Data ^{X}

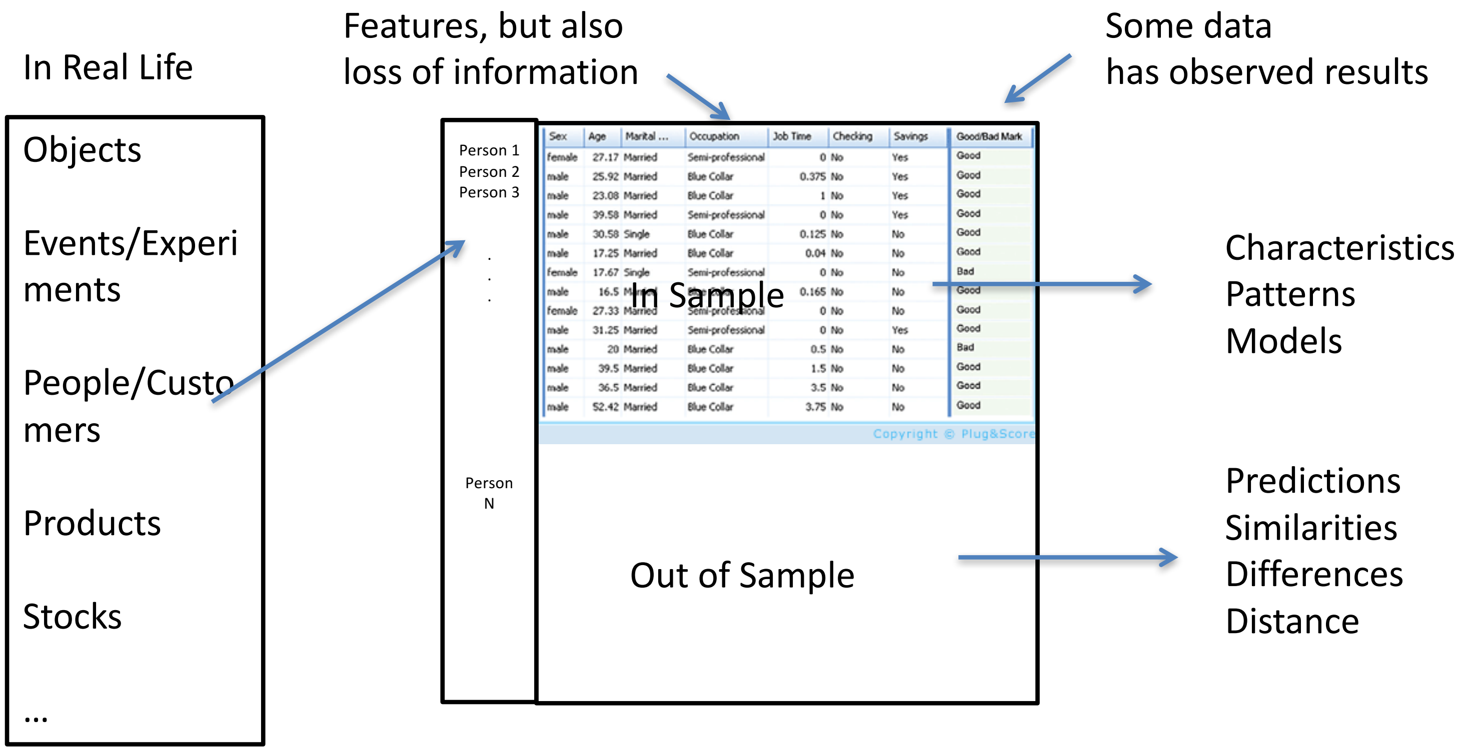
About Me:

Data as a Signal

Data X: A Course on Data, Signals, and Systems

Ikhlaq Sidhu
Chief Scientist & Founding Director,
Sutardja Center for Entrepreneurship & Technology
IEOR Emerging Area Professor Award, UC Berkeley

A High Level Framework

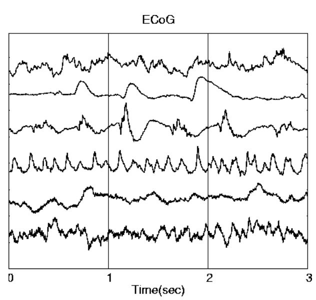


Sex	Age	Marital ...	Occupation	Job Time	Checking	Savings	Good/Bad Mark
female	27.17	Married	Semi-professional	0	No	Yes	Good
male	25.92	Married	Blue Collar	0.375	No	Yes	Good
male	23.08	Married	Blue Collar	1	No	Yes	Good
male	39.58	Married	Semi-professional	0	No	Yes	Good
male	30.58	Single	Blue Collar	0.125	No	No	Good
male	17.25	Married	Blue Collar	0.04	No	No	Good
female	17.67	Single	Semi-professional	0	No	No	Bad
male	16.5	[illegible]	[illegible]	0.165	No	No	Good
female	27.33	Married	Semi-professional	0	No	No	Good
male	31.25	Married	Semi-professional	0	No	Yes	Good
male	20	Married	Blue Collar	0.5	No	No	Bad
male	39.5	Married	Blue Collar	1.5	No	No	Good
male	36.5	Married	Blue Collar	3.5	No	No	Good
male	52.42	Married	Blue Collar	3.75	No	No	Good

