

Condensation algorithm for the 2-D mapping of gene expression patterns

Xijin Ge^{1,2}, Shuichi Tsutsumi², Hiroyuki Aburatani² and Shuichi Iwata¹
¹*Research into Artifacts, Center for Engineering (RACE);* ²*Department of Life Sciences, Research Center for Advanced Science and Technology, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan*

Abstract: Motivated by the physical phenomena of condensation, where atoms self-organize and display various structures that minimize the total energy, we developed a so-called condensation algorithm for the discovery of correlation and structures hidden in multi-dimensional data sets. Each data point is represented by a virtual atom that interacts with others in accordance with pairwise similarities. Total energy of the system is minimized by a simulated annealing process, and the resultant configuration reveals features of the statistical distribution of the raw data. Some artificial data sets and the Iris database are used to verify the algorithm. The method is then applied to the analysis of gene expression patterns for detecting new subclasses of cancers. The result is compared with those obtained by principal component analysis (PCA) and multidimensional scaling (MDS).

Key words: Gene expression profiling, multidimensional scaling, condensation algorithm, principal component analysis

1. INTRODUCTION

In addition to class prediction (assigning tumors to known classes), class discovery (the identification of new cancer subclasses) is also an important issue in the analysis of the gene expression patterns of a number of samples[1,2]. In ref. [3], Self-organizing maps (SOMs) have been used for unsupervised classification of samples. When a SOM with two nodes was

used, a collection of samples was successfully divided into two groups. One contains mostly acute lymphoblastic leukemia (ALL) samples and the other contains acute myeloid leukemia (AML) samples. Nonetheless, like many other algorithms for cluster analysis, SOM itself provides no mechanism to determine *how many* nodes are appropriate for a given data set. With a four-node SOM, Golub *et al.* found four classes largely corresponded to AML, B-lineage ALL, T-lineage ALL, and again, B-lineage ALL, respectively. Thus it is important to have a good guess of the number of subclasses presented in a given data set. In ref. [3], this problem is tackled by combining SOM with a supervised classification technique called neighborhood analysis, which searches a small number of predictor genes and uses the typical expression levels of these genes to classify samples. The strategy is that if putative classes reflect true structure, a class predictor based on these classes should perform well. In addition to this kind of *trial-and-error* approach, it is tempting that the number of subclasses be obtained in a more efficient way.

In this context, a 2-D mapping of the raw data can be helpful. Gene expression patterns can be considered as points in an n -dimensional Euclidean space. The distribution of these points constitutes a data manifold in that space whose geometrical features indicate regularities and correlation. In order to discover such regularities, one of the conventional strategies is to construct a low-dimensional representation while preserving certain topological features. Using such methods, it is possible to observe the data structure directly through a snapshot in 2 or 3 dimensions. Those snapshots can be helpful in determining the appropriate number of subclasses in a data set.

Principal component analysis (PCA) is a classical linear method for dimension reduction. It introduces a k -dimensional ($k < n$) “hyperplane” lying in the n -dimensional data space, the location and orientation of which are chosen such that the majority of data points can be approximated well by points of the hyperplane. The hyperplane is defined by a set of orthogonal axes along the directions of maximum covariance. Especially useful is a 2- or 3-dimensional plot of the raw data by using the first two or three principal components. Although PCA has been widely used in a variety of problem domains, its limitations as a linear dimension-reduction method are obvious: the distribution of data points may deviate from a hyperplane and be better described by a curved “hyper-surface”. Especially for very high dimensional data, the chance that a data set can be faithfully represented by a 2 or 3 dimensional linear projection is often small.

In this paper, we introduce a nonlinear algorithm for the mapping of the multi-dimensional data on lower-dimensional spaces. The algorithm is developed by taking advantage of some concepts from solid state physics[4]. When a collection of atoms are cooled down from high-temperature gaseous

or liquid state, atoms often self-organize into solids with particular structures, ranging from a snow blade at the macroscopic level to the beautiful buck ball C60 molecule and a face-centered cubic crystal lattice at the microscopic level. Nature arrives at these structures because they are optimal (at least near to optimal)¹ from a total energy perspective, given the physical properties of these atoms. The specific structure reflects the nature of interactions between atoms. Simulating this phenomenon, we use atoms moving in 2 or 3-dimensional space to represent data points in n -dimensional space and define the pairwise *interaction* between atoms according to the pairwise *similarities* of data points. The interaction is defined in such a way that the aggregation of similar patterns and the separation of dissimilar patterns will lower the total energy. By studying the configuration with the lowest energy, we hope to obtain topological features of the data manifold, and thereby make the observation of these features feasible by two or three-dimensional representation.

