Visual Terrain Analysis of High-dimensional Datasets (Technical Report, TRC04/06)

Wenyuan Li[1], Kok-Leong Ong[2], and Wee-Keong Ng[1]

[1] Nanyang Technological University, Centre for Advanced Information Systems
Nanyang Avenue, N4-B3C-14, Singapore 639798
liwy@pmail.ntu.edu.sg, awkng@ntu.edu.sg

[2] School of Information Technology, Deakin University
Waurn Ponds, Victoria 3217, Australia
leong@deakin.edu.au

Abstract. Most real-world datasets are, to a certain degree, skewed. When considered that they are also large, they become the pinnacle challenge in data analysis. More importantly, we cannot ignore such datasets as they arise frequently in a wide variety of applications. Regardless of the analytic, it is often that the effectiveness of analysis can be improved if the characteristic of the dataset is known in advance. In this paper, we propose a novel technique to preprocess such datasets to obtain this insight. Our work is inspired by the resonance phenomenon, where similar objects resonate to a given response function. The key analytic result of our work is the *data terrain*, which shows properties of the dataset to enable effective and efficient analysis. We demonstrated our work in the context of various real-world problems. In doing so, we establish it as the tool for preprocessing data before applying computationally expensive algorithms.

1 Introduction

The subfield of data analysis is essentially a collection of algorithms that focused on analyzing large datasets of high-dimensionality. Often than not, the cornerstone of these algorithms is to address the dimensionality curse when trying to provide effective and efficient results for a given user query. Towards this, there have been many research done; including cluster analysis to find clusters embedded in subspaces (also known as *subspace clustering* or *biclustering*), and dimensionality reduction.

In cluster analysis, most models are based on distance or similarity measures, or correlation measures of feature subsets or objects. While they unveil the details of subspace clusters, most are of no interest to the user. For example, more than $10,000$ clusters were obtained through OP-clustering [1] on a drug activity dataset with a dimension of $10,000 \times 30$. Clearly, this is overwhelming to the user trying to find insights about the data in question, e.g., the relationship among patterns rather than a list of patterns. Usually, closer inspection would suggest close relationships among clusters. And if high level insights is what the user is after, then this level of pattern redundancy would be inappropriate. Yet, a combinatorial explosion of patterns (satisfying the query) occur as the size and dimensionality of the dataset increases.

Dimension reduction is one alternative to 'curb' the combinatorial explosion of patterns by passing a reduced space to the analytical algorithms. This reduced space often produce results that are close to that of analyzing the original dataset. In fact, it is possible (and often desirable from the users' standpoint) to visualize data involving a smaller number of dimensions. The drawback, however, is the loss of patterns embedded in the subspace of the original space. This happens because most reduction techniques made use of distance or similarity measures over the full dimension, and therefore lack the mechanism to find the embedded patterns that are subtle but important. From the users' standpoint, this meant rerunning time-consuming dimension reduction on the original data each time a different level of detail is desired.

In this paper, we introduce the concept of *data terrain* to visualize high-dimensional datasets while overcoming the limitations of subspace clustering and dimension reduction. Our proposal effectively reveals the relationship among subspace clusters, and allows the user to explore the data at different levels of details. We show, by means of real-world applications (e.g., biclusters, outliers, and frequent itemsets), how the data terrain can help discover generic patterns that can be utilized to effectively analyze the patterns embedded in the original space. Unfortunately, to find this data terrain under varying conditions proved to be NP-hard. Thus, our contribution in this paper includes the proposal of efficient techniques to find the data terrain.

We next show a motivating example to illustrate the relevance of data terrains in analysis. We then prove that finding this data terrain is NP-hard in Section 3, and therefore motivates alternative approaches. To find the terrain at the macro-level, we introduced the resonance model in Section 4 and in Section 5, we show how the terrain at the micro-level can be obtained. We discuss related works in Section 6 and finally, we conclude our work in Section 7.