Converting From Time Sequence Data to Features

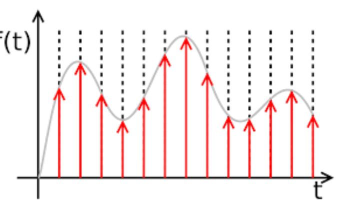
Many Types of data are signals in time

- Stock market
- Temperature
- Instrument readings



Continuous signals
$x(t)$

Sometimes we sample them, record at intervals of T



Sampled signals (data)
$x(nT)$

We get a list in a table, array, or vector

Rec	Observed
1	60.323
2	61.122
3	60.171
4	61.187
5	63.221
6	63.639
7	64.989
8	63.761
9	66.019
10	67.857
11	68.169
12	66.513
13	68.655
14	69.564
15	69.331
16	70.551

Discrete data
x_n = x1, x2, x3, ...

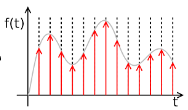
(might lose time reference)

What we want (for now): features and characteristics

For example:

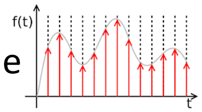
- Means
- Variances
- Patten matches
- Changes
- accumulation
- Frequency

Data Sequence in Tables Example

Apple 

.

.

Google 

Discrete data for each source
A, B, C..
x_n = x1, x2, x3, ...

data

Preprocess:
Filter/
Convolution
Correlation
Mean, var, stats

Windows of time

features

	Price	Price[n-20]	20 day MA	1 year average	Expected Price?
APPL	appl[n]	appl[n-20]	mva(appl, 20)	mva(appl, 200)	
FB	fb[n]	fb[n-20]	mva(fb,20)	mva(fb,200)	
GOOG	goog[n]	goog[n-20]	mva(goog, 20)	mva(goog, 200)	

Data Input and Temp Storage

Preprocess
(and lose
some information)

ML for Decisions / Predictions

DataX

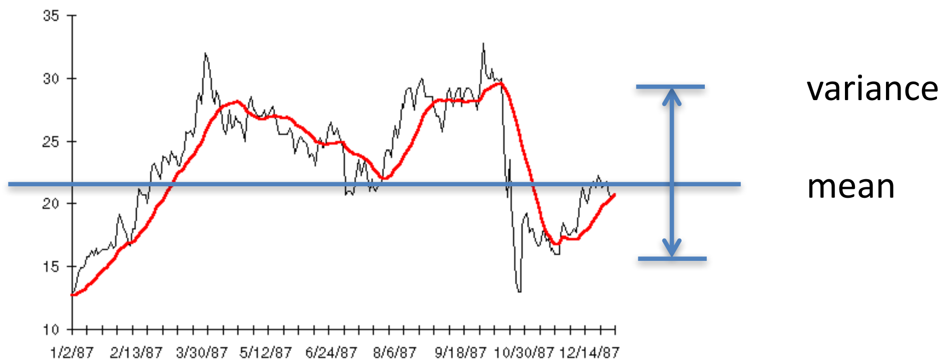
Signal Statistics and Windows

Getting statistics if you have a sequence: Mean, Variance

Discrete data

x_n = x1, x2, x3, …

Rec	Observed
1	60.323
2	61.122
3	60.171
4	61.187
5	63.221
6	63.639
7	64.989
8	63.761
9	66.019
10	67.857
11	68.169
12	66.513
13	68.655
14	69.564
15	69.331
16	70.551



Sample Mean	Population Mean
$\bar{x} = \frac{\Sigma x}{n}$	$\mu = \frac{\Sigma x}{N}$

where Σx is sum of all data values

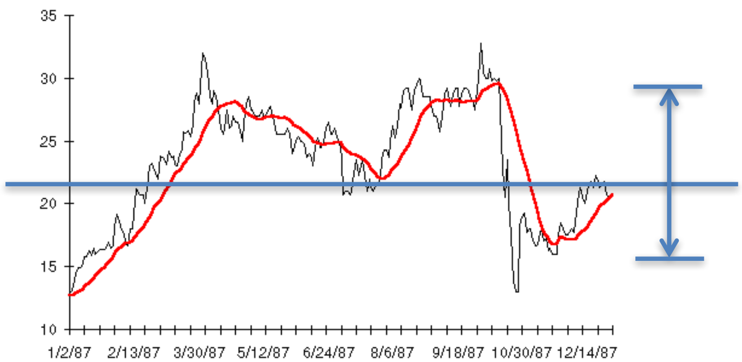
N is number of data items in population

n is number of data items in sample

Getting statistics if you have a sequence: Mean, Variance

Discrete data
x_n = x1, x2, x3, …

Rec	Observed
1	60.323
2	61.122
3	60.171
4	61.187
5	63.221
6	63.639
7	64.989
8	63.761
9	66.019
10	67.857
11	68.169
12	66.513
13	68.655
14	69.564
15	69.331
16	70.551



variance

mean

```
a = [15, 18, 2, 36, 12, 78, 5, 6, 9]
import numpy as np

m = np.mean(a)
v = np.var(a)
```

