A Spectral Clustering Approach Based on Newton's Equations of Motion

K. Blekas,* I.E. Lagaris†
Department of Computer Science, University of Ioannina, Ioannina 45110, Greece

In this article, we introduce *Newtonian spectral clustering,* a method that employs Newtonian preprocessing to promote cluster perspicuity and trajectory analysis to gain valuable affinity information. A simple two-body potential is used to model the interaction under the influence of which the points move according to Newton's second law. This procedure produces a transformed data set with reduced cluster overlap, which favors the spectral clustering approach. This is so, because the affinity matrix can be enriched with information derived from the underlying interaction model. Special care is also given to estimate the Gaussian kernel parameter, since its role is important for the clustering procedure. The method is further extended appropriately to treat problems of high dimensionality. We have tested the proposed methodology on several benchmark data and compared its performance to that of rival techniques. The superiority of the new approach is readily deduced by inspecting the reported results. © 2013 Wiley Periodicals, Inc.

1. INTRODUCTION

Given a set of data points, the problem of clustering is to discover a number of subsets, called *clusters*, that contain points with similar properties. In the literature, there is a plethora of clustering approaches that have been proposed rather recently. In this work, we concentrate on the class of methods which are based on spectral clustering.[1,2] Spectral clustering has become increasingly popular during the past decade. Such algorithms are based on similarity information between data points. That is, similar data points (or points with high affinity) are more likely to belong to the same cluster than points with low affinity. These kinds of algorithms have proved to be quite successful in numerous application domains, such as computer vision,[3–5] speech recognition,[6] bioinformatics,[7,8] and text mining.[9]

Spectral clustering techniques make use of information obtained from an appropriately defined affinity matrix. Their primary strength is the ability to treat complex patterns where other well-known methods (such as k-means) either cannot

*Author to whom all correspondence should be addressed: e-mail: kblekas@cs.uoi.gr.
†e-mail: lagaris@cs.uoi.gr.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 28, 394–410 (2013)

View this article online at wileyonlinelibrary.com. • DOI 10.1002/int.21582

be directly applied or fail. The similarity matrix must be built in such a way so as to reflect the topological characteristics of the data set. In the case where the affinity is represented by a Gaussian kernel function, the choice of the scaling parameter (σ) is crucial. Points close to each other must have a high affinity measure, whereas the affinity should be reduced for distant points. To face this issue, a self-tuning methodology is presented in Ref. 10, for example, based on a nearest neighbor distance.

Sparsity is another desired property, since it offers computational advantages.[1,11] In computer vision and related problems, where spectral clustering excels in performance, the similarity matrix is sparse due to the local character of the similarity measure. Methodologies leading to sparse affinity matrices have been proposed in the past.[1] For instance, the ϵ-neighborhood technique connects only points whose pairwise distances are smaller than a prespecified threshold ϵ. Another similar method is the (mutual) t-nearest neighbor, where every point is connected only with its t nearest neighbors. However, these methods heavily depend on the choice of the control parameter (ϵ or t) that acts as a threshold for cutting some edges of the associated graph.

We present here an alternative spectral clustering method that consists of two phases. First, the data points are preprocessed according to a motion scheme, which has been introduced in an earlier work of ours.[12] During this phase, a dynamical transformation is performed that forces each data point to move toward the center of its host cluster. In this way, the cluster formation becomes more apparent. At a second level, the affinity matrix is constructed but not in the usual manner. We have developed a novel special technique that takes into account information obtained during the previous phase. The resulting affinity matrix has a sparse structure and at the same time represents the topological features of the data set with a high degree of fidelity.

We have further extended the method rendering it capable to handle high-dimensional data sets and treat problems such as document clustering and object recognition. This is achieved by choosing a suitable potential model. The proposed method has been evaluated on a suite of established benchmarks, ranging from synthetic and real continuous data sets, to computer vision applications and problems with high-dimensional data. For comparison, we also report results from the application of a self-tuning method, the standard spectral clustering method, as well as the classical k-means algorithm.

The remaining of this paper is organized as follows. In Section 2, we lay out an algorithmic description of the proposed methodology, along with its extension to high-dimensional spaces. Section 3 presents numerical results in several data sets, and finally in Section 4 we summarize our conclusions.

2. NEWTONIAN SPECTRAL CLUSTERING

The overall framework of our method consists of an initial dynamic process that gathers useful information to strengthen and sparsify the data affinity matrix. Spectral analysis is performed next, and the clustering solution is found.

2.1. Newtonian Motion Scheme

Let $X = \{x_1, \ldots, x_N\}$ be the set of N observations that we want to partition into K groups. We consider that the data points correspond to particles of unit mass ($m_i = 1$) located at position x_i, interacting via a two-body attractive, short-range potential. Under this consideration, the data points x_i move under the influence of a *force* F_i as dictated by the Newton's second law:

$$F_i(t) = m_i \frac{d^2 x_i(t)}{dt^2} = \frac{d^2 x_i(t)}{dt^2}, \forall i = 1, 2, \ldots, N . \tag{1}$$

Owing to its complicated nature, there is no analytical solution to the above equations of motion and therefore they must be solved numerically. In particular, we integrate them in small time steps δt considering that the forces F_i remain constant during this short-time interval. In molecular dynamics, the most commonly used integration algorithm is due to Verlet.[13] The basic idea is to write two third-order Taylor expansions for the positions $x_i(t)$, one forward and one backward in time:

$$x_i(t + \delta t) = x_i(t) + v_i(t)\delta t + \tfrac{1}{2} F_i(t)\delta t^2 \tag{2}$$

$$x_i(t - \delta t) = x_i(t) - v_i(t)\delta t + \tfrac{1}{2} F_i(t)\delta t^2 , \tag{3}$$

where $v_i(t)$ is the velocity of particle, i.e., the first derivative with respect to time ($dx_i(t)/dt$). Hence, summing these two equations, we simulate the motion by the following scheme:

$$x_i(t + \delta t) = 2x_i(t) - x_i(t - \delta t) + \delta t^2 F_i(t) . \tag{4}$$

