Understanding Musical Performance

Using Self-Organising Maps to Characterise Performance Timing

Nicolas Gold, Neta Spiro, John Rink

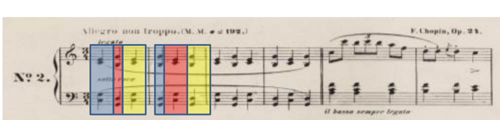
Data Analysis

- Absolute Values
  - Advantages
    - dealing with the “real” data
  - Disadvantages
    - have to use thresholds and tolerances (derived from where?)
- Relative Values
  - Advantages
    - dealing with the “shape” of expression
  - Disadvantages
    - not dealing with the “real” values directly

Relative Timing Analysis

- Need to define a range over which the shape will be drawn
  - e.g. a single bar

Beat 1	Beat 2	Beat 3
Long	Short	Long
Short	Long	Long



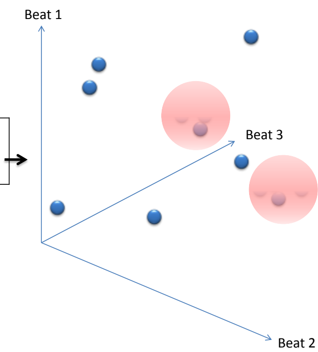
Example

Beat 1	Beat 2	Beat 3	Bar	Performance
39.34859	31.23548	29.41593	1	Rubinstein 1939
34.35139	32.39354	33.25507	2	Rubinstein 1939
33.56017	32.3172	34.12263	3	Rubinstein 1939
31.25251	39.47685	29.27064	4	Rubinstein 1939
41.68263	26.49714	31.82023	5	Rubinstein 1939
33.33333	33.33333	33.33333	6	Rubinstein 1939
34.12675	31.43626	34.43699	7	Rubinstein 1939
28.37537	36.37481	35.24982	8	Rubinstein 1939
32.48396	36.25022	31.26582	9	Rubinstein 1939
28.92779	37.27807	33.79414	10	Rubinstein 1939
31.641	35.48523	32.87377	11	Rubinstein 1939
30.57223	35.31093	34.11684	12	Rubinstein 1939
36.71971	28.76377	34.51652	13	Rubinstein 1939
34.1354	29.44178	36.42282	14	Rubinstein 1939
31.69572	36.60856	31.69572	15	Rubinstein 1939
32.9588	34.0824	32.9588	16	Rubinstein 1939
34.79624	30.40752	34.79624	17	Rubinstein 1939
32.5804	30.13687	37.28273	18	Rubinstein 1939
31.64366	39.1468	29.20954	19	Rubinstein 1939
29.44481	33.07997	37.47521	20	Rubinstein 1939
31.11139	34.97017	33.91844	21	Rubinstein 1939

Relative Timing Analysis

- How do we find similarities in both the shapes and the various degrees of similarity without imposing an *a priori* decision on what might occur?
  - Need to be able to cluster similar things together and then determine appropriate boundaries between the clusters
- So, regard each triple (b_1,b_2,b_3) as a vector
  - Has the advantage that additional values could be considered by increasing the dimensionality of the vector space.

Beat 1	Beat 2	Beat 3	Bar	Performance
39.34859	31.23548	29.41593	1	Rubinstein 1939
34.35139	32.39354	33.25507	2	Rubinstein 1939
33.56017	32.3172	34.12263	3	Rubinstein 1939
31.25251	39.47685	29.27064	4	Rubinstein 1939
41.68263	26.49714	31.82023	5	Rubinstein 1939
33.33333	33.33333	33.33333	6	Rubinstein 1939



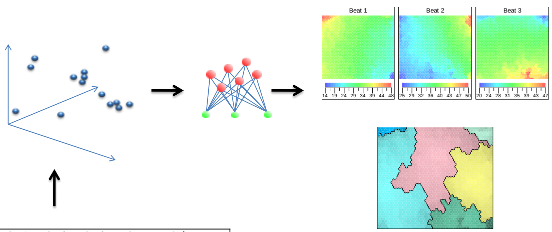
Self-Organising Maps

- A way to reduce the dimensionality of data
- The high dimensional space is represented by a set of nodes, each containing a "prototype" representation of part of that space e.g.
  - Node 1 (5, 10, 5)
  - Node 2 (7, 3, 15)
- For every vector in the input space (e.g. 4.5, 9, 4.5), one of the "prototype" nodes will match most closely (Euclidean distance)
- In training, each of the matching "prototype" nodes is moved slightly closer to the input vector that triggered it, along with its neighbours
- After many training cycles, the "prototype" nodes have been pulled into a topological representation of the input space
- New (or existing) data can then be classified by presenting it to the map and determining which "prototype" node (or nodes) trigger.