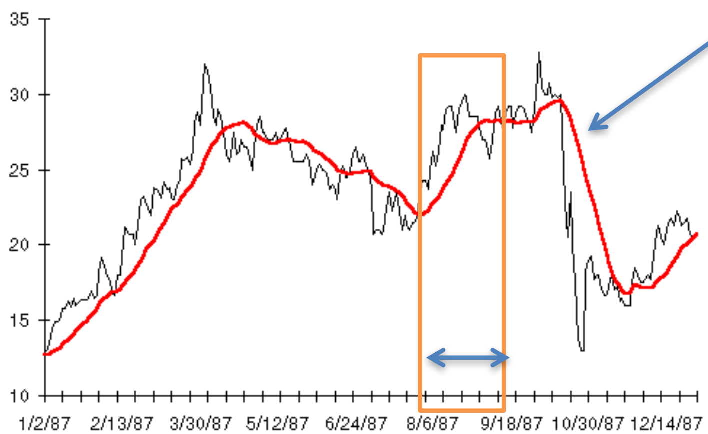
$$\sigma^2 = \frac{\sum (X-\mu)^2}{N}$$

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

And of course, standard deviation is s or sigma

Statistics with Windows
Moving Average of Sequence

- Discrete data x_n = x1, x2, x3, …
- Moving Average



$$MA(n) = \frac{1}{W} \sum_{i=0.}^{W-1} x_{n-i}$$

(Simple) Moving Average

W =size of window
MA(n) is red line, at the right edge of the window

Note:
- It's a function , not a number.
- It's a function of index n (and Window size W)

- Its smoother than the original (low pass filter)

```
a = [15, 18, 2, 36, 12, 78, 5, 6, 9]
import numpy as np

m = np.mean(a[n-W:n])

# use slices for windows
# note m is actually dependent on n
```

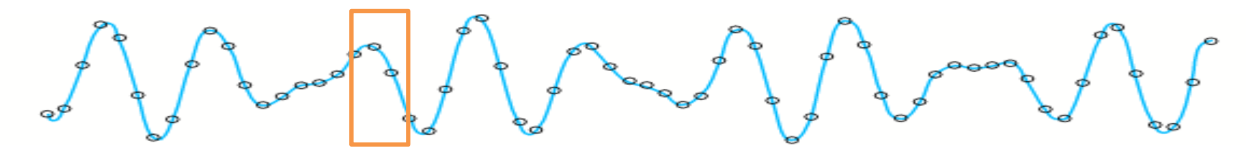
- We could have applied other functions over this window: deviation, mode, max/min, …

LTI Approaches

Introducing Convolution as another type of Window Function

For example, Moving Average calculated by Convolution of a Window of height 1/W

x(t)



x(t) → LTI: h(n)=1,1,1,1 → y[n] is similar to moving average with window of 4

x(n) = array with sequence of numbers = [10, 3, 6, 12, …. … 43, 12, 1, 4]

h(n) = impulse response function from linear systems. = [1, 1, 1, 1]

$MA_W(n)$ = y(n) = x(t) * h(t) (convolution)