2 Motivating Example

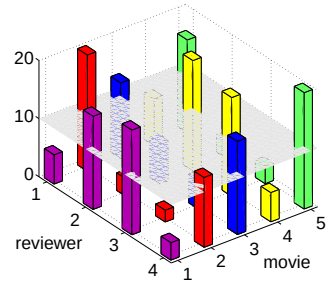
We begin by introducing the concept of data terrain and show by means of an example, how it facilitates better data analysis; and why it is better than other techniques like subspace clustering and dimension reduction. Our example is based on the survey of popular movies. Figure 1(a) shows the rating matrix W of 4 reviewers (R_i) on 5 movies (M_j), where each movie is rated on a scale of 1 to 20.

We first use biclustering to analyze the relationship between the reviewers and the movies. If we require each bicluster to involve at least 2 rows and columns, more than 10 biclusters can be discovered. Figure 1(c) – (f) are the distinct biclusters found in this case. While these biclusters precisely characterized the reviewers' 'rating style' on a movie, there is too much redundancy in the solution for such a small dataset. In real-world situations where the dataset is much larger, it will take much longer before the analyst is able to come to a conclusion about the characteristics of reviewers' 'rating style'. Worse, the analyst is likely to become confused on the real relationship between the reviewers and the movies if they were to work at this level of detail.

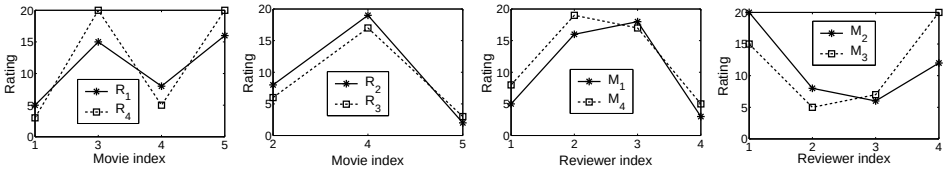
The other approach would be to use dimension reduction techniques. We consider using a popular technique known as Principal Component Analysis (PCA). The PCA's output is shown in Figure 1(g) and (h). Again, the relationship between the reviewers

	M_1	M_2	M_3	M_4	M_5
R_1	5	**20**	**15**	8	**16**
R_2	**16**	8	5	**19**	2
R_3	**18**	6	7	**17**	3
R_4	3	**12**	**20**	5	**20**

(a) original matrix W



(b) 3D of W



(c) bicluster 1 (d) bicluster 2 (e) bicluster 3 (f) bicluster 4

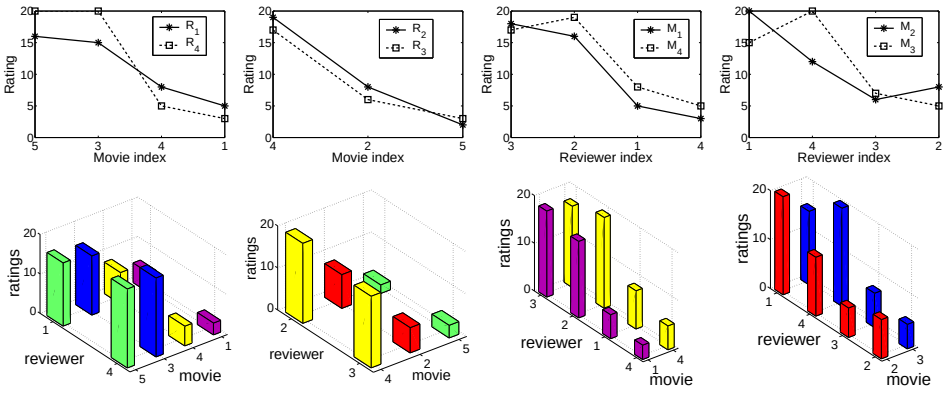
(g) relationships among movies (h) relationships among reviewers

Fig. 1. The rating matrix for 4 reviewers and 5 movies

and movies is revealed. However, as PCA runs across the full dimension of the data, we loose many subtle and local insights about reviewers and movies. For example, movies M_2, M_4 and M_5 are perceived as different in Figure 1(g). Yet, if we check back on our analysis using biclustering, we can see from Figure 1(d) that they are actually quite similar if we consider the ratings from reviewer R_2 and R_3. Thus, if only PCA is performed, we will not be able to arrive at this conclusion.

