



# Cluster Analysis of Stock Returns

---

Michael Tan, Ph.D., CFA

Copyright © 2002 Michael Tan, Ph.D., CFA

[www.michaeltanphd.com](http://www.michaeltanphd.com)

Apothem Capital Management, LLC

330 East 38<sup>th</sup> Street 14L

New York, NY 10016

Tel: 212-922-1265

[mltan@apothemcapital.com](mailto:mltan@apothemcapital.com)

All rights reserved



## Copyright Notice and Disclaimer of Liability

---

The material in this document is copyrighted by Michael Tan and Apothem Capital Management, LLC for which all rights are reserved.

This document may not be reproduced or distributed in any form, in whole or in part, or by any means, electronic or otherwise, including photocopying or by means of any computer storage and retrieval system without the express written permission of Michael Tan and Apothem Capital Management, LLC. Nonetheless, permission is hereby granted to the recipient of this document who downloaded it from the website of Michael Tan with the URL [www.michaeltanphd.com](http://www.michaeltanphd.com) to retain a printed copy or copy stored on a computer for his or her personal use.

Michael Tan, Apothem Capital Management LLC and its affiliates assume no responsibility for anyone's use of the information contained in this document and shall not be held liable for any direct, indirect, incidental or consequential damages including, but not limited to loss of profits and opportunities or business interruption, however caused and arising in any way out of the use of the information contained in this document.

This document was expanded and edited from a corporate lecture given by Michael Tan at Maple Securities USA in 2002.

Copyright © 2002 Michael Tan, Ph.D., CFA

[www.michaeltanphd.com](http://www.michaeltanphd.com)

Apothem Capital Management, LLC

330 East 38<sup>th</sup> Street 14L

New York, NY 10016

Tel: 212-922-1265

[mltan@apothemcapital.com](mailto:mltan@apothemcapital.com)

All rights reserved



## What is cluster analysis?

---

- Clustering is the process of organizing objects into classes or clusters whose members are similar in some way.
- The “classical” approach, well-developed by the mid-1970’s, deals with the automatic discovery of classes in data, where the classes reflect causal mechanisms making some objects more similar to each other than the rest.
- The advent of the World Wide Web in the 1990s brought about a resurgence of interest in clustering algorithms that can organize the vast amounts of data produced by search engines.
- Clustering is the natural research and automation tool for studying stock prices since many short-term trading strategies are based on relationships among stocks.
- The tools available in Matlab cover only the classical approach.



## Two styles of clustering data

---

- Two major styles of clustering in the classical approach:
  - *Partitioning* (“k-clustering”)
  - *Hierarchical Clustering* (“tree clustering”)
- *Partitioning* divides a set of objects into  $k$  clusters, where  $k$  is known or given a priori.
- *Hierarchical clustering* aims at discovering the “natural” classes in data where no “obvious” partitions can at first be seen.



## How to quantify the similarity between two objects?

---

- In cluster analysis, the causal relationship between two objects is quantified by a *distance function* giving the degree of association or similarity between the objects.
- Other names for distance function are “metric function”, “dissimilarity measure”, etc.
- For a function  $d(x_i, x_j)$  of  $x_i$  and  $x_j$  drawn from an input set  $S$  to qualify as a distance function, it must satisfy the “Euclidean metric”:

$$d(x_i, x_i) = 0 \quad \text{(i)}$$

$$d(x_i, x_j) = d(x_j, x_i) \quad \text{(ii)}$$

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j) \quad \text{(iii)}$$

- Metricable distance functions can be designed creatively to capture desired causal aspects of relationships among objects provided they satisfy (i), (ii), and (iii) above.
- In practice, one can simply verify numerically that a trial distance function satisfy (i), (ii), and (iii) for a large sample of objects drawn from the universe under study.



## Distance functions for analyzing security return relationships

---

- Some distance functions useful for the study of security prices are listed below.
- Let  $x_{it}$  and  $x_{jt}$  denote the returns of stock  $i$  and stock  $j$  at time  $t$  respectively;  $\sigma_i$  and  $\sigma_j$  denote the standard deviation of these returns; and the vectorized notation  $\mathbf{x}_i$  denote the time series of returns of stock  $i$ .

- *Euclidean metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^2}$$

The distance tends to be small for stocks with similar return volatility or beta and therefore groups stocks with similar betas together.

