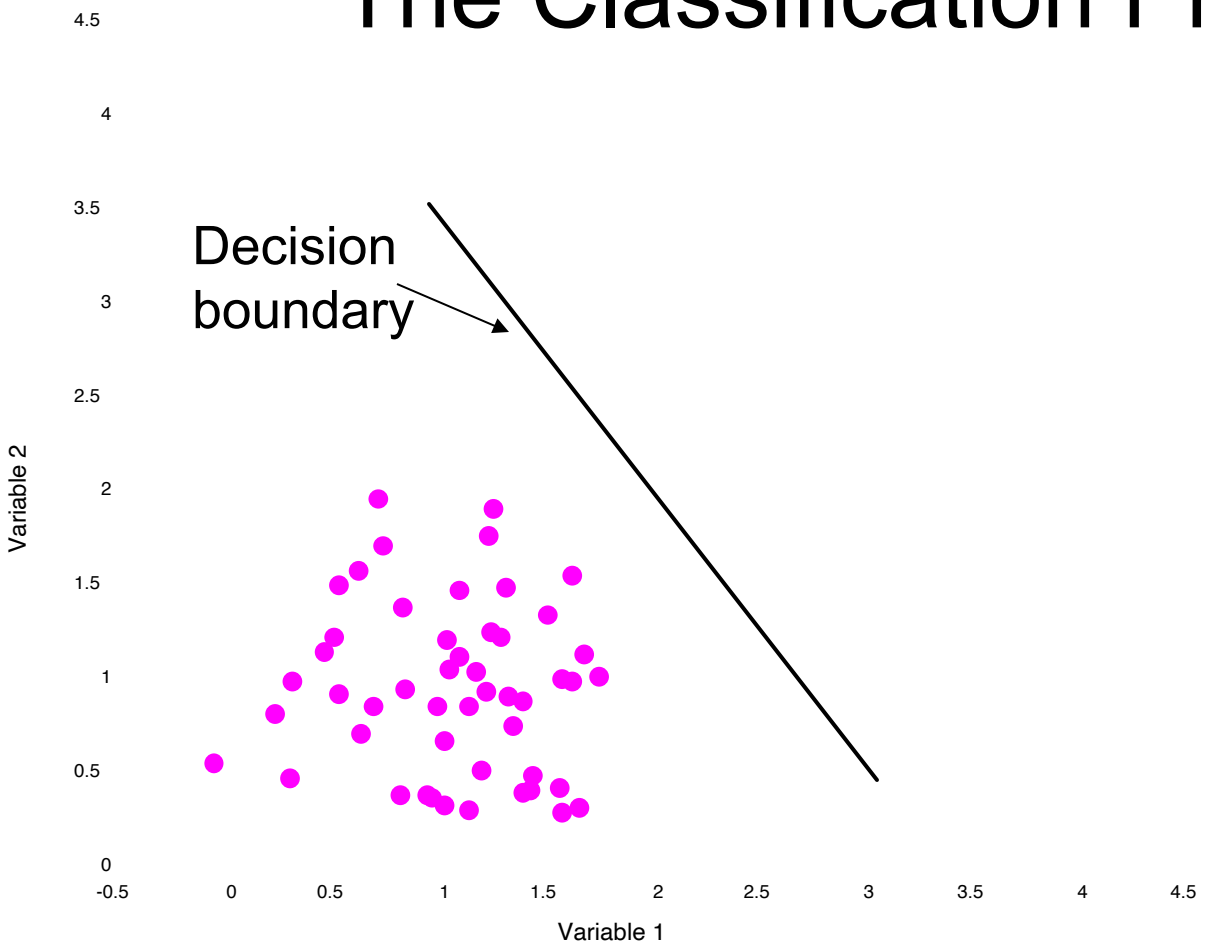
Supervised Learning

- Supervised learning is the most common approach in Biomedical Engineering applications.
- During training, classifier free parameters are adjusted adaptively to minimize classification errors.
- During training a training set is used where the answer, i.e., the correct classification, is known for each pattern.
- Classifier performance can be evaluated using a validation set for which the correct classification is also known but is not used to modify classifier parameters.
- The idea is that the classifier should perform with minimum error on data that it has never seen.

Supervised Learning (cont.)

- When training is complete, the classifier is applied to a test set where it performs its designed function to determine the most likely condition based on a given data pattern.
- It is only the classification error that occurs during the testing phase that really matters.
- A common failure of classifiers is that they perform well on the training set (after training, of course), but then do poorly on the test data, their real world application. Such classifiers are said to generalize poorly and/or be overtrained.

The Classification Problem



This figure shows a graphical representation of a typical classification problem.

There are 2 classes and 2 descriptive variables.

The 2 classes can be easily identified from the 2-variable scattergram.

A decision boundary separates measurement patterns associated with the two classes.

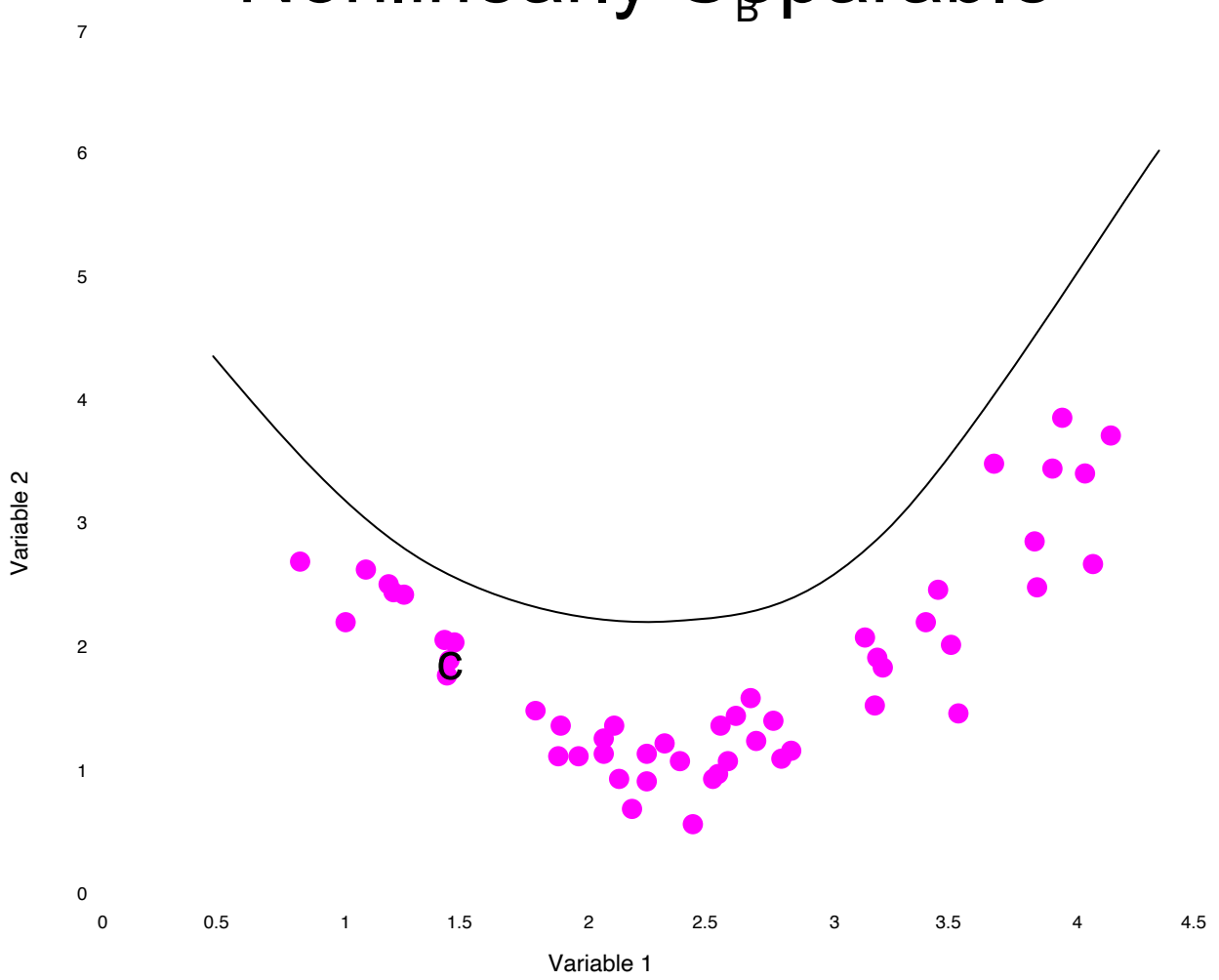
The Classification Problem

- If a straight line (or plane or hyperplane) can completely separate the classes, they are said to be linearly separable.