Interestingly, if we view W in 3D space (i.e., the *rating* becomes the third dimension), we can capture the relationships that both biclustering and PCA revealed. As Figure 1(b) shows, a direct 3D 'plot' of W does not seem to reveal any interesting insights – but if we were to reorder W into W' as shown in Figure 2(e) (and also Figure 2(a) – (d); where every bicluster can be shown in this manner [1]), we have a 3D terrain of W' as depicted in Figure 2(f). Notably, this terrain provides the insights that earlier requires both biclustering and PCA analysis.

To illustrate this, notice that any bicluster from Figure 2(a) – (d) can be obtained by selecting some points from the 'mountains' and 'plains' in this terrain. At the same time, we can also make conclusions that would otherwise be obtained through PCA: (i) there are primarily two groups of movies and reviewers; (ii) M_3 and M_5 have higher



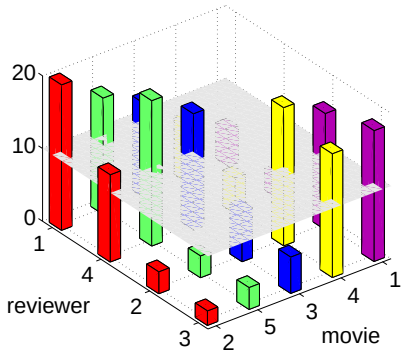
(a) bicluster 1 (b) bicluster 2 (c) bicluster 3 (d) bicluster 4

	M_2	M_5	M_3	M_4	M_1
R_1	**20**	**16**	**15**	8	5
R_4	**12**	**20**	**20**	5	3
R_2	6	3	7	**19**	**16**
R_3	8	2	5	**17**	**18**

	M_5	M_3	M_2
R_4	20	20	12
R_1	16	15	20

	M_2	M_3	M_5
R_2	8	5	2
R_3	6	7	3

Mountain 1



(e) the matrix W' of reordered W (f) 3D of W'

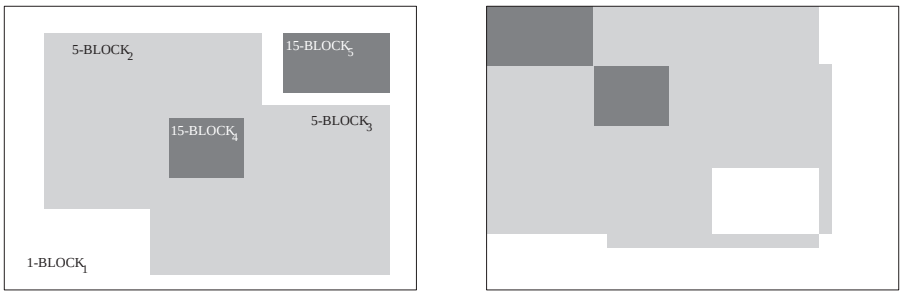
Fig. 2. The rating matrix reordered for 4 reviewers and 5 movies

similarity than M_2 despite being in the same group; and (iii) the 'rating style' of R_2 and R_3 is opposite that of R_1 and R_4. Thus, the terrain captures both local and global relationships about the data in an intuitive and effective manner. Of course, real-world datasets are much more complex that result in more complicated terrains. Consequently, trying to discover such a terrain in large and complex dataset proved to be a difficult problem. As we will show next, this problem is NP-hard.

3 Preliminaries

Conceptually, moving from W to W' is simply the reordering of the matrix to form the 'mountains' and 'plains'. Yet, this ordering on both dimensions can be difficult to achieve efficiently on massive datasets. To prove the hardness of this problem, we first give the following definitions.

