Tel Aviv University

The Raymond and Beverly Sackler

Faculty of Exact Sciences

Quantum Clustering of Large Data Sets

Thesis submitted towards the M.Sc. Degree

of Tel Aviv University

School of physics and astronomy

Submitted by

Guy Shaked

This research work has been carried out under the supervision of

Professor David Horn

January, 2013

Acknowledgments

I would like to thank my supervisor, Professor David Horn, for the patient guidance, encouragement and advice he has provided throughout my thesis as his student. I have been lucky to have a supervisor who cared so much about my work.

I wish to thank Dr, Erez Persi, Dr. Rami Hoffstetter and Prof. Marvin Weinstein for listening commenting and contributing ideas.

Also, I wish to thanks my parents, for their love and support.

And last but definitely not least, I would like to thank my better half Michal, without her encouragement and support this work could not have been done.

Table of Contents

Abstract

Unsupervised learning in general and clustering in particular, are the starting point of many exploratory studies, in order to find interesting patterns with no prior knowledge in new data. Big data sets become more common in many fields and the search for novel tools to extract and store knowledge is intensified.

We try to find new insights on two existing large data sets by using the quantum clustering method. On financial data, our method shows clustering of stocks to different groups, with correlation to their industry sector. Also, we show that by looking at the market as a time series we can distinguish between different states which the market shifts between.

On a catalog of earthquakes in the Eastern Mediterranean Region and the Dead Sea Fault, we demonstrate that earthquakes can be meaningfully clustered with respect to geophysics features alone. Correlating these clusters with time and location information then leads to novel insights regarding the characteristics of major faults. We conclude that our methodology has been validated, and our unsupervised analysis has led to a new perspective and understanding of this complex data set.

On the technical side we implemented three new applications. First, a new preprocessing step for quantum clustering , which leads to reduction in the algorithm complexity and thus running it on big data sets is feasible. Second, a newer version of COMPACT, with implementation of support vector clustering, and few enhancements for the quantum clustering algorithm. Third, an implementation of quantum clustering in Java.

Chapter 1

Introduction

Several authors in different fields have shown that by using clustering techniques, one can extract important and previously unknown understanding of the data in question. Samples of this might include work done on gene expression data[1], social network analysis[2], medical imaging[3], chemistry[4], and many other diversified fields of interest.

Quantum Clustering[5] and Dynamic Quantum Clustering[6] are two clustering methods which were inspired by quantum mechanics. Both of these methods showed great promise in exposing hidden patterns of data structures.

In this work we will perform exploratory search using these methods on new data sets, trying to find new conclusions on the related topics. We will also suggest a new way allowing the QC method to tackle big-data problems.

1.1 Background

1.1.1 Pattern recognition:

From the dawn of life on earth, all creatures with sense organs depend deeply on the ability to analyze quickly data that comes from their sense organs for survival, causing them to evolve highly sophisticated neural and cognitive systems for such pattern recognition tasks – taking in raw data and performing an action depending on the category of the recognized pattern.

The ease of which humans can identify a face in a crowd, understand poor handwriting, recognize words over a bad telephone connection or sense danger by a faint smell, sometimes falsely lead us to think that such actions are "easy"[7], and might be easily reproduced by machines.

In order to understand the problem of creating a computer program which can recognize patterns, let's think of a real-world problem – a system for handwritten zip code recognition[8]. Using constraints, like that the digits are written in a specific place on the envelope and with fixed space between them, simplifies the problem, but the bigger problem remains of assigning the correct digit (0-9) to each written character. If the computer had a base of all handwritings in the world this would be a simple exercise, the program would only need to scan through all of the possible images and match the digit to the exact image. This of course is not realistic; assuming every literate person agrees to provide such a sample, the resulting data-base would be large enough to make querying it impossibly long, and of course there is no guarantee that one's handwriting stays constant over time.

Instead, the system should be able to "learn" to recognize handwriting, by mimicking humans' ability to deduce the right digit based on prior knowledge. This means that we do not provide the system set of rules to differentiate between digits, rather we give a set of examples with their correct labels, and let the system find the rules by itself. Thus, the system might return a different set of rules based on different example sets.

1.1.2 Supervised Learning:

The problem described above is part of a subfield of machine learning called *Supervised Learning*.[9] In these cases the algorithm has prior knowledge about the problem in the form of a training set. The training set consist of examples composed of both the *features* of the problem (each sample has one or more different features), and a label assigning each sample to the right class. This means that the system knows both the number of different classes, which can be two or more, and the correct assignment of each sample.

The training data is given to the learning algorithm which produces a *classifier*. A classifier is a function that maps between a new sample and a class. In some cases the classifier can also return an indecisive answer saying the result may not be determined in a good confidence level.

This means that after using the training set to build the classifier it is time to use a real data-set with an unknown classification, and compare the results from the classifier to the data-set’s samples.

Techniques like this are used in a variety of different fields; we have already mentioned the use in computer vision to recognize handwriting or computer prints. Another important application is to diagnose diseases; in this case the features might be different patient physical parameters and the label will be if he is sick or healthy. A more common case is to perform DNA tests on a tumor and based on those figure out if the tumor is malignant or not. Other applications may be found in speech recognition, on-line marketing etc.

1.1.3 Unsupervised Learning:

The key assumption in the supervised learning algorithms is that a known training set is present. Such a training set is not always present, and sometimes the prior knowledge is enforced by the expert and is not really manifested in the data. To tackle those problems the unsupervised learning techniques assume no labeling knowledge on the input data, and just try to find the natural groupings of the input patterns.

The result of the fact that there is no prior knowledge is that there is no need for a training set. Rather the algorithm unravels the underlying similarities and groups “similar” vectors together.

1.1.4 Clustering

Clustering is an unsupervised learning technique; it is the process of dividing a set of data into "natural" groups. The input to a clustering algorithm is a set of N data points x_i in *d* dimensional space. The output is a proposed classification of the data points into groups.

In some cases the number of desirable clusters is given as a constraint to the algorithm, in others there are parameters that affect the number of clusters and their character. Clustering algorithms can be either plain classification or hierarchical. In hierarchical algorithms the output is a dendogram which is a tree that describes the classification of the data into groups and the breakdown of these groups into further smaller groups.

Examples of well-known clustering algorithms are K-means[10], hierarchical clustering[11], SOM[12] and mean-shift clustering[13].

1.2 Purpose of research

In information age, when new data sets of various fields emerge in an overwhelming speed, techniques able to handle large data sets should be developed. We apply clustering techniques to large data, trying to find hidden details which might be exposed only with this kind of analysis. We also suggest a new step in the Quantum Clustering algorithm, to enable it to work on big data problems

1.3 Thesis organization

The rest of the thesis is organized as follows: chapter 2 describes the algorithms and formalism we use to analyze the data, singular value decomposition formalism[14], the support vector clustering[15], quantum clustering[5], approximate quantum clustering and dynamic quantum clustering[6] algorithms.

In chapter 3 we analyze a financial data set using our algorithms.

In chapter 4 we analyze the complete Israel seismic network catalog, containing all the earthquakes which took place in the region since 1990.

In chapter 5 we compare the quantum clustering algorithm[5] with the support vector clustering[15] one, and present the use of the approximate quantum clustering algorithm on real data.

Part of chapter 2 and chapter 4 are based on a manuscript that has been submitted for publication (Shaked, Weinstein, Hofstetter, Horn)

Chapter 2

Algorithms and related formalism

2.1 Singular Value Decomposition (SVD)[14]:

Our study concerns different types of $m \times n$ data matrices X with rank $k = \min(m, n)$. The equation for the singular value decomposition of X is as follows:

$$X = USV^T$$

Where S is a (non-square) diagonal matrix, and U, V are orthogonal matrices.

This can be re-written in a sum representation of k=min(m,n) unitary matrices of rank 1

$$X = \sum_{i}^{k} u_i s_i v_i^T$$

Ordering the non-zero elements of S in descending order, and taking only the first r values give us

$$X^r = US^rV^T = \sum_{i}^{r} u_i s_i v_i^T$$

Which is the best approximation of rank r to X, i.e. it leads to the minimal sum of square deviations

$$D = \sum_{i}^{m} \sum_{j}^{n} \left(X_{ij} - Y_{ij}\right)^2$$

Once SVD is applied to a given matrix X, two spaces dual to each other emerge.

The matrix U has orthogonal columns that serve as axes for representing the rows of U, while the matrix V^T has orthogonal columns that serve as axes for representing all rows of V. Truncating these representations to r dimensions leaves the truncated rows of U and the truncated columns of $V^T$with non-equal norm. This leads to many vectors accumulating near the origin, which then leads to problems in the clustering algorithm that is applied on these spaces. Therefore, we project each vector onto a unit sphere in r-space (each vector is rescaled to a unit vector in r-space)

2.2 Support Vector Clustering (SVC) [15]:

Support Vector Clustering (SVC) is a clustering method using the approach of support vector machine[16] (a classification approach). In the algorithm, data points are mapped from the data space to a high dimensional feature space using a Gaussian kernel. In this feature space, the smallest sphere enclosing the data is looked for. This sphere is then mapped back to the data space, forming a set of contours which are interpreted as cluster boundaries. As the width parameter of the Gaussian kernel is decreased, the number of disconnected contours in data space increases, leading to an increasing number of clusters. Outliers can be dealt with by using the soft margin approach. With this approach the sphere in the feature space is allowed to not enclose all data points, leaving only the cluster cores. In this way overlapping clusters can also be dealt with.

The calculation uses the SVM mechanism with the following soft margin constraint

$$\|\Phi(x_i) - a\|^2 \leq R^2 + \xi_i \ \forall i$$

Using the Gaussian kernel on the dual problem introduces the following Lagrangian

$$\tilde{\mathcal{L}} = \sum_{i=1}^{n} \left(e^{-q\|x_i - x_i\|^2}\right)^2 \beta_i - \sum_{i,j}^{n} \beta_i \beta_j e^{-q\|x_i - x_j\|^2} = 1 - \sum_{i,j}^{n} \beta_i \beta_j e^{-q\|x_i - x_j\|^2}$$

$$\text{subject to } 0 \leq \beta_i \leq C, \qquad \sum_{i=1}^{n} \beta_i = 1, \qquad \sum_{i=1}^{n} \beta_i y_i = a$$

Solving this set of equations –one derives the distance from any point to the hyper-sphere center

$$R^2 = \|\Phi(x) - a\|^2$$

Using the Gaussian kernel gives

$$R^2 = 1 - 2\sum_{i}^{n} \beta_i e^{-q\|x - x_j\|^2} + \sum_{i,j}^{n} \beta_i \beta_j e^{-q\|x_i - x_j\|^2}$$

Using any one of the support vectors gives the radius of this hyper-sphere.

So far there was no differentiation between points that belong to different clusters. A geometric approach involves the radius calculation for each point that is used. Given any two data points which belong to different clusters, any path that connects them must exit from the sphere in feature space. To calculate the relation between any two points, we sample the shortest path between them (around 20 points).

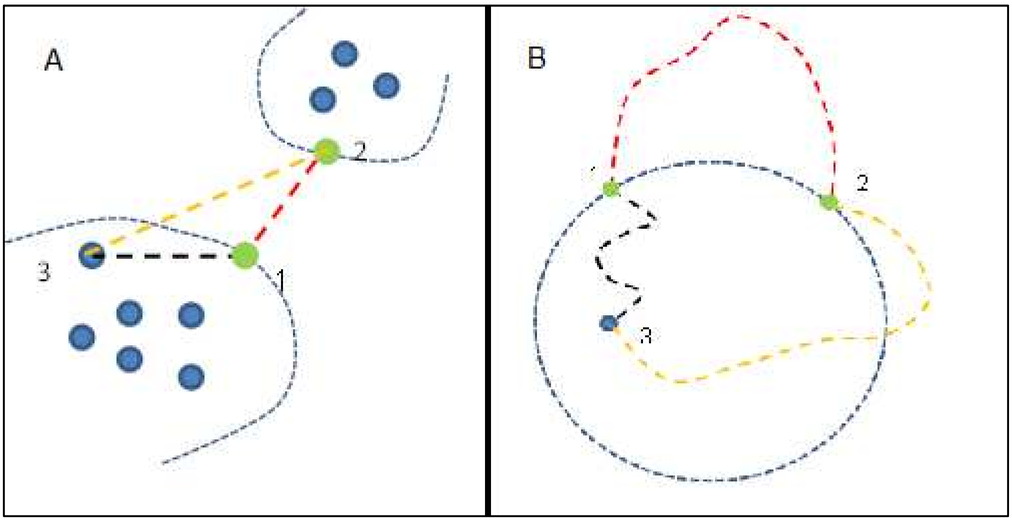


Figure 1 - A- points in data space, green represent support vectors, and shortest paths between them. B – the shortest paths in feature space. Paths connecting different clusters exit the hyper-sphere.

An adjacency matrix can be defined as

$$\mathrm{A}_{i,j} = \begin{cases} 1, & If\ for\ all\ y\ on\ the\ line\ segment\ connecting\ x_i\ and\ x_j\ R(y) \leq R \\ 0, & \text{Otherwise} \end{cases}$$

Clusters are now defined as the connected components of the graph induced by A.

The outliers are not classified by this procedure; after finding the cluster centers, they can be assigned to the closest cluster.

This approach of building the adjacency matrix gives the exact solution. But it becomes infeasible when dealing with "big data" problems (> ~10^4 data points), with a large number of dimensions.

For these situations, we develop a Heuristic SVC approach . We first consider a significant amount of points to be outliers. SVC is being used to separate the data into outliers and core points. The latter have to be grouped into ‘core clusters’ which should be quite separate from each other making. In order to avoid the costly adjacency matrix pairings of points, we use the advantage of large separations and employ the K-means[10] algorithm to the core points. Since there is no notion of the "real" number of clusters, we use a technique named Silhouette[17] which attempts to determine it.