- *Standardized Euclidean metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\left[ \frac{(\mathbf{x}_i - \bar{\mathbf{x}}_i)}{\sigma_i} - \frac{(\mathbf{x}_j - \bar{\mathbf{x}}_j)}{\sigma_j} \right]^2} = \sqrt{2(1 - \rho_{ij})}$$

This metric can be expressed as a function of the correlation  $\rho_{ij}$  between stock  $i$  and  $j$  which vanishes when they are perfectly correlated.

- *City block metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_t |x_{it} - x_{jt}|$$

Dampens large return differences and thus gives more weight to small differences.



## More distance functions

---

- More distance functions:

- *Minkowski metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_t |x_{it} - x_{jt}|^\alpha \right)^{1/\alpha}$$

Large (small) emphasizes large (small) return differences.

- *Chebychev metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \max_t |x_{it} - x_{jt}|$$

This metric is susceptible to idiosyncratic events affecting stock prices.

- *Cumulated difference metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \max_{t_1 \leq t \leq t_2} \left| \int_{t_1}^{t_2} dy \right| \quad \text{where } y = x_i(t) - x_j(t)$$

This metric is used in the paper by Gatev, Geotzmann & Rouwenhorst, "Pairs trading: Performance of a relative value arbitrage rule".

- *Percent disagreement metric:* 
$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{\text{count } x_{it} \neq x_{jt}}{\text{total time steps}}$$

Not useful for analyzing price data but may be used to analyze categorical data when is taken to be a category index.



## Even more distance functions

- Some more distance functions:

- *Correlation metric:*  $d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \rho_{ij}^2$

Treats both correlated and anti-correlated stocks as close neighbors (Mantegna, "Hierarchical Structure in Financial Markets", *Eur. Phys. J. B* **11**, 193-197 (1999)).

- *Information Entropy metric:*  $d(\mathbf{x}_i, \mathbf{x}_j) = \frac{2H(\mathbf{x}_i, \mathbf{x}_j)}{H(\mathbf{x}_i) + H(\mathbf{x}_j)} - 1$

where  $H(\mathbf{x}_i, \mathbf{x}_j)$  is the mutual entropy of stock  $i$  and stock  $j$ , and  $H(\mathbf{x}_i)$  is the entropy of stock  $i$ , given by:

$$H(x_i, x_j) = -\sum_i \sum_j P(x_i, x_j) \log_2 P(x_i, x_j)$$

$$H(x_i) = -\sum_i P(x_i) \log_2 P(x_i)$$

Here  $P(x_i, x_j)$  is the joint probability of observing a return  $x_i$  for stock  $i$  and a return  $x_j$  for stock  $j$ ;  $P(x_i)$  is the probability of observing a return  $x_i$  for stock  $i$ . The *mutual information*  $I(x_i, x_j)$  is defined as:

$$\begin{aligned} I(x_i, x_j) &= H(x_i) + H(x_j) - H(x_i, x_j) \\ &= H(x_i) - H(x_i|x_j) \\ &= H(x_j) - H(x_j|x_i) \end{aligned}$$

where  $H(x_i|x_j) = -\sum_i \sum_j P(x_i, x_j) \log_2 P(x_i|x_j)$  is the conditional entropy.  $I(x_i, x_j)$  is

interpreted as the reduction in the uncertainty of  $x_i$  due to the knowledge of  $x_j$ . The

distance function is proportional to  $-I(x_i, x_j)$  and thus the more uncertainty is

reduced the closer is  $i$  to  $j$ .



## Agglomerative clustering

---

- One way to perform hierarchical clustering is by *agglomeration*:
  - Step 1: Start with a set  $S$  of  $n$  objects.
  - Step 2: Place elements of  $S$  into singleton sets  $S_1, S_2 \dots S_n$ .
  - Step 3: Devise a *cost function* which determines the pair of sets  $\{S_i, S_j\}$  that is “cheapest” to merge.
  - Step 4: Remove  $S_i, S_j$  from the list of sets and replace with  $S_i \cup S_j$ .
  - Step 5: Repeat steps 3 and 4 until only one set remains.
- Agglomerative clustering then differ only in the definition of the cost function or *linkage algorithm*.
- A linkage algorithm links objects together into clusters based on the “cost” of each link which in turn depends on the distances between objects.