Let $\mathcal{O}$ be a set of objects, where $o \in \mathcal{O}$ is defined by a set of attributes $\mathcal{A}$. Further, let w_{ij} be the magnitude of o_i over $a_j \in \mathcal{A}$. Then we can represent the relationship of



(a) BLOCKs and ideal *terrain* of data

(b) Approximation of ideal *terrain* of data by Macro-view

Fig. 3. Illustrations of σ-BLOCKs

all objects and their attributes in a matrix $W = (w_{ij})_{|\mathcal{O}|\times|\mathcal{A}|}$ for the weighted bipartite graph $G = (\mathcal{O}, \mathcal{A}, E, W)$, where E is the set of edges. Thus, discovering the 'mountains' transforms into the problem of evaluating subgraphs where the magnitude of all its edges are above some 'altitude', i.e., $w_{ij} \geqslant \sigma$. Formally, the concept of a 'mountain' in this data terrain is called a BLOCK.

Definition 1. *Given a weighted bipartite graph G, a σ-**BLOCK** (or simply σ-B) is a subgraph $G' = (\mathcal{O}', \mathcal{A}', E', W')$ of G satisfying $w_{ij} \geqslant \sigma$ for any $i \in \mathcal{O}'$ and $j \in \mathcal{A}'$.*

From Definition 1, σ-B can be intuitively viewed as a plane (or a transverse section) with a specified altitude σ that 'cuts' across W. In the case of Figure 1(b), we set the plane at σ=10 to obtain two 10-Bs as shown in Figure 2(f): $\{R_1, R_4, R_2\} \times \{M_2, M_5\}$ and $\{R_2, R_3\} \times \{M_4, M_1\}$. Therefore, a series of σ-Bs can be generated when considering planes with different σ values. Once this set of BLOCKs relevant to G is found, we can order them to find the data terrain.

Definition 2. *Given a bipartite graph $G = (\mathcal{O}, \mathcal{A}, E, W)$ and a set of BLOCKs $\{B_1, B_2, \ldots, B_k\}$ found from G, the terrain of W is two ordered sequences of $\mathcal{O}$ and $\mathcal{A}$, such that these BLOCKs are placed consecutively in the reordered W.*

It is interesting to note that sorting both dimensions, i.e., $\mathcal{O}$ and $\mathcal{A}$, is an extension of sorting a single dimensional array to determine its distribution. However, sorting both dimensions simultaneously to get the 2-dimensional distribution is practically infeasible, i.e., finding the σ-BLOCKs by iteratively decreasing σ from the maximum value of W is NP-hard. In fact, finding a single σ-B is NP-hard.

Theorem 1. *Finding the largest σ-BLOCK ($|\mathcal{O}'|\times|\mathcal{A}'|$) is NP-hard.*

Proof. Our problem can be reduced from the *maximum edge biclique* [2], which is NP-complete. Given a bipartite graph $B = (\mathcal{O}, \mathcal{A}, E^B)$ and a positive integer k, we form a weighted bipartite graph $G = (\mathcal{O}, \mathcal{A}, E^G, W)$, where $w_{ij} = 1$ if and only if $e_{ij} \in E^B$; and $w_{ij} = 0$ otherwise for any $i, j > 0$. We then set $\sigma = 1$ to find the largest σ-BLOCK, i.e., $G' = (\mathcal{O}', \mathcal{A}', E^{G'}, W')$ of G. Clearly, in this largest σ-BLOCK, if $|\mathcal{O}'|\times|\mathcal{A}'| \geqslant k$, then there exists a biclique in B with at least k edges.

4 Discovering Data Terrains: Macro-Views

Given the difficulty of finding σ-Bs, we seek alternative methods to discover the data terrain. Since our objective is to find the 'mountains' and 'plains' but *not* where they are on the terrain, then some approximation to the actual terrain (that is computationally efficient) should suffice. The insignificance of the specific locations of the 'mountains' and 'plains' can be demonstrated from Figure 3(a) and (b), where the same set of insights are obtained from both figures. As this terrain is approximated, we called it the macro-view. To obtain the macro-view of a terrain for a dataset, we used a novel model inspired by the physics of resonance. This resonance model is very efficient even on very large and high-dimensional datasets.