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

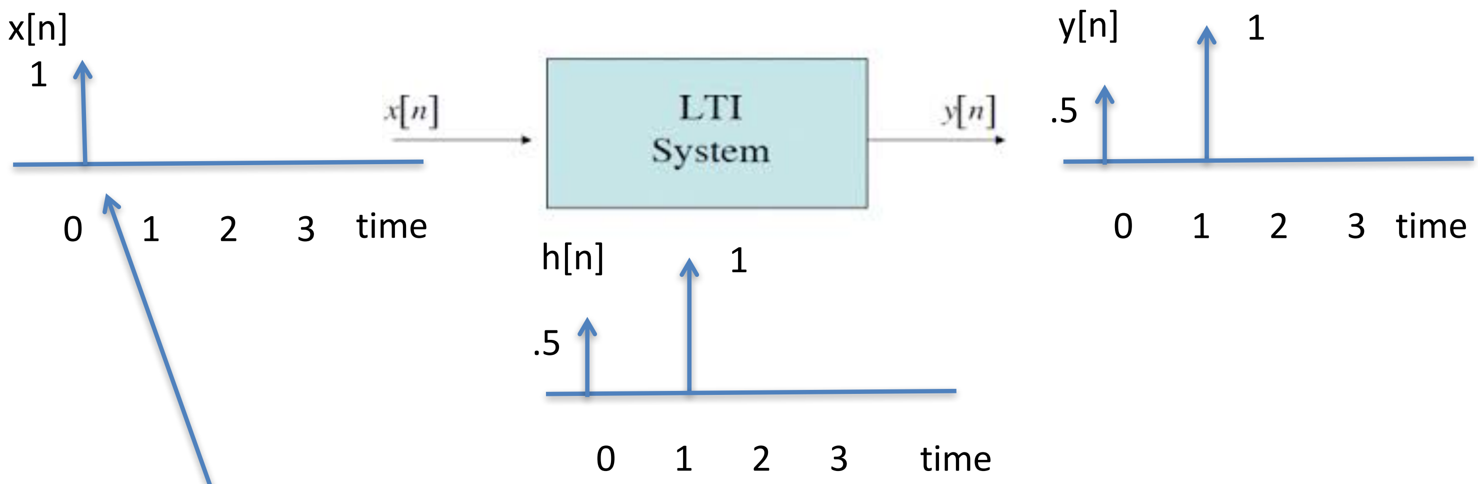
Results in a y for every index n

Continuous $y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau = x(t) * h(t)$

Circular $\boxed{(x \circledast y)_n \triangleq \sum_{m=0}^{N-1} x(m)y(n-m)}$

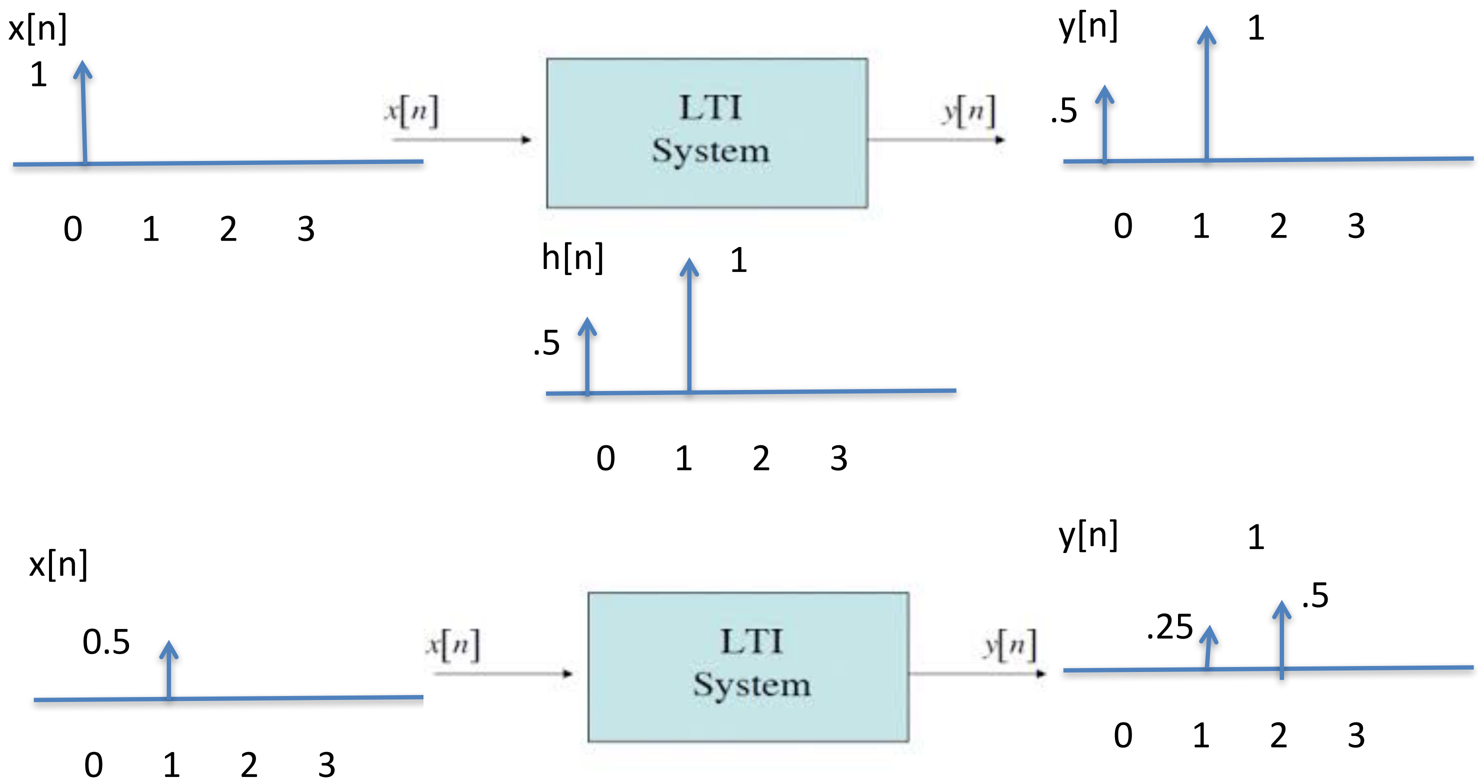
DataX

Linear Time Invariant System



Impulse (delta function: area = 1, width -> 0, height -> ∞)
In discrete systems, its just a value of "1"

Linear Time Invariant System



Data X

Convolution is actually really simple, but powerful

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

- x[[n] = 1 2 3 2 4 0 0 0 ...
- h[n] = 1 1 1
- y[n] =

y[n] =	1	1	1					h[n-0] x[0]
		2	2	2				h[n-1] x[1]
			3	3	3			h[n-2] x[2]
				2	2	2		h[n-3] x[3]
					4	4	4	h[n-4] x[4]
y[n] =	1	3	6	7	9	6	4 ...	