Self-Organising Maps

- Neural network that maps high-dimensional input data to 2-dimensional visualisation space
- No supervised training required
- Preserves topological relationships between items in the (high-dimensional) input space.
- Generic Process
  - Define a map of size p x q
  - Define how units on the map relate to each other
  - Initialise units to random initial values
  - Present all input data many times, each time moving every closest matching unit slightly nearer to the current input
  - On each pass decrease the amount by which the units "move"
  - The representation of the input space thus slowly "cools" and settles into a pattern

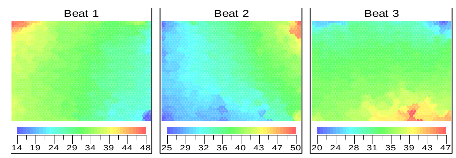
3D -> 2D



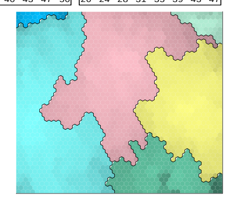
Beat 1	Beat 2	Beat 3	Bar	Performance
39.34859	31.23548	29.41593	1	Rubinstein 1939
34.35139	32.39354	33.25507	2	Rubinstein 1939
33.56017	32.3172	34.12263	3	Rubinstein 1939
31.25251	39.47685	29.27064	4	Rubinstein 1939
41.68263	26.49714	31.82023	5	Rubinstein 1939
33.33333	33.33333	33.33333	6	Rubinstein 1939

Images from Eudaptics Viscovery SOMine

Viscovery SOMine Representation

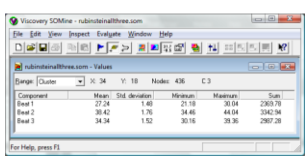


<u>SOM-Ward Clustering</u>
Initially, every node is a cluster
Clusters merged based on shortest distance accounting for map adjacency
Quality measure indicates "best" clustering

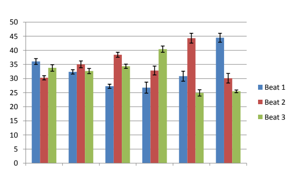


From Map to Motive

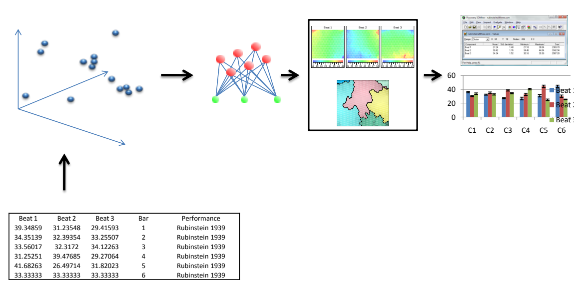
- For each cluster, take the mean value of each component e.g.



- Plot in Excel for shape

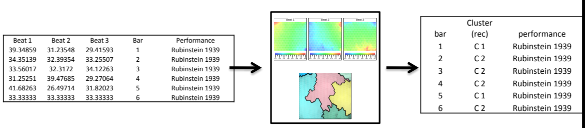


From Performance to Motive



From Map to Performance Characterisation

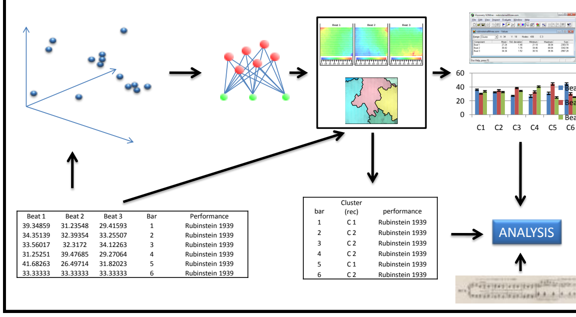
- Present the performance data to the map, recalling cluster membership



Beat 1	Beat 2	Beat 3	Bar	Performance
39.34859	31.23548	29.41593	1	Rubinstein 1939
34.35139	32.39354	33.25507	2	Rubinstein 1939
33.56017	32.3172	34.12263	3	Rubinstein 1939
31.25251	39.47685	29.27064	4	Rubinstein 1939
41.68263	26.49714	31.82023	5	Rubinstein 1939
33.33333	33.33333	33.33333	6	Rubinstein 1939

bar	Cluster (rec)	performance
1	C 1	Rubinstein 1939
2	C 2	Rubinstein 1939
3	C 2	Rubinstein 1939
4	C 2	Rubinstein 1939
5	C 1	Rubinstein 1939
6	C 2	Rubinstein 1939

From Performance to Motive and Characterisation



Beat 1	Beat 2	Beat 3	Bar	Performance
39.34859	31.23548	29.41593	1	Rubinstein 1939
34.35139	32.39354	33.25507	2	Rubinstein 1939
33.56017	32.3172	34.12263	3	Rubinstein 1939
31.25251	39.47685	29.27064	4	Rubinstein 1939
41.68263	26.49714	31.82023	5	Rubinstein 1939
33.33333	33.33333	33.33333	6	Rubinstein 1939

bar	Cluster (rec)	performance
1	C 1	Rubinstein 1939
2	C 2	Rubinstein 1939
3	C 2	Rubinstein 1939
4	C 2	Rubinstein 1939
5	C 1	Rubinstein 1939
6	C 2	Rubinstein 1939