To understand the rationale behind its efficiency, we can discuss a simple analogy. Suppose we are interested in finding the characteristics of students who are fans of thriller movies. One way is to poll each student. Clearly, this is time-consuming. A better solution is to gather a sample but we risk acquiring a wrong sample that leads to a wrong finding. A smarter approach is to announce the free screening of a blockbuster thriller. In all likelihood, the fans of thrillers will turn up for the screening. Despite the possibility of 'false positives', this sample is easily and quickly obtained with minimum effort. Hence, rather then finding single σ-B iteratively, we could recreate a similar effect as in the case of collecting a sample for thriller fans.

The scientific model that corresponds to the above is the principle of resonance. In other words, instead of checking every σ-B, we can simulate a resonance experiment by injecting a response function to elicit objects of interest to the analyst. In our analogy, this response function is the blockbuster thriller that fans automatically react to by going to the screening. In sections that follow, we present the model and discuss its properties and support practicality of the model by discussing how it improves analysis using some real-world applications.

4.1 The Model

To simulate a resonance phenomenon, we require a forcing object $\tilde{o}$, such that when an appropriate response function $\mathtt{r}$ is applied, $\tilde{o}$ will resonate to elicit those objects $\{o_i, \dots\} \subset \mathcal{O}$ in G, whose 'natural frequency' is similar to $\tilde{o}$. This 'natural frequency' represents the characteristics of both $\tilde{o}$ and the objects $\{o_i, \dots\}$ who resonated with $\tilde{o}$ when $\mathtt{r}$ was applied. For the weighted bipartite graph $G = (\mathcal{O}, \mathcal{A}, E, W)$ and $W = (w_{ij})_{|\mathcal{O}|\times|\mathcal{A}|}$, this 'natural frequency' of $o_i \in \mathcal{O}$ is $\mathbf{o_i} = (w_{i1}, w_{i2}, \dots, w_{i|\mathcal{A}|})$. Since a one-dimensional array (or vector) can be sorted to obtain its own terrain, we also refer $\mathbf{o_i}$ as the terrain of the object o_i. Likewise, the terrain of the forcing object $\tilde{o}$ is defined as $\tilde{\mathbf{o}}_\mathbf{i} = (\tilde{w_1}, \tilde{w_2}, \dots, \tilde{w_{|\mathcal{A}|}})$.

Put simply, if two objects of the same 'natural frequency' will resonate and therefore, should have a similar terrain. The evaluation of resonance strength between objects o_i and o_j is given by the response function $\mathtt{r}(\mathbf{o_i}, \mathbf{o_j}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. We defined this function abstractly to support different measures of resonance strength. For example, one existing measure to compare two terrains is the well-known *rearrangement inequality theorem*, where $\mathtt{I}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i y_i$ is maximized when the two positive sequences $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ are ordered in the same way (i.e.

$x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and $y_1 \geqslant y_2 \geqslant \cdots \geqslant y_n$) and is minimized when they are ordered in the opposite way (i.e. $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and $y_1 \leqslant y_2 \leqslant \cdots \leqslant y_n$).

Notice if two vectors maximizing $\mathbf{I}(\mathbf{x}, \mathbf{y})$ are put together to form $M = [\mathbf{x}; \mathbf{y}]$ (in MATLAB format), we obtain the terrain. More importantly, all σ-Bs are immediately obtained from this terrain with the need to search every σ-B! This is why the model is efficient – it only needs to consider the resonance strength among objects once the appropriate response function is selected. For example, the response function $\mathtt{I}$ is a suitable candidate to characterize the similarity of terrains of two objects. Likewise, $\mathtt{E}(\mathbf{x}, \mathbf{y}) = \exp(\sum_{i=1}^{n} x_i y_i)$ is also an effective response function.