## Linkages

---

- Some common linkage algorithms and their cost functions are given below:

Algorithm

Cost Function

*Single Linkage*

$$\min_{x_i \in S_i, x_j \in S_j} d(x_i, x_j)$$

*Complete Linkage*

$$\max_{x_i \in S_i, x_j \in S_j} d(x_i, x_j)$$

*Ward Linkage*

$$c(S_i) = \sum_{p=1}^{n(S_i)} \sum_{q=1}^{n(S_i)} [d(x_p^{(i)}, x_q^{(i)})^2] \quad \text{where } n(S_i) = \text{number of elements in } S_i$$

- Ward linkage is the hierarchical clustering counterpart to k-clustering:

Step 1: Start with a set  $S$  of  $n$  objects.

Step 2: Place elements of  $S$  into singleton sets  $S_1, S_2, \dots, S_n$ .

Step 3: Let  $W = \sum_{i=1}^n c(S_i)$  whose initial value is 0.

Step 4: Merge the pair  $\{S_i, S_j\}$  such that  $W$  increases the least after merging. If  $S_i$  or  $S_j$  is not singleton, then the distance between them is that between their centroids.

Step 5: Remove  $S_i, S_j$  from the list of sets and replace with  $S_i \cup S_j$ .

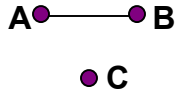
Step 6: Repeat steps 4 and 5 until only one set remains.



## Spherical versus elongated clusters

---

- Complete, Ward, and average linkage tend to produce “spherical” clusters, i.e. those whose members are classified together because they are close to each other or to a “bellwether” member:



C must be close to both A and B to make {A, B, C} a cluster

- Single linkage tend to produce “elongated” clusters:



C need only be close to B (but not necessarily to A) to make {A, B, C} a cluster

- Complete linkage is “furthest neighbor” linkage while single linkage is “nearest neighbor” linkage.
- To the extent that there are bellwether stocks and lead-lag relationships among stocks, single linkage is probably not suited for the analysis of stock clusters.



## ***k*-clustering**

---

- *k*-clustering or *k*-means clustering will form *k* clusters that are as distinct as possible, where *k* is stipulated or known a priori.

- Minimize the sum of costs  $W = \sum_{i=1}^k c(S_i)$  where the cost is defined as

$$c(S_i) = \sum_{p=1}^{n(S_i)} \sum_{q=1}^{n(S_i)} [d(x_p^{(i)}, x_q^{(i)})^2] \quad \text{where } n(S_i) = \text{number of elements in } S_i$$

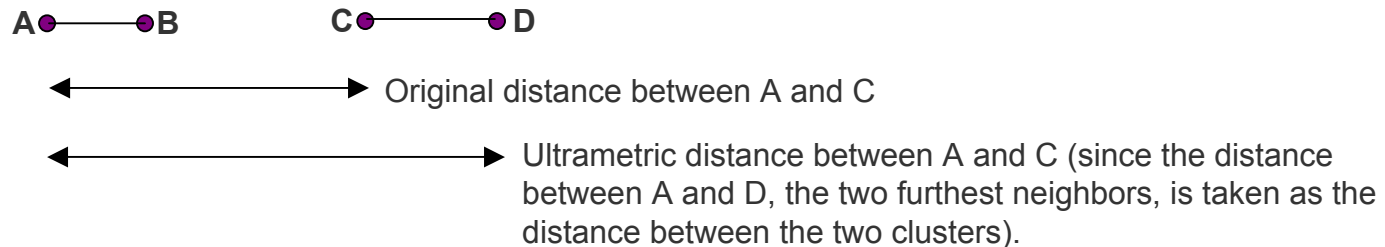
- The sum of costs criterion above is same as one used in Ward linkage.
- *k*-clustering will minimize variability within clusters and maximize variability among clusters.
- This is the same problem as the partitioning of the sum of squares in ANOVA, i.e. into sum of squared distances from the mean of each cluster plus sum of squared distances of cluster means about overall mean.
- *k*-clustering maximizes the corresponding ANOVA test statistic for significance of the variability among cluster means.