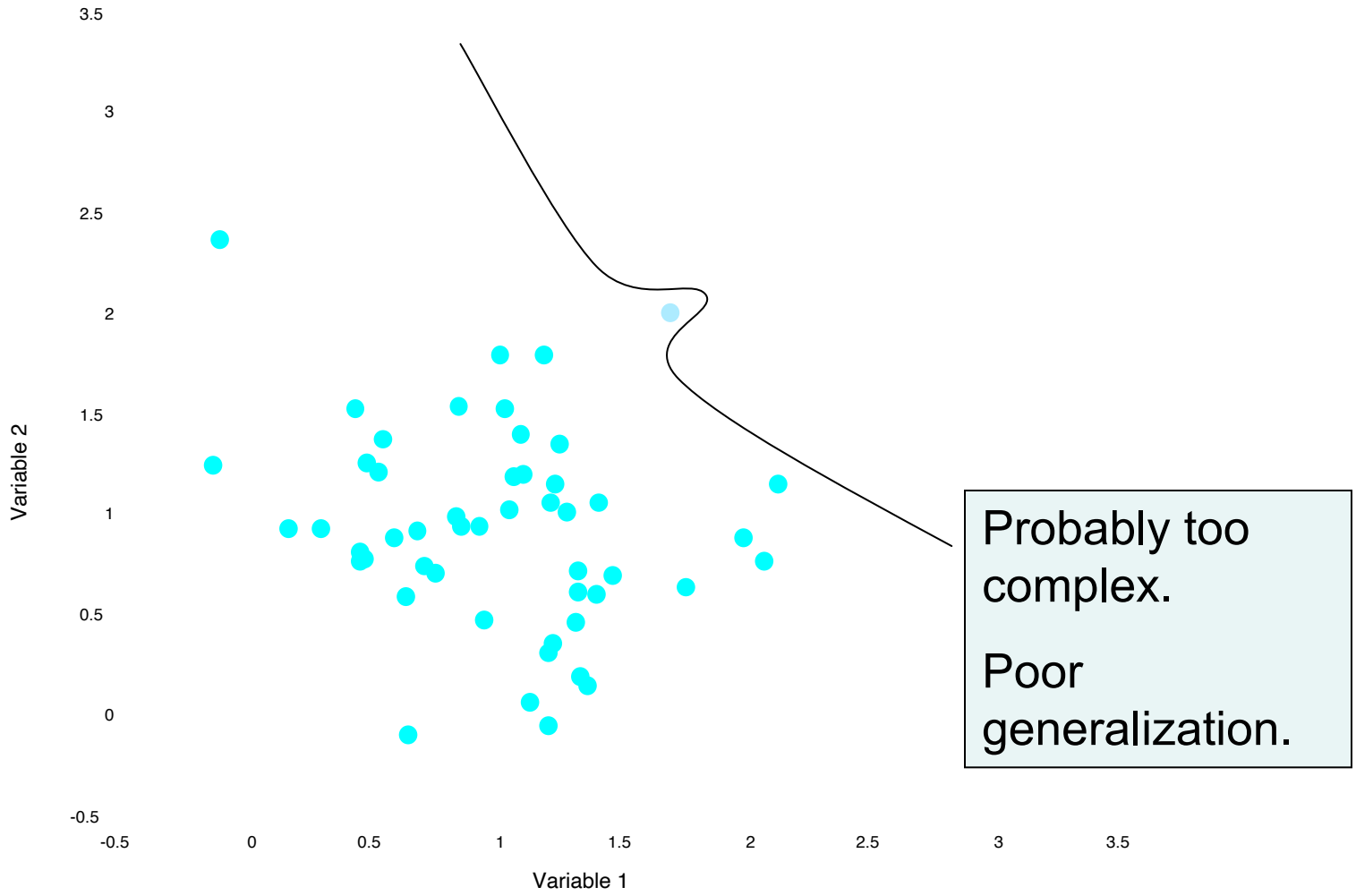
- Although classifiers may deal with a large number of input variables, it is easiest to visualize them when only two input variables are involved, so most of the examples use only two input variables.

- This is not really a limitation as the classification methods described here extend to multiple inputs in a straightforward manner.

Nonlinearly Separable

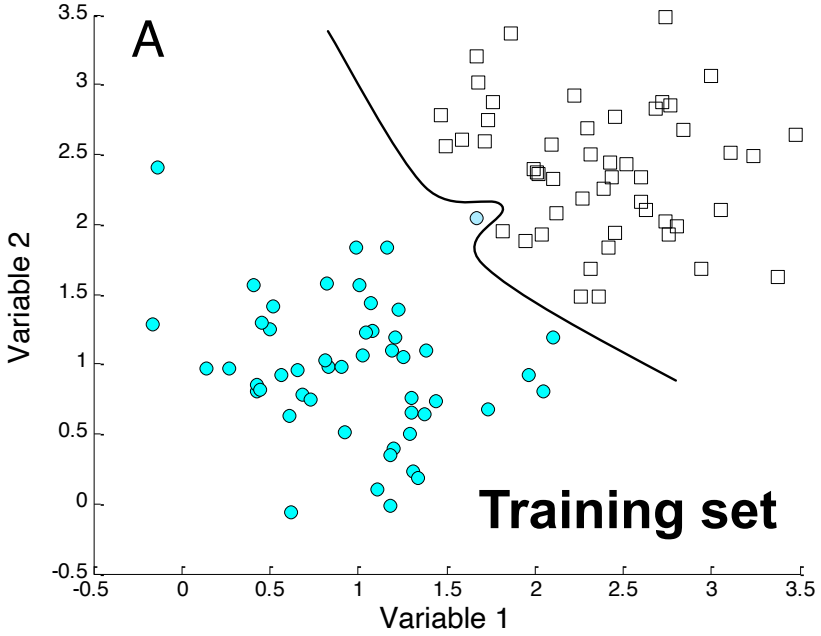


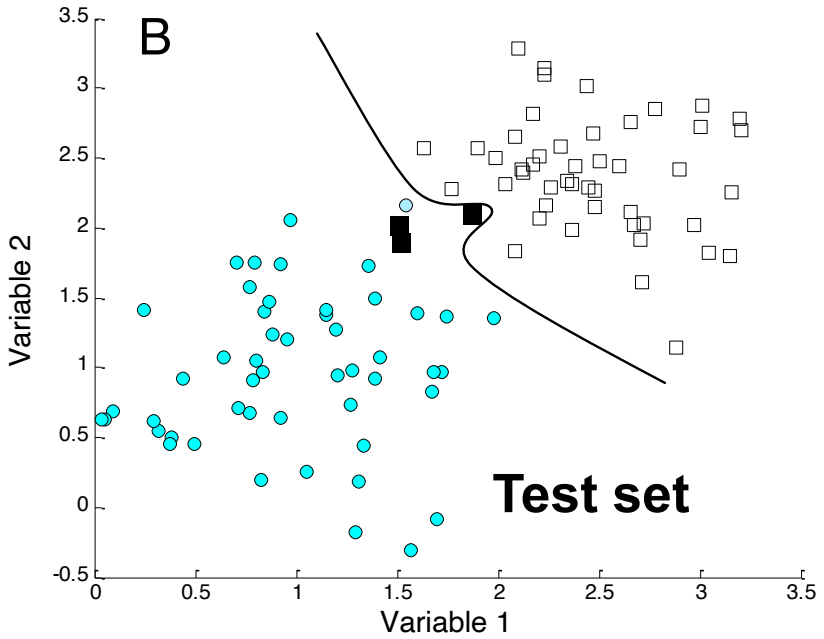
These two classes can still be correctly identified, but a curve is needed to separate them.



The two classes overlap somewhat, and a very complicated boundary is required to separate them.