To find the 'mountains' and 'plains', the forcing object $\tilde{o}$ evaluates the resonance strength of every objects o_i against itself to locate a 'best fit' based on the contour of its terrain. By running this iteratively, those objects that resonated with $\tilde{o}$ are discovered and placed together to form the 'mountains' within the 2-dimensional matrix W. In the same fashion, the 'plains' are discovered by combining those objects that resonated weakly with $\tilde{o}$. This iterative learning process between $\tilde{o}$ and G is outlined below.

Initialization Set up $\tilde{o}$ with a uniform distribution: $\tilde{\mathbf{o}} = (1, 1, \ldots, 1)$; normalize it as $\tilde{\mathbf{o}} = \mathtt{norm}(\tilde{\mathbf{o}})$[3]; then let $k = 0$; and record this as $\tilde{\mathbf{o}}^{(0)} = \tilde{\mathbf{o}}$.

Apply Response Function For each object $o_i \in \mathcal{O}$, compute the resonance strength $\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_i})$; store the results in a vector $\mathbf{r} = \big(\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_1}), \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_2}), \ldots, \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o}_{|\mathcal{O}|})\big)$; and then normalize it, i.e., $\mathbf{r} = \mathtt{norm}(\mathbf{r})$.

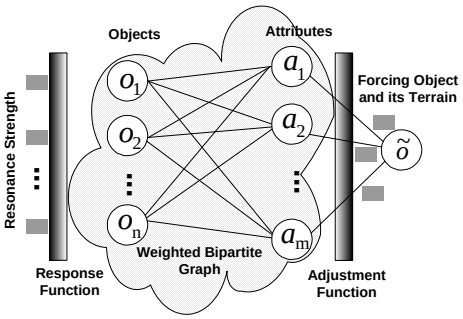
Adjust Forcing Object Using $\mathbf{r}$ from the previous step, adjust the terrain of $\tilde{o}$ for all $o_i \in \mathcal{O}$. To do this, we define the adjustment function $\mathtt{c}(\mathbf{r}, \mathbf{a_j}) : \mathbb{R}^{|\mathcal{O}|} \times \mathbb{R}^{|\mathcal{O}|} \to \mathbb{R}$, where the weights of the j-th attribute is given in $\mathbf{a_j} = (w_{1j}, w_{2j}, \ldots, w_{|\mathcal{O}|j})$. For each attribute a_j, $\tilde{w_j} = \mathtt{c}(\mathbf{r}, \mathbf{a_j})$ integrates the weights from $\mathbf{a_j}$ into $\tilde{o}$ by evaluating the resonance strength recorded in $\mathbf{r}$. Again, $\mathtt{c}$ is abstract, and can be materialized using the inner product $\mathtt{c}(\mathbf{r}, \mathbf{a_j}) = \mathbf{r} \bullet \mathbf{a_j} = \sum_i w_{ij} \cdot \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_i})$. Finally, we compute $\tilde{\mathbf{o}} = \mathtt{norm}(\tilde{\mathbf{o}})$ and record it as $\tilde{\mathbf{o}}^{(k+1)} = \tilde{\mathbf{o}}$.

Test Convergence Compare $\tilde{\mathbf{o}}^{(k+1)}$ against $\tilde{\mathbf{o}}^{(k)}$. If the result converges, go to the next step; else apply $\mathtt{r}$ on $\mathcal{O}$ again (i.e., forcing resonance), and then adjust $\tilde{o}$.

Macro-view of *terrain* Sort the objects $o_i \in \mathcal{O}$ by the coordinates of $\mathbf{r}$ in descending order; and sort the attributes $a_i \in \mathcal{A}$ by the coordinates of $\tilde{\mathbf{o}}$ in descending order.

