SLAC-PUB-13435
October 13, 2008

Dynamic quantum clustering: a tool for unsupervised exploration of structures in data

Marvin Weinstein[1] and David Horn[2]
[1]Stanford Linear Accelerator Center, Stanford, CA, USA
[2]School of Physics and Astronomy, Tel Aviv University, Tel Aviv 69978, Israel

October 13, 2008

Corresponding author: Marvin Weinstein, SLAC, , USA. Tel: 650-926-2266; fax: 650-926-2525; email: niv@slac.stanford.edu

Classification: PHYSICAL SCIENCES (Applied Mathematics)

Manuscript information: 25 pages, 4 figures.

Counts: abstract 131 words.

Abbreviations: QC = quantum clustering; DQC = dynamic quantum clustering; SVD = singular value decomposition

Published in the Proceedings of the National Academy of Sciences

Work supported in part by US Department of Energy contract DE-AC02-76SF00515

Abstract

A given set of data-points in some feature space may be associated with a Schrödinger equation whose potential is determined by the data. This is known to lead to good clustering solutions. Here we extend this approach into a full-fledged dynamical scheme using a time-dependent Schrödinger equation with a small diffusion component. Moreover, we approximate this Hamiltonian formalism by a truncated calculation within a set of Gaussian wave functions (coherent states) centered around the original points. This allows for analytic evaluation of the time evolution of all such states, opening up the possibility of exploration of relationships among data-points through observation of varying dynamical-distances among points and convergence of points into clusters. This formalism may be further supplemented by preprocessing, such as dimensional reduction through singular value decomposition or feature filtering.

Clustering of data is, in general, an ill-defined problem. Nonetheless it is a very important one in many scientific and technological fields of study. Given a set of data-points one looks for possible structures by sorting out which points are close to each other and, therefore, in some sense belong together. This is a preliminary stage taken before investigating what are the common properties of these data.

It has recently become quite popular to investigate such questions in a dynamic framework, thus allowing for more fluid associations of data-points, rather than rigid clusters. Diffusion geometry [2, 7, 9] is such a method, based on a discrete analog of the heat equation

$$-\frac{\partial \Phi}{\partial t} = H\Phi \tag{1}$$

where H is some operator with positive eigenvalues, guaranteeing that the temporal evolution of $\Phi(\vec{x}, t)$ is that of diffusion. Thus, starting out with $\Phi(\vec{x}, 0)$, e.g. a Gaussian concentrated around some data point one would expect $\Phi(\vec{x}, t)$ to spread over all space that is occupied by the data points.

Here we advocate the use of a Schrödinger Hamiltonian H that is intimately connected to the data-structure, as defined by the quantum clustering method [5] and summarized below. We extend it into a time-dependent Schrödinger equation which contains a small component of diffusion:

$$\frac{i}{1 - i\epsilon} \frac{\partial \Psi(\vec{x}, t)}{\partial t} = H\Psi(\vec{x}, t) \tag{2}$$

The ensuing Dynamic Quantum Clustering (DQC) formalism allows us, by varying a few parameters, to study in detail the temporal evolvement of wave-functions, representing the original data-points, and the associations they form with each other. It is well suited for exploration of the data and obtaining various clustering solutions.

The solution of the Schrödinger equation in a large number of dimensions is a formidable task. We formulate an approximation scheme, based on a truncated Hamiltonian defined within a system of coherent states, that allows for analytic expressions of all the terms that are needed for such computations. Thus we may simultaneously follow the time-evolution of all Gaussian wave-functions centered at the original data points. This is incorporated into visual tools allowing the user to search for cluster creation under the dynamics defined by the Hamiltonian.

Quantum Clustering

Given a set of data one may use a Parzen-window estimator[3] of the probability distribution leading to the data at hand. The estimator is constructed by associating a Gaussian with each of the n data points in a Euclidean space of d dimensions and summing over all of them. This can be represented, up to an overall normalization factor by

$$\psi(\vec{x}) = \sum_i e^{-\frac{(\vec{x}-\vec{x}_i)^2}{2\sigma^2}} \tag{3}$$

where $\vec{x}_i$ are the data points. Conventional scale-space clustering [11] views the maxima of this function as determining the locations of cluster centers. The Quantum Clustering (QC) method looks, instead, at the Schrödinger potential for which $\psi(\vec{x})$ is a ground-state. The minima of the potential are candidates for cluster centers. Its Hamiltonian is defined by

$$H\psi \equiv (-\frac{\sigma^2}{2}\nabla^2 + V(\mathbf{x}))\psi = E_0\psi. \tag{4}$$

The potential $V(\mathbf{x})$ can be uniquely determined, up to a constant, from $\psi(\vec{x})$ [5]. For a single data-point it is the quadratic harmonic potential, whose quantum mechanical ground-state is the Gaussian wave-function. The Hamiltonian incorporates the interplay of two effects: attraction of points to the minima of V and their scattering as modeled by the second derivative (kinetic term). Thus H is a model framework for data distribution, based on the known experimental realization.