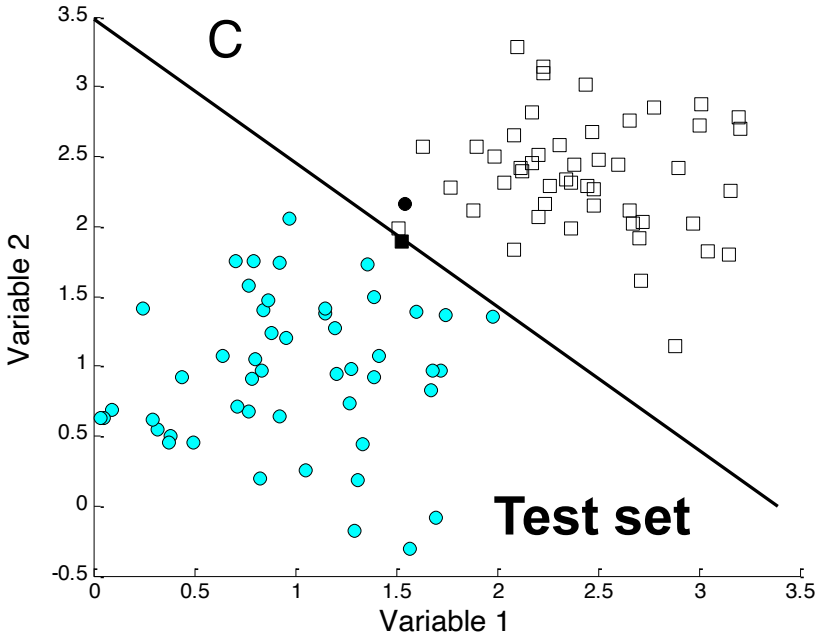
If this is a training set, the boundary shown is unlikely to generalize well.





Overtraining

A simple straight line produces only two errors.



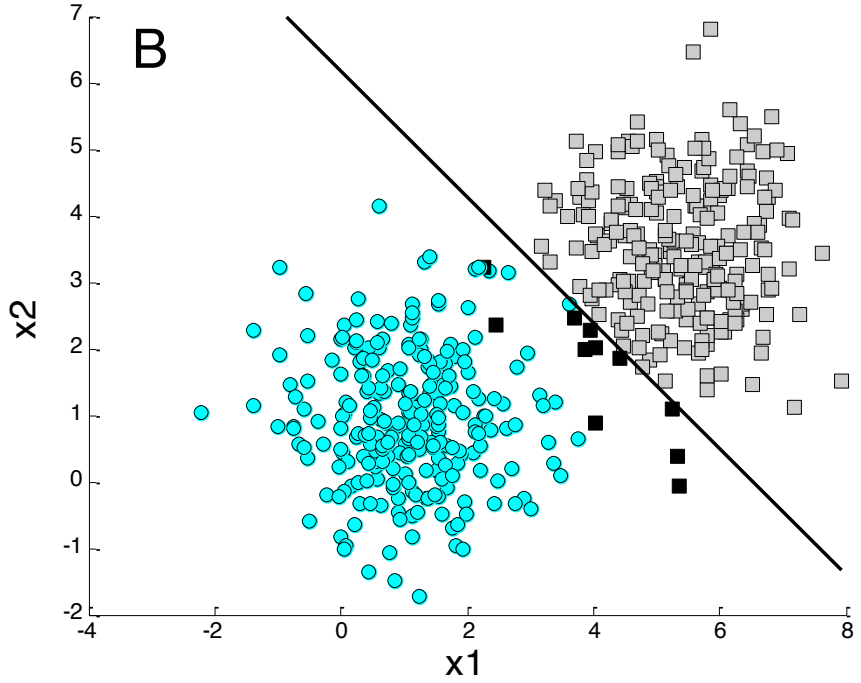
Three errors occur when this complicated boundary is applied to a test set.

Machine Capacity

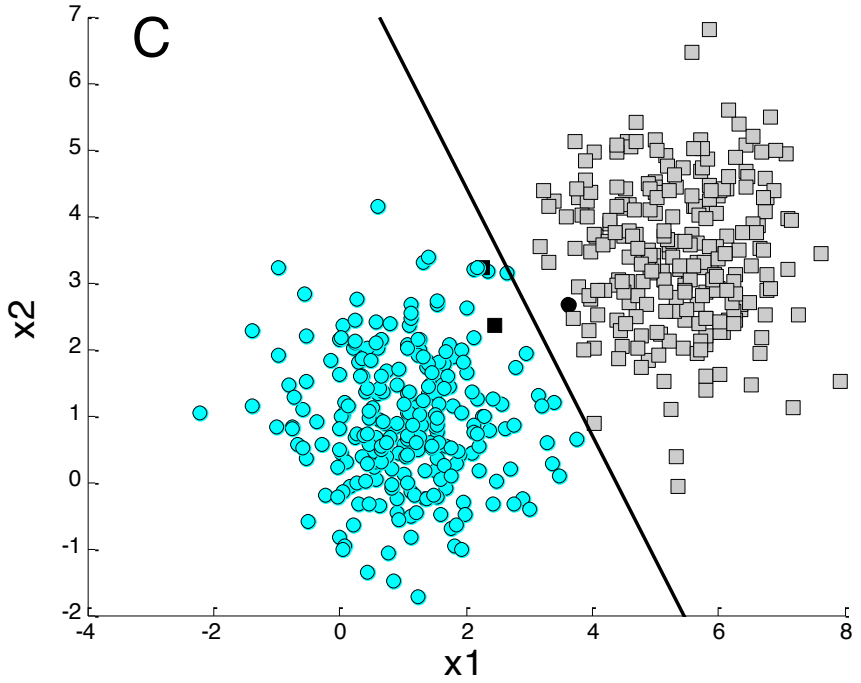
- The complexity of the boundary is determined by the classification algorithm.
- Since classification is a form of machine learning, classification algorithm complexity is termed machine capacity.
- Machine capacity is a major design factor of a classification algorithm and determines the complexity of the decision boundary.
- If capacity is too large for the data, the classifier will overtrain. It will perform well on the training set, but will not generalize well and perform poorly on the test set.
- A machine with too little capacity will show excessive errors in training and sub-par performance in classifying the test set.

Problems in Generalizing

12 classification errors occur if trained on a smaller (*N*=20) training set.



Only 2 classification errors occur if trained on a larger training set. (*N* = 200).



Evaluating Classifier Performance

Confusion Matrix

True Class	Predicted Class		
	Class 0	Class 1	Class 2
Class 0	% correct	% error	% error
Class 1	% error	% correct	% error
Class 2	% error	% error	% correct

Evaluating Performance:

Sensitivity and Specificity for 2-class classification

Sensitivity: Percent correct detections

$$Sensitivity \square 100 \frac{True\ Positives}{True\ Positives \square False\ Negatives} \square 100 \frac{True\ Positives}{Total\ Abnormal}$$

Specificity: Percent correct rejections

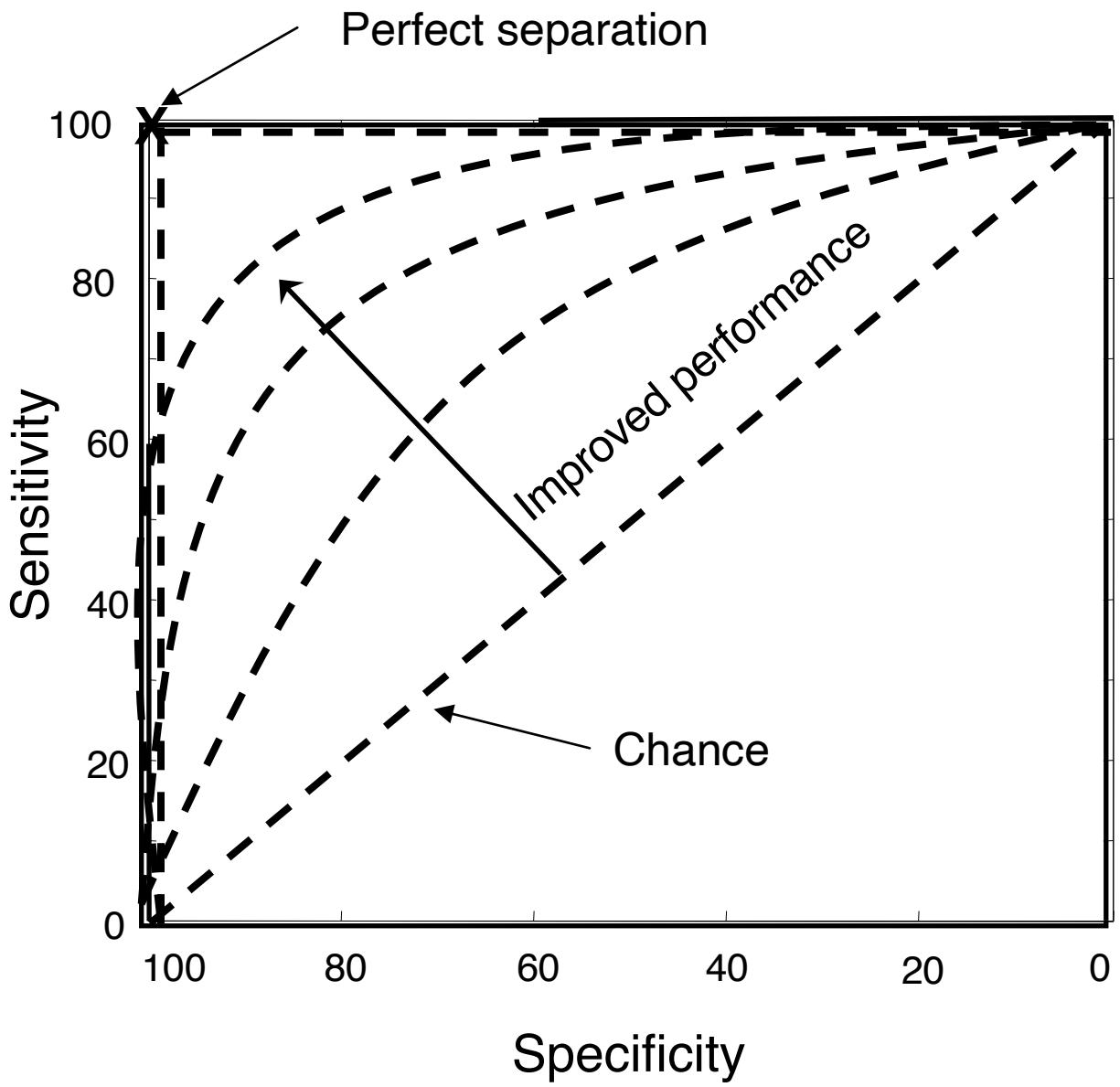
$$Specificity \square 100 \frac{True\ Negatives}{True\ Negatives \square FalsePositives} \square 100 \frac{True\ Negatives}{Total\ Normal}$$