Note: if h[n] = 1/3, 1/3, 1/3, then y[n] = 1/3, 1, 2, 7/3, 3, 2, 4/3

Note, MA with Window = 3
(if divided by 3)

Another Convolution, different impulse response

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

- x[[n] = 1 2 3 2 4 0 0 0 …
- h[n] = 3 2 1
- y[n] =

y[n] =	3	2	1					← h[n-0] x[0]
		6	4	1				← h[n-1] x[1]
			9	6	1			← h[n-2] x[2]
				6	4	1		← h[n-3] x[3]
					12	8	4	← h[n-1] x[4]
y[n] =	3	8	14	13	17	9	4 …	

This is called filtering the data with an FIR filter with impulse response h[n] = 3 2 1

Code Example of Convolution

```
>>> np.convolve([1, 2, 3], [0, 1, 0.5])

array([ 0. ,  1. ,  2.5,  4. ,  1.5])
```

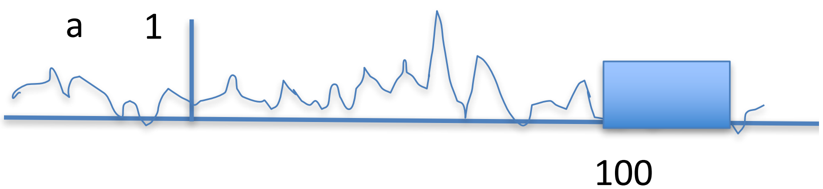
numpy.convolve(a, v, mode='full')[source]

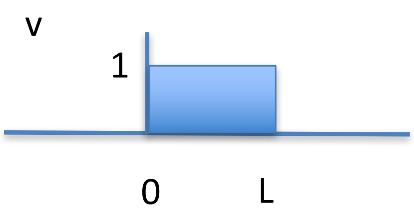
- Returns the discrete, linear convolution of two one-dimensional sequences.
- If *v* is longer than *a*, the arrays are swapped before computation.

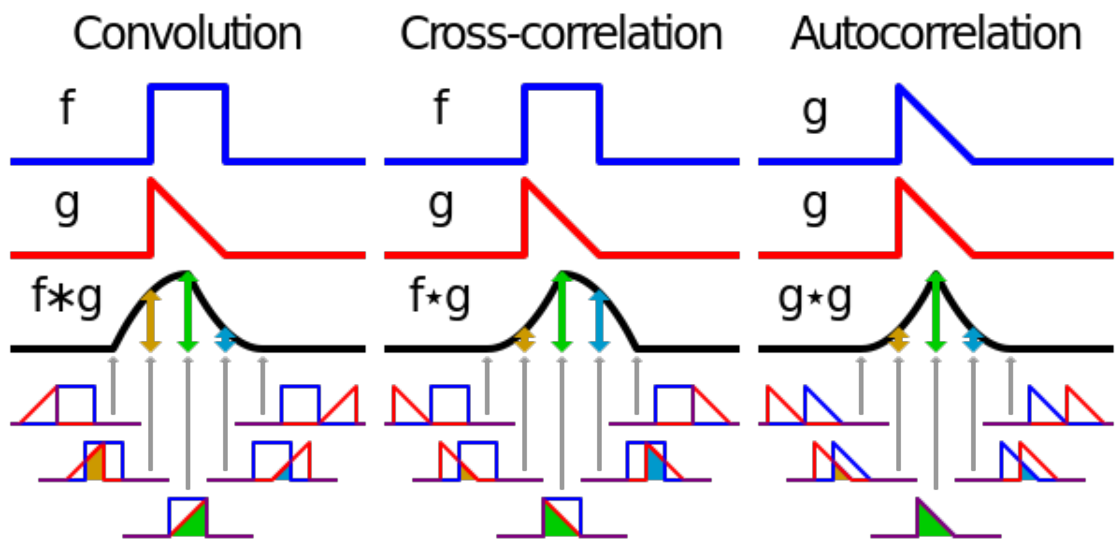
- Applications:
- LTI – system, effect on a with impluse v response v
- Prob theory: distribution of X + Y if f(x) * f(y), X, Y must be independant

Convolution vs Cross Correlation

First, recall Cross Correlation as a function:







np.corr(x1,x2[n:n+w])?