In QC applications it turns out that the minima of V are very good indicators of cluster centers. We refer to [5, 13] for examples. It should be noted that in [5] the function ψ assumed the role of the probability function within the Parzen-window approach. This choice, while unconventional in quantum mechanics, was adopted in [5] for computational convenience. Here we switch to the conventional quantum-mechanical interpretation of $\psi^*\psi$ representing a probability function, thus modifying slightly the interpretation of the Parzen-window derivation of ψ.

Dynamic Quantum Clustering (DQC)

Converting the static QC method to a full dynamical one, let us consider the time evolution of a wave function $\Psi(\vec{x})$ determined by the Schrödinger equation for a particle of mass m moving in d-dimensions under the influence of the potential $V(\vec{x})$:

$$i\frac{\partial\Psi(\vec{x},t)}{\partial t} = H\Psi(\vec{x},t) = \left(-\frac{\nabla^2}{2m} + V(\vec{x})\right)\Psi(\vec{x},t) \tag{5}$$

If we set $m = 1/\sigma^2$ then, by construction, $\psi(\vec{x})$ of Eq. 3 is the lowest energy eigenstate of the Hamiltonian.

Employing conventional quantum-mechanical formalism, we may represent the time evoltion of the wave-function by

$$\Psi(\vec{x},t) = e^{-iHt}\Psi(\vec{x}), \tag{6}$$

and the expectation value of the operator $\vec{x}$ becomes

$$\langle\, \vec{x(t)}\,\rangle = \int d\vec{x}\Psi^*(\vec{x},t)\,\vec{x}\Psi(\vec{x},t) \tag{7}$$

and satisfies Newton's law:

$$\frac{d^2\langle\,\vec{x}(t)\rangle}{dt^2} = -\frac{1}{m}\int d\vec{x}\Psi^*(\vec{x},t)\,\vec{\nabla}V(\vec{x})\,\Psi(\vec{x},t). \tag{8}$$

This is just Ehrenfest's theorem [8] and it is significant because, if we let $|\Psi_i\rangle$ be a Gaussian localized around the i^{th} data point, then we may explore the relation of this data point to the minima of $V(\vec{x})$ by following the time-dependent trajectory $\langle\,\vec{x}_i(t)\,\rangle = \langle\Psi_i(t)|\,\vec{x}\,|\Psi_i(t)\rangle$. Given Ehrenfest's theorem, we expect to see any points located in, or near, the same local minimum of $V(\vec{x})$ to oscillate about that minimum, coming together and moving apart. In order to reduce somewhat the oscillatory behavior we introduce a diffusion component ϵ into the dynamics, as shown in Eq. 2. This means that the evolution equation

$$|\Psi(t)\rangle = W|\Psi\rangle = e^{-itH}\, e^{-\epsilon t H}|\Psi\rangle \tag{9}$$

is no longer unitary. In order to keep using the expectation values we have to modify the definition of $|\Psi(t)\rangle$ as follows:

$$|\Psi(t)\rangle = \frac{W(t)|\Psi\rangle}{\sqrt{\langle\Psi|W(2t)|\Psi\rangle}}. \tag{10}$$

For problems with multiple minima, all states will asymptotically diffuse to the ground-state of H. We advocate therefore using small values of ϵ and evolving the dynamics for a finite time only, in order to trace the clustering of points associated with each one of the potential minima.

By introducing m different from $1/\sigma^2$ we allow ourselves the freedom of employing low σ, which introduces large numbers of minima into V, yet having also low m which guarantees efficient tunneling, thus connecting points that may be located in different potential minima. Under this more general Hamiltonian, we expect to reduce the sensitivity of the calculation to the specific choice of σ.

The Calculation Method

Assume there are n states, $|\Psi_i\rangle$, corresponding to the n points in the data-set. They form a basis within which we calculate the evolution of our model. We will denote by N, the $n \times n$ matrix formed from the scalar products

$$N_{i,j} = \langle\Psi_i|\Psi_j\rangle, \tag{11}$$

by $\mathcal{H}$, the $n \times n$-matrix

$$\mathcal{H}_{i,j} = \langle \Psi_i | H | \Psi_j \rangle, \tag{12}$$

and by $\vec{X}_{i,j}$ the matrix

$$\vec{X}_{i,j} = \langle \Psi_i | \vec{x} | \Psi_j \rangle. \tag{13}$$

The calculation process can be described in five steps. First, begin by finding the eigenvectors of the symmetric matrix N which have non-vanishing eigenvalues. These vectors are linear combinations of the original Gaussians which form an orthonormal set. Second, compute $\mathcal{H}$ in this truncated basis, $\mathcal{H}^{tr}$. Do the same for $\vec{X}_{i,j}$. Fourth, find the eigenvectors and eigenvalues of $\mathcal{H}^{tr}$, construct $e^{-it\mathcal{H}^{tr}}$ and use it to evolve the original states $|\Psi_i\rangle$. Finally, construct the desired approximate trajectories

$$\langle \vec{x}_i(t) \rangle = \langle \Psi_i | e^{it\mathcal{H}^{tr}} \, \vec{X} \, e^{-it\mathcal{H}^{tr}} | \Psi_i \rangle \tag{14}$$

through repetition of small steps of Δt until clustering of points occurs.