The Receiver Operation Curve (ROC)

Plot of Sensitivity versus Specificity

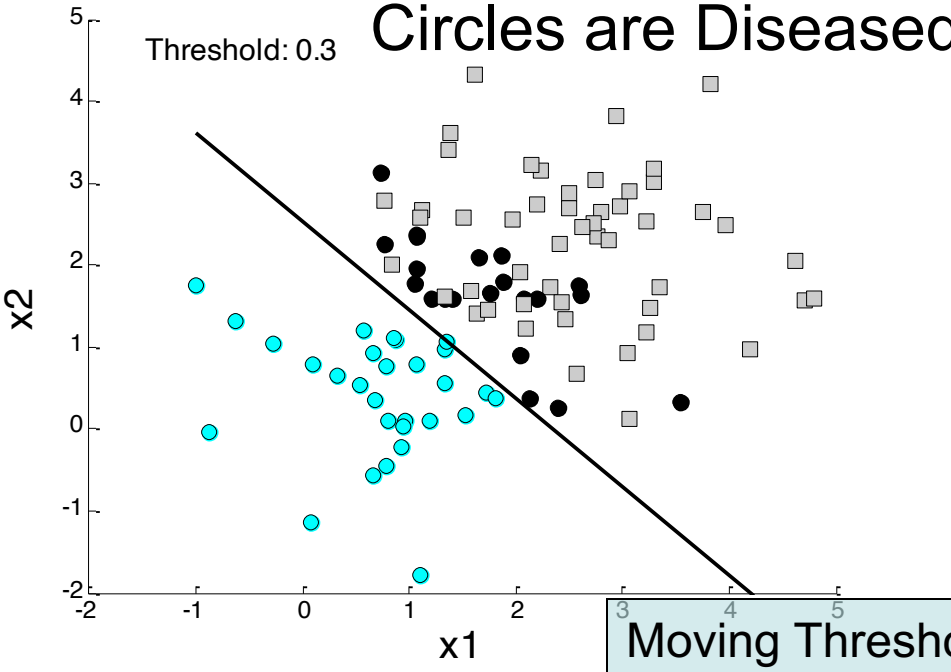
- Sometimes it is possible to vary the decision boundary to increase or decrease the detection of abnormals.
- Increasing the detection of true positives will usually increase the number of false positives and lower the number of true negatives.
- There is a trade-off between sensitivity and specificity.
- A curve showing this tradeoff between sensitivity and specificity is called the *ROC* (receiver operator characteristics) curve.

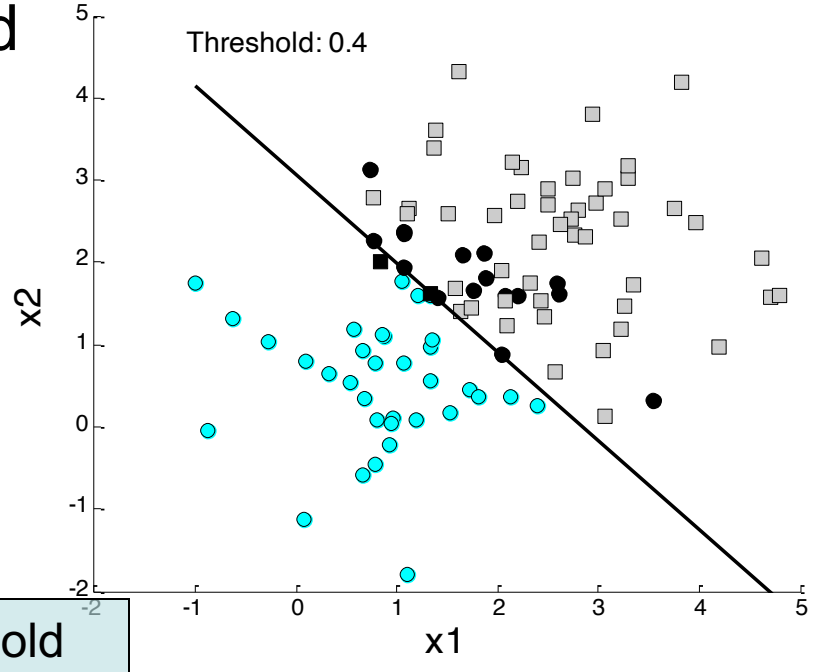
Example ROC Curve



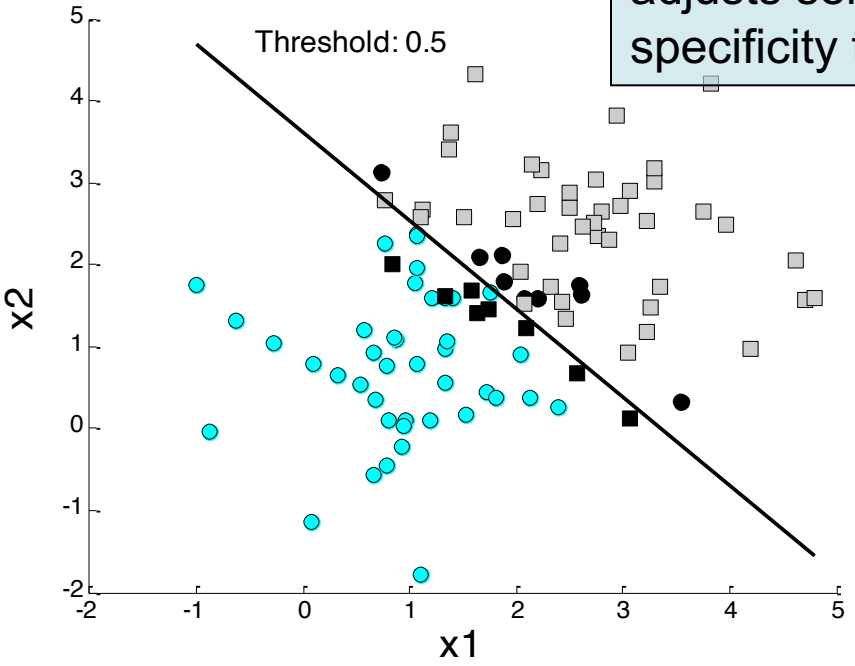
Note that Specificity is plotted in reverse