## Cophenetic coefficient

---

- The cophenetic coefficient is the correlation between distances in the original set of objects and their corresponding “ultrametric” distances.
- For example, suppose 4 objects A, B, C and D are agglomerated into 2 clusters {A, B} and {C, D} via complete linkage:

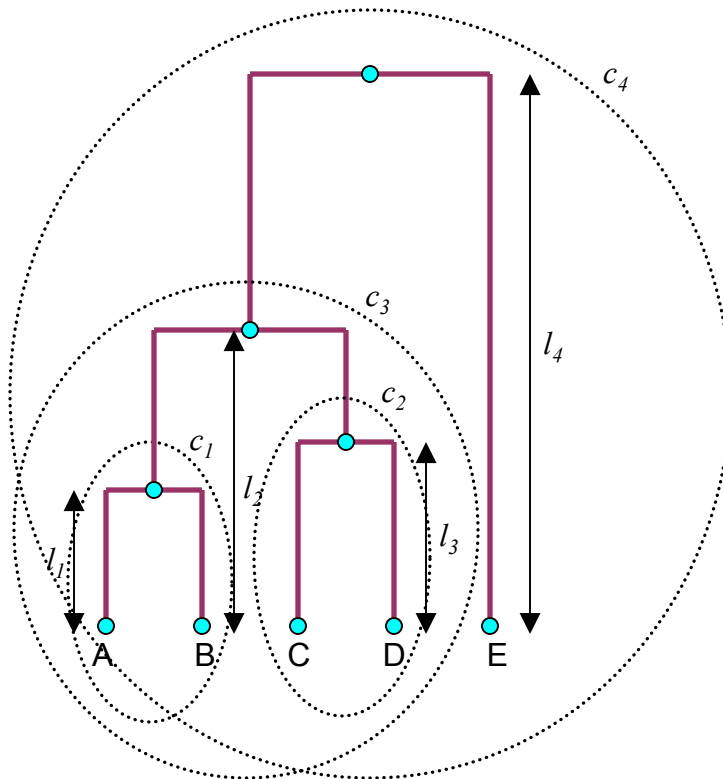


- The original object-to-object distance AC is to be correlated with the distance between the clusters to which A and C belong.
- The cophenetic coefficient measures the distortion of the distance information in the original data set caused by linkage into clusters.



## Dendrograms

- A *dendrogram* or *hierarchical cluster tree* is a graphical depiction of the distances between clusters.
- The tree ends in leaves at the bottom of the graph corresponding to singleton clusters, i.e. single stocks.
- Branches are drawn such that height of node at which one branch meets another is equal to the distance between clusters at the end of the branches.



● represents a node

The objects A, B, C, D, E are *leaf nodes*.

Number of objects is 5.

Number of clusters  $c_1, c_2, c_3, c_4$  is always one less than the number of objects.

The height of the links  $l_1, l_2, l_3, l_4$  correspond to distances.



## Inconsistent coefficient

---

- An *inconsistent coefficient* can be computed for each link in a cluster tree.
- It compares the length of the link to the average length of links below it to a specified depth (number of levels in the hierarchy).
- For example, the inconsistent coefficient to depth one for link  $l_2$  in the tree on the previous page is

$$\frac{l_2 - (\text{mean of } l_1, l_2, \text{ and } l_3)}{(\text{standard deviation of } l_1, l_2, \text{ and } l_3)}$$

- For depth  $m$ , the mean and standard deviation in the above formula are taken over the lengths of all links stemming from  $l_2$  down  $m$  levels in the hierarchy.
- By definition, links connecting leaf nodes have inconsistent coefficients of zero.
- A large inconsistent coefficient implies that the link connects two very distinct clusters.
- A small inconsistent coefficient implies that the link connects two clusters that are not differentiated.



## Generating clusters

---

- Clusters can be generated by grouping together the leaves at the bottom of a link whose inconsistent coefficient is larger than a specified *cut-off value*.
- The node from which the link emanates may in principle be several levels above the leaves if the links below it have inconsistent coefficients smaller than the cut-off.
- In this case, a large cluster may be generated.
- Thus, a large cut-off value (say larger than 1) will usually produce large clusters while a small cut-off value will produce small clusters.
- Inconsistent coefficients typically have a numerical range between 0 and 3.
- Different clustering configurations can be obtained by adjusting the cut-off value.



## Example of a dendrogram created using single linkage

- This is a dendrogram showing the hierarchical structure of the top 226 stocks ranked by market capitalization from the S&P 500 Index produced using single linkage of the standardized Euclidean metric for returns.