2. METHODOLOGY

Let \mathbf{X} be a set of data vectors, namely,

$$\mathbf{X} = \{X_1, X_2, \dots, X_m\}, \quad (1)$$

where $X_i = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}\}$ denotes an n -dimensional data point, here the expression pattern of n genes in a sample. The degree of similarity can be measured in several ways, such as the Euclidean distance, angle, or inner product of the two vectors, etc. For example, one can use the Euclidean distance to measure the similarity between X_i and X_j ,

$$S_{ij} = \sqrt{\frac{1}{n} \sum_{k=1}^n (x_i^{(k)} - x_j^{(k)})^2}. \quad (2)$$

¹ Nature is good at finding optimal solutions. What is the densest packing of identical spheres? Many mathematicians believe and most physicists know that no packing of spheres can be denser than a face-centered-cubic lattice, the crystal structure shared by many elemental crystals, such as copper. But the proof of the so-called Kepler's conjecture has kept mathematicians busy for centuries until 1998 when Thomas C. Hales claimed that he had proved it. Refer to N. J. A. Sloane, Nature, vol. 395, pp. 435-436 (1998) for a review. Related to the Kepler's problem, another interesting problem is "what is the longest piece of ropes you can pack into a particular box?". Surprisingly, computer simulations (A. Maritan et al., Nature, vol 406, pp.287-290, 2000) seem to suggest that it is the familiar, naturally occurring, helical structures of proteins and DNA. It is uncertain how long it will take until a mathematical proof is given.

It has been shown that this simple definition works well for a variety of problem domains. For a specific problem, however, the form of similarity metric should be chosen according to the nature of the problem.

Using whatever a similarity metric, we eventually obtain a $m \times m$ similarity matrix \mathbf{S} . Note that \mathbf{S} is symmetrical as $S_{ij} = S_{ji}$, and $S_{ii} = 0$. For convenience of further calculation, matrix elements are linearly scaled such that the mean and standard deviation of all matrix elements are 0 and δ , respectively. The standard deviation δ is usually chosen to be 0.2 to 0.5 in the our calculation. After normalization, S_{ij} would be a small negative number if X_i and X_j are more similar to each other than the average level, and a small positive number if they are not.

Each data vector is then represented by an ‘atom’ in some low dimensional space. Usually a 2 or 3 dimensional space is used for the convenience of visualisation, although the algorithm does not have any limitation on the dimension. These virtual atoms move around and interact with each other through pairwise interaction characterised by a potential energy. The state of this collection of atoms is described by its geometrical arrangement of atoms. For every state, the total energy is given by

$$E_{total} = \frac{1}{2} \sum_{i,j(i \neq j)} E_{ij}(r_{ij}, S_{ij}), \quad (3)$$

where $E_{ij}(r_{ij}, S_{ij})$ is the pairwise potential energy between atom i and j . Here it is assumed that these atoms are massless, as the expression for total energy only includes potential energy, excluding kinetic energy.

The pairwise potential energy $E_{ij}(r_{ij}, S_{ij})$ between two atoms is a function of the inter-atomic distance r_{ij} as well as the similarity S_{ij} between the corresponding data vectors. Since the purpose is just to let atoms representing similar data vectors attract each other and finally find closer stable positions while doing the opposite for those dissimilar ones, the form of the interaction can be defined in various ways. Here the potential energy between atoms is given by

$$E_{ij} = E_0 (\mathbf{1} + x^*) e^{-x^*}, \quad (4)$$

$$x^* = \beta \left(\frac{r_{ij}}{e^{\alpha S_{ij}}} - x_0 \right),$$

where E_0 , α , β , and x_0 are adjustable parameters. The value of E_0 is negative while the other parameters are all positive. The prototype of this function ($S_{ij} = 0$) is an empirical equation describing the dependence of