The technique provides a succinct graphical representation of how well each object lies within its cluster.
Let us define a(i) = average Euclidian distance to all other nodes within the same core cluster.
b(i) = minimum d(i,Cj) - the minimum Euclidian distance to other clusters (the nearest cluster is chosen).
From these two numbers we can define:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Finally, averaging on all s(i) gives us the dissimilarity value for this choice of cluster numbers. Comparing s(i) to different cluster arrangements will give the best one.

Since we assume that the cluster centers are relatively distinct, the best silhouette score will be defined as the "correct" number of clusters.

The algorithm will be:

```
1) Use the SVC algorithm to get the core points
2) For k = 2 to N (the number of max clusters to check)
   2.1) run Kmeans with k as the number of clusters
   2.2) check the silhouette value of the Kmeans solution
3) Find the highest silhouette value and use this to define the core clusters
4) Go over all outliers o(i)
   4.1) find nearest core point c(i)
   4.2) assign o(i) to the same cluster as c(i)
```

2.3 Quantum Clustering (QC)[5]:

The main clustering algorithm we are going to use is the Quantum Clustering (QC) algorithm originally suggested by Horn and Gottlieb[5].

It starts by assigning a Gaussian, ψ_i, with width σ to each data points in the Euclidean SVD coordinates.

$$\Psi(\mathrm{x}_i) = \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{e}^{-\frac{(\mathrm{x}_i - \mathrm{x}_j)^2}{2\sigma^2}}$$

Then constructing the sum of the individual Gaussian function to obtain what is known as the Parzen window estimator[18].

$$\Psi = \sum\nolimits_i \Psi(x_i)$$

Finally define the potential function associated with the Parzen function to be

$$V = \frac{\sigma^2}{2} \frac{\nabla^2 \Psi}{\Psi} + E$$

where

$$E = -min \frac{\sigma^2}{2} \frac{\nabla^2 \Psi}{\Psi}$$

V has the unique property that it serves as the potential function of the Schrödinger equation

$$H\Psi = \left(-\frac{\sigma^2}{2} \nabla^2 + V(x) \right) \mathit{\Psi} = E\Psi$$

for which Ψ is the ground state. In this equation, the potential function V(x) can be regarded as the source of attraction, whereas the first Lagrangian term is the source of diffusion, governed by the parameter σ. (In the earthquake section a potential of this kind will be shown.). The QC looks for minima in V, since these correspond to regions where the density of the data is a local maximum, thus define cluster centers.

In Fig. 2 we present examples of two Gaussians centered at 0 and A and having width=1, and their corresponding potentials. It is clear that the potential can distinguish between overlapping Gaussians, and it intensifies the differences between them

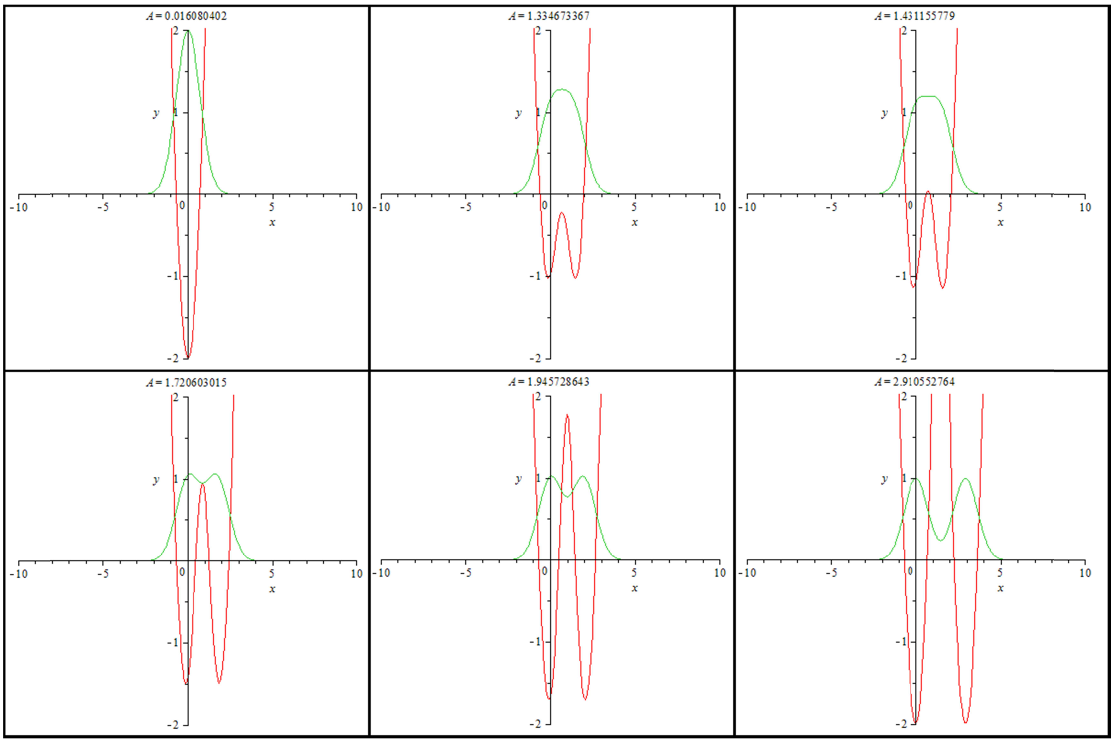


Figure 2 - a function of two Gaussians (green) and its potential (red). A represents the distance between them

σ is a parameter of QC which has to be chosen by the user to satisfy subjective criteria, such as limiting oneself to relatively small numbers of clusters

Once V is constructed we can use the gradient descent algorithm[19], where the power moving the data-points is the classical force that is given by $-\nabla V$. This leads the data-points to follow the dynamics of

$$y_i(t+\Delta t) = y_i(t) - \eta(t)\nabla V\big(y_i(t)\big)$$

Data-points that descend to the same minimum are declared to be in the same cluster.

The time complexity of calculating the potential function at a certain point is $\mathcal{O}(r \cdot N)$ where r is the number of truncated dimensions after the SVD, and N is the number of data-points, since the potential at each point is a function of all original data points. A complete step of the gradient descent is of order $\mathcal{O}(r \cdot N^2)$ because all of the moving points need to be processed. Therefore the complete calculation is of order

$$\mathcal{O}(m \cdot r \cdot N^2)$$

where m is the number of iterations of the gradient descent.

2.4 Approximate Quantum Clustering (AQC):

As the field of machine learning developed and tools to extract data sets improved, it is no longer rare to find data sets with more than 10^6 samples and features. As mentioned before with the help of SVD a reduction in feature space is possible but because the complexity of the QC algorithm is of order $\mathcal{O}(N^2)$, it is still infeasible to run on big data sets.

In order to improve the complexity we first need to analyze a QC step. In each step it is required to calculate the effect of each data-point on every other moving point. Since each data-point is represented by a Gaussian, all of the points taken together form an over-complete set. If we could find a smaller set of Gaussians with different coefficients that might form $\widetilde{\Psi}$ which will approximate Ψ, the complexity of the AQC will be of order $\mathcal{O}(c \cdot N)$, where c is the number of Gaussians in this reduced set.

To calculate this set we will employ the bra-ket notation

$$\Psi(\mathrm{x}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \sum_{\mathrm{i}=1}^{M} e^{-\frac{(\mathrm{x}-x_i)^2}{2\sigma^2}} = \sum_i \langle x|i\rangle$$

where $|i\rangle$ is the set of original Gaussians and d is the number of dimensions. We introduce another set $|\alpha\rangle$ which will serve as the approximate set.

We define the matrix N as

$$N_{\alpha\beta} = \langle\alpha|\beta\rangle$$

And the projection operator

$$\mathcal{P} = \sum_{\alpha\beta} |\alpha\rangle (N^{-1})_{\alpha\beta} \langle\beta|$$

This operator obeys $\mathcal{P}^2 = \mathcal{P}$ and projects the original set of vectors $|i\rangle$ onto the approximated set $|\alpha\rangle$

Now we use this projection operator to get $\widetilde{\Psi}$

$$\widetilde{\Psi} = \sum_{i} \langle x|\mathcal{P}|i\rangle = \sum_{i\alpha\beta} \langle x|\alpha\rangle\, (N^{-1})_{\alpha\beta} \langle \beta|i\rangle$$

With the definition of

$$C_{\alpha i} = \sum_{\beta} (N^{-1})_{\alpha\beta} \langle \beta|i\rangle$$

$$C_{\alpha} = \sum_{\iota} C_{\alpha i}$$

$\widetilde{\Psi}$ can be written as:

$$\widetilde{\Psi} = \sum_{\alpha} \langle x|\alpha\rangle\, C_{\alpha}$$

From $\widetilde{\Psi}$ we can calculate $\tilde{V}$as in the QC algorithm.

The coefficients C_α are calculated only once, as a precursor to the gradient descent phase.

Of course the choice of $|\alpha\rangle$ has crucial importance, since picking a set which does not span the data space correctly will harm the approximation. In order to get a reasonable choice of Gaussian base, the following heuristic is employed

1. Find min-max of all of the dimensions
2. Divide the space into voxels
3. Go over all voxels
   - 3.1. If there is one or more data points in the voxel take one
   - 3.2. Else do nothing

The size of the voxels is left for the user to choose, depending on how rough does he want the approximation to be, with respect to each dimension range and σ chosen. Usually, since SVD and renormalization is performed, each dimension is bound in the $[-1..1]$ range, therefore dividing each dimension into 10 voxels works reasonably well.

2.5 Dynamic Quantum Clustering (DQC)[6]:

Dynamic Quantum Clustering (DQC) is a method which is based on QC but replaces the gradient-descet dynamics with that of a time-dependent Schrödinger equation. It lets each ψi develop for a short while under the Schrödinger equation, and then constructs a new corresponding ψi and proceeds for many such steps using the original V, thus this dynamics may be regarded as the Schrödinger equation analog of gradient descent.

In this formalism each data point will be viewed as the expectation value of the position operator in a Gaussian wave function $\psi_i(\vec{x}) = Ce^{-(\vec{x}-\vec{x}_i)^2/2\sigma^2}$ where C is the appropriate normalization factor.

Thus the expectation value of the operator $\vec{x}$, is simply the coordinates of the original data point

$$\vec{x}_i = \langle\psi_i|\vec{x}|\psi_i\rangle = \int d\vec{x}\psi_i^*(\vec{x})\vec{x}\,\psi_i(\vec{x})$$

Now the time evaluation of each state $\psi_i(\vec{x})$ can be determined by the time-dependent

Schrödinger equation

$$i\frac{\partial\psi_i(\vec{x},t)}{\partial t} = H\psi_i(\vec{x},t) = \left[-\frac{\nabla^2}{2m} + V(\vec{x})\right]\psi_i(\vec{x},t)$$

where $V(\vec{x})$ is the potential function and m is an arbitrarily chosen mass parameter. It allows for tunneling between near-by valleys of V, thus connecting between data points in nearly degenerate potential minima.

Chapter 3[*]

Financial data

3.1 Data:

We analyze all 440 stock data of the Standard and Poor's (S & P) 500 list that were recorded daily throughout the period January 1st, 2000 – February 24th, 2011. The total number of active trading days was 2803. Notably this includes the crises of 2002 and of 2008

3.2 Analysis:

We use this data to examine various features of the QC algorithm and to demonstrate how we can extract information from data matrices.

We start by building the daily relative return matrix R (440X2803). Submitting R to SVD we truncate it to 10 dimensions, and project the data onto the unit sphere in the 10 dimensional SVD space. The QC algorithm, when applied on this matrix (440X10), results in 9 clusters. We will refer to these clusters as **sections**. Half of them have high overlaps with the 10 **sectors** into which these stocks are traditionally classified (one of the sectors – Diversified - has only one member).

Figure 3 displays the correspondence between the financial sectors (ordered sequentially on the x-axis) and the nine sections (on the y-axis). The ten sectors are: 1. Basic Materials, 2. Communications, 3.Consumer, Cyclical, 4. Consumer, Non-cyclical, 5. Energy, 6. Diversified, 7. Financial, 8. Industrial, 9. Technology, 10. Utilities.

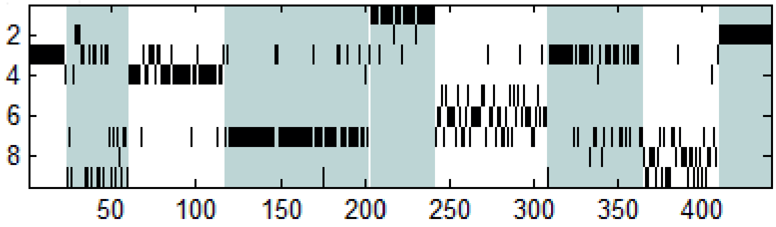


Figure 3 - Correspondence between stock classifications into nine financial sectors, ordered along the x-axis, and their QC clustering into nine clusters (sections). Each short vertical bar represents one stock

[*] Based on joint work with Erez Persi, Marvin Weinstein and David Horn (unpublished)

It can be seen that this kind of clustering finds that some sectors have high correlation between the stocks associate to them (like the Utilities or Financial sectors), but in some cases there exists mixing between sectors which leads to cross sectors correlations.

Performing a similar exercise on a matrix composed of daily data of weekly returns, we find, for the same sigma, only 6 clusters as presented in figure 4. Moreover, these clusters group some sectors together; in other words, the mixing of basic materials and industrials observed in cluster 3 of figure 3 becomes a much more spread phenomenon. Evidently this means that the behavior of daily returns, that characterizes some sectors, washes out by correlating related sectors within a few days.