Let V_{ij} be the potential between particles located at points x_i and x_j. The force F_i consists of two parts: One is due to the mutual interaction with the rest of the points, given by $-\nabla_i \sum_{\substack{j=1 \\ j \neq i}}^{N} V_{ij}$, and the other is a dissipative term that slows down the motion, given by $-cv_i(t)$, where c is a positive constant. Thus we can obtain the following equation:

$$F_i(t) = -\nabla_i \sum_{\substack{j=1 \\ j \neq i}}^{N} V_{ij} - cv_i(t) . \tag{5}$$

By substituting then Equations 4 and 5 to the basic motion Equation 1, we can obtain the following general form:

$$x_i(t + \delta t) = \frac{2x_i(t) - (1 - c\delta t)x_i(t - \delta t) - \delta t^2 \sum_{\substack{j=1 \\ j \neq i}}^{N} \nabla_i V_{ij}}{1 + c\delta t} . \tag{6}$$

Here, it must be noted that we have used the common central difference approximation for calculating the first and the second derivatives, i.e.,

$$\frac{dx_i(t)}{dt} \approx \frac{x_i(t+\delta t) - x_i(t-\delta t)}{2\delta t}, \tag{7}$$

$$\frac{d^2x_i(t)}{dt^2} \approx \frac{x_i(t+\delta t) + x_i(t-\delta t) - 2x_i(t)}{\delta t^2}. \tag{8}$$

During all experiments, the value of constant c and the time step δt were set as $c = 2$ and $\delta t = 10^{-4}$, respectively.

In our study, we have considered a potential of Gaussian form given by

$$V_{ij} = -\exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right). \tag{9}$$

The scaling parameter σ plays an important role of the process, since it determines the range of the interaction. Section 2.3 presents a mechanism for estimating the proper value for this parameter.

Since the interaction is attractive, after a time period T, particles belonging to the same cluster will get closer together, whereas data that move toward different clusters either repel each other or they are too far away to interact. So an initially spread-out cluster is being shrunk as a result of the above procedure. The simulation terminates either after a preset number of steps, or when the positions seem to remain essentially unaltered. Some typical examples are shown in the first column of Figure 3, where the original data points (grey) are concentrated (black) after $T = 100$ steps.

2.2. Constructing a Sparse Affinity Matrix

Integration of the equations of motion (Equation 6) then yields a trajectory that describes the positions and accelerations of the particles as they vary with time. From this trajectory, useful information can be gathered for constructing a richer affinity matrix A.

In particular, at the end of each step t of the dynamic process, a new transformed data set $X^{(t)} = \{x_1(t), x_2(t), \ldots, x_N(t)\}$ is obtained as the points have been moved into new positions $x_i(t)$ from their reset positions $x_i = x_i(0)$. Let $d_{ij}(t) = ||x_i(t) - x_j(t)||$ denote the distance between two points at this time step t. If this distance becomes larger, i.e., $d_{ij}(t) > d_{ij}(0)$, means that these points have moved apart. Furthermore, two points may tend to move apart, which is expressed by the condition $(x_i(t) - x_j(t))^T(F_i(t) - F_j(t)) > 0$. In these situations, it is considered that these points belong to different clusters and hence their affinity is set to zero ($A_{ij}^{(t)} = 0$). On the other hand, points that either cluster together or have the tendency to come closer they automatically earn an affinity reward.

Taking into account the above considerations, we can obtain the following expression for calculating the elements of affinity matrix at time step t:

$$A_{ij}^{(t)} = b_{ij}^{(t)} \exp\left(-\frac{d_{ij}^2(t)}{2\sigma^2}\right). \tag{10}$$

The coefficients $b_{ij}^{(t)}$ are binary and are determined by the following rule:

$$b_{ij}^{(t)} = \begin{cases} 0 \text{ if } d_{ij}(t) > d_{ij}(0) \text{ or } (x_i(t) - x_j(t))^T (F_i(t) - F_j(t)) > 0 \\ 1 \qquad\qquad \text{otherwise} \end{cases}. \tag{11}$$

Obviously, these parameters enhance the sparsity of the affinity matrix A.

It must be noted that the required number of Newtonian steps T for obtaining the motion points trajectories is not critical and does not affect the quality of the affinity matrix and sequentially the clustering solution. Experiments have shown that in most cases, only a few steps (less than 10) were enough to obtain all the necessary information. Figure 1 illustrates a typical example of the effect of the Newtonian dynamic procedure to a simulated data set that has been generated by two spherical Gaussian distributions (150 points per class). In particular, we show (a) the transformed data $X^{(t)}$, (b) the binary sparse coefficients $b_{ij}^{(t)}$, and (c) the