It is obvious that restricting attention to the truncated Hamiltonian perforce loses some features of the original problem, however its advantage is that we can derive analytic expressions for all operators involved (see Appendices A and B). As a result, the numerical computations can be done very quickly.

Example: Ripley's Crab data

To test our method we apply it to a five-dimensional dataset with two-hundred entries, used in Ripley's text book [10]. This dataset records five measurements made on male and female crabs that belong to two different species. This dataset has been used in the original paper on quantum clustering by Horn and Gottlieb[5]. Our main motivation is to provide a simple example which exhibits the details of the DQC method. In particular, we wish to show that the simplest computational scheme for implementing the general program captures the essential features of the problem and does as well as one can reasonably expect to do.

The data is stored in a matrix M which has 200 rows and 5 columns. As is typical in clustering computations, we preprocess it with a singular-value decomposition

$$M = U\,S\,V^{\dagger}, \tag{15}$$

where U is a unitary 200×200 matrix and S is the 200×5 matrix of singular values, the latter occurring on the diagonal of its upper 5×5 entries. The sub-matrix of U consisting of the first five columns, the so-called five principal components (PCs), can be thought of as assigning to each sample a unique point in a five-dimensional vector space. We may study the problem in the full five-dimensional space, or within any subspace by selecting appropriate principal components. In [5] QC was applied to this problem in a 2-dimensional subspace, consisting of PC2 and PC3. Here we demonstrate DQC on a 3-dimensional manifold composed of the first three PCs. Since we work within a sub-space of the original data-space, normalization of these vectors is not guaranteed. Hence we employ the conventional approach of projecting all points onto the unit sphere [14].

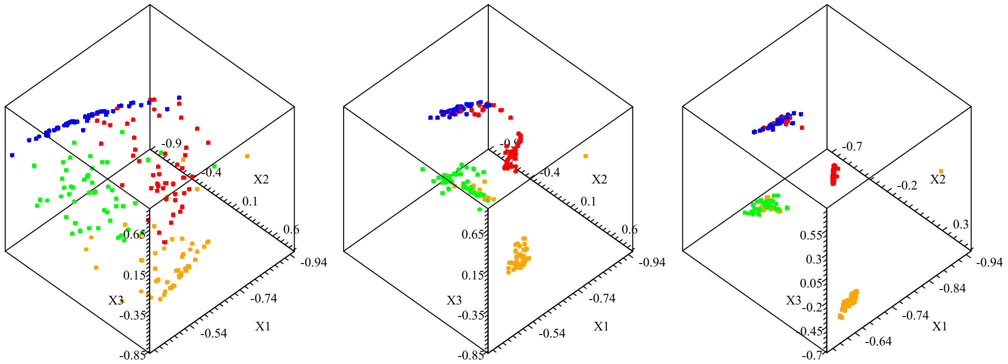

Figure 1: The left hand plot shows three-dimensional distribution of the original data points before quantum evolution. The middle plot shows the same distribution after quantum evolution. The right hand plot shows the results of an additional iteration of DQC. The values of parameters used to construct the Hamiltonian and evolution operator are: $\sigma = 0.07$, $m = 0.2$, and $\epsilon = 10^{-6}$. Colors indicate the expert classification of data into four classes, unknown to the clustering algorithm.

We study the temporal behavior of $\langle \vec{x}_i(t) \rangle$, for all i, to which we will henceforth refer as the 'motion of points'. Figure 1 shows the distribution of the original data points plotted on the unit sphere in three dimensions. This is the configuration before we begin the dynamic quantum evolution. To guide the construction of our tool we color the data according to its known four classes, although this information is not incorporated into our unsupervised method. The two species of crabs ((red,blue) and (orange,green)) are fairly well separated; however, separating the sexes in each species is problematic. The middle plot in Figure 1 shows the distribution of the points after quantum evolution, stopped at a time when some convergence into clusters has occurred. It is immediately obvious that the quantum evolution has enhanced the clustering and made it trivial to separate clusters by eye. Once separation is accomplished, then extracting the clusters can be performed by any conventional technique, e.g. k-means.

An alternative way of displaying convergence is shown in Figure 2, where we plot the Euclidean distance from the first point in the dataset to each of the other points. The clusters lie in bands which have approximately the same distance from the first point.