cohesive energy of solids on its lattice parameter[5], widely used in the computer simulation of solids(see [6] for an example). This function defines a stable distance x_0 at which the potential energy between two atoms is the lowest (E_0). If they come closer than x_0 the interaction will be repulsive; when they get farther away, the interaction becomes attractive. Both cases give rise to a potential energy higher than E_0 . In order to let similar atoms aggregate, one could simply modify this distance x_0 according to the similarity, but we find it more efficient to scale the distance of atoms in the features space. In Eq. 4, the distance S_{ij} is normalised by a factor $\exp(\alpha S_{ij})$. Qualitatively, this is equivalent to enlarging x_0 for dissimilar atoms and shortening x_0 for similar ones. Figure 1 shows the potential function for different values of similarity. The parameters are $E_0 = -2.0$, $x_0 = 1.0$, $\alpha = 1.0$, $\beta = 1.0$. These values are used in the following calculation. From the figure, it can be seen that atoms representing similar data points have shorter stable distance. According to the above definitions, it is obvious that the aggregation of similar atoms and the separation of dissimilar ones will lower the total energy. In the next step we minimize this total energy by adjusting positions of atoms.

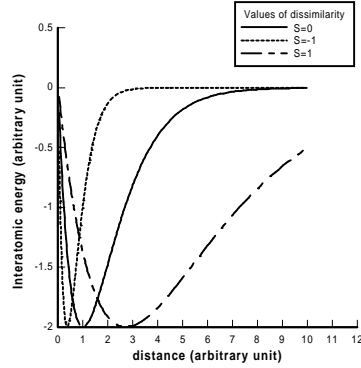


Figure 1. Interaction potential between virtual atoms depending on measures of similarity.

A very simple move-and-evaluate technique is employed to optimize the positions of atoms. At the very beginning of the simulation, atoms are placed randomly. During each iteration, these atoms make a small step of random movements. After each move the total energy is calculated according to Eq. 3. If the total energy is lowered, the move is accepted. Otherwise, the probability to accept the move is given by the Boltzmann distribution,

$$P\{E\} = e^{-E/(k_B T)}, \quad (5)$$

where T and k_B are parameters corresponding to temperature and Boltzmann constant, respectively. In our calculation, the value of k_B is arbitrarily chosen to be 0.002 and T can range from one to two thousand. These values are decided only for the sake of computation. These up-hill moves are accepted in order to avoid the trapping of the system by local minima. Nevertheless, the probability of acceptance $P\{E\}$ is often very small. It is easy to imagine that the total energy will be gradually lowered by simply repeating this process. To achieve a faster convergence, the temperature T is linearly decreased during calculation, resulting in a decreased probability of accepting up-hill moves. According to Eq. 5, it is more probable to accept high-energy moves at high temperature than at low temperature. Also the average distance that an atom can jump is dependent on T . This provides a large mobility of atoms to explore the space at the early stage of the simulation and the ability of fine-tuning at the late stage. When the decrease in total energy becomes very slow and atoms are relatively stabilized, we obtain an optimal configuration for the system of virtual atoms.

In fact, this method of optimization is a very simple form of simulated annealing. Simulated annealing is an optimization method applicable to searching for global minimum of objective functions. Here the objective function is just the total energy of the system and the variables to be optimized are the locations of atoms. Simulated annealing is based on the annealing process in the physics of solids. Annealing denotes a physical process, where the solid is first heated to a high temperature and then cooled slowly down to the original temperature. The high annealing temperature provides the particles of the solid with a very high mobility. Consequently, the particles can reach locations all around the solid. If the cooling happens sufficiently slowly, all the particles of the solid arrange themselves such that the system will have minimum total energy. It has been proved mathematically that simulated annealing is able to find global minimum in the solution space, given sufficiently slowing cooling rate[7].

The self-organization of atoms upon cooling parallels the physical process of condensation where randomly located gaseous or liquid atoms form solids, which often have a low-energy configuration, such as a regular crystal structure. For this reason, we named the algorithm as *condensation algorithm*.

3. VERIFICATION

Before any practical application, it is necessary to test the algorithm by using artificial data sets and some standard databases with known structures.

3.1 Artificial data

The algorithm is first applied to a set of artificially generated data vectors. For the purpose of comparison, the raw data are chosen to be 2-dimensional (Fig. 2). These data are generated by adding small random vectors to four proto-types, (0.1, 0.1), (0.1, 0.8), (0.8, 0.1), and (0.8, 0.8). Fifteen vectors are generated from each prototype. So the data set is composed of four clusters that form a square. This raw data is used to calculate the similarity matrix, according to which inter-atomic potentials between 60 atoms are defined.