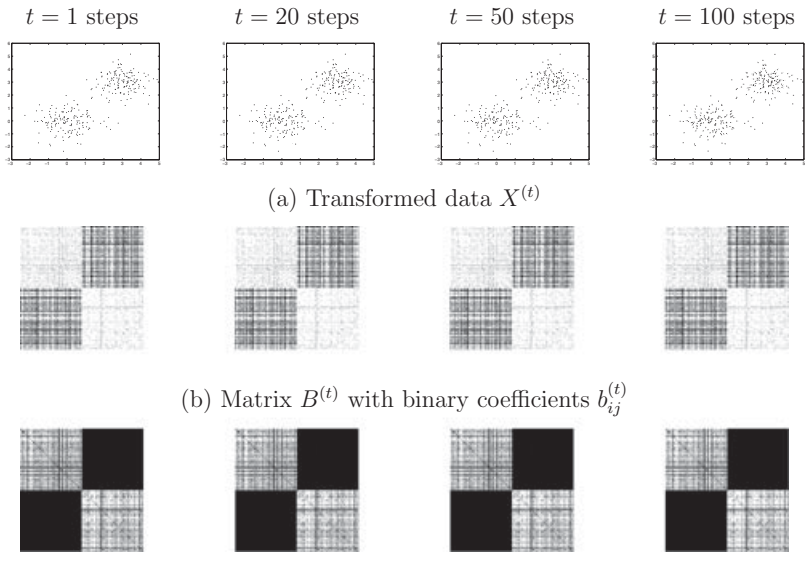


Figure 1. The evolution of the proposed Newtonian motion scheme in the case of a simulated two-class data set.

constructed affinity matrix $A^{(t)}$ at four time different steps. As it is obvious, even from the first step ($t = 1$), the affinity matrix becomes sparse (more than 20% of its elements are zeroed) without being modified significantly during the next $T = 100$ steps. Moreover, the clustering performance is exactly the same for all cases.

After constructing the affinity matrix A, the next step to our method is to operate a spectral clustering algorithm. Spectral clustering techniques takes a graph theoretical approach toward data clustering. The given N data points are viewed as nodes of a graph, and they are connected to each other by edges with weights equal to the degree of similarity. In the literature, there are several variations of the standard methodology, described in Ref. 2, that we have followed in our study. According to this scheme, the *Laplacian* matrix L is then constructed as

$$L = D^{-1/2} A D^{-1/2}\,, \tag{12}$$

where D is a diagonal matrix with elements $D_{ii} = \sum_{j=1}^{N} A_{ij}$. The Laplacian matrix is known to be symmetric and positive semidefinite. Next, the K normalized eigenvectors $u_1, \ldots, u_K$ of matrix L (where K is the desired number of clusters) that correspond to the largest eigenvalues are computed and eventually fed into the typical k-means algorithm to estimate the final clustering solution.

2.3. Estimation of the Scaling Parameter

As mentioned before, the determination of the scaling parameter σ is crucial and has to be chosen carefully. Intuitively, its value depends on the nature of data; sparse data sets require a longer range than dense data sets. An automatic selection of this parameter was suggested in Ref. 2 by running the clustering algorithm repeatedly for a number of values of σ and selecting the one which provides the least distorted clustering solution. However, this may increase significantly the computation time. Another approach proposed in Ref. 10 assumes that each data point x_i has its own local scaling parameter σ_i given by the distance to its nth neighbor, where n being manually selected. A systematic methodology has been initially presented in a previous work of ours[12] that does not need the manual tuning of any parameter. It is based on order statistics over the nearest-neighbor distance of data points.

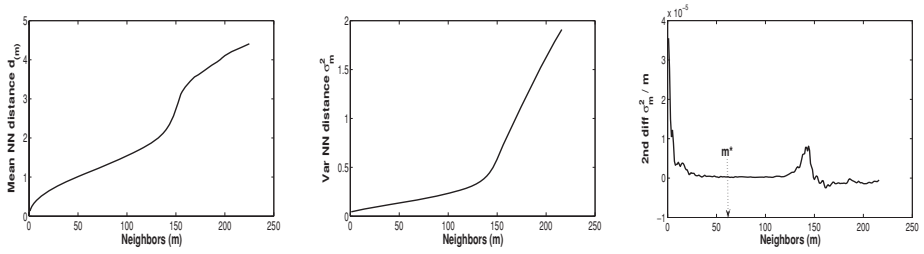
Let d_m^i be the distance between point at x_i and its mth nearest neighbor. Then, the mean value of mth nearest neighbor distance is defined as

$$d_{(m)} = \frac{1}{N} \sum_{i=1}^{N} d_m^i\,. \tag{13}$$

Obviously, we have that

$$d_{(1)} < d_{(2)} < \cdots < d_{(N-1)}\,. \tag{14}$$

These ordered values are known as *order statistics*. Studying the statistics of this quantity is useful since it can provide a measure of the appropriate range of the



(a) (b) (c)

Figure 2. Plots of (a) mean distance $d_{(m)}$, (b) variance $\sigma^2_{(m)}$ and (c) second difference of $\tilde{\sigma}^2_{(m)}/m$ in the case of two-class data set of Figure 1.

potential around every point particle. In particular, when there are more than one clusters in an examined data set we can observe some discontinuities of $d_{(m)}$ with respect to the number of neighbors m. This is happened because, as m grows the mean nearest-neighbor distance $d_{(m)}$ would be calculated using points outside the range of the cluster. In Figure 2, (a) we give one such example by plotting the quantity $d_{(m)}$ for the simulated two-class data set of Figure 1 (150 points per class). The discontinuity is appeared as m approaches the size of two clusters (150).

Assuming that the nearest neighbor distances are i.i.d. random variables following a distribution with the probability density function $f_d(y)$ and a cumulative distribution function $F_d(y) = \int_0^y f_d(t)dt$, it is known that the ordered statistics have the next pdf:

$$f_{d_{(m)}}(y) = \frac{N!}{(m-1)!(N-m)!} f_d(y)[F_d(y)]^{m-1}[1-F_d(y)]^{N-m} . \tag{15}$$

Given a probability density function $f_d(y)$, one can then calculate the variance $\sigma^2_{(m)}$ for the mth nearest neighbor distance. We have studied this quantity for several choices of pdf $f_d(y)$, where the required integrations are made numerically. In all cases, we have observed that it follows a quadratic form with respect to the number of neighbors m. When there are two or more clusters within the data set, $d_{(m)}$ acquires discontinuities and the cumulative quantity $\sigma^2_{(m)}$ is given by a superposition of translated quadratics; see Figure 2b.