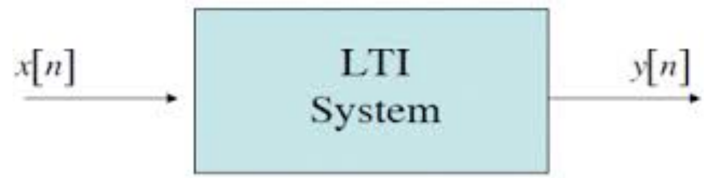
This is important to understand because
f(n) * g(-n) is the cross correlation (a function of n)

dsp.stackexchange.com

Convolution: why we care

FIR:Finite Impulse Response Filter

- Take Any Input Sequence x[n]
- Filter for what you want with h[n]
- Get result y[n]



h[n]	**Result**
1/W,1/W,1/W,1/W	MA or Low pass filter or moving average, with 1/W cutoff
(a1,a2,…a10)	Any linear weighted sum
1 1, -1 -1	Differential, locates an edge
Any sequence	Cross-correlation, allows you to look for similar pattern

Properties of Convolution

EQUATION 7-6
The commutative property of convolution. This states that the order in which signals are convolved can be exchanged.

$$a[n] * b[n] = b[n] * a[n]$$

EQUATION 7-8
The distributive property of convolution describes how parallel systems are analyzed.

$$a[n] * b[n] + a[n] * c[n] = a[n] * (b[n] + c[n])$$

Visual Example

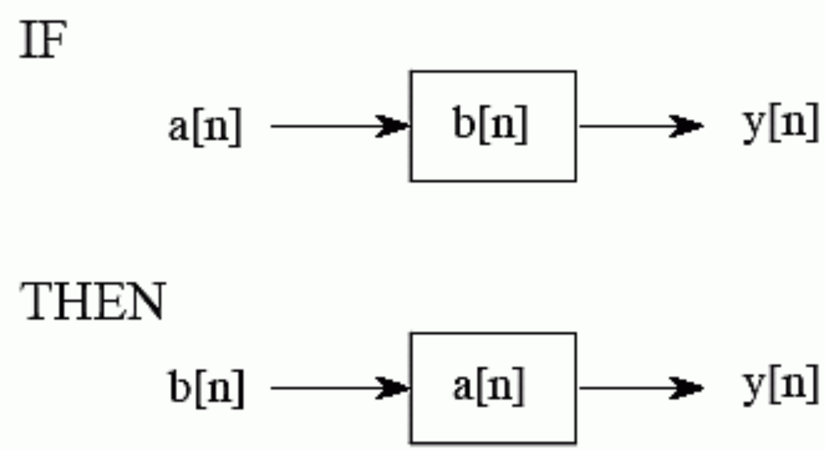


FIGURE 7-8
The commutative property in system theory. The commutative property of convolution allows the input signal and the impulse response of a system to be exchanged without changing the output. While interesting, this usually has no physical significance. (A signal appearing inside of a box, such as b[n] and a[n] in this figure, represent the *impulse response* of the system).

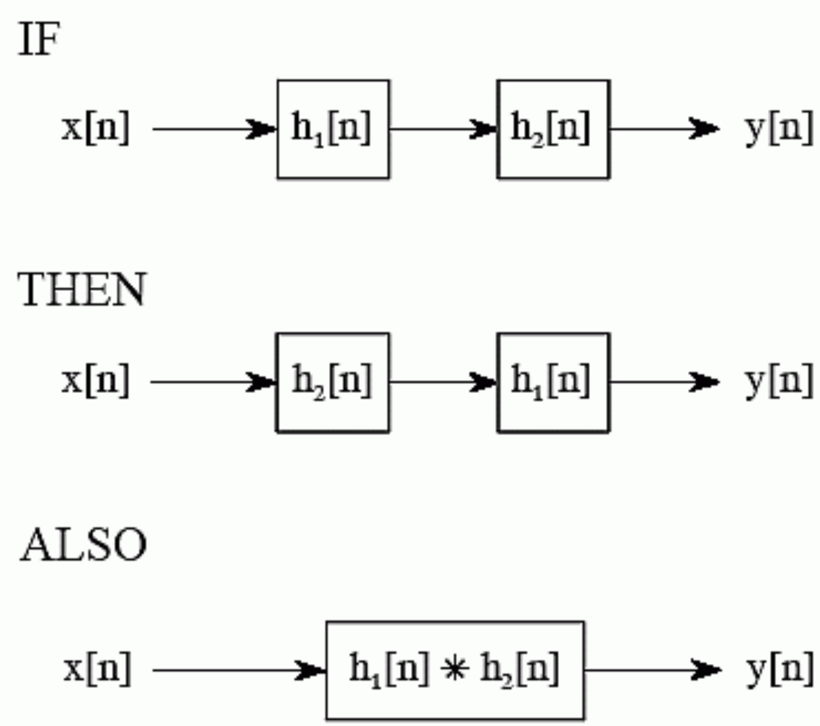


FIGURE 7-9
The associative property in system theory. The associative property provides two important characteristics of *cascaded* linear systems. First, the order of the systems can be rearranged without changing the overall operation of the cascade. Second, two or more systems in a cascade can be replaced by a single system. The impulse response of the replacement system is found by convolving the impulse responses of the stages being replaced.