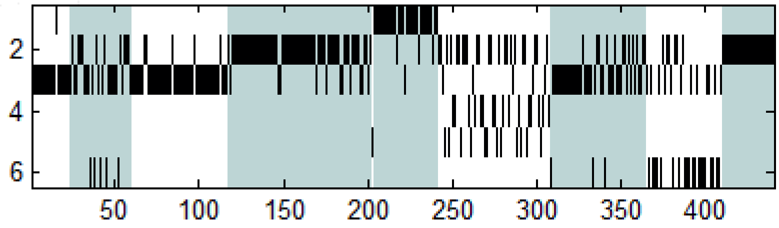


Figure 4 - Correspondence between stock classifications on a weekly return into six financial sectors

Next we try to look at this problem the other way around, trying to cluster the temporal domain into **epochs**. This requires considering the 2803 days as individual variables. Reducing once again R into ten-dimensional SVD space, and projecting the points onto a unit sphere, the DQC algorithm finds two major clusters that contain days from all along the temporal domain, and many other clusters with scarcer content. These results do not suggest any reasonable interpretation other than that R is a matrix of almost random fluctuations with zero average; hence it does not allow for simple clustering boundaries to appear in the 440-dimensional stock-space (or within its 10-dimensional SVD reduction).

In order to find temporal clusters we need to find a different representation which will not fluctuate too much. A suitable choice is the matrix P of daily stock prices (relative to the starting price on Jan 1st, 2000). Since the time series of each stock has relatively small fluctuations, it is possible for (close-by) time-points to exhibit similar vectors within the 440 stock-space, and thus fit into the same temporal cluster.

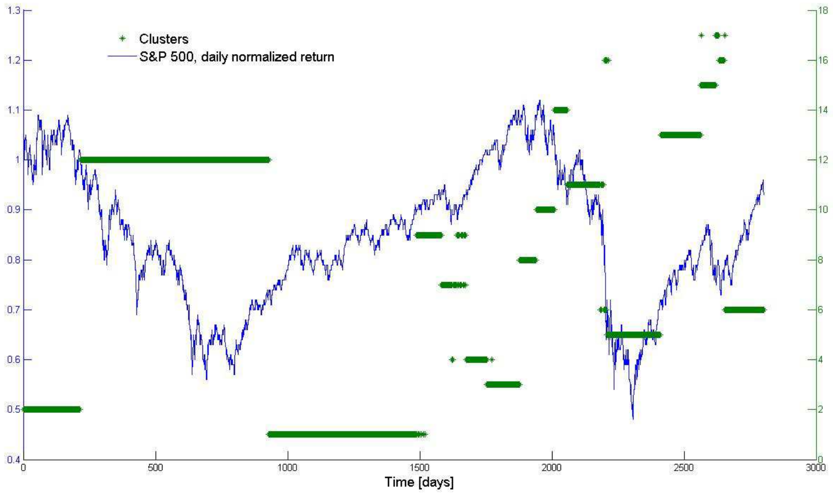


Figure5 - Temporal DQC clustering of the matrix P into 17 epochs, represented by bars. For comparison we plot the S&P 500 index for the same days, just to serve as an indicator of the known market behavior with its crises of 2002 and 2008.

The results, displayed in Fig. 5, show the existence of many clusters for the second half of the studied period (including the 2008 crisis), but only three epochs during the first five years (with only one covering the 2002 crisis). Each temporal cluster has its unique characteristics in stock prices. One way of displaying this property is by plotting the daily prices of stock averaged over different sectors. This is displayed in Fig. 6 on a 3-d plot spanned by three dominant sectors. The 17 epochs are distinguished by different colors.

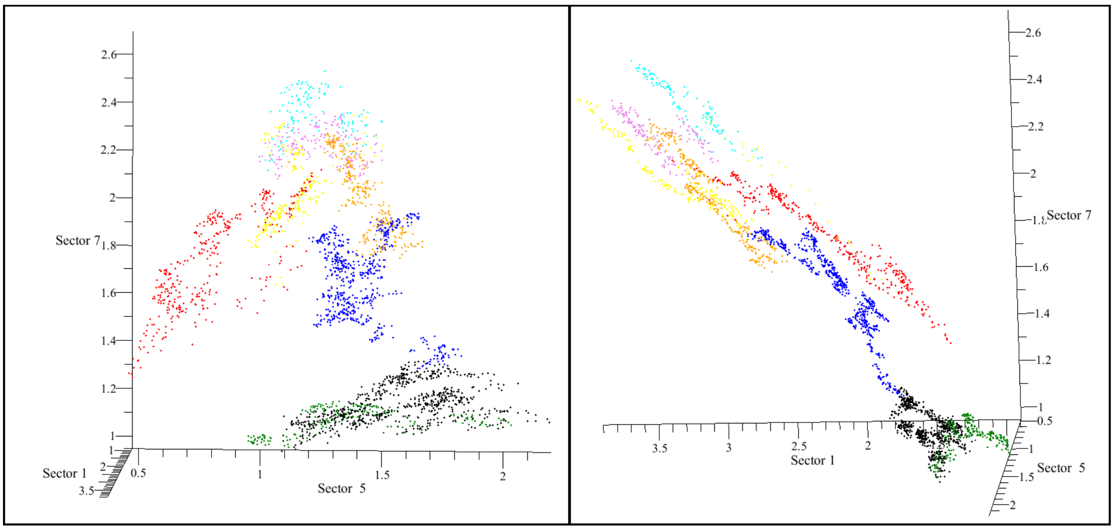


Figure 6 - Average daily prices for three sectors (1- Basic Materials, 5 - Energy, 7 - Financial) are plotted in a space spanned by these sectors in two different orientations. The data are seen to cluster into different epochs, distinguished by the different colors

An alternative is to investigate the Pearson correlations between the return values of stocks during each one of the 17 epochs. These are displayed in Fig. 7, with stocks arranged such that those that belong to the same sectors lie near each other. The ordering of the matrix plots corresponds to the order of appearance of the epochs. The first three epochs cover more than half of the temporal span, extending from 2000 to 2005. These epochs are characterized by significantly lower correlations among stocks of different sectors than during the following epochs. Each of the 17 epochs is characterized by a different correlations matrix, thus exhibiting its unique behavior.

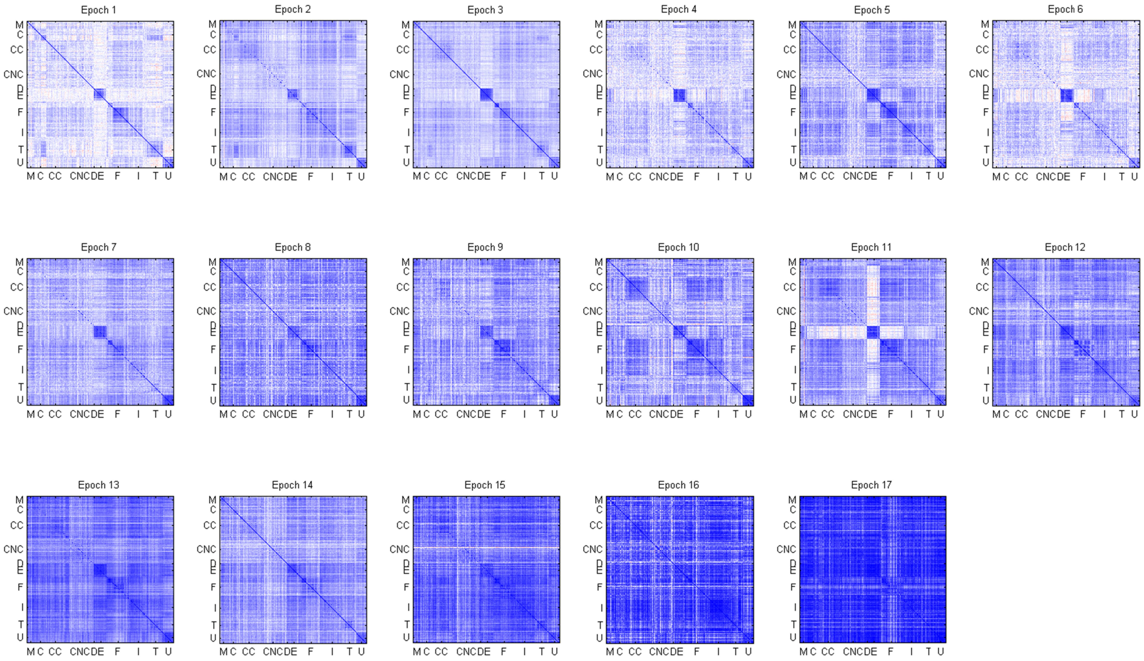


Figure 7 - Pearson correlations among all the different stocks for the each one of the 17 epochs. the order of the displayed by heat-map matrices (darker implies stronger correlations) corresponds to the temporal ordering of the epochs.

We can also see that some sectors exhibit higher correlations over all of the different epochs like the energy and utility sectors. There is no surprise that these two sectors are the ones best matched to the QC sections, since SVD also calculates the correlation between the different stocks.

3.3 Discussion

When studying complex systems one often tries to categorize their phenomena and cluster or classify their different dynamical variables. We have seen that our clustering approach helps us attain such goals. Using the daily return matrix we have obtained clustering of stocks, into what we called sections, and using the daily price matrix we have obtained temporal clustering into epochs. It should be emphasized that neither the sections nor the epochs should be viewed as clusters with rigid boundaries. Our clustering procedure depends on the density of data points, and clustering results depend on the parameter sigma that defines the neighborhood to which each point is sensitive. Nonetheless, the qualitative structure of these clusters remains the same even if the details of the boundaries may vary or clusters may merge with one another as sigma increases.

Clustering allows us to discuss market phenomena in terms of discrete categories, in analogy with market sectors (for stock characterization) or market periods characterized by increases or decreases of the index. Whereas the latter are quite common intuitive descriptions, clustering allows us to put them on a more objective mathematical footing.

In a recent paper, Munnix et al[20] proposed, using correlations between stocks, over short time periods (e.g. week or month), to define states of the market. In other words, different states will be associated with considerably different correlation matrices. Our proposal of clustering the temporal domain into epochs is an alternative: each epoch may be viewed as a state of the market realized within stock-space. In detail the two methods may disagree about the specific division into epochs, because they are based on different mathematical manipulations of the data. But both serve as alternatives for discretizing the complex system into categories that allow for a quantified analysis of the market structure.

Chapter 4[*]

Earthquakes data

4.1 Background:

The Dead Sea Fault separates the African-Sinai plate from the Arabian plate[21,22,23,24]. The Gulf of Aqaba, constituting the southern part of the Dead Sea Fault, is seismically active[25,26,27,28,29,30,31,32]. The strongest earthquake in the 20th century occurred in the gulf on November 22, 1995, at 04:15 GMT at 28.76^{0} N, $34.66\ ^{0}$ E, according to data from Cyprus, Egypt, Israel, Jordan, and Saudi Arabia. It measured $M_W = 7.2$ on the moment magnitude scale and was strongly felt in neighboring countries. The Gulf of Aqaba earthquake was followed by an intense swarm of aftershocks that reached well over 5,000 recorded events, which lasted for about one year, including several strong earthquakes with the largest magnitude $M_W = 5.6$ (on Feb. 26, 1996). The Eastern Mediterranean Region (EMR) is also seismically active, i.e. the Cypriot Arc with a major earthquake on Oct. 9, 1996. These major earthquakes, their following aftershocks and the activity in between the main sequences are listed in the catalog of the Israel Seismic Network. The general activity in between major earthquakes has on average about one event per day over the period of the last 30 years somewhere in the EMR, if we do not take into account the aftershock sequences that are characterized by outstanding intense activity. These events are described not only in terms of magnitude and geographic location of the earthquake, but also in terms of various geophysical and seismic parameters, these parameters are described in the catalog of the Israel Seismic Network. This catalog includes all earthquakes with $M_d > 2$ with a confidence level of 90%. It is based on information collected from stations in Israel and neighboring countries. The general activity in between major earthquakes has on average about one event per day over the period of the last 30 years somewhere in the EMR, if we do not take into account the aftershock sequences that are characterized by outstanding intense activity. These events are described not only in terms of magnitude and geographic location of the earthquake, but also in

[*] Based on a manuscript by Guy Shaked, Marvin Weinstein, Rami Hofstetter and David Horn, submitted for publication

terms of various geophysical and seismic parameters. Absence of such parameters in the catalog is in no way correlated with the characteristics of the event.

Out of this catalog we have analyzed 5,693 full earthquake records (out of 18850 recorded earthquakes) that contain, besides the location and time of occurrence, also the following features describing each event: M_d - the coda duration magnitude of the earthquake, M_0 - seismic moment, stress drop, source radius and f_0 - corner frequency. Thus we attempt to cluster earthquakes in terms of their geophysical properties alone

4.2 Earthquake data

Several authors have inverted the teleseismic body-wave seismograms to obtain the seismic moment and the source mechanism of the main shock, which is found to be in agreement with InSAR observations[26,30,32] *Hofstetter et al*[27]. have also inverted 57 moderate to strong aftershocks, using the method of[33], which were clearly observed by the regional broad band stations BGIO, JER, and KEG. In the case of small earthquakes, usually *Md* < 4.0, the seismic moment, M_0, corner frequency, f_0, stress drop, and source radius, r_0 are determined based on the dislocation model of *Brune*[34,35] using the spectra of S-waves recorded by short period stations (three-components or vertical component). Details of the application of the method to the Israel Seismic Network are presented in *Shapira et al*[36].

4.3 QC analysis

Since in this case the dimensions of the problem (5 features) are already small we will not reduce the number of SVD dimensions for the QC algorithm, so the input matrix will be of size 5693×5.