In our approach, we calculate the sample variance $\tilde{\sigma}^2_{(m)}$ of any value of m as obtained by

$$\tilde{\sigma}^2_{(m)} = \frac{1}{m}\sum_{k=1}^{m}\left(d^2_{(k)} - (d_{(k)})^2\right), \tag{16}$$

where $d^2_{(k)}$ denotes the second moment of ordered mean nearest neighbor distance as computed by samples, i.e., $d^2_{(k)} = \frac{1}{N}\sum_{i=1}^{N}(d^i_k)^2$. Considering the quadratic form of the variance, the quantity $\frac{\tilde{\sigma}^2_{(m)}}{m}$ is linear in m and thus its second difference vanishes.

Hence, the value for the range of the potential σ^2 is estimated by finding the number of neighbors m^* for which the second difference of $\frac{\tilde{\sigma}^2_{(m)}}{m}$ (with respect to m) vanishes. An example is shown in Figure 2c. Note that for the determination of σ, one may choose a range of m values around m^*, without affecting the method's performance.

2.4. Algorithmic Description

The overall scheme of the proposed approach is summarized in Algorithm 1.

Algorithm 1. Newtonian Spectral Clustering

Input: Data set $X = \{x_1, \ldots, x_N\}$; Number of clusters K.

1. Estimate the scaling parameter σ^2, by finding the value m^* where the second difference of $\frac{\tilde{\sigma}^2_m}{m}$ (Equation 16) vanishes. Set $\sigma^2 = \tilde{\sigma}^2_{m^*}$.
2. Integrate the Newton's equations of motion (Equation 6) for T steps.
3. At the end of the process, calculate the *sparse* affinity matrix A (Equations 10 and 11) and then the Laplacian matrix L (Equation 12).
4. Construct the matrix $U = [u_1, u_2, \ldots, u_K]$ from the normalized eigenvectors of L that correspond to its K largest eigenvalues.
5. Apply the k-means algorithm to the N lines of U and obtain the final clustering solution.

2.5. Working with High-Dimensional Data

An important challenge in clustering is treating high-dimensional data, such as documents, gene expressions, or images. Since spectral clustering is a common technique used for this purpose, we have tried to adjust the proposed method to deal with such problems. For this purpose, we have selected the kernel cosine similarity measure for computing the proximity between each pair of data:

$$V_{ij} = x_i^T x_j \,, \tag{17}$$

where we assume that the vectors are first normalized. The above rule is also used as the potential function during the Newtonian preprocessing (see Equation 6).

The introduction of such kind of potential requires an alternative motion scheme of data points. Now, the interaction is not always attractive as in the case of Gaussian kernel. Naturally, any particle is influenced positively from similar data (that belong to the same cluster) and their interaction is attractive (positive force). In the opposite case, dissimilar data vectors have a repulsive effect to the particle and thus offering a negative sign force within its motion update rule. Taking into account the above observations, the formulation of the force F_i (Equation 5) now becomes as

$$F_i(t) = -\sum_{\substack{j=1 \\ j \neq i}}^{N} \gamma_{ij}^{(t)} x_j - c\frac{dx_i(t)}{dt} \,, \tag{18}$$

where the coefficients $\gamma_{ij}^{(t)}$ correspond to the status of the interaction (sign of force) among pairs of data, i.e.

$$\gamma_{ij}^{(t)} = \begin{cases} +1 & \text{if } V_{ij}^{(t)} > \overline{V^{(t)}}/2 \\ -1 & \text{otherwise} \end{cases} \tag{19}$$

The quantity $\overline{V^{(t)}}$ is the mean similarity value among all pairs of data in the current motion step. In fact, $\overline{V^{(t)}}/2$ acts as a threshold similarity value for distinguishing between attractive and repulsive data points. As it is obvious, this threshold is modified at each step during the recursive dynamic procedure due to the shrinking effect of data.

3. EXPERIMENTAL RESULTS

Several experiments have been performed in an attempt to examine the effectiveness of the proposed Newtonian spectral clustering approach (NSC). We have considered both simulated data sets and other widely used benchmarks. Comparison has been made with the self-tuning spectral clustering approach (STSC) that was proposed in Ref. 10 as well as with the standard spectral clustering (SC)[a] and the traditional k-means algorithm.[b] In the case of the STSC method, the local scaling parameter σ_i for each point x_i was calculated from the distance of its nth nearest neighbor. As it was expected, choosing the proper value of n significantly affects the performance of spectral clustering and therefore we have tried to tune it manually in all experiments. A satisfactory value was around a 10% of the total size N of data set.

Since we were aware of the true class label of data, all clustering methods were evaluated using two criteria:

- the *purity*, which is the percentage of correctly classified data after labeling each cluster with the label of its dominant class (most frequent among the data belong) and
- the *normalized mutual information* (NMI), which is an information-theoretic measure based on the mutual information of the true labeling (Ω) and the clustering ($\mathcal{C}$) normalized by their respective entropies:

$$NMI(\Omega, \mathcal{C}) = \frac{I(\Omega, \mathcal{C})}{[H(\Omega) + H(\mathcal{C})]/2}\,, \tag{20}$$

where

$$I(\Omega, \mathcal{C}) = \sum_k \sum_j P(\omega_k, c_j) \log \frac{P(\omega_k, c_j)}{P(\omega_k)P(c_j)}\,, \tag{21}$$

[a]Both methods NSC and SC use the same value of scaling parameter σ^2 as estimated from the proposed technique.

[b]The performance of k-means was obtained by selecting the best result among 50 different runs with different initialization.

Table I. Comparative results using five known experimental data sets.