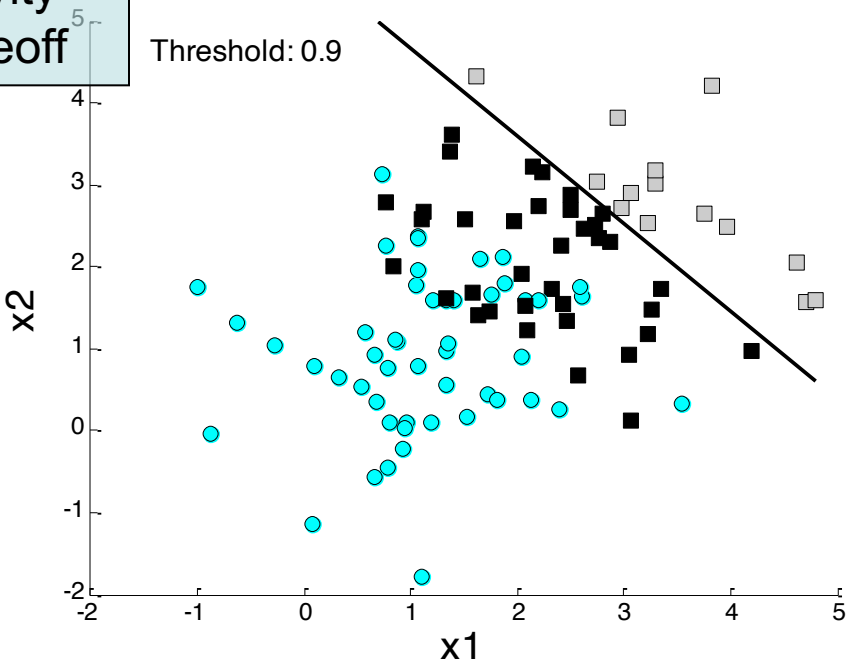
Circles are Diseased

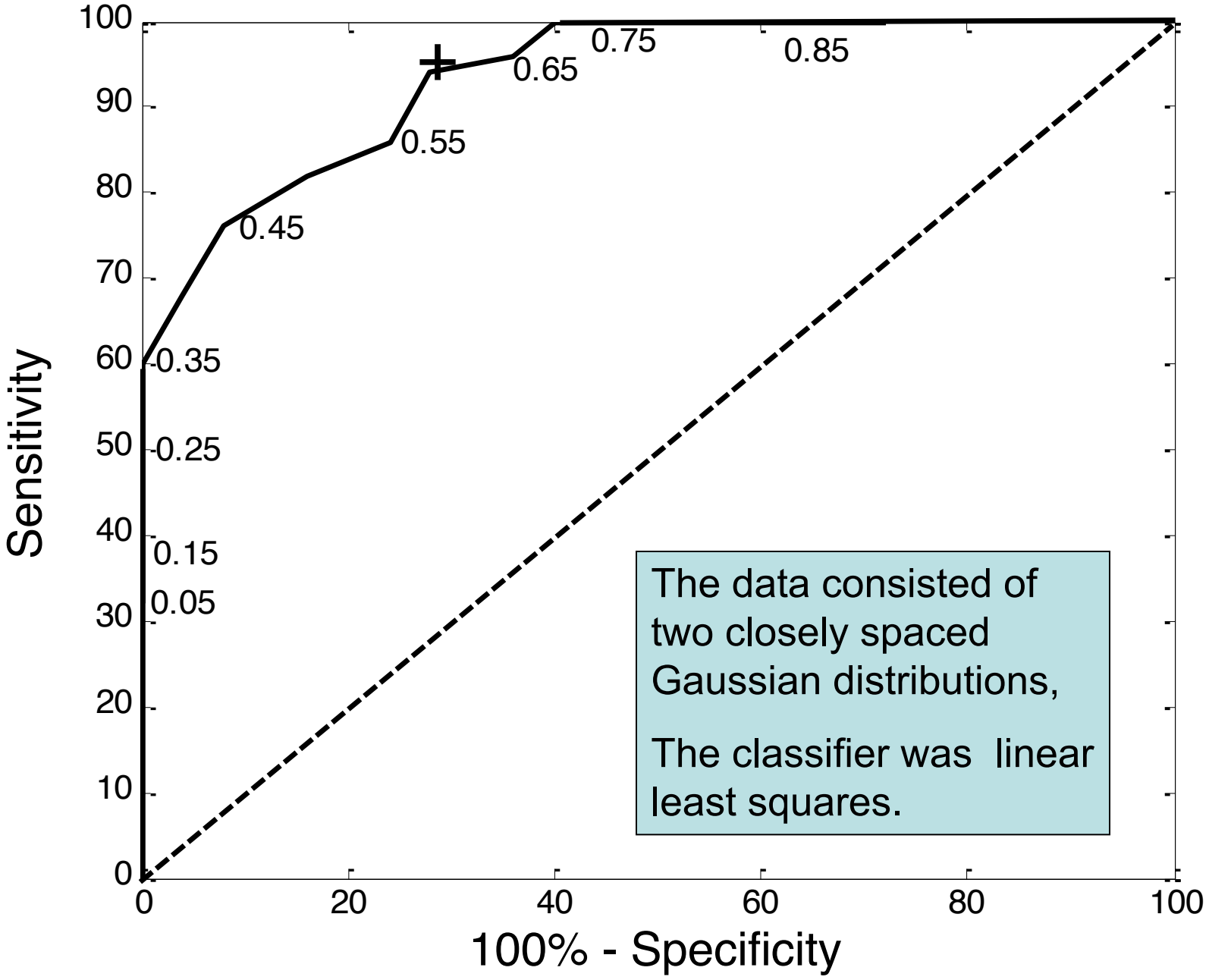




Moving Threshold adjusts sensitivity specificity tradeoff





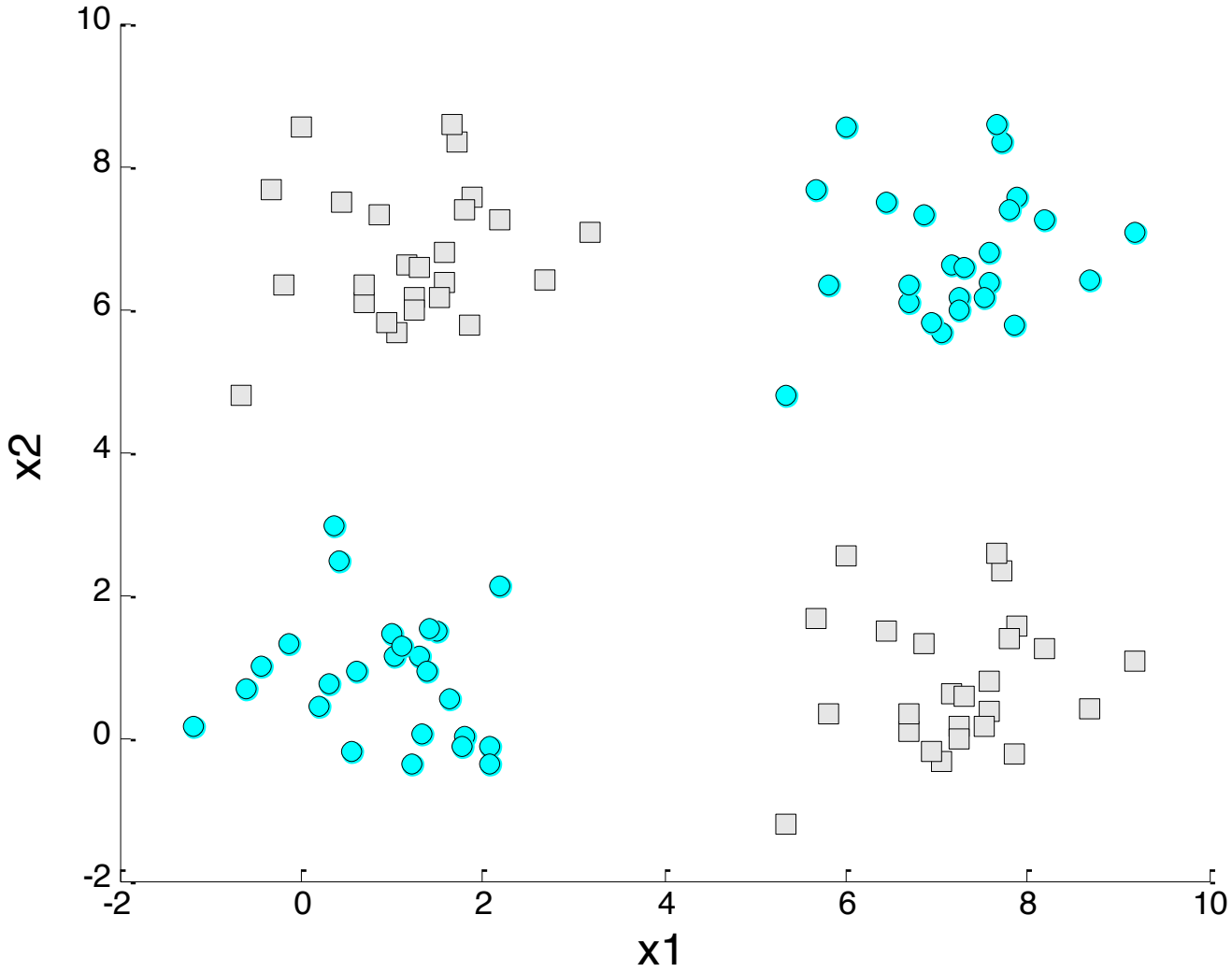

Sensitivity
100% - Specificity
0
10
20
30
40
50
60
70
80
90
100
0
20
40
60
80
100
0.35
0.25
0.15
0.05
0.45
0.55
0.65
0.75
0.85
The data consisted of two closely spaced Gaussian distributions,
The classifier was linear least squares.