Fig. 8 presents the data points in the first 3 dimensions of SVD

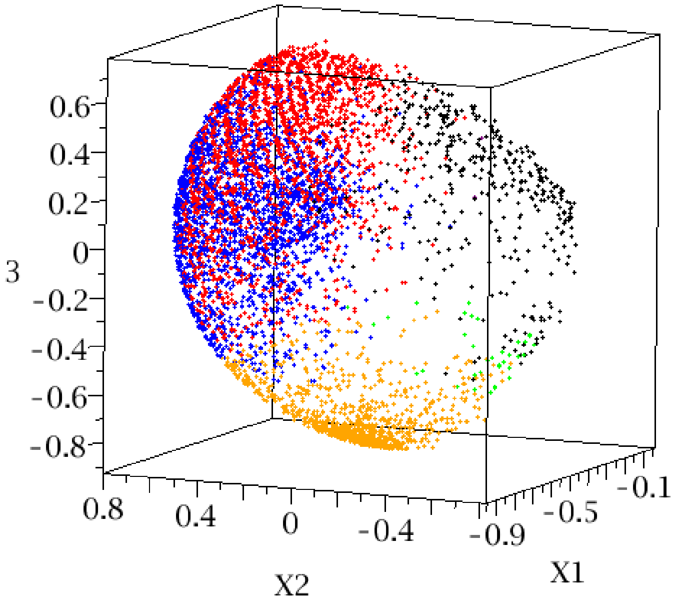


Figure 8 - first 3 dimensions of the SVD matrix. The different colors represent the different clusters that are identified at the end of the gradient-descent process with σ = 0.3

As we have seen in the QC algorithm explanation, we can think of the potential like a topographic map. In Fig. 9, we illustrate the potential topographic map in the original feature space. Here we draw the potential values for points belonging to different clusters displayed for the original feature M_d and F_0.

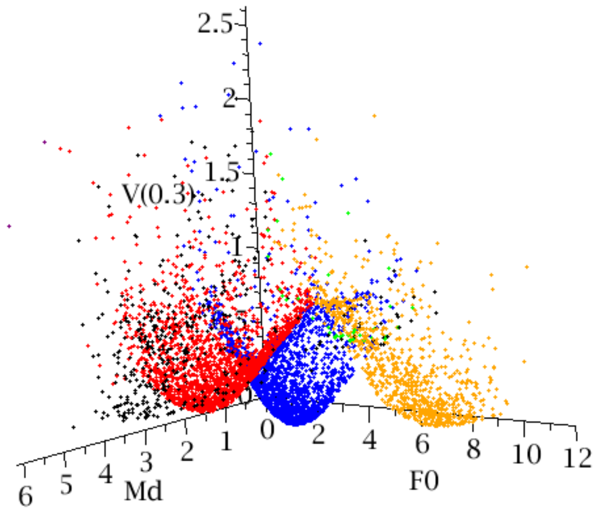


Figure 9 - The topographic plot of the potential calculation. Each point is drawn in the real Md, F0 space with its potential value on the Z-axis. The colors represent the different clusters that are identified at the end of the gradient-descent process with σ = 0.3

The different minima are clearly visible, and the different cluster colors suggest that the gradient descent algorithm does in fact move the data points to their closest point of attraction.

Since the potential is a scalar, it cannot be projected on a sub-space for better visualization and understanding. For this reason we will use "force" induced by the potential or simply $-\nabla V$. This vector can be projected on different SVD dimensions, so we can see the direction of convergence in all dimensions as seen in figure 10.

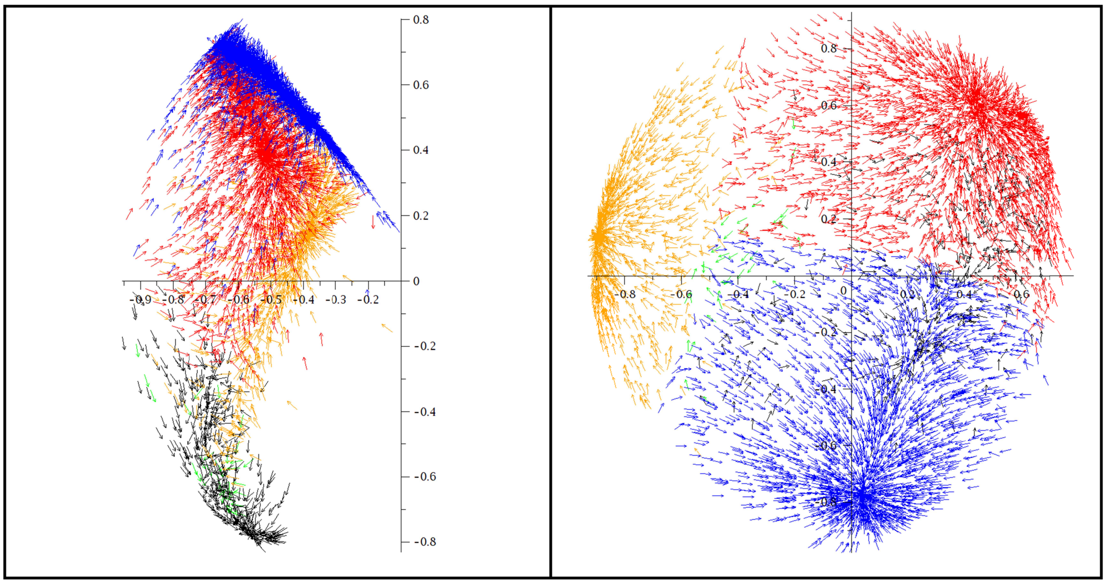


Figure 10 - Unit vectors, designating the directions of the force derived from the potential in SVD space, exhibit four clear centers of attraction. A(left): PC=1 and 2. B(right): PC=3 and 4. The QC parameter was chosen as σ=0.3. The arrows are colored according to the clusters that are identified at the end of the gradient-descent process.

The dynamics of gradient descent make all points converge onto cluster centers at the various minima of the potential; the number and identity of these clusters depend on the value of the parameter σ. For σ = 0.3 we obtain five major clusters, which we designate by colors red (2,116 events), blue (2199), orange (951), green (36) and black (389), and one additional small cluster containing 2 strong earthquakes.

Once we have carried out clustering we can return to the original features of the data, in order to get some understanding on how the different clusters can be characterized. In Fig. 11 we display our results within a 3-dimensional parameter space spanned by M_d, f_0 and stress drop. Clearly the red cluster contains events with large magnitude and low to medium stress drop, while the black cluster contains events with similar large magnitude but much larger stress-drop. The blue cluster contains events with medium magnitude, medium corner frequency and low stress drop. The orange cluster contains earthquakes of small magnitude, large corner frequency and low stress drop, while the green cluster has small magnitude, large corner frequency and high stress drop. The central values of the parameters for all clusters are presented in Table 1.

It is quite evident that the boundaries between the clusters are not due to clear separation between the corresponding events. Rather, they are due to differences in the weights of the distributions and to the choice of the parameter σ. The same is, of course, true of the boundaries between the different color fields in SVD space, as shown in Figure 11.

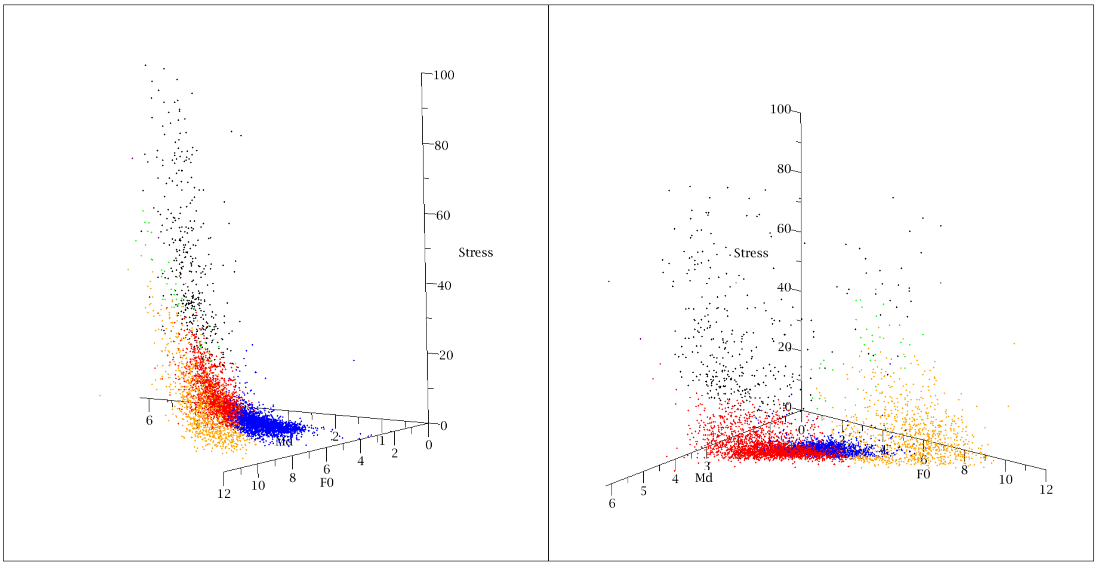


Figure 11 - Two perspectives of a three-dimensional parameter space spanned by Md, f0 and stress drop

clusters	Md	Stress drop	f_0	log M_0	Source radius	M_w
		bar	Hz	dyne-cm	km	
2199 Blue	2.0	2	4.04	19.00	0.33	2.0
2116 Red	3.3	7	3.36	19.86	0.41	2.5
951 Orange	1.8	12	8.19	18.90	0.16	1.9
389 Black	3.9	50	3.25	20.86	0.45	3.2
36 Green	2.8	44	7.27	19.71	0.18	2.4
2	5.9	58	0.40	24.89	10.97	5.9

Table 1: cluster centers for QC σ=0.3 in feature space.

4.4 Geographic and temporal characterization.

Once we have obtained clusters and interpreted their meanings in terms of earthquake features, we look at the correlations of these clusters with their recorded locations and times of occurrence.

The distribution of the various clusters of earthquakes along the Dead Sea fault and the Gulf of Aqaba is plotted in Fig. 12 Some general consequences are that the smaller earthquakes clearly trace the Dead Sea fault, and most strong earthquakes, belonging to the red and black clusters, occur in the Gulf of Aqaba. Of particular interest are the orange and green clusters, since they are mainly concentrated in the Gulf of Aqaba.

The amazing observation that the two clusters are concentrated in a single region is compounded by the additional observation that they are also localized in time: *i.e.*, most of their events followed the major Gulf of Aqaba earthquake of 1995, during several months. The temporal distribution of earthquakes is displayed in Fig. 13A, with a fine-grained expansion in Fig. 13B.

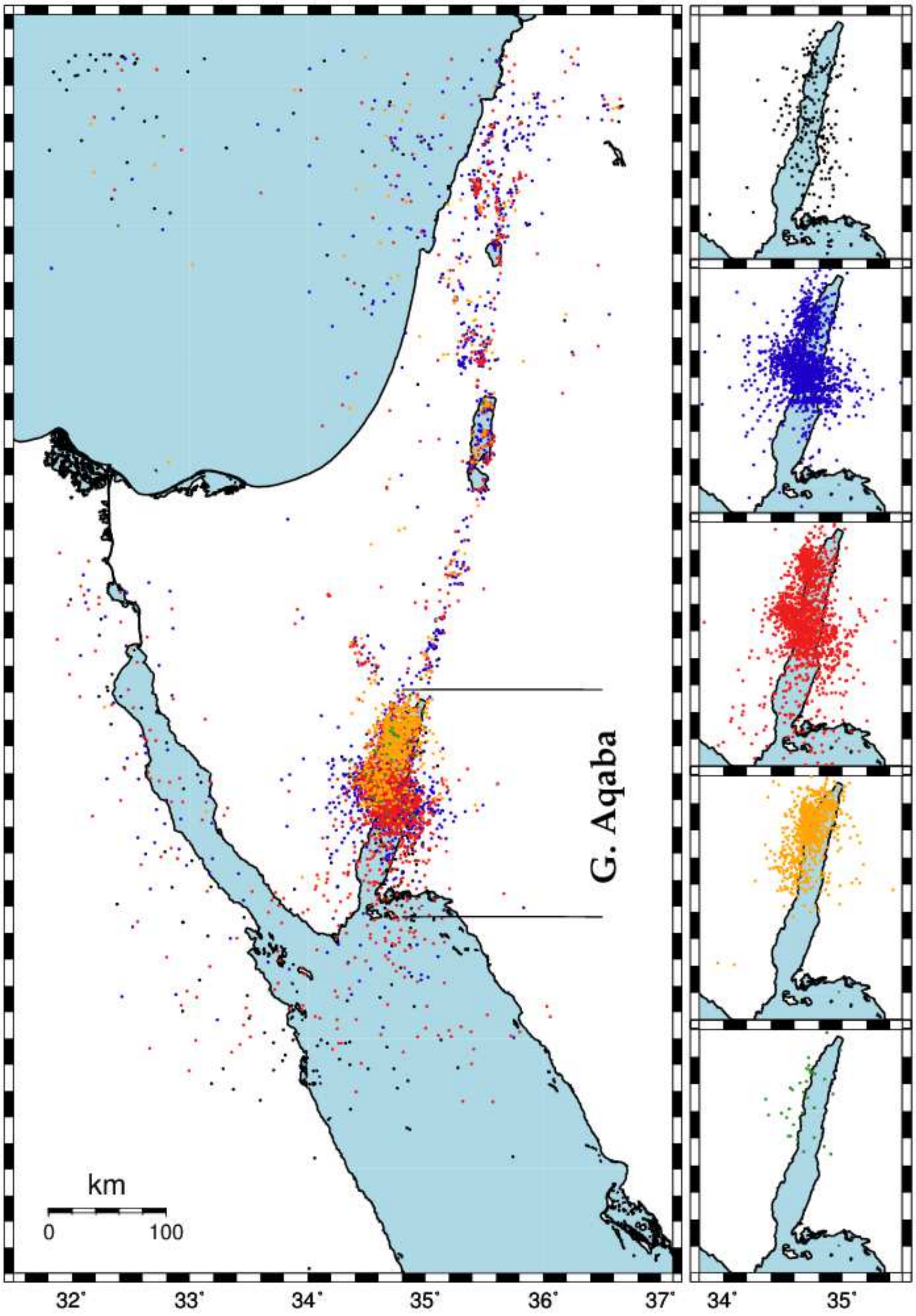


Figure 12 - earthquake events, classified by our clustering colors. The region of the Gulf of Aqaba is marked on the regional map on the left side. The "cloud" in the gulf is composed of five clusters (black, blue, red, orange and green) that are shown on the right side[37].