Experimental data set	Performance of							
	NSC		STSC		SC		k-means	
	Purity	NMI	Purity	NMI	Purity	NMI	Purity	NMI
CRABS ($N = 200$, $K = 4$, $M = 2$)	0.94	0.84	0.93	0.83	0.93	0.83	0.93	0.82
Iris ($N = 150$, $K = 3$, $M = 4$)	0.95	0.83	0.91	0.79	0.94	0.83	0.89	0.76
Wine ($N = 178$, $K = 3$, $M = 13$)	0.98	0.93	0.98	0.91	0.97	0.88	0.97	0.88
Breast Cancer Wisconsin ($N = 569$, $K = 2$, $M = 32$)	0.92	0.59	0.90	0.55	0.91	0.56	0.90	0.55
Heart ($N = 768$, $K = 2$, $M = 8$)	0.70	0.15	0.70	0.13	0.69	0.12	0.64	0.06

$$H(\Omega) = -\sum_k P(\omega_k) \log P(\omega_k)\ ,\ H(\mathcal{C}) = -\sum_k P(c_k) \log P(c_k)\ . \tag{22}$$

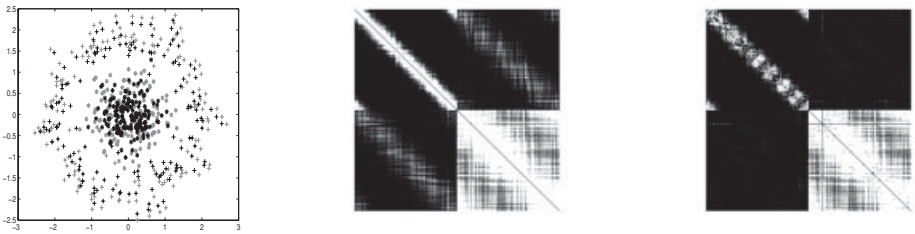
$P(\omega_k)$, $P(c_j)$, and $P(\omega_k, c_j)$ are the probabilities of a data point being in class ω_k, cluster c_j, and in their intersection, respectively, and are calculated based on set cardinalities (frequencies).

The first series of experiments was performed on four two-dimensional simulated data sets presented in Figure 3. By considering different levels of noise, we performed 50 experiments for each noise value and kept record of the mean value of both evaluation criteria for every method. In the same figure, we also compare the normal with the proposed sparse Laplacian matrix L as calculated after T steps. The depicted results are illustrated in Figure 4. As it was expected, all methods displayed almost identical behavior when studying the first two Gaussian data sets, since they were generated by K Gaussian densities that have the same spherical-type covariance matrix. The superiority of our method is apparent in the plots of the other two data sets, which are non-Gaussian and more complex. These results show that choosing the proper value of the Gaussian kernel parameter and simultaneously appropriate sparsifying the data affinity matrix has a potential impact to the performance of spectral clustering and can improve significantly the quality of clustering.

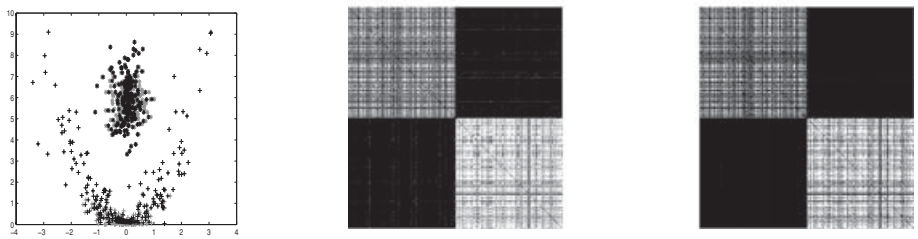
Additional experiments were made using five known benchmarks. The first one is the CRAB data set of Ripley,[14] that contains $N = 200$ data belonging to four clusters ($K = 4$). Here, we have created a two-dimensional data set by projecting the data on the plane defined by the second and third principal components. We have also studied four known benchmarks from the UCI repository[15]: the Fisher iris, the wine, the breast cancer Wisconsin, and the heart data set. Table I summarizes the results of the four comparative approaches. As it is obvious, the proposed clustering method gives slightly better performance than the others, where in some cases the difference is quite noticeable.

(a) Gaussian data with $K = 2$ classes (150 per class)

(b) Gaussian data with $K = 4$ classes (150 per class)



(c) Concentric data with $K = 2$ classes (150 per class)



(d) Moon & sun dataset $K = 2$ classes (250 per class)

Figure 3. The effect of the Newtonian procedure to spectral clustering in the case of four simulated data sets. Obviously, the proposed sparsification of the Laplacian matrix has an increased discrimination ability.

