Trends (inc. or dec)
periodicity.
etc.

Lag-K TS

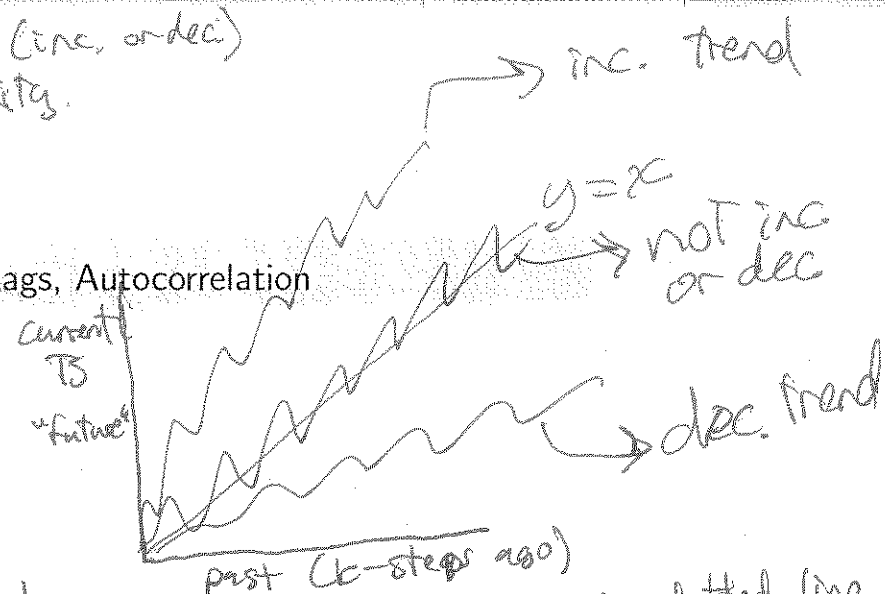
What did the TS look like K-steps ago
↓
time units

Recap: Time Series, Lags, Autocorrelation

Lags and Lag Plots:

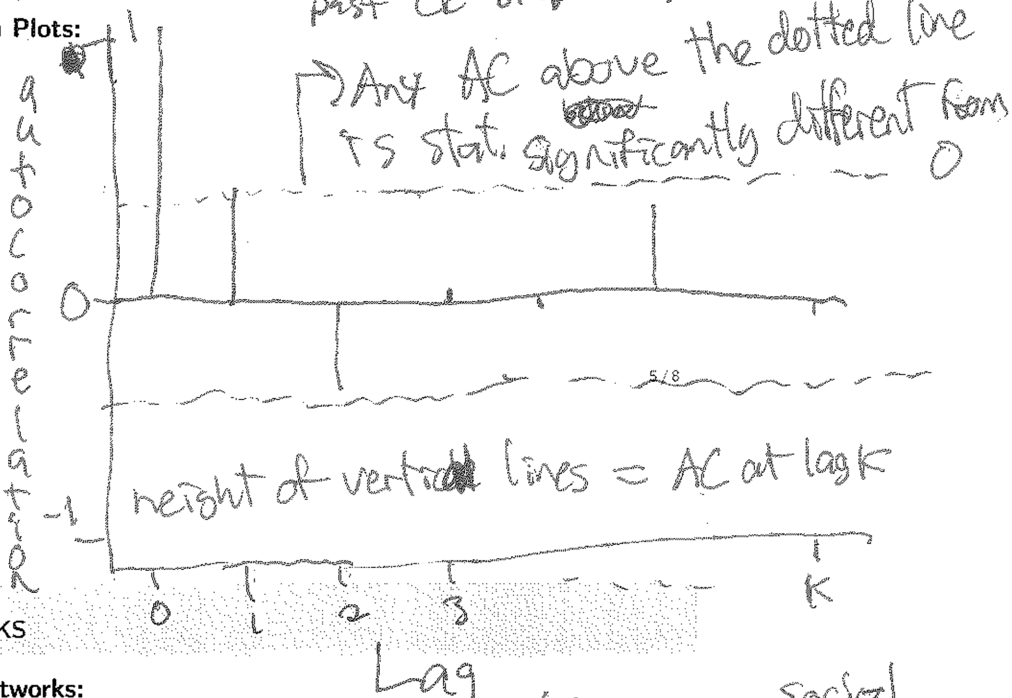
Lag-K plot

TS vs. itself K-steps ago.



Autocorrelation Plots:

Autocorrelation: Correlation between the TS and itself at a specified lag



5/8

Today: Networks

"Graphs"

Examples of networks:

networks of neural cells → send signals to each other / the body

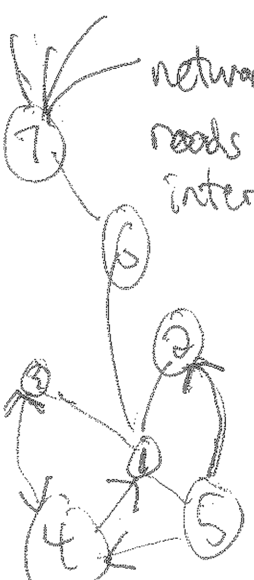
social

roads / people on roads

citations / wikipedia

internet / servers / cell phone networks

Basic Network Structure:



Nodes: 1, 2, ... 7: these can be people, servers, phones, neurons

Links/Edges: undirected, directed
→ linkedin
→ Twitter, instagram
→ ~~# of twitter followers~~
mutual
↳ in-degree
↳ out-degree → # following

weighted → "facebook news feeds" → strength of the link

Path: min. Distance between two nodes ("6 degrees of separation")

6/8

Cliques ≃ "clusters", hubs / "center", bottlenecks

Adjacency ≈ Distance (but opposite)

$A_{ij} = g(d_{ij})$, where g is a monotonically dec. fxn, non-neg.

Adjacency Matrices

n = # of nodes / observations

Adjacency Matrix: representation of the network in matrix form

$$\underset{n\times n}{A} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ i \\ n \end{array}\begin{array}{c} 1 \; 2 \; \cdots \; n \\ \left[\begin{array}{c} A_{ij} \approx \text{edge weight} / \\ \text{"link strength"} / \\ \text{presence/absence of} \\ \text{a link between} \\ \text{nodes } i \text{ and } j \end{array} \right] \end{array}$$

$A_{ij} \in \{0, 1\}$ binary link Twitter

$A_{ij} \in [0, 1]$

↳ "probability of a link"

$A_{ij} \in [0, \infty)$

A_{ij} = link → represents directed association between two nodes

$A_{ij} \neq A_{ji}$ (they can be the same)

Visualizing Networks

Layout: Trying to position the nodes to "maximize" visualization corresponding to specific criteria

* force directed algorithm: position nodes such that
 1) edges are approximately the same length
 2) few crossing edges

Assign "forces" to the edges / nodes, then minimize their "energy"

~~**Layout**: Trying to position the nodes to "maximize" visualization corresponding to specific criteria~~

Kamada-Kawai: (smaller graphs) → good for connected graphs, poor for unconnected graphs

Fruchterman-Reingold: use for very large graphs / networks

adds extra emphasis on even vertex distribution