Higher Dimensions – Kernel machines

- Many classification problems involve data that are separable, but not by a single straight line (see data in next slide).

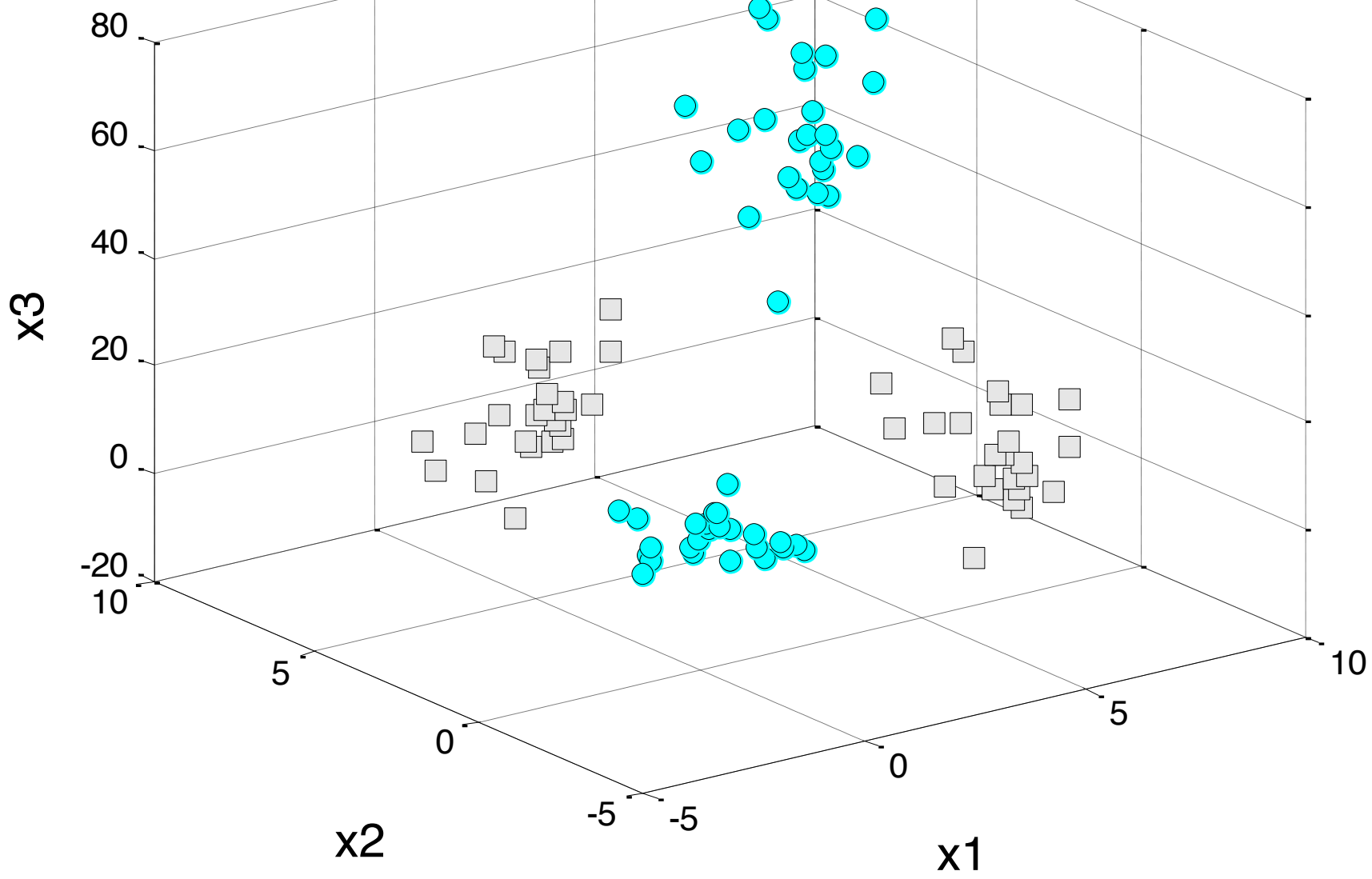
- In more complex data sets, it is still possible to separate the classes using a linear boundary if the data are transformed into a higher-dimensional space.

- In fact, if the number of dimensions is high enough, you can always find a linear boundary (hyperplane) that will separate the data without error (Cover's Theorem).

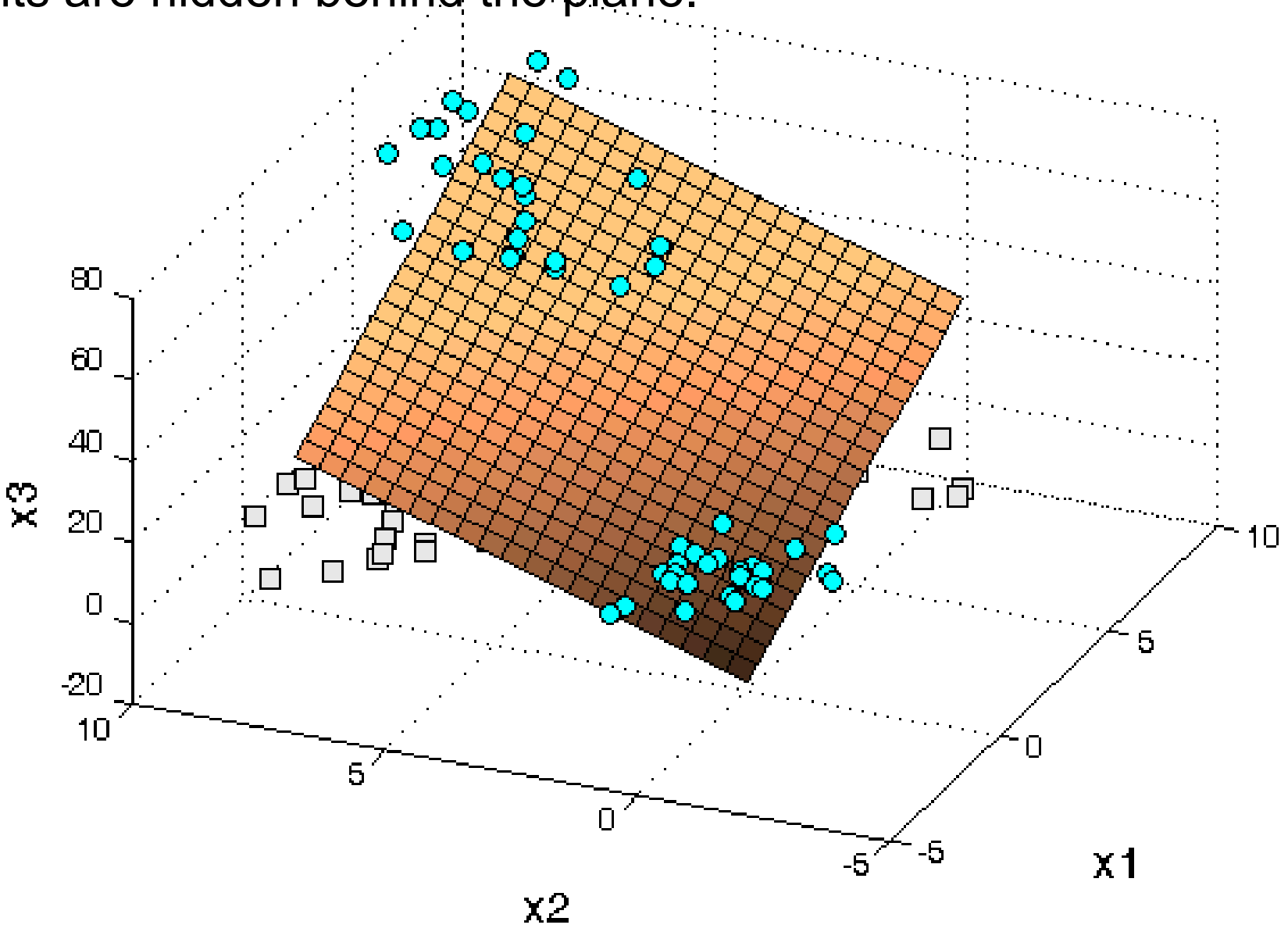


A two-class data set that is clearly separable, but not by a single linear boundary.