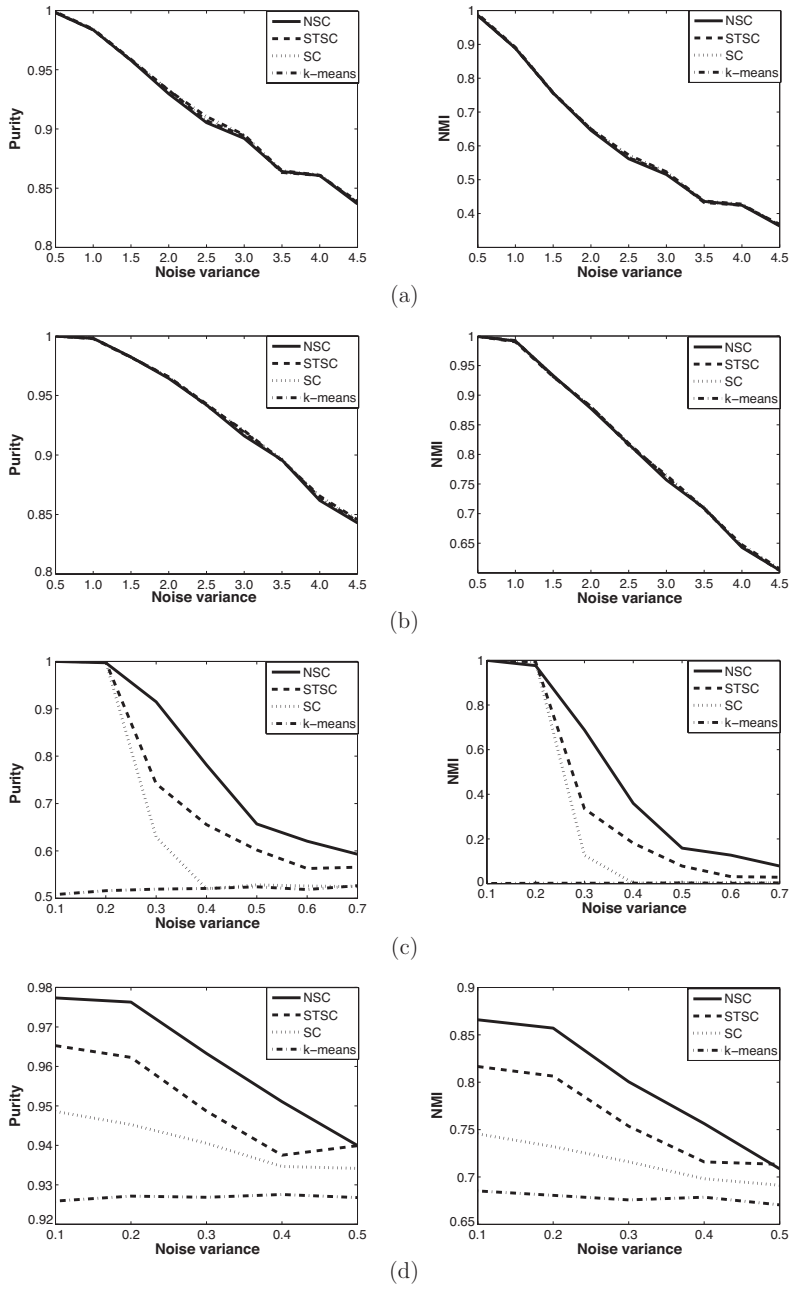


Figure 4. Comparative results on the simulated data sets of Figure 3 in terms of the degree of noise level (variance).

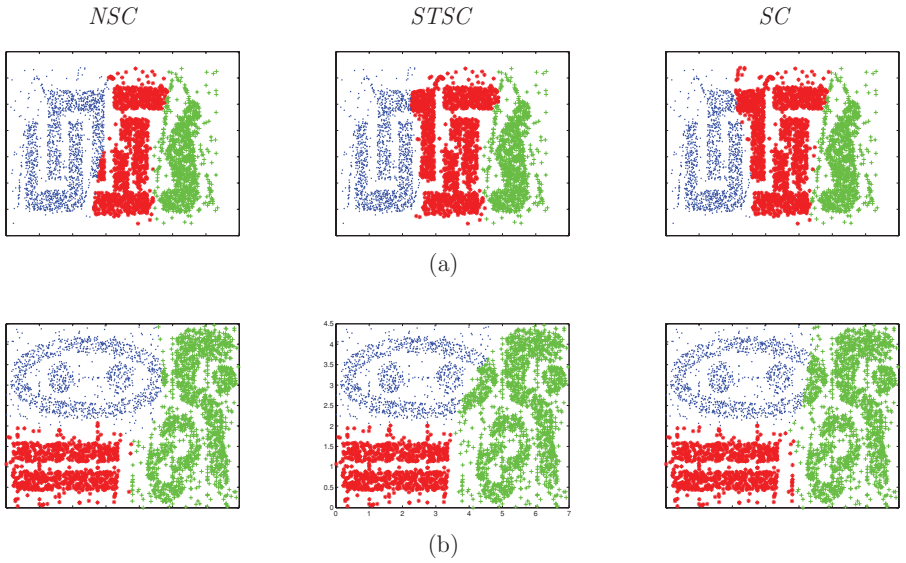


Figure 5. Comparative clustering results in two synthetic data sets.

We have also tested the effectiveness of our method to computer vision applications. At first, we have used two synthetic data sets with $K = 3$ different objects. Figure 5 illustrates the comparative clustering results. As it is obvious, our method manages to effectively discriminate three target objects, in comparison with the other two methods that provide objects with significant overlapping areas. Moreover, we have considered the task of intensity-based image segmentation. For this purpose, we have selected six colored images from the Berkeley segmentation database[c] presented in Figure 6, all with resolution of size 150×150. The segmentation results of each method are illustrated in Figure 6, where in the reconstructed images every pixel takes the intensity value of the cluster center that belongs. It is interesting to notice here that our method creates much smoother regions in comparison with the other two spectral methods. We believe that if we take into account additional features, such as spatial and texture. the quality of the resulting segmentation will be improved.