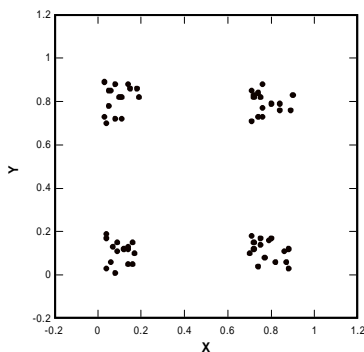


Figure 2. A set of artificially generated 2-dimensional vectors as a database for testing.

Simulated annealing is then performed to search for the optimal configuration of this. Figure 3 shows the process of optimization. Initially ($t=0$), all atoms are confined into a small region near the origin. According to the pairwise potential defined by Eq. 4, there are strong repulsive forces between atoms and the system has a very high total energy. At the early stage of the simulation (for example at $t=2$), the system underwent an “explosion” process as more and more atoms jumped out of the limited small area. At $t=100$, the explosion ended and the system entered a self-organized “condensation” phase. At $t=260$, well-defined clusters formed, but the relative position of clusters are still not optimal. Finally ($t=5000$), the systems condensed into a stable configuration, which reveals the topological features of the raw data. The decrease of total energy is shown in Fig. 4.

Actually, the map is similar to a rotated version of the raw data set shown in Fig. 2². The interpretation of Fig. 3-(f) should be made based on the notion that a shorter distance between two atoms indicates a greater degree of similarity between the data vectors that they represent. Therefore, a cluster

² Practically, we are not interested in building rotated version of 2-dimensional maps as the raw data set itself can be directly visualized.

of atoms implies that the corresponding data points are close to each other in multi-dimensional space. The absolute values of coordinates of atoms shown in Fig. 3-(f) are just parameters for visualization and do not convey information on their own rights. Instead, we should infer essential features of the raw data from the relative positions of atoms and the overall structure shown by the whole collection of atoms.

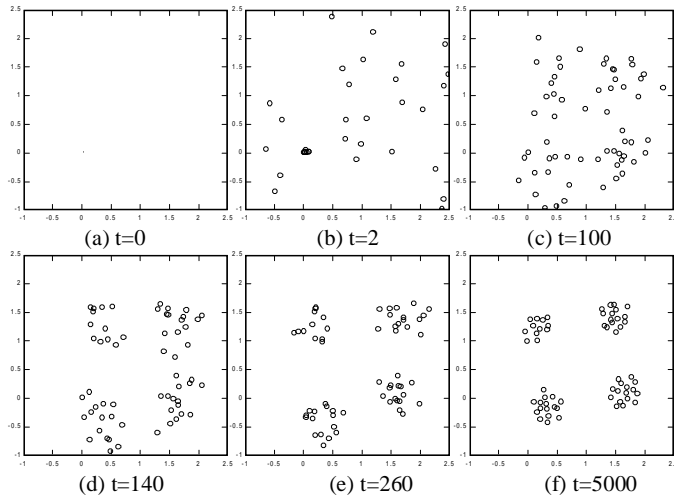


Figure 3. Learning process of the condensation algorithm when applied to the data set shown in Fig.2.

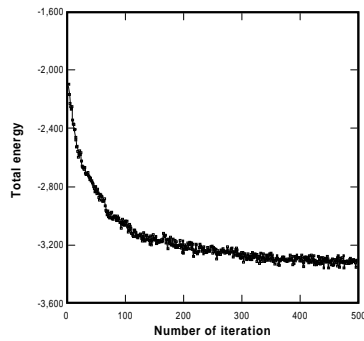


Figure 4. Decrease of total energy in the process of simulated annealing. The curve dropped quickly at the beginning of the simulation and then slowly converged to a minimum value.

Small fluctuations are observed as up-hill jumps are accepted with small probability.