We denote the resonance model as $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{r}, \mathtt{c})$, where the instances of functions $\mathtt{r}$ and $\mathtt{c}$ can be either $\mathtt{I}$ or $\mathtt{E}$. Interestingly, the instance $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ is actually the HITS algorithm [3], where W is the adjacency matrix of a directed graph. However, this instance is actually different from HITS in 3 ways: (i) the objective of our model is to obtain an approximate terrain of the dataset (through the resonance simulation), while HITS is designed for Web IR and looks at only the top k authoritative Web pages (a reinforcement learning process); (ii) the implementation is different by the virtue that our model is able to use a non-linear instance, i.e., $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$, to discover data terrains while HITS is strictly linear; and (iii) we study a different set of properties and functions from HITS, i.e., *maximum edge biclique*, and the data terrain.

[3] $\mathtt{norm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$, where $\|\mathbf{x}\|_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$ is 2-norm of vector $\mathbf{x} = (x_1, \ldots, x_n)$.



(a) architecture of resonance model

	M_3	M_2	M_4	M_5	M_1
R_1	**15**	**20**	8	**16**	5
R_4	**20**	**12**	5	**20**	3
R_3	7	6	**17**	3	**18**
R_2	5	8	**19**	2	**16**

(b) reordered matrix from Figure 1(a) given by the linear model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$

Fig. 4. Resonance model and effectiveness of linear model

4.2 Properties of the Model

We shall discuss some important properties of our model in this section. In particular, we show that the model gives a good approximation to the actual terrain, and that its iterative process converges quickly. We will also discuss optimization, and provide the preliminaries to obtaining the micro-view of the data terrain.

Approximation to Actual Terrain The abstract view of the general model is given in Figure 4(a). Depending on the response and adjustment function, the abstract model instantiates into different implementations. In practice, we have the linear model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$, and the non-linear model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$. Using the synthetic data from Figure 3(a), we can see how well both implementations approximate the actual terrain.

After 4 iterations of the linear model, it converges to a precision of $\epsilon = 0.001$, i.e., once $\|\tilde{\mathbf{o}}^{k+1} - \tilde{\mathbf{o}}^{k}\| \leqslant \epsilon$, terminates. The reordered matrix is the same as Figure 3(b). In running the non-linear model, it converges after 2 iterations with the same precision and the results is the same as that of the linear model. We then performed the same test on the movie-rating example. This time, the linear model converges in 4 iterations – Figure 4(b), and the non-linear model in Figure 2(e) converges in 3 iterations. Obviously, the two results reordered by $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$ gives a better approximation, where the 'mountains' and 'plains' are easily distinguishable.

Thus, we can conclude that the different instances of R may give an approximate of the actual terrain, but they have different bias. In cases where the matrix is sparse, the linear model is preferred. Otherwise, the non-linear model performs better. These conclusions are also empirically proven in the later section.

Convergence Since the resonance model is iterative, it is essential that it converges quickly to be efficient. Essentially, the model can be seen as a type of *discrete dynamical system* [4]. Where the functions in the system is linear, then it is a *linear dynamical system*. For linear dynamical systems, it corresponds to eigenvalue and eigenvector computation [3–5]. Hence, its convergence can be proved by eigen-decomposition for

R where the response and adjustment functions are linear. In the non-linear case (i.e., $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$), its convergence is proven below.

Theorem 2. $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{r}, \mathtt{c})$*, where* $\mathtt{r}$*,* $\mathtt{c}$ *are* $\mathtt{I}$ *or* $\mathtt{E}$*, converges in limited iterations.*

Proof. When $\mathtt{r}$ and $\mathtt{c}$ are $\mathtt{I}$, we get $\tilde{\mathbf{o}}^{(k)} = \mathtt{norm}\big(\tilde{\mathbf{o}}^{(0)}(W^TW)^k\big)$ by linear algebra [5]. If A is symmetric and $\mathbf{x}$ is a row vector that is not orthogonal to the first eigenvector corresponding to the first largest eigenvalue of A, then $\mathtt{norm}(\mathbf{x}A^k)$ converges to the first eigenvector as k increases. Thus, $\tilde{\mathbf{o}}^{(k)}$ converges to the first eigenvector of W^TW. As the exponential function has Maclaurin series $\exp(x) = \sum_{n=0}^{\infty} x^n/n!$, the convergence of the non-linear model with $\mathtt{E}$ functions can be decomposed to the convergence of the model, when $\mathtt{r}$ and $\mathtt{c}$ are simple polynomial functions x^n.