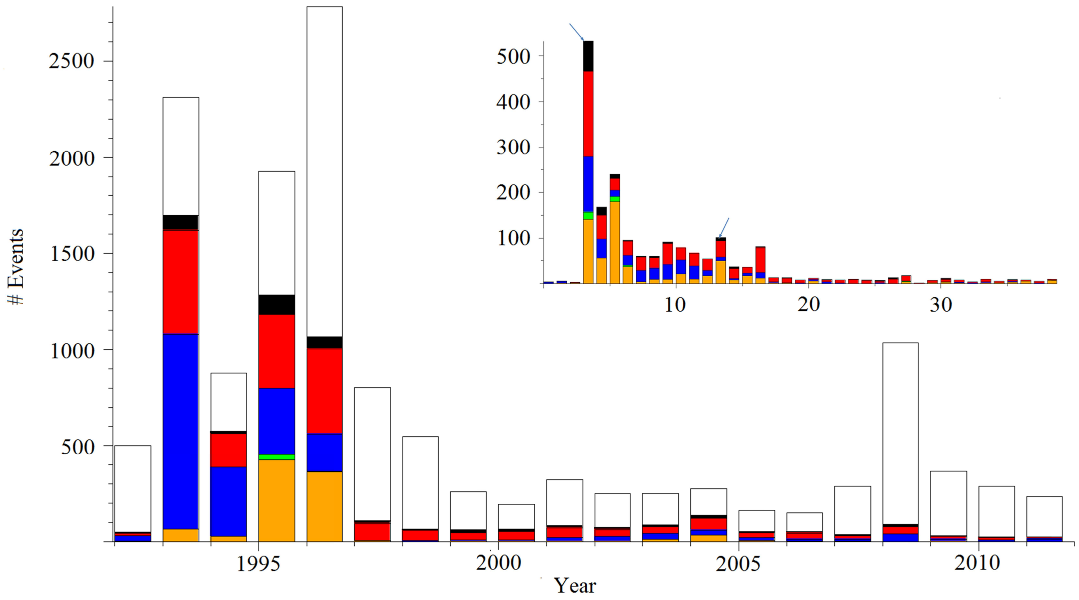


Figure 13 - A. Yearly histogram of earthquakes throughout the period of 1992-2011. Colors correspond to earthquakes belonging to the different clusters. Open columns stand for registered earthquakes that lack all or some of the features that we have employed in our analysis. B (inset). Weekly histogram of fully recorded events covering the period Nov 1995 – June 1996. The major earthquake event of Nov. 22nd 1995, with magnitude $M_W = 7.2$ is marked by an arrow. A second arrow designates another earthquake of large magnitude ($M_W = 5.6$) that occurred in Feb. 1996.

4.5 Interpretation of the orange cluster

The events we have characterized as belonging to the orange cluster are particularly abundant following the November 1995 major earthquake whose center was in the Gulf of Aqaba. Their occurrences started in conjunction with this major quake, and continued with a temporal signature that shows ups and downs which is quite different from simple decline. Many orange events continue to show up until the end of February 1996, and they are all concentrated in the Eilat and Aragonese basins[28,27]. The green cluster is confined mostly to the first and third week following the major 1995 earthquake.

The natural interpretation of the orange and green events is that they represent ruptures that have occurred following the major earthquake. It is interesting to note that the analysis of events in the relevant period carried out by *Baer et al.*[38], reproduced observed slip distributions which were quite unique to just this period.

4.6 Comparative analysis of the 1993 and 1995 data

Given the interpretation of the orange cluster it is interesting to reanalyze the geographic distribution of the different clusters separately for the 1993 events (the eight months following the major earthquake that triggered them) and the 1995 events (the eight months following the major 1995 earthquake). Details are shown in Fig. 14.

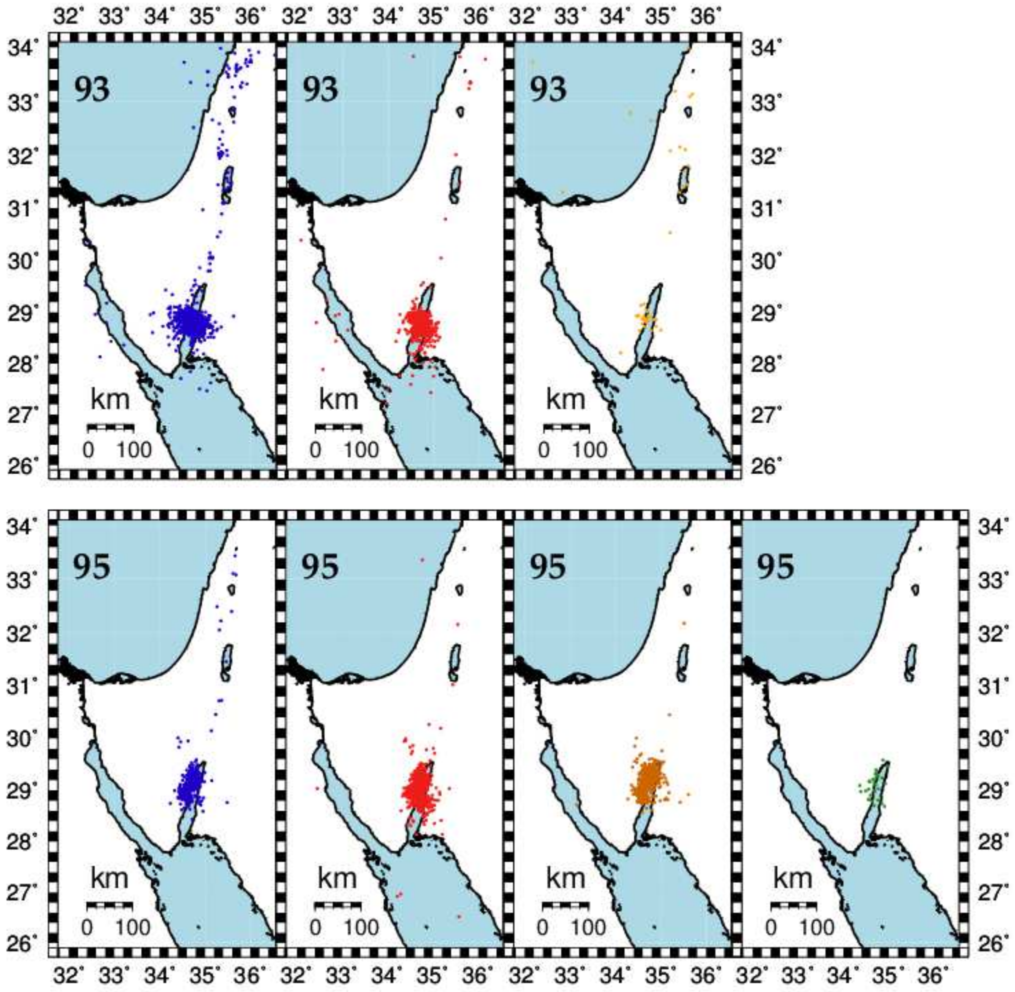


Figure 14 - Geographic distributions of blue, red and orange and green events following the major earthquakes of 1993 and 1995. All available data during the 8 months following the major earthquakes are displayed. Note that there are only a few orange events and no green ones after the 1993 earthquake. Most orange events and all the green ones occur after the major earthquake of Nov 22nd, 1995[37].

We observe that there are various differences between these groups, other than the scarcity of orange events in 1993. First, the 1993 activity clearly took place in a region lying to the south of where the 1995 activity occurred. Second, the relative shape of the blue cluster of events in 1993 is globular whereas that of 1995 is elongated along the Gulf of Aqaba. Third, the green events occur only in the Gulf of Aqaba (and only following the 1995 earthquake). Finally, we find that there are many more blue aftershocks along the Dead Sea Fault in 1993 than in 1995, in spite of the fact that the latter had a stronger trigger. Note that there are many more blue events than red and orange ones along the Dead Sea Fault in both eight months periods of 1993 and 1995. In fact we may conclude from it that the basin of the blue cluster in parameter space, as depicted in Fig. 14 is the natural major characteristic of the frequent weak earthquakes occurring along the Dead Sea Fault.

4.7 Inter Quake Intervals

Next, we examine the temporal characteristics of the earthquake occurrences for each of the five clusters. The different types of earthquakes have different rates of occurrences. Measuring the inter-earthquake-time intervals (IQI) we find that the mean interval for black events is 1.5×10^6sec, while for the blue, red and orange events the mean IQI values are 3, 3.1 and 6.2×10^5 sec respectively. The group of green events has the largest average IQI of 1×10^7sec.

IQI of earthquakes have been shown to follow log-normal distributions. *Lomnitz*[39] has provided a model leading to such distributions for major earthquakes. A comparison with a Brownian model of recurrent earthquakes has been provided by *Matthews et al.*[40].

We find the log-normal to adequately describe some of the IQI distributions. The best example of a log-normal distribution is the IQI distribution for all blue events, displayed in Fig. 15-right. In contrast, the distribution of the black events (Fig. 15-left) is quite skewed. The two events that led to the single entry of the left tail in the black distribution have occurred south of Cyprus in October 1996 and will be discussed in section 4.10, their details are given in Table 8.

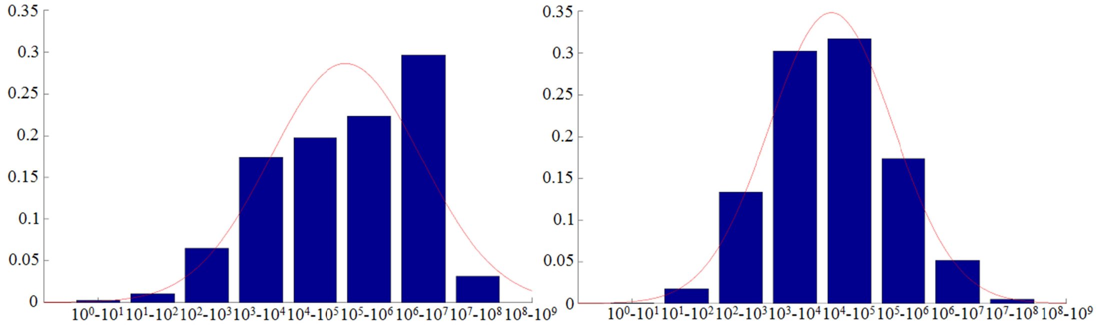


Figure 15 - IQI distributions of all blue events (right) and all black events (left) since 1990 on a logarithmic time-difference scale. The plotted curve is a Gaussian with the correct average and variance. The blue distribution is very close to log-normal, whereas the black distribution is quite skewed.

Next we turn to IQI statistics where time-intervals of events of one kind (e.g. orange) are measured with respect to the closest event of a different kind (e.g. red or black). The most interesting correlations that we have found are displayed in Fig. 16. Whereas the IQI of orange events are quite log-normal, when triggered by the most recent red event, their own distribution is shifted to the left, and when triggered by the most recent black event, the distribution is shifted to the right. The orange triggered on red distribution implies that an orange earthquake is more likely to appear within the first thousand seconds of a red one, which indicate a weak causal relationship. The orange triggered on black distribution is shifted to larger temporal windows because the time span between adjacent black events (1.5×10^6 sec) is an order of magnitude higher than the average orange rate.

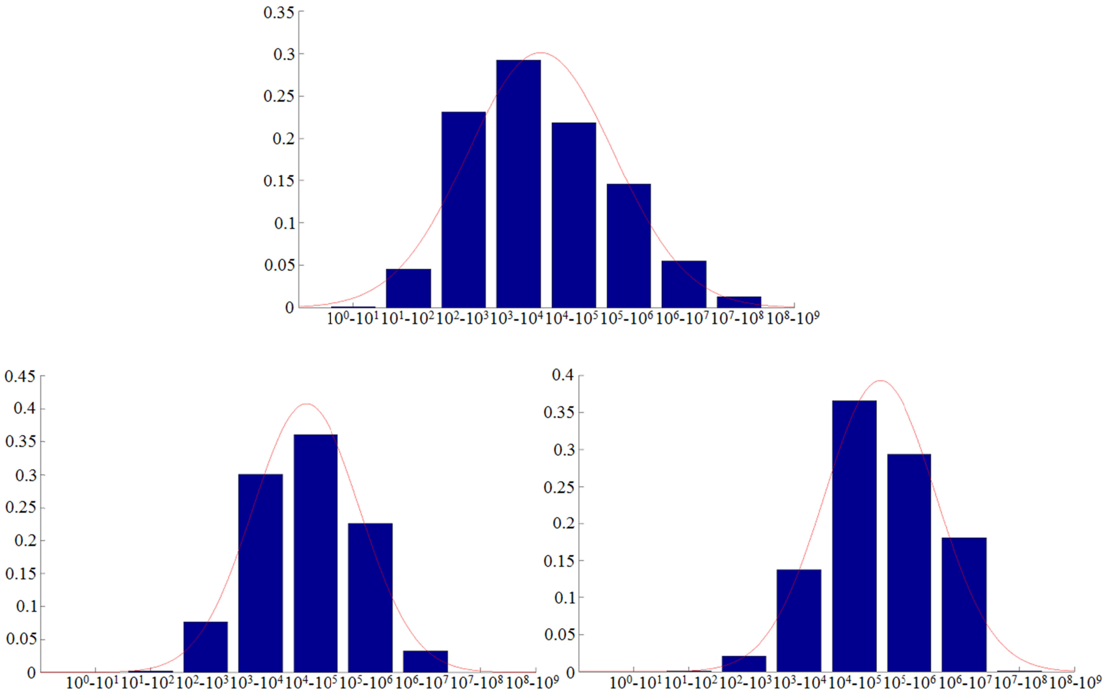


Figure 16 - IQI distributions of all orange events (top middle) on a logarithmic time-difference scale. Down left: orange events with IQI triggered on the closest red event. Down right: orange events with IQI triggered on the closest black event.