3.2 Real data: Classification of plants

For further verification, the algorithm was applied to the Iris database, which is a classical example in the pattern recognition literature[8]. The Iris database was originally constructed for the taxonomy of plants by the use of multiple measurements[9]. Each sample is characterized by four attributes, namely sepal length, sepal width, petal length, and petal width. So each sample can be represented by a data point in 4-dimensional Euclidean space. The database was originally constructed by Anderson in 1935[9] and was used in a classical paper by Fisher in 1936[10]. The data set contains 3 classes of 50 samples each, where each class refers to a subtype of Iris. It is known that one class is linearly separable from the other two, which are not linearly separable from each other.

By using the condensation algorithm we obtained a 2-dimensional map given in Fig. 5. The first type is clearly separated from the other two that are not linearly separable. It is interesting to compare this result with the PCA projection by the first two principal components (Fig. 6). Two maps reflect very similar distributions of samples, not only in overall features like the separability of one class from the other two, but also in the relative location of individual samples with respect to others. As the third eigenvalue of PCA is very small compared with the first two, the PCA projection reveals true structures of the data. Obtaining a map similar to PCA projection in the case of Iris database thus shows the reliability of our method for building feature maps of multi-dimensional data sets.

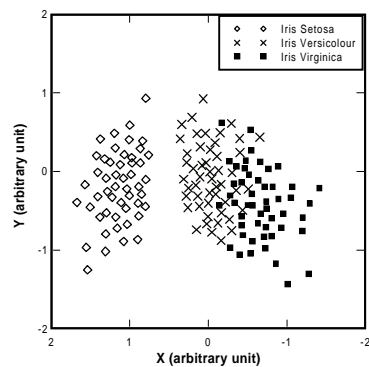


Figure 5. A map of the Iris database produced by using the condensation algorithm.

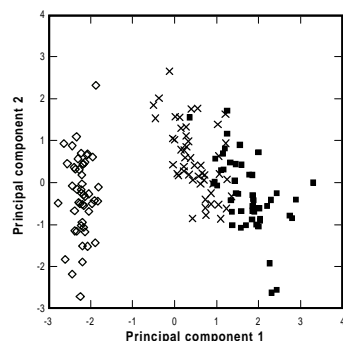


Figure 6. Projection of the Iris database by the first two principal components. Refer to Fig.5 for the meaning of symbols.

3.3 Repeatability

The condensation algorithm begins with a random configuration of atoms, in which the coordination of atoms are assigned by a pseudo-random number generator. In addition, the generator is frequently used to generate random moves of atoms for the purpose of simulated annealing. At the beginning of calculation, a seed is provided to the generator, which generates pseudo-random numbers one by one.

So what happens if we calculate using the same database for several times with different seeds for the random number generator? To answer this question, such calculations have been carried out and the results are given in Fig.7. The raw database is a database of gene expression patterns from [3], which will be discussed in more details in the next section. Instead of discussion on the detailed data structure, the focus here is whether the algorithm can discover maps with similar features. As shown by Fig.7, repeated calculations result in maps that reveal similar data structures. Note that the map has an extra degree of freedom with respect to rotation, which is being decided randomly and has no direct relation with the raw database. Despite differences in the details of individual data points, all calculations suggest the existence of roughly three classes. This means that the map of condensation algorithm reflects true data structures, which is repeatable to some extent.

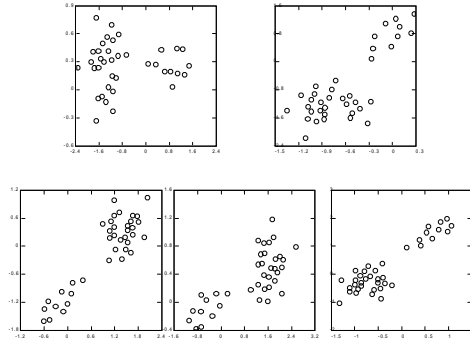


Figure 7. Repeatability of the condensation algorithm.

4. APPLICATION TO GENE EXPRESSION PATTERNS

As a practical application, the method is applied to the classification of cancer according to gene expression patterns. Here we select acute leukemia as an example. It has been recognized that this disease has two subtypes, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Distinguishing ALL from AML is critical for successful treatment. We used the data of 72 patients (47 ALL and 25 AML) from reference [3], including both the initial set and the independent set. For each sample, the data set contains the expression values of 7129 genes obtained by using DNA microarrays. As most of these genes are inactive and show no variation among samples, the raw data is subject to a variation filter. We first restricted all the expression levels to a limited positive range, (200, 10000), and then excluded genes with less than fivefold variation across the collection of samples. Genes are also eliminated if the standard deviation across samples is less than 300. Finally, 1606 genes left and the values of expression levels are log-transformed using base 10. We then calculated the similarity function between these expression patterns. Following Eisen *et al.*[11], we found the standard correlation coefficients (i.e., the dot product of two normalized vectors) yields better results for distinguishing subtypes of cancers than Euclidean distance does. This is because the dot product captures similarity in “shape” but places no emphasis on the magnitude.