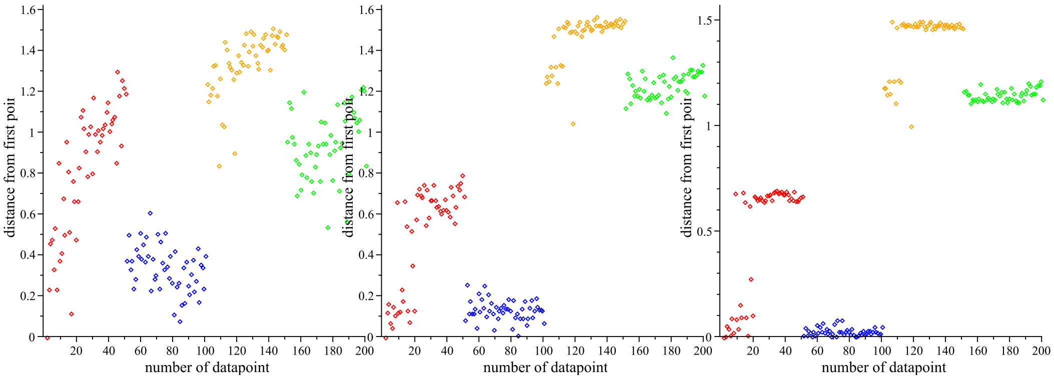

Figure 2: A plot of Euclidean distance of each point i from the first data point. Again, the left hand plot shows the distances for the initial distribution of points. The middle plot shows the same distances after quantum evolution. The right-hand plot shows results after another iteration of DQC. The numbering of the data-points is ordered according to the expert classification of these points into four classes containing 50 instances each.

It is difficult to get very tight clusters since the points, while moving toward cluster centers, oscillate around them, and arrive at the minima at slightly different times. Given this intuition, it is clear that one way to tighten up the pattern is to stop DQC evolution at a point where the clusters become distinct, and then restart it with the new configuration, but with the points redefined at rest. We refer to this as iterating the DQC evolution. The right-hand plots in Figure 1 and Figure 2 show what happens when we do this. The second stage of evolution clearly enhances the clusters significantly, as was expected.

By the end of the second iteration, there can be no question that it is a simple matter to extract the clusters. As is quite evident, clustering does not agree completely with the expert classification, i.e. points with different colors may be grouped together. This is, however, the best one can do by color-blind treatment of the information provided in the data-matrix.

The full 5-dimensional study of the crab data-set can proceed in the same manner, although it does not lead to new insights. It is provided in the Supplementary Material. Another data-set is studied there, that of viruses discussed in a paper by Varshavsky et. al.[12]. Once again DQC is shown to be a versatile tool.

Dynamic Distances

The fact that data-points of different classes happen to lie close to each other in the data-matrix can be due to various factors: errors in data measurements, errors in the expert assignment to classes, true proximity of data-points in spite of differences of origin (extreme example would be similarities of phenotypes in spite of differences in genotypes) or - the simplest possibility - the absence of some discriminative features in the feature-space that spans the data measurements. But there is another important conceptual message to be learned here - clustering and/or classification may not capture all the interesting lessons that may be derived from the data. A similar message is included in the Diffusion Geometry approach [2, 7] that advocates measuring diffusion-distances among points rather than Euclidean ones. Diffusion distances are influenced by the existence of all other points. In our DQC analysis this may be replaced in a straightforward manner by defining dynamic distances among points

$$d_{i,j}(t) = ||\, \langle \vec{x}_i(t) \rangle - \langle \vec{x}_j(t) \rangle \,|| \tag{16}$$

with the norm being Euclidean or any other suitable choice.

Clearly $d_{i,j}(0)$ is the geometric distance as given by the original data-matrix or by its reduced form that is being investigated. As DQC evolves with time $d_{i,j}(t)$ changes, and when some semi-perfect clustering is obtained, it will be close to zero for points that belong to the same cluster. Figure 2 shows this change in time for all $d_{i,1}(t)$ in the crab-data example studied above. It is quite obvious that, in addition to the few cases in which clustering disagrees with classification, there are many intermediate steps where different data-points are close to each other in spite of eventually evolving into different clusters and belonging to different classes. Thus a close scrutiny of the dynamic distances matrix $d_{i,j}(t)$ may lead to interesting observations regarding the relationships among individual pairs of points in the original data, a relationship that is brought out by DQC as result of the existing information about all other data-points. It may be used to further investigate the reason for such proximities, along any one of the lines mentioned above, and thus may lead to novel insights regarding the problem at hand.

Feature Filtering

Data exploration involves not only the instances, or data-points, but also the features (coordinates) with which the instances are defined. By performing SVD, and selecting a sub-set of coordinates, we define superpositions of the original features within which we search for clustering of the instances. In problems with very many features, it is adventageous to also perform some feature filtering, employing a judicious selection of subsets of the original features. Here we wish to demonstrate the power of feature filtering, as well as its iterative employment in conjunction with iterations of DQC. This will be demonstrated on the dataset of Golub *et al.* [4], consisting of gene chip measurements on cells from 72 leukemia patients with two different types of Leukemia, ALL and AML. The expert identification of the classes in this data set is based upon dividing the ALL set into two subsets corresponding to T-cell and B-cell Leukemia. The AML set was divided into patients who underwent treatment and those who did not. In total the Affymetrix GeneChip used in this experiment measured the expression of 7129 genes. The feature filtering method we employ is based SVD-entropy, and is a simple modification of a method introduced by Varshavsky *et al.*[12] and applied to the same data.