Another temporal aspect of earthquakes that is worth noting is the frequency of aftershock occurrences as function of the time elapsed since the major earthquake that has triggered the seismic activity. Omori[41] has proposed that they drop like a power law with an exponent near -1, and Christnesen *et al*[42]. have argued that such behavior is consistent with self-organized criticality of earthquakes. In Fig. 17 we show results of our analysis for aftershocks following the earthquake of Nov. 22nd, 1995, for a period of 8 months. For comparison see also Fig. 13 and its inset. The data in Fig. 17 include all recorded aftershocks, i.e. also those for which not all seismological parameters are recorded, demonstrating a behavior of $T^{-0.5}$ setting in at 10^4 sec and continuing until 10^7 sec after the major earthquake. The frequencies of aftershocks occurring within the different clusters follow suit with their own decreasing behavior, mostly displaying clear power decline after 10^4 sec. The orange cluster displays different behavior, with frequencies that stay at the same level for about two temporal decades.

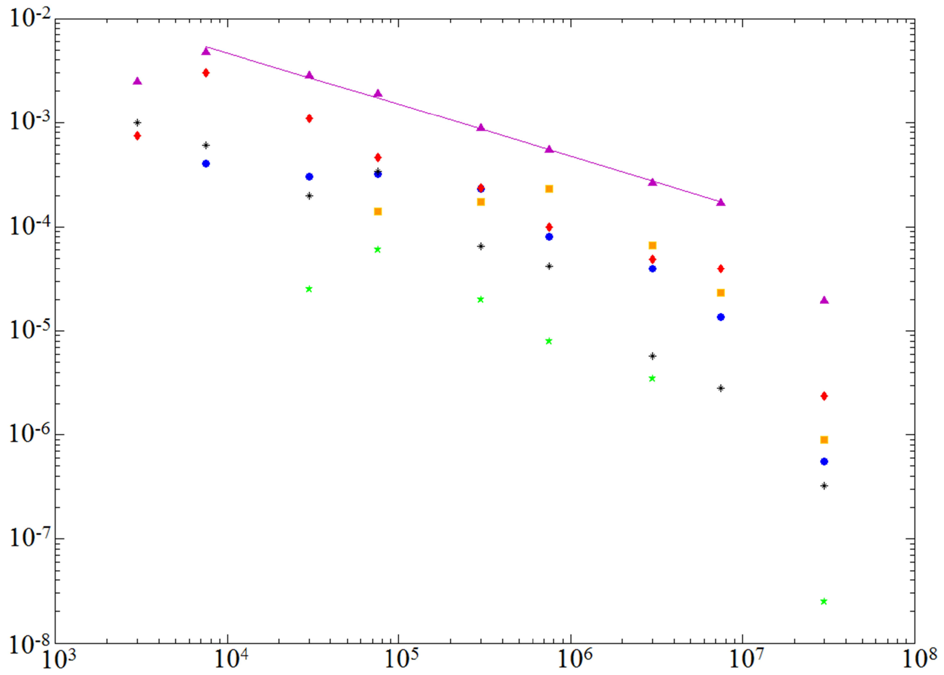


Figure 17: Frequencies of aftershock occurrences as function of time (in sec) elapsed since the major earthquake of Nov. 22nd, 1995. Top data (purple triangles) include all recorded aftershocks, and are well fit by a behavior of $T^{-0.5}$. Other colors represent aftershocks that are classified into our five clusters.

4.8 Stability and variability of clustering assignments.

QC has one parameter, σ, whose value determines the pattern of clustering, i.e. the number of clusters and the association of each instance with one of the clusters. In our analysis we have varied σ from 0.2 to 0.5. The numbers of clusters have varied accordingly from 8 to 4. The clustering patterns we have presented so far belong to σ=0.3, containing four major clusters, a minor one (green) containing 36 events, and one minute cluster containing 2 events. The major four clusters dominate the scene for all σ from 0.2 to 0.5, but as we reduce σ we increase the number of smaller clusters. We illustrate this phenomenon in Table 2, where we compare the clustering patterns of σ=0.25 and σ=0.3. Interestingly σ=0.25 has three minute clusters, two of which are singletons and one containing 4 events, but its green cluster is considerably larger than that of σ=0.3, acquiring additional events from orange and from black clusters in the latter.

0.25\0.3	2199 Blue	2116 Red	951 Orange	389 Black	36 Green	2
2302 Blue	2192	88	22	0	0	0
1989 Red	0	1974	15	0	0	0
865 Orange	1	1	863	0	0	0
412 Black	6	49	0	356	1	0
119 Green	0	0	51	33	35	0
4	0	3	0	0	0	1
1*	0	0	0	0	0	1
1	0		0	0	0	0

Table2 - comparison between QC clustering assignments of σ=0.25 and σ=0.5. The singleton labeled with * is the strongest recorded earthquake in the data (Nov 22nd, 1995). Small clusters are labeled just by the numbers of their corresponding events.

Table 3 contains the values of the cluster centers for σ=0.25, and should be compared with the analog information for σ=0.3 which is contained in Table 1. We note that the centers of all large clusters for both σ values are very close to each other. The 4-cluster includes large-magnitude earthquakes, 3 from the red cluster of 0.3, and one from the 2-cluster of 0.3. The other earthquake of the 2-cluster is the major earthquakes of Nov. 22nd, 1995, which is classified as a singleton in σ=0.25.

0.25 cluster centers	Md	Stress drop	f_0 Hz	log M_0 dyne-cm	Source radius km	M_w
2302 Blue	2.1	2	4.05	19.01	0.33	2.0
1989 Red	3.3	6	3.37	19.86	0.41	2.5
865 Orange	1.8	11	8.32	18.87	0.16	1.9
412 Black	4.0	45	2.88	20.90	0.47	3.2
119 Green	2.6	45	7.55	19.67	0.17	2.4
4	5.1	36	0.78	22.48	1.57	4.3
1*	6.2	69	0.17	26.86	20.00	7.2
1	3.6	3	0.90	21.40	1.56	3.6

Table 3 - cluster centers for QC σ=0.25 in feature space. The singleton labeled by * is the strongest recorded earthquake in the dataset.

4.8.1 The small σ=0.25 clusters

As we have seen changing the QC parameters, like lowering σ, one can pick smaller clusters out of otherwise larger clusters, thus finding particular groups that might be of major interest. The next tables (4-5) puts focus on some aspects which can be identified from these small clusters

YearMoDyHrMn	Md	Lat. (^0N)	Lon. (^0E)	log Mo dyne-cm	Stress drop bar	Source radius km	fo Hz
199511232228	4.6	28.658	34.908	22.01	27	1.18	0.97
199703260422	5.5	33.864	35.391	22.91	47	1.95	0.62
199703261320	5.2	33.703	35.565	22.52	33	1.63	0.75
200004060637	5	28.802	34.824	22.49	37	1.53	0.79

Table 4: Data of the 4-cluster of the σ=0.25 clustering scheme

YearMoDyHrMn	Md	Lat. (^0N)	Lon. (^0E)	log Mo dyne-cm	Stress drop	Source radius km	fo Hz
199511220415	6.2	28.762	34.682	26.86	69	20	0.17
200202240956	3.6	32.069	35.469	21.4	3	1.56	0.9

Table 5 : Data of singletons of the σ=0.25 clustering scheme

As it can be seen the cluster of four is identified by high magnitude values, also it's all of the 6 events describe here have large source radius values in comparison to the rest of the data suggesting why these were picked up by finer clustering approach.

4.9 Comparison to DQC

The DQC algorithm is based on the same potential function V as in QC, but its convergence pattern of instances into clusters are somewhat different, since it is based on a quantum analog of the gradient descent algorithm. It has two parameters: the Gaussian width σ and a quantum tunneling parameter m. Table 6 shows a comparison between the clustering assignment obtained from DQC with the parameters σ=0.25 and m=0.35 and the QC clustering we have used throughout the paper. The four major clusters remain more or less intact, but for the black QC one which splits into black and red in DQC. The QC Green cluster merges into the DQC orange one, and a new DQC cluster of size 255, labeled cyan, turns out to be composed of some of the red and some of the blue events in the QC clustering. The central parameter values of the DQC clustering are presented in Table 7, which may be compared with Table 1, where we have shown the analogous results for the QC clusters. Note that the DQC orange cluster is mostly composed of the QC orange and green clusters. The events in the DQC orange cluster occur in the same location of the Gulf of Aqaba in Fig. 12. Hence, our observation of its unique interpretation remains the same.

DQC\QC	2199 Blue	951 Orange	2116 Red	389 Black	36 Green	2
2052 Blue	2046	5	1	0	0	0
1142 Orange	73	946	77	10	36	0
2012 Red	2	0	1842	168	0	0
229 Black	11	0	7	211	0	0
255 Cyan	67	0	187	0	0	1
1*	0	0	0	0	0	1
2	0	0	2	0	0	0

Table 6: : comparison of a DQC clustering result, using σ=0.25 and m=0.35, with our standard QC of σ=0.3.

Clusters	Md	Stress drop bar	f_0 Hz	log M_0 dyne-cm	Source radius km	M_w
2052 Blue	2.0	2	4.05	18.97	0.32	1.9
1142 Orange	1.9	13	7.78	18.95	0.17	1.9
2012 Red	3.4	9	3.33	19.96	0.41	2.6
229 Black	3.8	62	3.46	20.85	0.44	3.2
255 Cyan	3.1	4	2.49	19.96	0.54	2.6
1*	6.2	69	0.17	26.86	20.00	7.2
2	2.8	12	4.51	19.75	0.27	2.5

Table 7: cluster center values of the DQC result in parameter space. 1* is the major earthquake of Nov 22nd, 1995.

4.10 Geographic clusters

Some of outstanding events in the clustering in feature space are also clustered temporally and geographically, thus related to the same faults. An example is shown in Table 8 All these events occurred south of Cyprus during and shortly after the strong earthquake on Oct. 9, 1996. All five events are black earthquakes. The first two occurred within a time-difference of 2 minutes. The following three events have very similar characteristics to each other, and all have occurred within four days. All may well have belonged to one particular geographic cluster of earthquake activities within the same fault structure.

YearMoDyHrMn	M_d	Lat. (^{0}N)	Lon (^{0}E)	log M_0 dyne-cm	Stress drop bar	Source radius km	f_0 Hz
199610091346	3.8	34.205	32.421	21.57	365.29	0.352	3.21
199610091348	4.6	34.095	31.843	22.28	830.88	0.463	2.37
199610202114	3.1	34.153	32.529	22	552.78	0.425	2.74
199610231029	3	34.072	32.048	22.08	543.49	0.456	2.47
199610240721	3.3	34.108	32.137	21.93	279.1	0.505	2.19

Table 8 - A group of outstanding black events that form a geographic cluster of earthquake activities that occurred in October 1996 south of Cyprus

Another example of a geographic cluster is given by two black events (M_d =5.5 and 5.2) which belong to the 4-cluster of σ=0.2 and are listed as the second and third entries of Table 4. They occurred within 9 hours of each other in 1997 on the shore of Lebanon and were followed within a few days by several aftershocks at the same location. These aftershocks were of smaller magnitudes and were not fully analyzed; hence they were not included in our data.

4.11 Summary

The discovery of a special cluster of events, characterized by low magnitude and high stress-drop and well defined in localization and time, corresponds nicely to the ruptures that have occurred following the major earthquake. The latter have been analyzed in the past with special attention to modeling the faults that have been involved[32]. We have been able to put these events into a different context, pointing out the fact that, by their association with a particular range of parameters, they define a new class of events, different in character from all other earthquake events. Had we had at our disposal a larger catalog of earthquakes from different regions of the globe, we could find out to what degree this particular class of ruptures occurs world-wide, and analyze their correlation with major earthquakes that they follow.

Chapter 5

Comparison between clustering algorithms

5.1 Comparison between SVC[15] and QC[5]

The SVC algorithm takes a more traditional way to cluster data, using the SVM method with Gaussian kernel. Since both SVC and QC methods start by using the transformation between data points to Gaussians, it is interesting to try and compare the two. For this purpose we will continue to work with the earthquake problem, and see how these two methods match.

As done for the QC, we start with applying SVD and normalizing the results on the unit 5-dimensional sphere. Running SVC with q=10 and p=0.7 (see appendix A), returns 3 core clusters as can be seen in Fig. 18 (right). The three cores are used to construct three clusters of the total data according to our method explained in 2.2. Comparing this clustering assignment to the result of the QC algorithm shows overall agreement between the two assignments. The SVC red blue and black map into the QC blue red and orange.