4.1 Results

Figure 8 shows the mapping discovered by using the condensation algorithm. Clinical information is also shown by different symbols. Samples of two subtypes, AML and ALL, approximately form two clusters. However, the two clusters are adjacent to each other. The distinction is not clear, since there are several samples in the middle of clusters. Our result is in agreement with neighborhood analysis which yields very small prediction strength (a parameter that measures the reliability of prediction [3]), or even mis-predictions, for those samples. For example, sample No. 67, which is ALL according to clinically diagnosis, was predicted as AML with prediction strength 0.15. Unsupervised grouping by SOM also mis-classified three ALL samples into the AML group and one AML sample into the ALL group[3]. Although the ALL-AML classification based on the analysis of gene expression patterns will not be 100% accurate, Fig.8 indicates a strong statistical correlation between ALL-AML distinction and gene expression pattern.

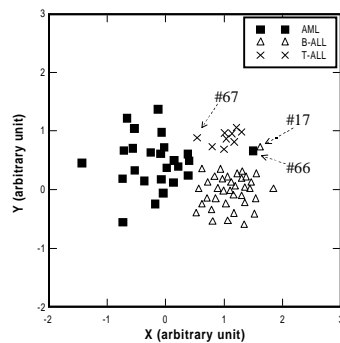


Figure 8. A map of gene expression patterns obtained by the condensation algorithm. Each point represents a patient sample. It can be seen that not only two subtypes of acute leukemia but also further difference within one subtype are featured.

As shown in Fig. 8, molecular heterogeneity is observed within both AML and ALL groups. However, the heterogeneity of AML samples is random, whereas ALL samples are further divided into two sub-regions, corresponding to T- and B-cell lineage. T-cell ALL samples are clearly separated from B-cell ALL. This shows that capability of the condensation algorithm to build feature maps that reflect the hierarchical structures of data sets.

Some samples are unreasonably mapped in Fig. 8. For example, despite its clinical diagnosis of AML, sample No. 66 was found to be more similar

to the ALL group than it is to the AML group. Additionally, its location is nearer to the T-cell side. Neighborhood analysis also incorrectly predicted this sample as ALL[3] and the prediction strength (0.27) is very close to the threshold value (0.3) for making predictions. We also found that sample No. 17, which is said to be a B-cell ALL, is more similar to T-cell ALL. When compared with the classification of a four-node SOM, we also found that one B-cell ALL sample is mis-classified into the T-cell ALL. Therefore, some of the irregularities in Fig. 8 reflect true irregularities in the raw data.

The above interpretation of Fig. 8 has been made with the help of clinical information. Without this information, it could be difficult to classify these samples merely from a plot like Fig. 8 with the same symbol for all patient samples. On the one hand, partitioning could be possible if more samples are available. On the other hand, this reflects the limitation of our present algorithm and the necessity of combining the technique with clustering algorithm such as SOM. The condensation algorithm provides means for direct observation of statistical distribution of raw data, from which one could possibly determine the suitable number of clusters to be used in SOM classification.

4.2 Comparison with PCA

For comparison, we also applied PCA to the same data set. Figure 9 shows a projection according to the first two principal components. It is found that samples of AML and ALL are also linearly separable by the projection. Even the first principal along would serve as a fairly good indicative. Again, this shows a strong correlation between AML-ALL distinction and gene expression patterns. The location of sample No. 66 and No. 67 are also irregular, similar to the plot obtained by the condensation algorithm in Fig. 8. Nevertheless, in the PCA projection B- and T-cell ALL samples are not separable. This may be because the hyperplane that separates them deviates from the plane defined by the first two principal components. This indicates the limitation of PCA as a linear method for dimensionality-reduction. As shown by Fig. 8, the condensation algorithm is essentially nonlinear and is able to show correlation among data points occurring at different hyperplanes.