The method begins by computing the SVD-based entropy [1] of a dataset M (matrix of n instances by m features of Eq. 15) based on the eigenvalues s_j of its diagonal matrix S. Defining normalized relative variance values $v_j = \frac{s_j^2}{\sum_k s_k^2}$, the dataset entropy is defined through

$$E = -\frac{1}{\log r} \sum_{j=1}^{r} v_j \log(v_j) \tag{17}$$

where r is the rank of the data-matrix, typically much smaller than m . Given the dataset entropy of the matrix M, define the contribution of the i^{th} feature to the entropy using a leave-one-out comparison; i.e., for each feature we construct the quantity

$$CE_i = E(M_{(n\times m)}) - E(M_{(n\times(m-1))}) \tag{18}$$

where the second entropy is computed for the matrix with the i^{th} feature removed. Our filtering technique will be to remove all features for which $CE_i \leq 0$.

Figure 3 displays the raw data in the 3-dimensional space defined by PCs 2 to 4, and the effect that DQC has on these data. In Figure 4 we see the result of applying feature filtering to the original data, represented in the same 3-dimensions, followed by DQC evolution. Applying a single stage of filtering has a dramatic effect upon clustering, even before DQC evolution. The latter helps sharpening the cluster separation. In the Supplementary Material we demonstrate the effects of further consecutive applications of

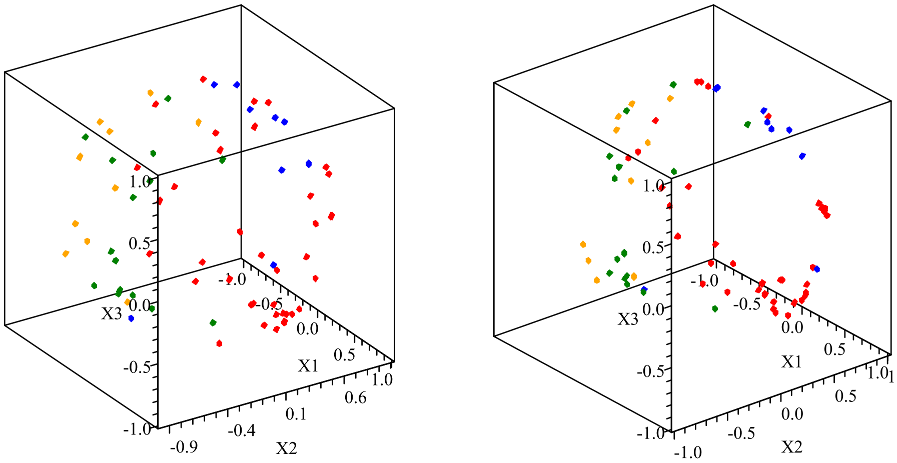

Figure 3: The left hand picture is the raw data from the Affymetrix Chip plotted for principal components 2,3,4. Clearly, without the coloring it would be hard to identify clusters. The right hand picture is the same data after DQC evolution using $\sigma = 0.2$ and a mass $m = 0.01$. The different classes are shown as blue, red, green and orange.

filtering and DQC evolution, improving clustering quality. But what is even more interesting, is that one may follow dynamical iterations tracing the merger of clusters, and use the feature filtering leading to the merger as an indication of which features are responsible for the previous distinction between the clusters.

Summary

We have proposed a dynamical theory for exploration of proximity relationships among data-points in large spaces. Starting with the potential function of quantum clustering [5] we have shown that its embedding into