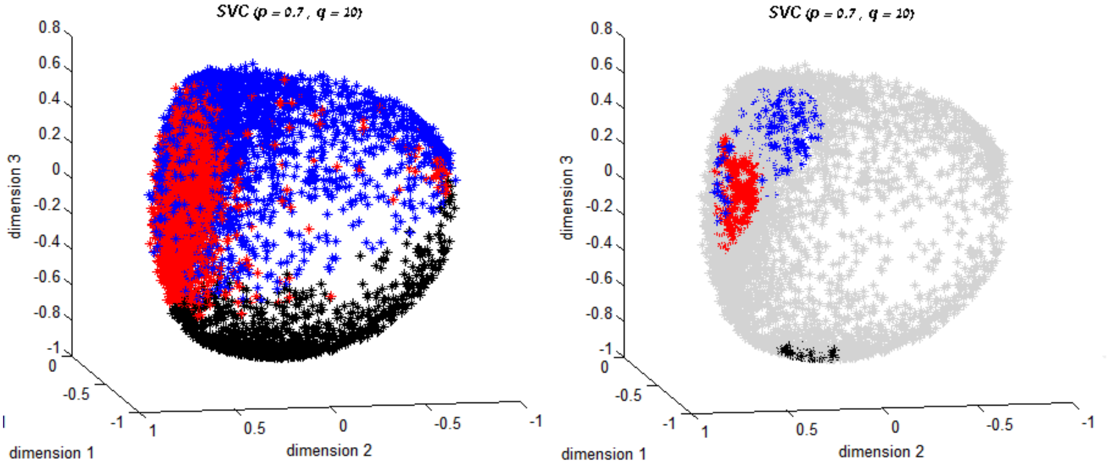


Figure 18 - on the left the cluster assignment, the different colors represent different clusters. on the right only the cluster cores (after throwing the outliers)

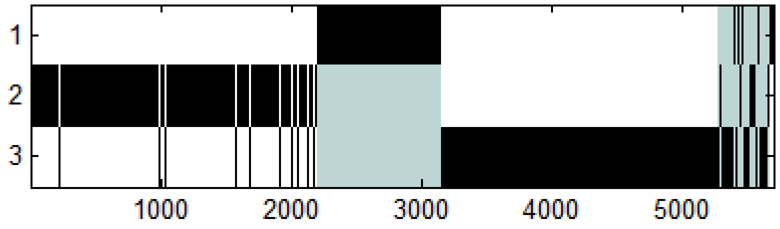


Figure 19 - Correspondence between QC earthquake classification, ordered along the x-axis (the order is blue, orange, red, black, green and 2), and their SVC clustering into 3 clusters. Each short vertical bar represents one earthquake

The QC green cluster is mapped to the SVC black cluster. We have seen before that the orange and green clusters are related (see 4.4) The QC black cluster has been divided between the three SVC clusters, with the red one getting the majority.

In order to compare the matching between the two classifications one can use the Jaccard score defined by

$$J = \frac{n_{11}}{n_{11} + n_{10} + n_{01}}$$

where n_{11} is the number of pairs of samples that appear in the same cluster both according to a known classification (in this case the regular QC classification as presented in chapter 4), and according to the new clustering algorithm (in this case the AQC). $n_{10} + n_{01}$ is the number of pairs that appear together in one classification and not in the other. This score should be 1 for perfectly matched clustering and decrease as the clustering quality decreases.

As one can see there is a good match between the two methods. The Jaccard score is 0.72, but unlike the QC method which can also detect small clusters by adjusting the sigma parameter, the SVC method is not as sensitive, and does not return small clusters no matter what are the q and p parameters. This is due to the fact that in order to get a classification there is a need of declaring large number of points as outliers, and thus letting the clustering be cruder.

5.2 Approximate QC

As presented before, the main idea of the Approximate QC is to reduce running time on big data. In this section we will use the earthquake data to evaluate the accuracy and benefits of AQC .

In Table 9 we compare several choices for the approximated potential base sizes. The size is determined by letting the user choose the number of voxels per dimension.

# voxels per dimension	# approximation data points	# clusters	# miss-matches	Jaccard score	Running time [min]
--------------	5693	6	-----------------	---------	8:20
10	1096	6	0	1	2:50
8	666	6	3	0.998	1:58
6	343	7	37	0.984	1:20
5	196	7	132	0.928	1:06
4	104	9	290	0.882	0:57

Table 9 - Comparison between different base sizes AQC runs with σ=0.3 and 100 iterations. The first row represents a regular QC run. The miss-matches and Jaccard score columns are calculated by comparing to the QC result with σ=0.3. The running time includes all of the process (building the base + running the gradient descent)

The first row represents a regular QC run (as was analyzed in previous sections), with σ=0.3 and 100 iterations. Remarkably one can see that taking ~20% of the data points as the base for the algorithm still gives an exact match to the regular classification in only 1/3 of the running time.

To describe the quality of the results we calculate, for each choice of base size, the Jaccard score (see 5.1).

For the 8 voxels case, the 6 cluster solution stays the same with only 3 miss-matches. Reducing the number of voxels leads to less accurate classification as the number of clusters increases. Although for all of them the big 5 clusters remain quite the same, the small cluster of 2 events breaks up. Some earthquakes switch clusters, but still the choice of about 6% of the data as the base returns a Jaccard score of 0.98!. For the choice of 4 voxels we can see breaking of some clusters but still there is a good match for the big 5 clusters with a Jaccard score of 0.88.

For this scenario dividing the space into 8 voxels per dimension gives a very accurate result while reducing the running time to less than 25% of the original.

As described in section 2.4 the complexity of AQC is of order $\mathcal{O}(m \cdot N)$, where m is the approximate base size, which is the number of occupied voxels. Since we carry out the analysis in a normalized SVD[14] space divided into M voxels, m will stay roughly the same size even if N is being increased.

The AQC method was employed on a big-data set, generated from High Energy Physics data, containing over 300,000 events in 6-dimensional parameter space. Carrying out the approximation using 6 voxels per dimension, has led to a base size of m=3888. The complete run for this setup over 100 gradient descent steps took around 18 hours, and has led to significant cluster structures. This shows that the approximation method can handle data sets that are even bigger than the ones studied in the previous chapters.

Appendices

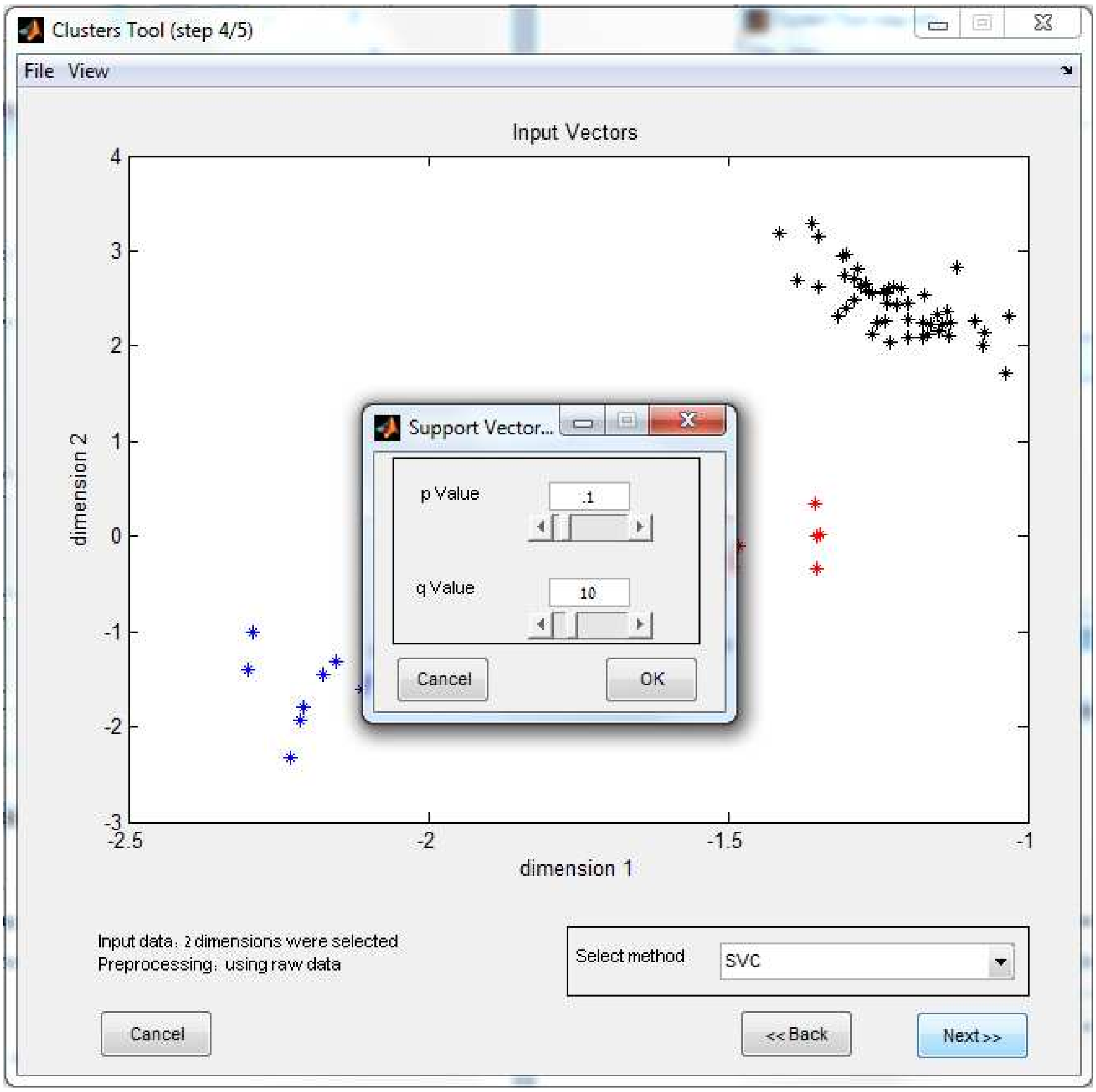
Appendix A – COMPACT 2.2

As part of this thesis enhancements were performed on the COMPACT 2.0[43], and a new version COMPACT 2.2 was introduced[44].

The main features that were added are:

A.1 SVC

The SVC algorithm as presented in section 2.2, was integrated into COMPACT, and now can be chosen from the method choices pool.



p Value .1

q Value 10

Input data: 2 dimensions were selected
Preprocessing: using raw data

Select method SVC

Figure 20 - The SVC dialog in COMPACT

The algorithm takes the following parameters as input

p value – the percentage of outliers.

q value – the Gaussians width.

Another feature which was added is the ability to view only the cluster cores as assigned to clusters, while the outliers are not assigned.

The solution for the SVM step was done with LIBSVM[45]

A.2 QC

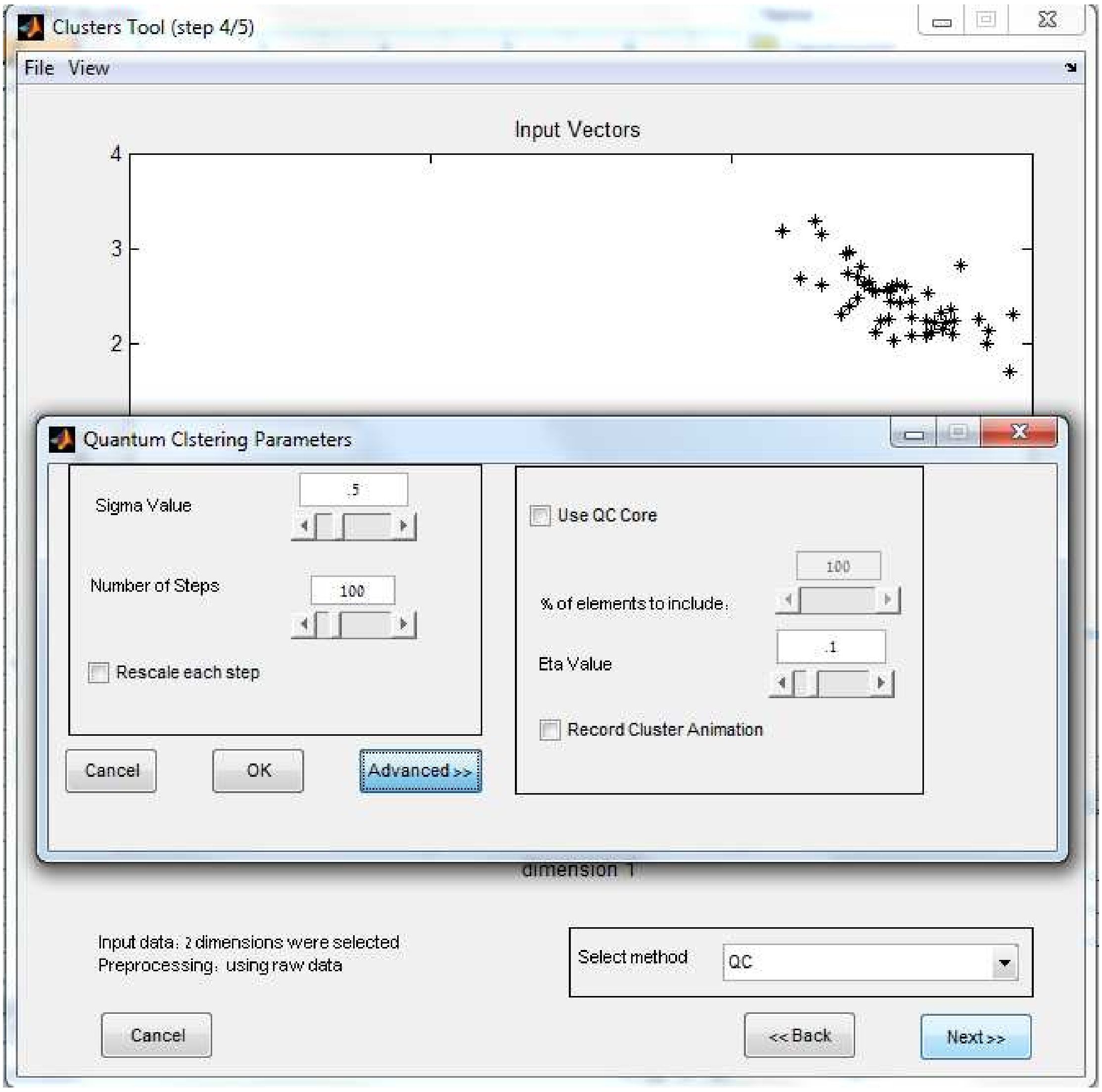
Two new input parameters were added to the QC algorithm

Rescale each step – whether or not to rescale the result after one gradient descent step in order to lie on the unit sphere.

eta – controls the eta value, which is the size of the gradient descent step

The other enhancement is the ability to record the gradient descent phase as a movie, letting the user see the movement of the data points. In this way the user may observe if the data points converged to a steady solution, or if changes should be applied to the input parameters.