3.1. Experiments with High-Dimensional Data

Finally, we have studied the performance of our method when dealing with high-dimensional data to study the impact of the proposed motion scheme in such spaces. At first, we have examined the digits and object image recognition problem. For this purpose, we have selected two known data sets:

[c] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html

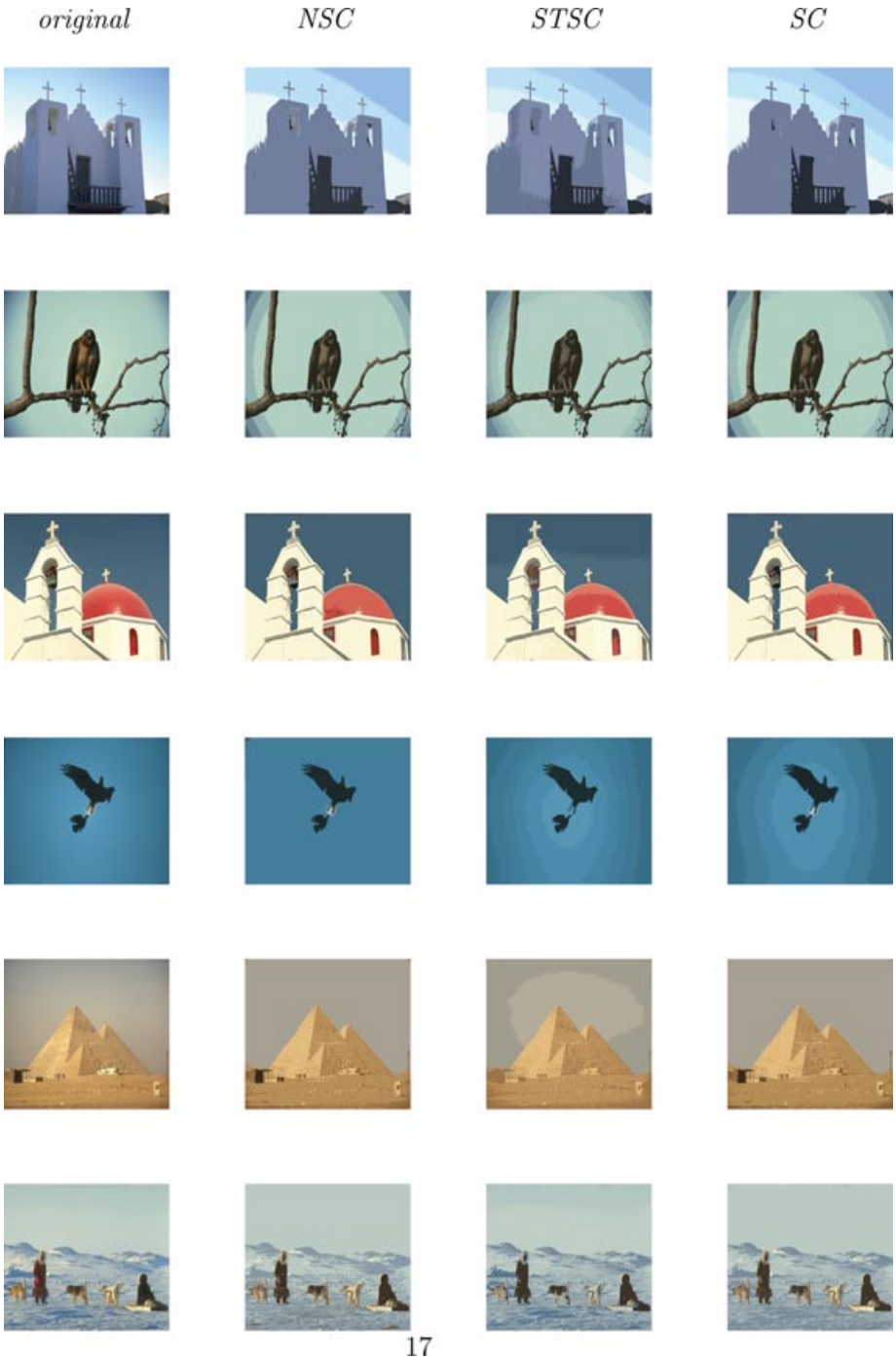


Figure 6. Segmentation results obtained by three comparative clustering methods in six real colored images.

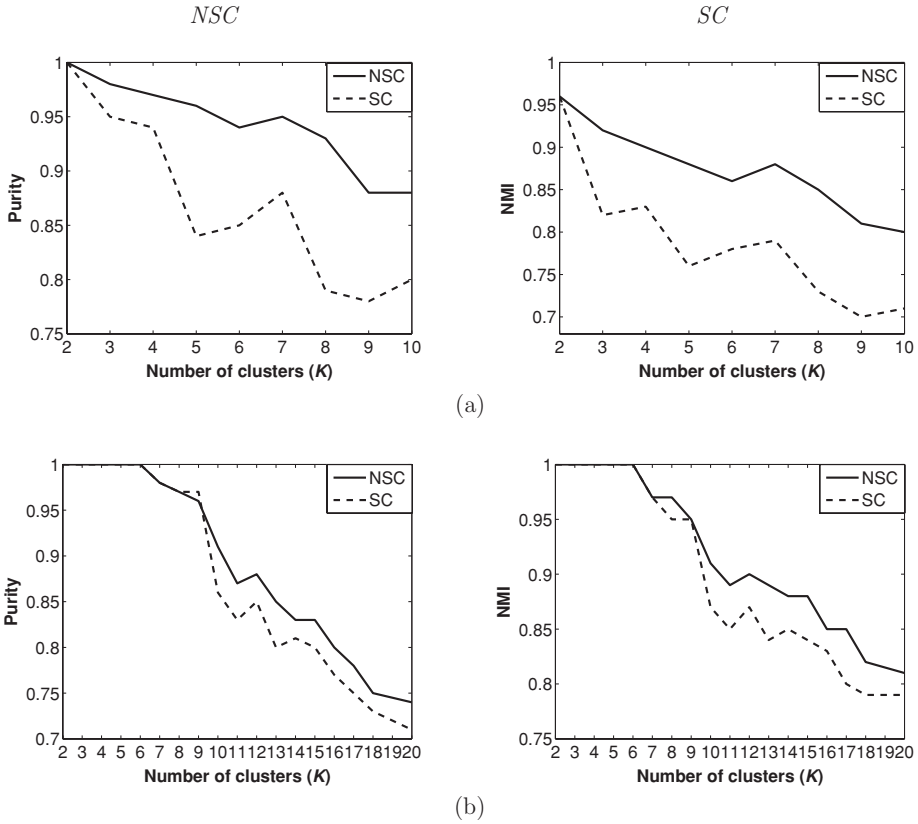


Figure 7. Evaluation of comparative methods in terms of the number of clusters for two high-dimensional data sets (a) the OCR and (b) the Coil-20.