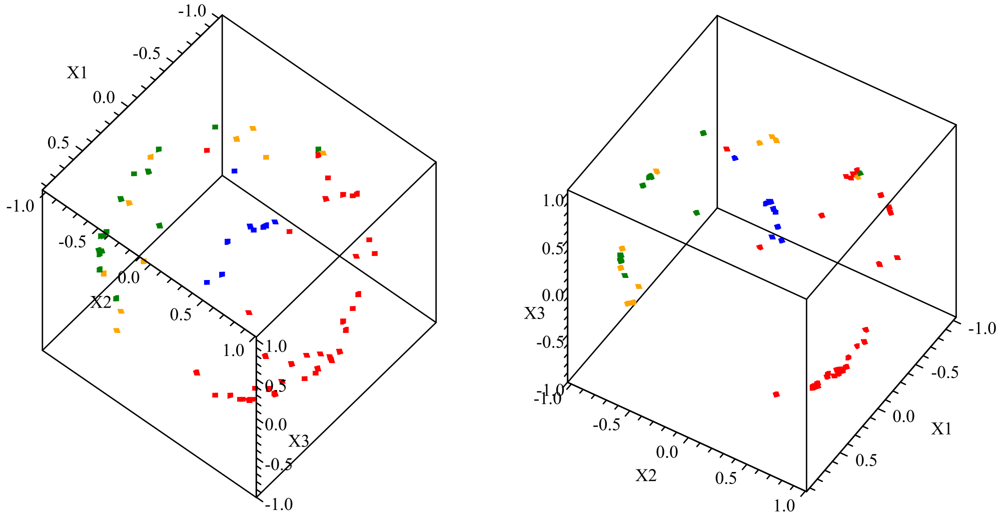

Figure 4: The left hand plot is the Golub data after one stage of SVD-entropy based filtering, but before DQC evolution. The right hand plot is the same data after DQC evolution.

a dynamical theory provides an exploratory tool. Formulating the theoretical treatment within coherent (Gaussian) states, we have derived analytic expressions for all necessary calculations of the temporal evolution. This allows us to treat quite complicated data and put them into a visual framework that can be easily manipulated by the user who wishes to search for structures in the data. We have tested the system on random data to make sure that it does not produce unwarranted clustering structures.

There are some preprocessing tools that we have employed. The first is SVD, which is being used for dimensional reduction. This is a must for handling data in very large dimensions and it helps to remove noise from the data. Dimensional reduction implies that we select a set of leading principal components, thus performing a selection of a few preferred axes composed of superpositions of the original features of the data. In addition, or as an alternative, one may wish to perform selection of individual features that are judged to be relevant to the data at hand. Since our problem is unsupervised, we employ a feature filtering method that depends on the contribution of the features to SVD-entropy. This method can be applied in tandem with iterative applications of our DQC technique.

The computational advantages of our method are its ease of use and the fact that, once one has formed the Hamiltonian of the system, the computational problem is carried out within a matrix which has no more rows and columns than the number of data points. Moreover, the simplest reduction of the analytic problem of assigning data points to minima of the multi-dimensional potential function works remarkably well. Going beyond the truncation procedure explained in Appendix B, while easily doable, seems unnecessary for most problems, and this allows us to greatly speed up the computations.

Finally, we observe that the DQC methods described in this paper can be easily extended to general classification problems that are usually resolved by supervised machine learning methods. The point is that given a training set, i.e., a data set that has been fully resolved by DQC once the appropriate stages of dimensional reduction and feature filtering has been applied, then one can use this set to classify new data. The key idea is that, since the training set has been successfully clustered we can assign distinct colors to points which lie in the training set to visually identify them in all subsequent studies. Once this has been done, the classification of new data points can been accomplished in two steps. First, reduce the SVD matrix containing both the training set and the new data points (using the previously determined features) to the an appropriate dimension, and construct the QC potential using only the the training set. Next, apply DQC to study the evolution of the full system using the new QC potential and see how the new points associate themselves with the points in the training set. Note, as always, both the intermediate dynamics and eventual coalescence of the full set into clusters can give useful information about the full data set. The fact that the old points have been colored according to the original classification scheme makes it possible to see if the SVD reduction of the full data set (training set plus new data) distorts the original classification. If this happens, i.e. if the original points fail to cluster properly, then one can go back and use the tools of feature filtering, etc. to analyze what has changed. This sort of visual identification of aspects of the data which distort clustering was already used in the case of the leukemia data set to see that the existence of a *strong* cluster can distort the clustering of the remaining data. Once this easily identified cluster was removed from the data set the clustering of the remaining data was significantly improved.

A computational package for carrying out DQC using Maple is available upon request from the authors.

APPENDIX A. USEFUL OPERATOR IDENTITES

Using conventional quantum-mechanical notation we represent the Gaussian wave function by

$$|\sigma\rangle = (\sqrt{\pi}\sigma)^{-\frac{1}{2}}\, e^{-x^2/2\sigma^2}, \tag{19}$$

where we adopted Dirac's bra and ket notation [8] to denote $|\psi\rangle = \psi(x)$ and $\langle\psi| = \psi(x)^*$. Employing the operators x and $p = \frac{1}{i}\frac{d}{dx}$ obeying the commutation relations $[x, p] = i$, we define the annihilation operator

$$A_\sigma = i\,\frac{\sigma}{\sqrt{2}}\,p + \frac{1}{\sigma\,\sqrt{2}}\,x \tag{20}$$

obeying $A_\sigma|\sigma\rangle = 0$. Its Hermitian adjoint creation operator $A_\sigma^\dagger = -i\,\frac{\sigma}{\sqrt{2}}\,p + \frac{1}{\sigma\,\sqrt{2}}\,x$ obeys $\left[A_\sigma, A_\sigma^\dagger\right] = 1$.