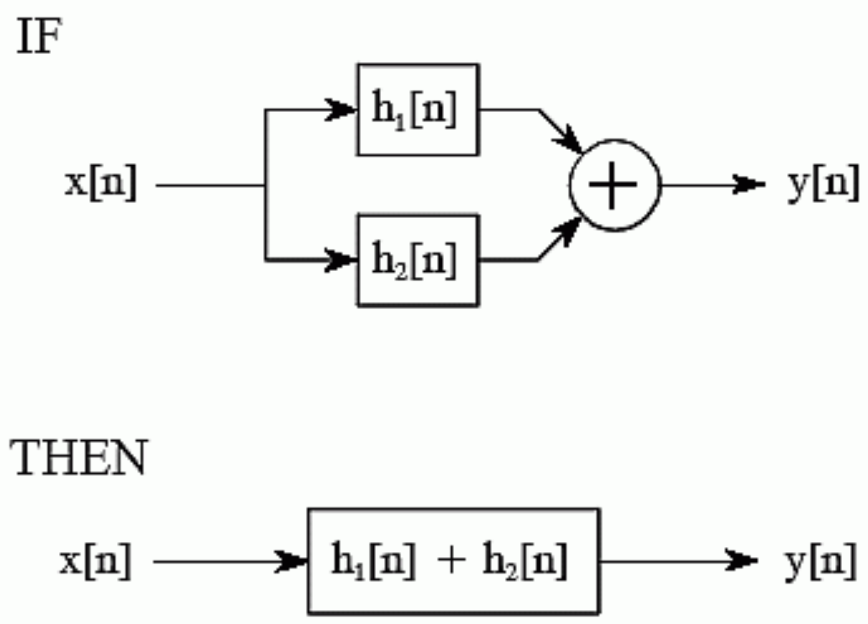
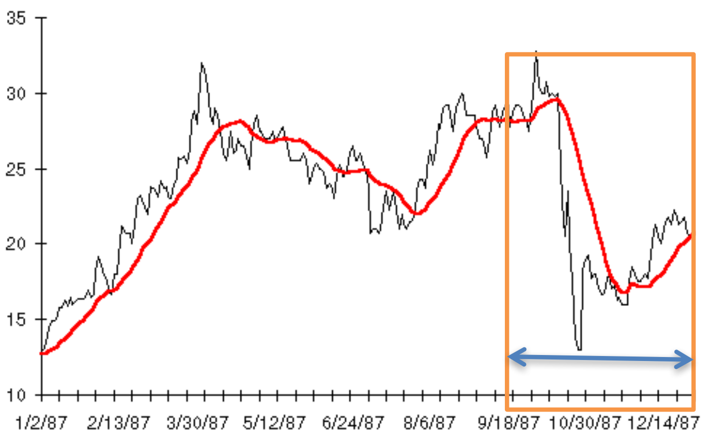


FIGURE 7-10
The distributive property in system theory. The distributive property shows that parallel systems with added outputs can be replaced with a single system. The impulse response of the replacement system is equal to the sum of the impulse responses of all the original systems.

Suppose you want to keep statistics but don't want to keep the past data



Example: Moving Average

$$s_t = \frac{1}{k}\sum_{n=0}^{k-1} x_{t-n} = \frac{x_t + x_{t-1} + x_{t-2} + \cdots + x_{t-k+1}}{k} = s_{t-1} + \frac{x_t - x_{t-k}}{k}$$

Example: Exponential Moving Average

$s_t = \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \alpha(1-\alpha)^3 X_{t-3}$ .... (6.2.1)

$s_{t-1} = \alpha X_{t-1} + \alpha(1-\alpha)X_{t-2} + \alpha(1-\alpha)^2 X_{t-3} + \alpha(1-\alpha)^3 X_{t-4}$ .... (6.2.2)

Multiply $(1-\alpha)$ though 6.2.2 yields:

$(1-\alpha)s_{t-1} = \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \alpha(1-\alpha)^3 X_{t-3} + \alpha(1-\alpha)^4 X_{t-4}$ ...