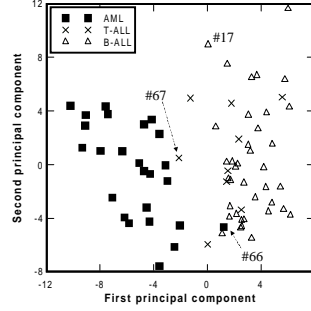


Figure 9. A projection of gene expression patterns by using principal component analysis.

5. DISCUSSION

In fact, the condensation algorithm are closely related to a group of nonlinear projection method called multidimensional scaling (MDS) which also construct low dimensional maps of multidimensional data items by using pairwise dissimilarity measures[12]. The key idea of MDS is to represent each data vectors by a point in lower dimensional space such that the distance between these points resembles the distance between data points in the multidimensional space, which is also interpreted as a measure of dissimilarity between patterns. In a standard metric MDS[13], the mapping is obtained by minimizing the following objective function,

$$E_{MDS} = \sum_{i,j(i \neq j)} (r_{ij} - S_{ij})^2, \quad (6)$$

where r_{ij} is the distance between point i and j in the feature map and S_{ij} is distance of the corresponding data vectors in the multidimensional space. The pairwise contribution to this cost function is

$$E_{ij} = (r_{ij} - S_{ij})^2. \quad (7)$$

Considering E_{MDS} in Eq. 6 as total energy of the system of “atoms”, we can treated E_{ij} as pair potentials in the condensation algorithm. The pair potential defined by Eq. 7 are plotted in Fig. 6 for some given values of S_{ij} . These curves can be obtained by simply transforming the parabolic curve $y = x^2$ along the x-axis, with one single minimum at $r_{ij} = S_{ij}$.

In comparison with the pair potential of the condensation algorithm shown in Fig. 1, this potential does not decay for large r . The advantage of parabolic function is that it enables more faithful maps. However, for complex data structures, it can lead to high levels of “stress” in the map. Like pair potentials widely used in the modelling of solids[4], the potential function of the condensation algorithm (Eq. 4) decays exponentially with distance and approaches zero when two atoms are separated by a large distance. Although this might result in large distances between otherwise “near” data points, it makes possible more flexible mapping with low levels of “stress”.

Another source of difference between metric MDS and the condensation algorithm comes from the dependence of the shape of curve on the parameter S_{ij} . The curves shown in Fig. 10 are simply transformation from the parabolic curve, thus the shapes of the curves are all the same. But in Fig. 1, curves are not only transformed but also deformed for different S_{ij} . The result of the deformation is that the range of interaction for smaller S_{ij} is shortened, while the range of interaction for larger S_{ij} is extended. As a consequence, the repulsive interaction between less similar patterns is emphasized in the condensation algorithm.

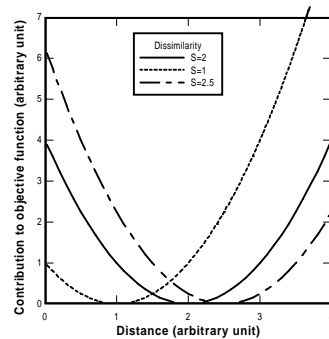


Figure 10. Pairwise contribution to the objective function in multidimensional scaling (MDS)[11]. This corresponds to the pair potential in the condensation algorithm given in Fig.1. These plots show the main difference between the condensation algorithm and conventional metric MDS algorithms.

These differences between potential functions are the main causes of differences between the condensation algorithm and conventional MDS algorithms. To demonstrate these differences, we also apply the conventional MDS algorithm to the database of gene expression patterns and obtain a map shown in Fig.11. The MDS map is closer to the PCA projection than the map of condensation algorithm. Taken the linear projection of PCA as a standard,

the MDS map is more faithful than the map of condensation algorithm. As discussed above, the reason for this is because the former uses a more faithful pair potential.

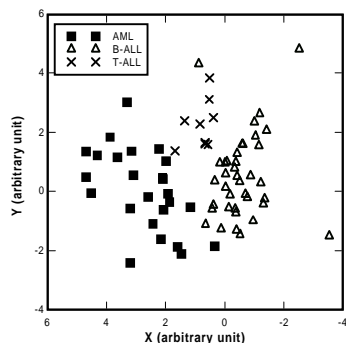


Figure 11. Map of gene expression patterns by using multidimensional scaling (MDS).