We will need a few identities to derive the matrix elements we have to calculate. First we note the normal ordering identity (meaning rewriting by using the operator commutation relations so that $A_\sigma{}'s$ appear to the right of all $A_\sigma^\dagger{}'s$):

$$e^{\alpha(A_\sigma^\dagger + A_\sigma)} = e^{\alpha^2/2}\, e^{\alpha A_\sigma^\dagger}\, e^{\alpha A_\sigma} \tag{21}$$

which may be proven by differentiation with respect to α. Next we note that

$$e^{g(\alpha)A_\sigma^\dagger}\, A_\sigma\, e^{-g(\alpha)A_\sigma^\dagger} = \sum_n \frac{g(\alpha)^n}{n!}[A_\sigma^\dagger, [A_\sigma^\dagger, [\ldots, [A_\sigma^\dagger, A_\sigma]]]\ldots]_n = A_\sigma - g(\alpha) \tag{22}$$

which is easily derived by differentiating with respect to g and noting that only the first commutator is non-zero. A similar calculation proves the equally useful result:

$$e^{\alpha(A_\sigma^\dagger - A_\sigma)} = e^{-\alpha^2/2}\, e^{\alpha A_\sigma^\dagger}\, e^{-\alpha A_\sigma} \tag{23}$$

Now, because the Parzen window estimator is constructed using Gaussian wavefunctions centered about points other than $x = 0$, it is convenient to have an operator expression which relates the Gaussian centered about $x = 0$ to the Gaussian centered about $x = \bar{x}$.

Theorem: $|\sigma, \bar{x}\rangle = e^{-ip\bar{x}}\, |\sigma\rangle$ is a normalized Gaussian wave-function centered at $x = \bar{x}$; i.e.

$$|\sigma, \bar{x}\rangle = (\sqrt{\pi}\sigma)^{-\frac{1}{2}}\, e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}. \tag{24}$$

This state is known as a coherent state [6], obeying

$$A_\sigma|\sigma, \bar{x}\rangle = \bar{x}|\sigma, \bar{x}\rangle. \tag{25}$$

The generalization to Gaussians in any number of dimensions is straightforward, since they are just products of Gaussians defined in each one of the different dimensions.

APPENDIX B. MATRIX ELEMENTS

The states we start out with $|\sigma, \bar{x}_i\rangle$ have norm one and are, in general, linearly independent; however, they are not orthogonal to one another. In what follows we will need an explicit formula for the scalar product of any such Gaussian $|\sigma, \bar{x}_i\rangle$ with another $|\sigma, \bar{x}_j\rangle$. This is easily derived given the operator form for the shifted Gaussian derived in Appendix A. Thus we find that

$$\langle \sigma, \bar{y} | \sigma, \bar{x} \rangle = \langle \sigma | \, e^{-ip(\bar{x}-\bar{y})} | \sigma \rangle = e^{-(\bar{x}-\bar{y})^2/4\sigma^2}, \tag{26}$$

which is needed for computing the matrix of scalar products $N_{ij} = \langle \sigma, \bar{x}_i | \sigma, \bar{x}_j \rangle$. Similarly, by employing $e^{ip\bar{y}} \, x \, e^{-ip\bar{y}} = x + \bar{y}$ we find that

$$\langle \sigma, \bar{y} | \, x \, | \sigma, \bar{x} \rangle = \frac{(\bar{x}+\bar{y})}{2} \, e^{-(\bar{x}-\bar{y})^2/4\sigma^2}. \tag{27}$$

It is straightforward to generalize this derivation to obtain

$$\langle \sigma, \bar{y} | \, V(x) \, | \sigma, \bar{x} \rangle = e^{-(\bar{x}-\bar{y})^2/4\sigma^2} \langle \sigma | \, V(x + \frac{(\bar{x}+\bar{y})}{2}) \, | \sigma \rangle, \tag{28}$$

for any function $V(x)$. Note that this expectation value can be evaluated by expanding V in a Taylor series about the point $(\bar{x}+\bar{y})/2$. The leading term is simply $e^{-(\bar{x}-\bar{y})^2/4\sigma^2} \, V\left(\frac{\bar{x}+\bar{y}}{2}\right)$ and the remaining terms, involving $\langle \sigma | \, x^n \, | \sigma \rangle$ can be evaluated from the identity

$$\langle \sigma | \, e^{\alpha \, x} \, | \sigma \rangle = \sum_{n=0}^{\infty} \frac{\alpha^n}{n!} \langle \sigma | x^n | \sigma \rangle = \sum_{p=0}^{\infty} \frac{\alpha^{2p} \sigma^{2p}}{4^p \, p!}. \tag{29}$$

To speed up computations we chose to approximate all expectation values of $V(x)$ by $V(\frac{\bar{x}+\bar{y}}{2})$, the first term in this series. One could obviously get a more accurate approximation to the original problem by including additional terms but explicit computation has shown that, for our purposes, this level of accuracy is sufficient.