- The UCI handwritten digits data set[15] consisted of nearly 750 samples per digit, where each one is described with a 64 (8×8) dimensional vector.
- The Coil-20 object image database,[16] which is a collection of gray-level images of 20 objects. For each object, there are 72 images copies (taken around the object at the pose interval of 5 deg) of size 128×128, i.e., vectors of 16,384 components.

Comparison has been made only with the typical SC algorithm, since the self-tuning spectral clustering (STSC) method considers only RBF kernels.

In this series of experiments, we have additionally examined the impact of the number of clusters on the performance of the method. For this reason, we considered several number of clusters, varying from 2 to $K_{\max}$ (10 and 20 for each data set, respectively), where we took several permutations of data from different classes. Figure 7 illustrates the obtained results for both evaluation criteria, where for each number of clusters (K) we show the mean value of both evaluation metrics on all random permutations. Our method significantly outperforms the standard spectral clustering approach, especially for a large number of clusters.

Table II. Document data used in our experiments and the accuracy results obtained by both NSC and SC methods.

Document data set		Performance of			
		NSC		SC	
Name	Description	Purity	NMI	Purity	NMI
Talk_3	$N = 300$, $K = 3$, $M = 4515$	0.80	0.48	0.71	0.35
Science_4-400	$N = 400$, $K = 4$, $M = 4855$	0.71	0.44	0.70	0.42
Science_4-2000	$N = 2000$, $K = 4$, $M = 10250$	0.76	0.47	0.73	0.42
Multi_5	$N = 500$, $K = 5$, $M = 5589$	0.75	0.57	0.63	0.50

Another interesting application in large spaces is document clustering that aims at the division of a collection of documents into groups based on their terms similarity. In our study, each input document has been transformed into a feature vector equal to the size of the corpus vocabulary, such that every feature denotes the weight of the corresponding term. We have applied the TF-IDF (term frequency, inverse document frequency) weighting scheme for creating feature vectors. During our experimental study, we have selected subsets of documents from the popular 20-newsgroup collection[d] described in Table II. The first set (Talk_3) consists of documents of the talk subjects (politics.guns, politics.mideast, politics.misc), the next two of scientific documents (crypt, electronics, med, space), and the fourth set has documents from five newsgroups (comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, and talk.politics.mideast). Table II presents the obtained comparative results. As it can be observed, our method improves, in some cases significantly, the performance of the typical SC framework, showing that the proposed scheme of constructing sparse affinity matrix is worthwhile in high-dimensional data. Several other experiments were made using other subsets from the same data collection where we took similar results.

4. CONCLUSIONS

In this study, we presented a novel clustering method, the Newtonian spectral clustering, that inherits from Newtonian clustering information such that renders possible the formation of a proper affinity matrix that is sparse and contains enriched information. An extension of this approach has also been presented to deal with high-dimensional data such as images and documents. We have applied the method to several benchmark problems, and we noticed performance superior to the standard spectral clustering approach. It is our intention to further pursue and develop the method to study interesting application areas with a complex type of data such as time series, multimedia data, and discrete sequences. Furthermore, another future research direction is to construct appropriate kernel functions for them so as to obtain better motion quality and therefore to improve the clustering performance.

[d]http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

References

1. Luxburg U. A tutorial on spectral clustering. Stat Comput 2007; 17(4): 395–416.
2. Ng A, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. Adv Neural Inform Process Syst 2001; 14: 849–864.
3. Chang H, Yeung D. Robust path-based spectral clustering with application to image segmentation. In: Proc Int Conf on Computer Vision; 2005. pp 278–285.
4. Park J, Zha H, Kasturi R. Spectral clustering for robust motion segmentation. In: 8th European Conf on Computer Vision; 2004. pp 390–401.
5. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Recog Mach Intell 2000; 22(8): 888–905.
6. Bach F, Jordan MI. Learning spectral clustering, with application to speech seperation. J Mach Learn Res 2006; 7: 1962–2001.
7. Higham DJ, Kalna G, Kibble M. Spectral clustering, and its use in bioinformatics. J Comput Appl Math 2007; 204: 25–37.
8. Pentney W, Meila M. Spectral clustering for biological sequence data. In: Proc of the 25th Annual Conference of AAAI; 2005. pp 845–850.
9. Dhillon IS. Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc 7th ACM SIGKDD Int Conf on Knowledge Discovery and Data mining (KDD); 2001. pp 269–274.
10. Zelnik-Manor L, Perona P. Self-tuning spectral clustering. Adv Neural Inform Process Syst 2004; 17: 1601–1608.
11. Chen W, Song Y, Bai H, Lin C, Chang EY. Parallel spectral clustering in distributed systems. IEEE Trans Pattern Anal Mach Intell 2011; 33(3): 568–586.
12. Blekas K, Lagaris IE. Newtonian clustering: an approach based on molecular dynamics and global optimization. Pattern Recog 2007; 40(6): 1734–1744.
13. Verlet L.. Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Phys. Rev. 1967; 159: 98-103.
14. Ripley BD. Pattern recognition and neural networks. Cambridge, UK: Cambridge University Press; 1996.
15. Merz CJ, Murphy PM. UCI repository of machine learning databases. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html. Irvine, CA.; 1998.
16. Nene SA, Nayar SK, Murase H. Tr-cucs-005096. Technical report, Columbia Object image library (COIL-20); 1996.