The data in the previous slide transformed into three-dimensional feature space by adding a cross-product term.

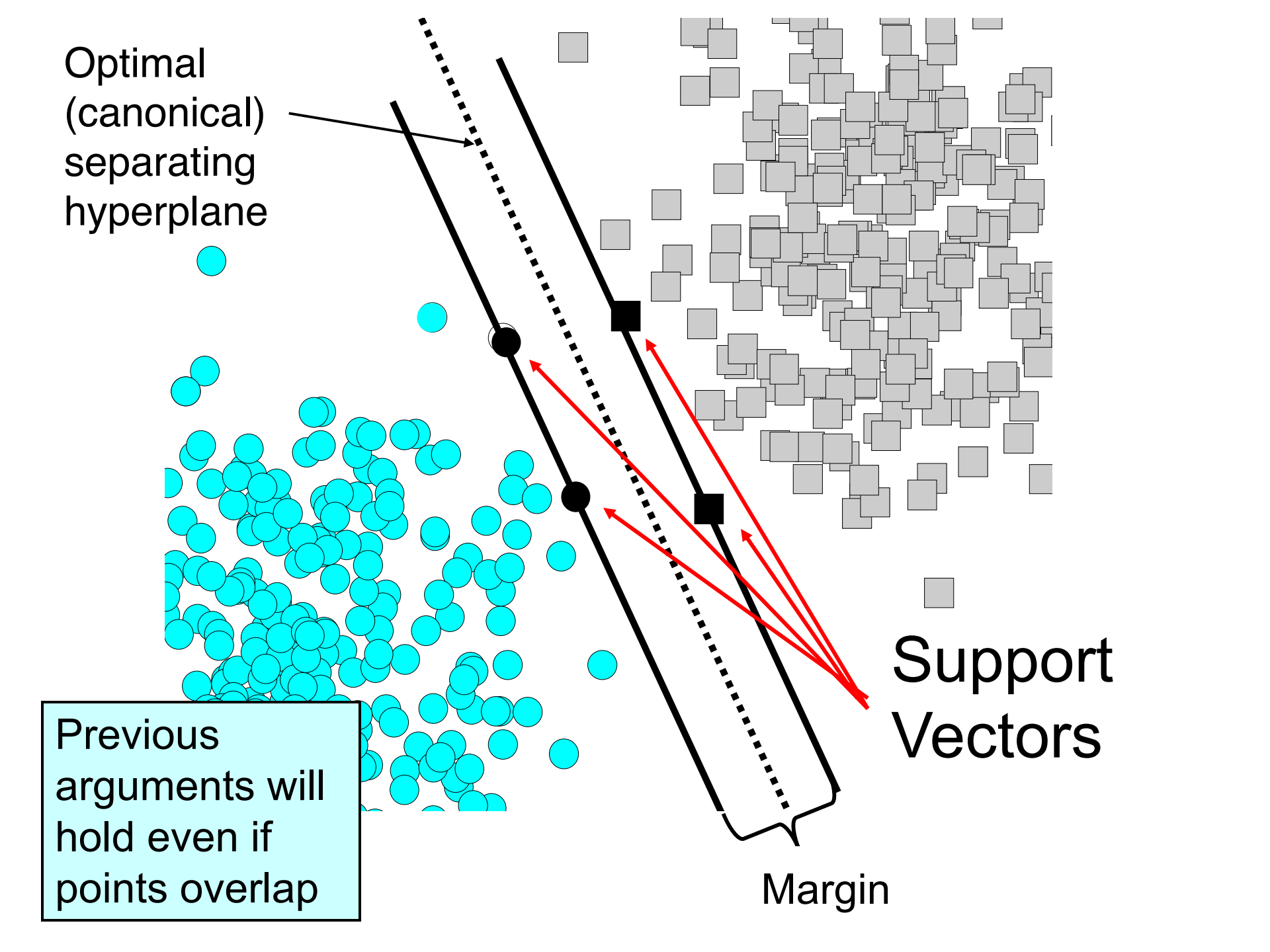


Example Results A plane gives perfect separation of the transformed data set. Some of the square data points are hidden behind the plane.



Support Vector Machines

- The problem with the least square method and similar methods is that they put too much emphasis on data points that are not critical.
- An approach that maximizes the distance between the critical data points should generalize better.
- The points closest to the boundary are called the support vectors.
- A support vector classifier determines the boundary that maximizes the distance between the critical support vectors.
- Since support vector classifiers maximize the *margin*, they are also known as maximum margin classifiers.
- Support vector classifiers have become quite popular because they produce excellent results in practical problems.


Optimal
(canonical)
separating
hyperplane
Support
Vectors
Margin
Previous
arguments will
hold even if
points overlap

Cluster Analysis

- Cluster analysis is particularly popular for unsupervised classification, especially when the number of classes is unknown. Here only two supervised versions of cluster analysis will be described:

  k-nearest neighbor classifiers and

  k-means clustering.

k-nearest neighbor classifier

- The *k*-nearest neighbor classifier operates directly off the testing set and does not need prior training.
- The classifier simply takes each test set point and determines the **distance** to the *k* nearest training points, where *k* is a constant.
- It then takes the class values of these nearest points and assigns the test point to the majority class.
- For example, if *k* = 5 and the training set points that were closest to the test points had class values of: 0, 1, 1, 0, 0; then the average would be 2/5, and class 0 would be assigned to the test point.
- This approach can be used for any number of classes and any number of input variables.

Distance Measurement

Distances between points can be measured using a variety of metrics but the most common and straightforward is the Euclidean distance. The Euclidean distance between two points is:

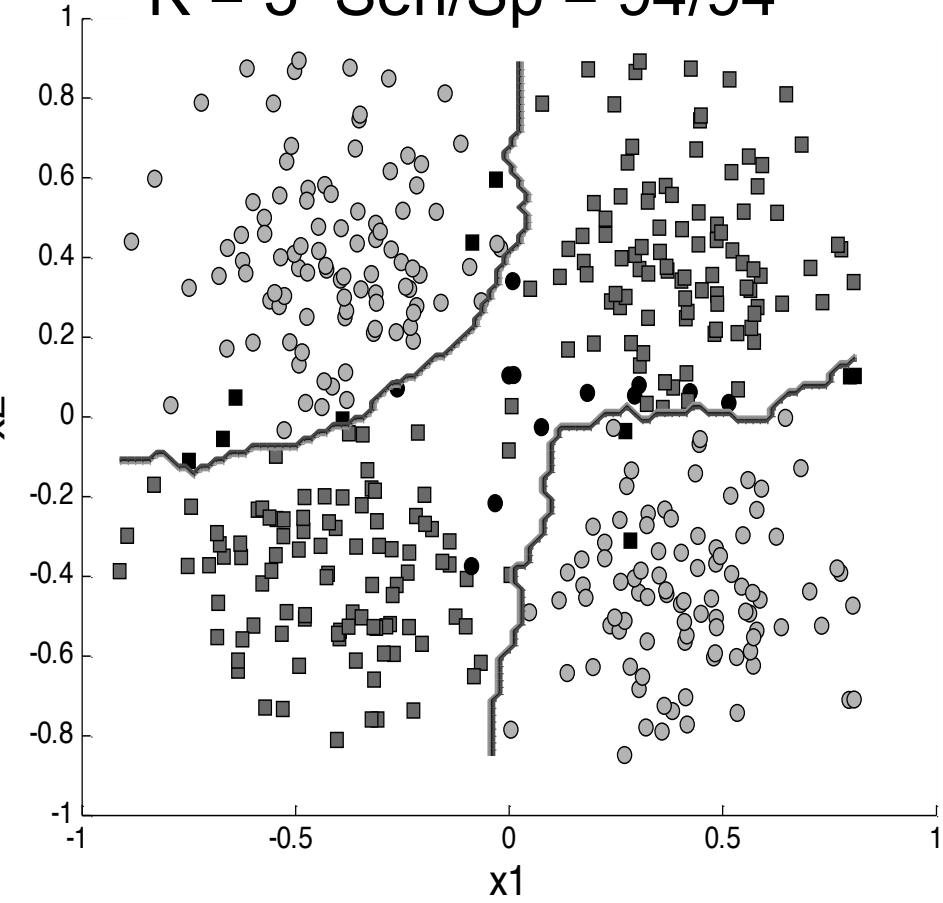
$$dist \square \sqrt{x_2 - x_2} \square \left\| x_2 - x_1 \right\|$$

where x_1 and x_2 are vectors and $||x_2 – x_1||$ is the norm of the vector that results after subtraction.