In Fig.11, three types of patient samples are also mapped into clusters of points. But in Fig. 10 these clusters are more visible as the distance between within-class members are shortened and those between members of different classes are enlarged. Our algorithm is more suitable for the detecting of the possible existence of clusters within a large set of samples. This property is due to the deformation of the pair potential curves as discussed above. This is a desirable property for the analysis of complex data structures when the goal is to get insights on more global distributions of the data set and fidelity is of less importance. For the class discovery of cancers using gene expression profiling, a map produced by the condensation algorithm would be more helpful in guessing the number of clusters presented in a given data set.

Our calculation converged quickly with up to several hundreds of data points and yielded repeatable results. For larger databases, however, the condensation algorithm may require more computational resources, as simulated annealing is a computationally expensive method for optimization. Also, the applicability of the algorithm should be limited to relatively simple data manifolds because it is almost impossible to map complex high-dimensional manifolds onto 2-dimensional maps.

Also, the similarity matrix only conveys a part of the information provided by the raw data. The final graph produced by the condensation algorithm, again, captures only a part of the information provided by the similarity matrix. While the loss of information is a general feature for techniques of abstraction and knowledge discovery, one should always be warned with the limitations of such techniques.

6. CONCLUSION

A condensation algorithm is developed for the visualization of multi-dimensional data by 2- or 3-dimensional maps. The objective function is inspired by the concept of pair potentials widely used in the modelling of solids. This algorithm enables the construction of low dimensional maps from dissimilarity measures. The nonlinearity of this algorithm makes it more suitable for the discovery of complex data structures, in comparison with the linear projection of PCA. Unlike MDS, the new algorithm emphasizes the repulsive interaction between less similar objects, and produces maps that are more likely to represent data items by several clusters. When applied to the leukemia data set[3], it produces a snapshot that reveals true data structures, which could be helpful in deciding the number of subclasses embedded in the data. The new algorithm thus can be used as a complement to clustering analysis methods like SOM.

ACKNOWLEDGEMENTS

We would like to thank Yiming Mi, Shin-ichi Yonamine, and Yoko Kobune for helpful discussions. X.J.G. wants to thank Simon Lin of Duke University Medical Center for discussions on multidimensional scaling (MDS). We also thank the anonymous reviewer for in-depth comments and suggestions on our manuscript.

REFERENCES

- [1] Lockhart, D. J. and Winzeler, E. A., Genomics, gene expression and DNA arrays, *Nature*, 405, 827 (2000).
- [2] Alizadeh, A. A. et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, *Nature*, 403, 503 (2000).
- [3] GOLUB, T. R. et al.: Molecular classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science*, 286, 531 (1999).
- [4] Kittel, C.: *Introduction to solid state physics*, 7th ed., Wiley, New York (1996).
- [5] Rose, J. H., Smith, J. R., Guinea, F., and Ferrante, J.: *Physical Review B* 29, 2963 (1984).
- [6] Ge, X.J., Chen, N.X, Zhang, W.Q., and Zhu, F.W.: Selective field evaporation in field-ion microscopy for ordered alloys, *J. of Appl. Phys.*, 85, 3488 (1999).
- [7] Van Laarhoven, P. J. M.: *Theoretical and computational aspects of simulated annealing*, Centrum voor Wiskunde en Informatica, Amsterdam (1988).
- [8] Duda, R. O.: & P. E. Hart, *Pattern Classification and Scene Analysis*, (John Wiley & Sons, NY, 1973).
- [9] Anderson, E. : The Irises of the Gaspe Peninsula, *Bulletin of the American Iris Society*, 59, 2(1935).

- [10] Fisher, R. A.: The use of multiple measurements in taxonomic problems, *Annual Eugenics*, 7, Part II, 179-188 (1936); also in *Contributions to Mathematical Statistics*, (John Wiley, NY, 1950).
- [11] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D.: Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci. USA*, 95, 14863 (1998).
- [12] Kruskal, J. B. & Wish, M. : *Multidimensional Scaling*. Sage Publications, Beverly Hills, Calif. (1978).
- [13] Torgerson, W. S.: *Multidimensional Scaling : I. Theory and method*, *Psychometrika*, 17 401 (1952).