The recording of the movie can be done through the QC parameters dialog, and the running is performed by a special button placed on the results screen and enabled only with the QC method.



Sigma Value .5

Number of Steps 100

□ Rescale each step

□ Use QC Core

% of elements to include: 100

Eta Value .1

□ Record Cluster Animation

Input data: 2 dimensions were selected
Preprocessing: using raw data

Select method QC

Figure 21 – The QC dialog in COMPACT

Appendix B – QC on JAVA

Since Matlab is not s suitable environment to run big-data on, we have decided to implement the QC algorithm on JAVA environment.

The decision to use JAVA was made due to:

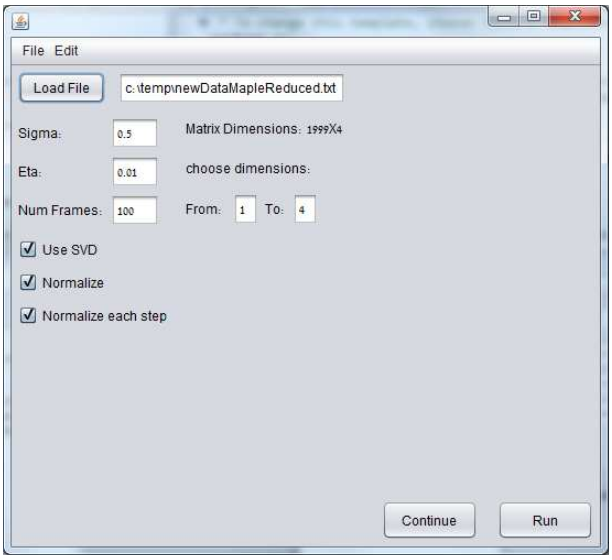
A) Good running time since it is a compiled program
B) Easy multi-threaded implementation
C) Existence of external libraries for data types and algorithms (like the SVD)
D) Existence of external libraries for 3D viewing
E) Compatibility for all platforms

Since the running time on big data was the major factor for the reason to export the algorithm, we have also designed the algorithm to run as multi-threaded, to exploit as much resources as possible.

Implementation

The porting to Java tries to save as much functionality of the COMPACT library, as can be seen in Fig 22, the basic possibilities of using SVD, normalizing, assigning parameters, and choosing dimensions are kept. The input file is a .txt one, delimited by "tab", first row and column are treated as headlines and are not taken into account for the algorithm.

The result screen as can be seen in Fig. 22, lets the user control all aspects like zoom/pan/rotate, and play the simulation of the moving data points. The 3D environment uses VTK wrapping for Java, and can handle large data sets. As in COMPACT each cluster is colored in a different color, unlike COMPACT the user may choose which dimensions he wants to plot. Saving the classification is also possible through the 'File' menu bar.



c:\temp\newDataMapleReduced.txt

Sigma: 0.5

Matrix Dimensions: 1999X4

Eta: 0.01

choose dimensions:

Num Frames: 100

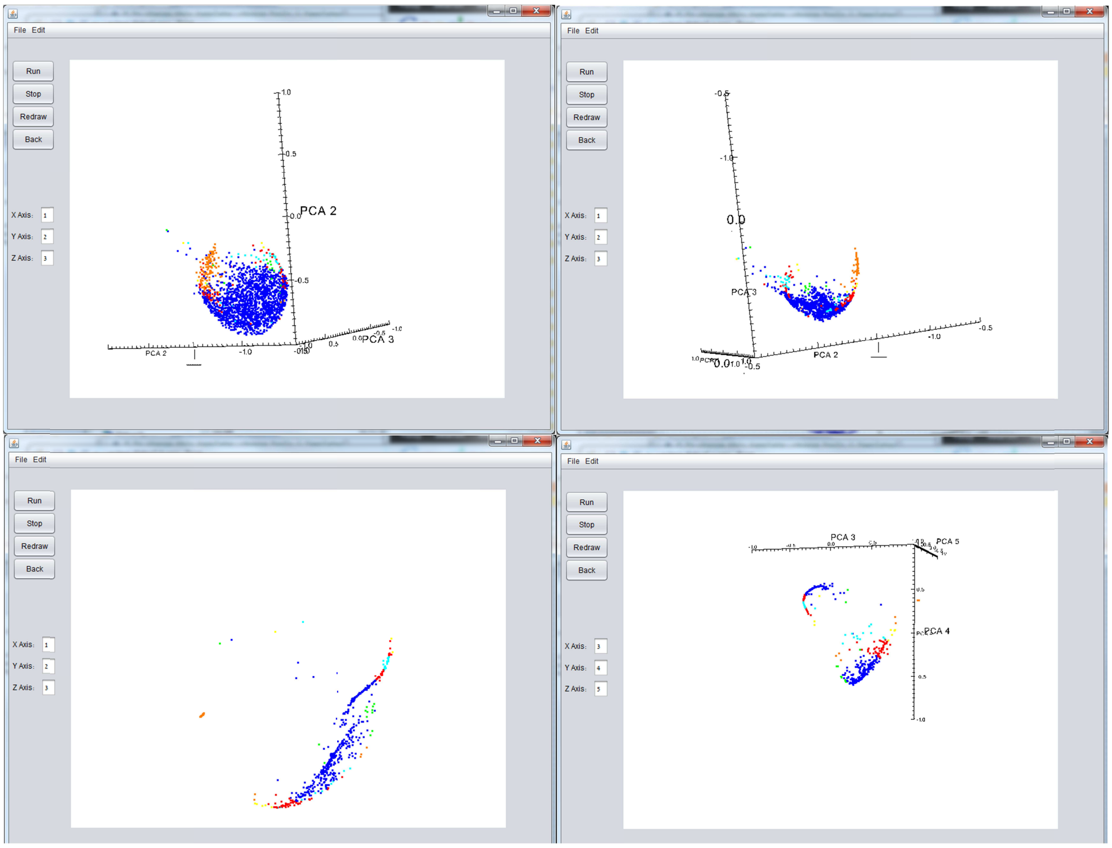
From: 1 To: 4

☑ Use SVD

☑ Normalize

☑ Normalize each step

Figure 22 - QC on Java opening dialog



X Axis: 1 Y Axis: 2 Z Axis: 3

X Axis: 1 Y Axis: 2 Z Axis: 3

X Axis: 1 Y Axis: 2 Z Axis: 3

X Axis: 3 Y Axis: 4 Z Axis: 5

Figure 23 Top Left: main Graphic viewer. Top Right: scene after several steps of animation and changing rotation. Buttom Left: zooming in on specific data.
Buttom right: scene plotted in different dimensions after several steps

In each step of gradient descent, the calculation of the induced "force" on a data-point is not affected by the same calculation for the other data-points, hence it is the logic place to perform parallelization. Indeed this is how we chose to implement the algorithm. On the earthquake data the speedup gained by the parallelization was of factor 3 on a quad core machine. Working on bigger data sets will improve the speedup, as each thread will live longer, the overhead of handling them will be less noticeable, and also will be ideal for more sophisticated machines or computer-clusters.

References

1. Ben-Dor, A.; Shamir, R.; Yakhini, Z., *JCoB*, **1999**, *6*, 281-297.

2. Wasserman, S.; Faust, K., *Cambridge University Press*, **1994**, *8*

3. Collignon, A.; Vandermeulen, D.; Suetens, P.; Marchal, G., *Computer Vision, Virtual Reality and Robotics in Medicine*, **1**.193-204 ,995

4. Basak, S. C.; Magnuson, V. R.; Niemi, G. J.; Regal, R. R., *Discrete Applied Mathematics*, **1988**, *19*, 17-44.

5. Horn, D.; Gottlieb, A., *PhRvL*, **2002**, *88*

6. Weinstein, M.; Horn, D., *Physical Review e*, **2009**, *80*

7. Ripley B.D., *Cambridge University Press*, **1996**

8. Theodoridis, S.; Koutroumbas, K., *Academic Press*, **2009**

9. R.O.Duda; P.E.Hart; D.G.Stork, *Wiley-Interscience*, **2001**

10. MacQueen J., *In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press*, **1965**, 281-297.

11. Eisen, M. B.; Spellman, P. T.; Brown, P. O.; Botstein, D., *Proc. Natl. Acad. Sci. U. S. A.*, **1999**, *96*, 10943.

12. Kohonen, T., *Springer*, **2001**, *30*

13. Cheng, Y. Z., *Ieee Transactions on Pattern Analysis and Machine Intelligence*, **1995**, *17*, 790-799.

14. Golub G.H.; Van Loan C.F., *John Hopkins University Press*, **1996**

15. Ben-Hur, A.; Horn, D.; Siegelmann, H. T.; Vapnik, V., *The Journal of Machine Learning Research*, **2002**, *2*, 125-137.

16. Cortes, C.; Vapnik, V., *MLear*.273-297 ,20 ,1995 ,

17. Rousseeuw, P. J., *JCoAM*, **1987**, *20*, 53-65.

18. Parzen, E., *The Annals of Mathematical Statistics*, **1962**, *33*, 1065-1076.

19. Press, W. H.; Teuklosky, S. A.; Vetterling, W. T.; Flannery, B. P., *Cambridge University Press*, **1992**

20. Munnix, M. C.; Shimada, T.; Schafer, R.; Leyvraz, F.; Seligman, T. H.; Guhr, T.; Stanley, H. E., *Scientific Reports*, **2012**, *2*

21. Ben-Avraham, Z., *Journal of Geophysical Research-Solid Earth and Planets*, **1985**, *90*, 703-726.

22. Freund, R.; GARFUNKE.Z; Zak ,I.; Goldberg, M.; WEISSBRO.T; Derin, B., *Philosophical Transactions of the Royal Society of London Series A-Mathematical and Physical Sciences*, **1970**, *267*, 107.&-

23. Garfunkel, Z., *Tectp*, **1981**, *80*, 81-108.

24. Garfunkel, Z.; Zak, I.; Freund, R., *Tectp*.1-26 ,80 ,1981 ,

25. Abdel-Fattah, A. K.; Hussein, H. M.; Ibrahim, E. M.; Abu El Atta, A. S., *Annals of Geophysics*, **1997**, *40*

26. Baer, G.; Sandwell, D.; Williams, S.; Bock, Y.; Shamir, G., *Journal of Geophysical Research-Solid Earth*, **1999**, *104*, 25221-25.232

27. Hofstetter, A., *Tectp*, **2003**, *369*, 21-36.

28. Hofstetter, A.; Thio, H. K.; Shamir, G., *JSeis*, **2003**, *7*, 99-114.

29. Klinger, Y.; Rivera, L.; Haessler, H.; Maurin, J. C., *Bulletin of the Seismological Society of America*, **1999**, *89*, 1025-1036.

30. Klinger, Y.; Michel, R.; Avouac, J. P., *GeoRL*, **2000**, *27*, 3651-3654.

31. Pinar, A.; Turkelli, N., *Tectp*, **1997**, *283*, 279-288.

32. Shamir, G.; Baer, G.; Hofstetter, A., *GeoJI*, **2003**, *154*, 731-744.

33. Dreger, D. S.; Helmberger, D. V., *Journal of Geophysical Research-Solid Earth*, **1993**, *98*, 8107-8125.

34. Brune, J. N., *JGR*, **1970**, *75*, 4997.&-

35. Brune, J. N., *JGR*, **1971**, *76*, 5002.&-

36. Shapira, A.; Hofstetter, A., *Tectp*, **1993**, *217*, 217-226.

37. Wessel, P.; Smith, W. H. F., *Eos Trans. AGU*, **1991**, *72*, 445.446-

38. Baer, G.; Funning, G. J.; Shamir, G.; Wright, T. J., *GeoJI*, **2008**, *175*, 1040-1054.

39. Lomnitz, C., *Tectp*, **1964**, *1*, 193-203.

40. Matthews, M. V.; Ellsworth, W. L.; Reasenberg, P. A., *Bulletin of the Seismological Society of America*, **2002**, *92*, 2.233-2250

41. Omori, F., *College Sci. Imper. Univ. Tokyo*, **1895**

42. Christensen, K.; Danon, L.; Scanlon, T.; Bak, P., *Proc. Natl. Acad. Sci. U. S. A.*, **2002**, *99*, 2509-2513.

43. Varshavsky, R.; Linial, M.; Horn, D., *Parallel and Distributed Processing and Applications - Ispa 2005 Workshops*, **2005**, *3759*, 159-167.

44. *http://neuron.tau.acil/~horn/compact.html*, **2012**

45. Chang, C. C.; Lin, C. J., *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2011**, *2*, 27.