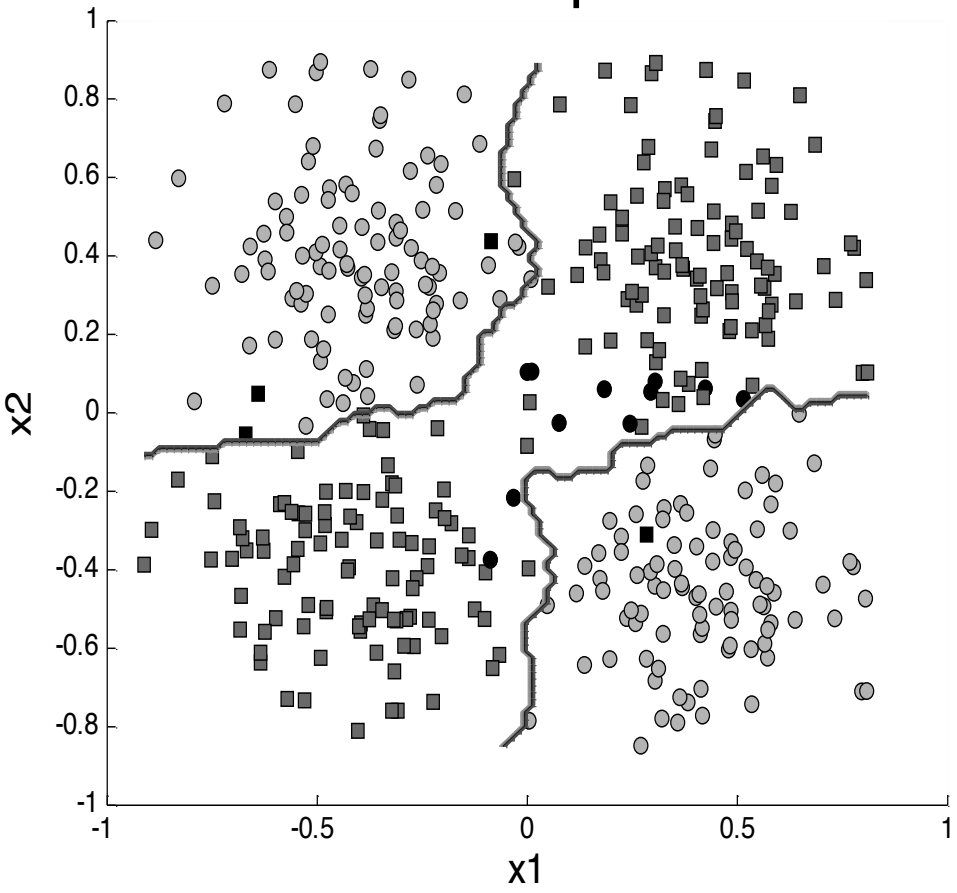
It is also common to normalize each variable so it has a variance between ± 1.

Boundaries obtained using the k-nearest neighbor

K = 5 Sen/Sp = 94/94



K = 15 Sen/Sp = 95/98



Confusion matrix is given in the next slide.

The *k*-means Clustering Classifier

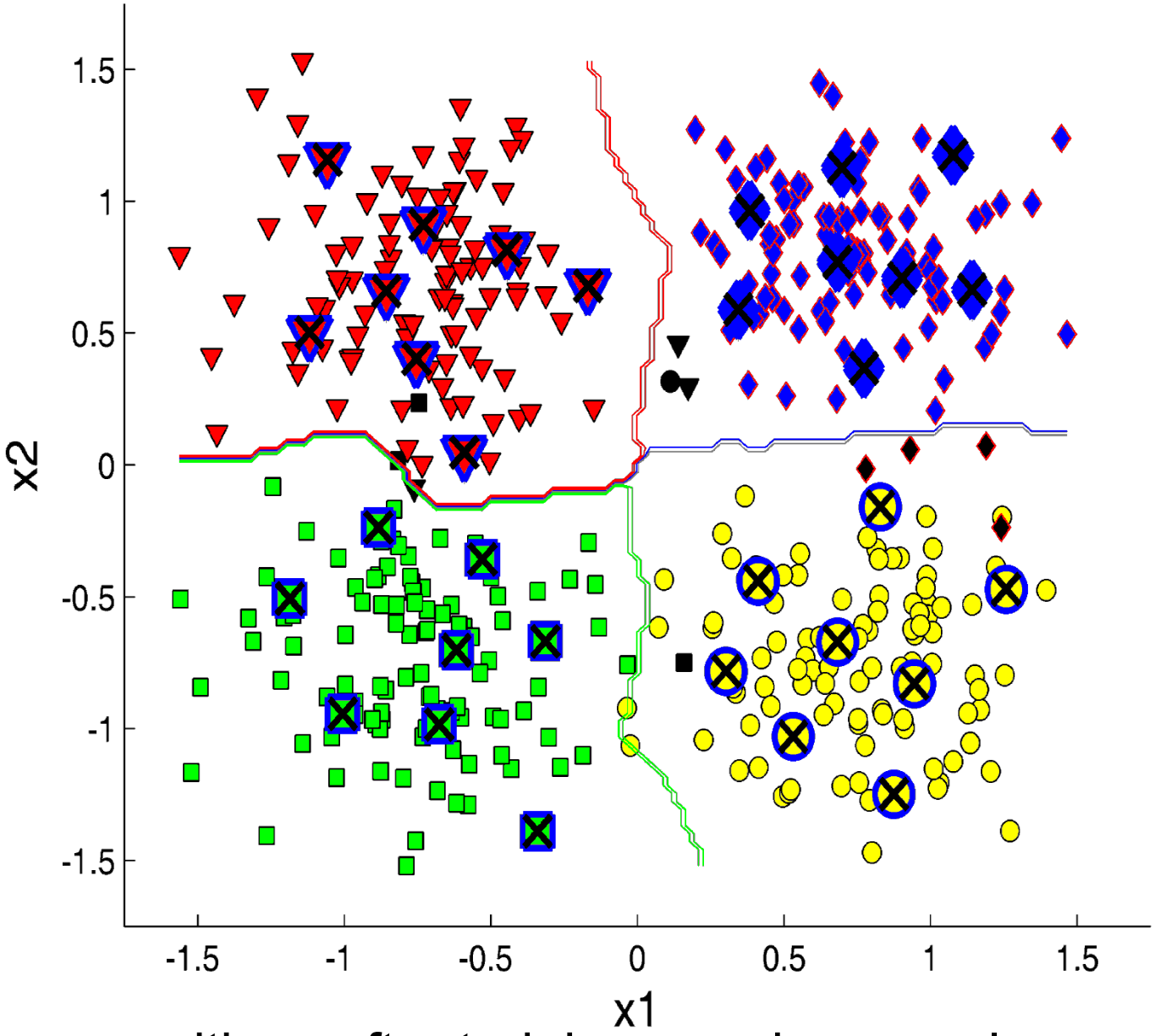
- The *k*-means clustering classifier is a related method that represents the training data with a number of data centers known as *prototypes*.
- The training data are reduced to a set of *k* representatives.
- Instead of comparing the test set points to the training set points, the test set point are compare to the prototypes.
- Once these prototype centers are established, the test data are assigned to the class of the closest prototype.

The *k*-means Clustering Classifier (cont)

- In this approach the *k* stands for the number of prototype centers.
- The position of the prototypes determines the boundary, and the number of prototypes chosen to represent each class determines the complexity of the boundary.
- The larger the value of *k,* the more complicated the boundary. The value of *k* is directly related to machine capacity.
- The number of prototypes is selected by the user and the prototype centers are positioned during a training period.

Classification of a 4-class, two-variable test set using *k*-means clustering.

Eight prototypes were used to represent each class (*k* = 8)



Prototype positions after training are shown as large symbols with an ‘x.’