6.2.3

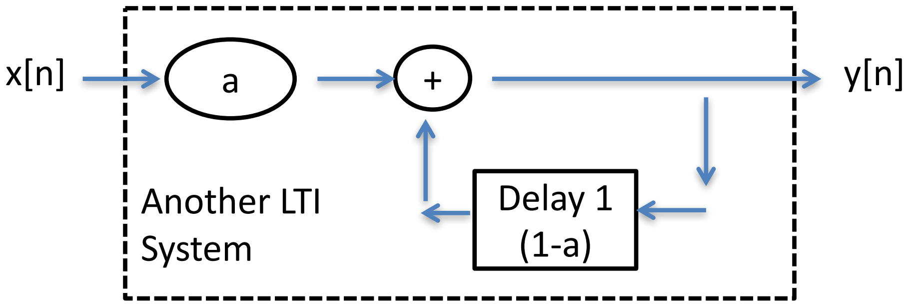
Subtract 6.2.3 from 6.2.1

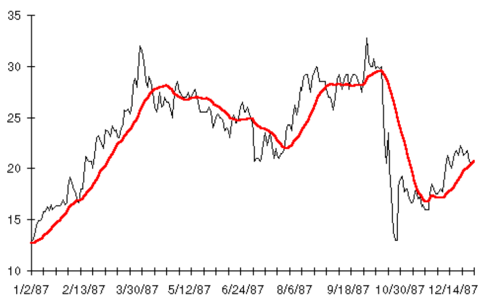
$s_t - (1-\alpha)s_{t-1} = \alpha X_t$

Or $st = \alpha X_t + (1-\alpha)s_{t-1}$

Another LTI System: Infinite Impulse Response

- Discrete data x_n = x1, x2, x3, ...





alpha = smoothing factor of 2/(W+1)

W is number of time periods

$$y(n) = \alpha x(n) + (1-\alpha) y(n-1)$$

Notice:

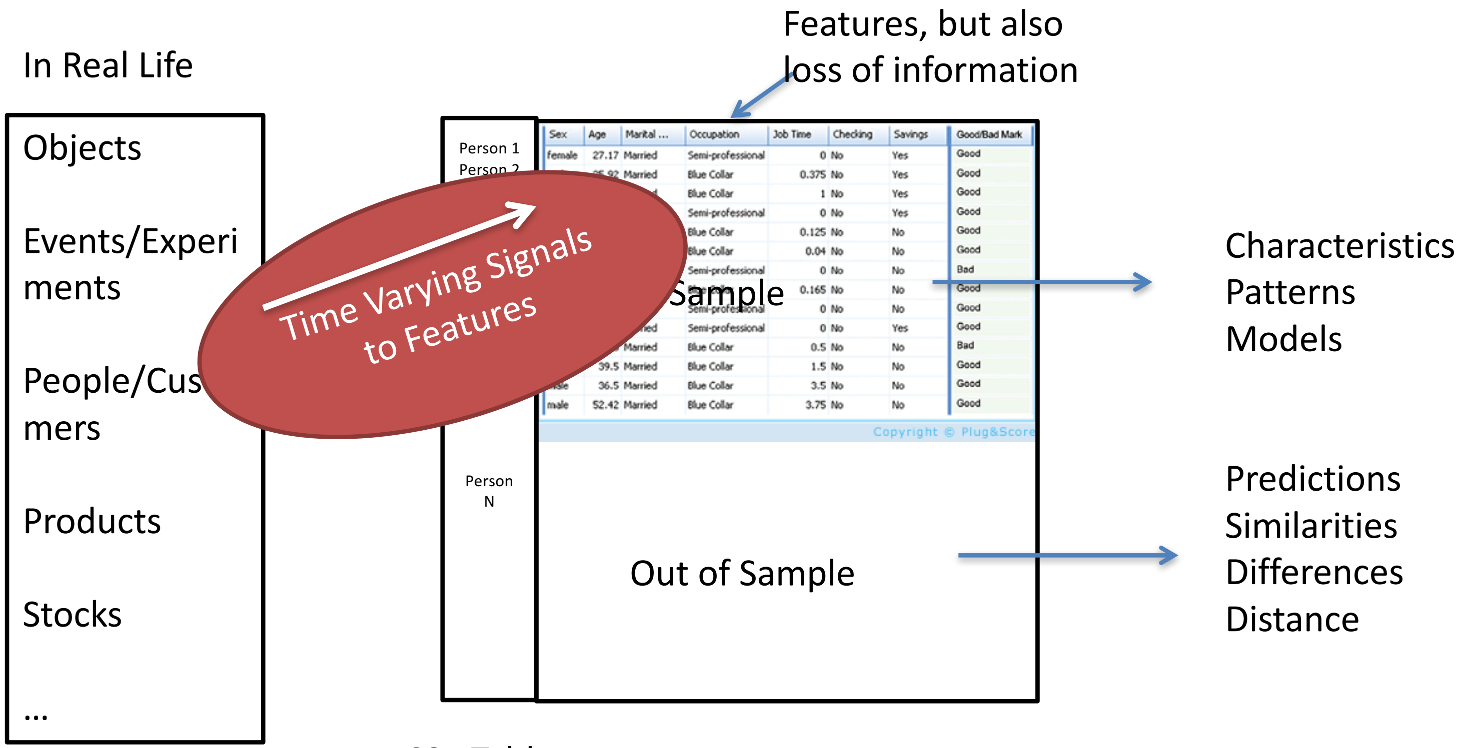
We only use need state:

- xn = the most recent input
- yn-1 = last estimate of exp. moving average

Equivalent to below, which technically uses all past values:

$$s_t = \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \alpha(1-\alpha)^3 X_{t-3} \ldots$$

Summary: Time Varying Data Signals to Features



CS: Table
Math: Matrix X, which is
N rows – each person
m columns, each feature (age, salary, ..)

End of Section