The final formula we need to derive is that for

$$\langle \sigma, \bar{y} | \, p^2 \, | \sigma, \bar{x} \rangle = \langle \sigma | \, p^2 \, e^{-ip(\bar{x}-\bar{y})} \, | \sigma \rangle = \frac{(\bar{x}-\bar{y})^2}{2\sigma^2} \, e^{-(\bar{x}-\bar{y})^2/4\sigma^2}. \tag{30}$$

With these preliminaries behind us it only remains to describe the mechanics of the DQC evolution process, where we evaluate the Hamiltonian truncated to an $n \times n$ matrix in the non-orthonormal basis of shifted Gaussians:

$$\mathcal{H}_{i,j} = \langle\sigma, \bar{x_i}|\, H \,|\sigma, \bar{x_j}\rangle. \tag{31}$$

The time evolution of our original states is computed by applying the exponential of the truncated Hamiltonian to the state in question; i.e., $|\sigma, \bar{x}\rangle(t) = e^{-i\mathcal{H}t}|\sigma, \bar{x}\rangle$. Computing the exponential of the truncated operator is quite simple, except for one subtlety: we have defined $\mathcal{H}$ by its matrix elements between a non-orthonormal set of states. Hence, to perform the exponentiation, we first find the eigenvectors and eigenvalues of the metric N_{ij} and use them to compute the matrix $N_{i,j}^{-1/2}$. [1] Then we construct the transformed $\mathcal{H}$ by

$$\mathcal{H}^{tr}_{i,j} = \sum_{k,l} N_{i,k}^{-1/2}\, \mathcal{H}_{k,l}\, N_{l,j}^{-1/2}. \tag{32}$$

Now we can construct the exponential of this operator by simply finding its eigenvectors and eigenvalues. In order to compute the time evolution of one of the original states we simply write them in terms of the orthonormal basis.

The only step which remains is to explain how we compute the expectation values of the operator x as functions of time: we first construct, for

[1] If our original set of states is not linearly independent, then $N_{i,j}$ will have some zero eigenvalues. Clearly, we throw their corresponding eigenvectors away when computing $N_{i,j}^{-1/2}$. In practice we discard all vectors whose eigenvalue is smaller than 10^{-5}.

each component, the operator

$$X_{i,j} = \langle \sigma, \bar{x_i} |\, x \,| \sigma, \bar{x_j} \rangle \tag{33}$$

and use $N_{i,j}^{-1/2}$ to put this into the same basis in which we exponentiate $\mathcal{H}$; i.e., construct

$$X_{i,j} = \sum_{k,l} N_{i,k}^{-1/2}\, X_{k,l}\, N_{l,j}^{-1/2}. \tag{34}$$

References

[1] O.Alter, P.O.Brown, D. Botstein. Singular value decomposition for genome-wide expression, data processing and modeling. *Proceedings of the National Academy of Sciences*, 97, 10101-10106 (2000).

[2] R.R.Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data. *Proceedings of the National Academy of Sciences*, 102(21), 2005.

[3] R.O. Duda, P.E. Hart and D.G. Stork. *Pattern Classification.* Wiley-Interscience, 2nd ed., 2001.

[4] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek et al: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science **286** 531 (1999).

[5] D. Horn and A. Gottlieb. Algorithm for Data Clustering in Pattern Recognition Problems Based on Quantum Mechanics. Phys. Rev. Lett. **88** 018702 (2002).

[6] J. R. Klauder and B.-S. Skagerstam. *Coherent States.* World Scientific Pub. Co. 1984.

[7] S. Lafon and A.B. Lee. Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning and data set Parameterization. *IEEE transactions on Pattern Analysis and Machine Intelligence*,28, 1393-1403 (2006).

[8] E. Merzbacher. *Quantum Mechanics.* Jon Wiley & Sons, Inc., 3rd. ed. 1998.

[9] B. Nadler, S. Lafon, R. Coifman, I. G. Kevrekidis. Diffusion Maps, Spectral Clustering and Reaction Coordinates of Dynamic Systems *Applied and Computational Harmonic Analysis* 21, 113-127, (2006).

[10] B. D. Ripley *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge UK, 1996.

[11] S.J. Roberts. Pattern Recognition, **30** 261 (1997).

[12] R. Varshavsky, A. Gottlieb, M. Linial and D. Horn. Novel Unsupervised Feature Filtering of Biological Data. Bioinformatics **22** no. 14 (2006), e507-e513.

[13] R. Varshavsky, M. Linial and D. Horn. COMPACT: A Comparative Package for Clustering Assessment. in G. Chen, Y. Pan, M. Guo and J. Lu (Eds.): *Lecture Notes in Computer Science (Springer)* 3759, 159-167 (2005).

[14] Wolfe, M. B., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W., and Landauer, T. K. Learning from text: Matching readers and text by Latent Semantic Analysis. *Discourse Processes*, **25**, 309-336 (1998).