So far, we note that either implementations converge quickly if a reasonable precision threshold ϵ is set. In practice, this is acceptable because we are only interested in the convergence of orders of coordinates in $\tilde{\mathbf{o}}^k$ and $\mathbf{r}^k$, i.e., we are not interested in how closely $\tilde{\mathbf{o}}^k$ and $\mathbf{r}^k$ approximate the converged $\tilde{\mathbf{o}}^*$ and $\mathbf{r}^*$. Furthermore, $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$ converges faster than $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$. Therefore, each iteration of learning is bounded by $\mathbf{O}(|\mathcal{O}| \times t_{\mathtt{r}} + |\mathcal{A}| \times t_{\mathtt{c}}))$, where $t_{\mathtt{r}}$ and $t_{\mathtt{c}}$ is the runtime of the response function $\mathtt{r}$, and the adjustment function $\mathtt{c}$ respectively. With k iterations, the final complexity is $\mathbf{O}(k \times (|\mathcal{O}| \times t_{\mathtt{r}} + |\mathcal{A}| \times t_{\mathtt{c}}))$. Since the complexity of $\mathtt{r}$ is $\mathbf{O}(|\mathcal{O}|)$ and $\mathtt{c}$ is $\mathbf{O}(|\mathcal{A}|)$, we have $\mathbf{O}(k \times |\mathcal{O}| \times |\mathcal{A}|)$. In our experiments, our model converges within 50 iterations even on the non-linear configurations giving a time complexity of $\mathbf{O}(|\mathcal{O}| \times |\mathcal{A}|)$. In all cases, the complexity is sufficiently low to efficiency handle large datasets.

Average Inter-resonance Strength $\frac{1}{\binom{k}{2}} \sum_{\substack{i \neq j \in \mathcal{O}' \\ |\mathcal{O}'|=k}} \mathtt{r}(\mathbf{o}_i, \mathbf{o}_j)$ among Objects Theorem 3 is in fact an optimization process to find the best k objects, whose average inter-resonance strength is the largest among any subset of k objects. In Section 4.3, we exploit this property to unveil the relationship between the macro-view of the data terrain and the biclusters. The following theoretical analysis provides the proof.

Lemma 1. *Given a row vector* $\mathbf{u} = (u_1, u_2, \ldots, u_n)$*, where* $u_1 \geqslant u_2 \geqslant \ldots \geqslant u_n \geqslant 0$*, we generate a matrix* $U = \lambda \mathbf{u}^T\mathbf{u}$*, where* $\lambda > 0$ *is a scale factor. We then define the* k*-sub-matrix of* U *as* $U_k = U(1:k, 1:k)$ *(in MATLAB format). Then,* U *has the following 'staircase' property*

$$|U_1| \geqslant |U_2| \ldots \geqslant |U_k| \ldots \geqslant |U_n| = |U| \tag{1}$$

where $|U|$ *of a symmetric matrix* $U = (u_{ij})_{n \times n}$ *is given as* $|U| = \frac{1}{\binom{n}{2}} \sum_{1 \leqslant i \neq j \leqslant n} u_{ij}$*.*

Proof. By induction, when $n = 2$ (base case), we prove $|U_2| \geqslant |U_3|$. Since $|U_2| = u_1u_2$, $|U_3| = \frac{1}{3}\big(u_1u_2 + u_1u_3 + u_2u_3\big)$ and $u_1 \geqslant u_2 \geqslant u_3 \geqslant 0$, we have $|U_2| \geqslant |U_3|$. When $n = k$, we prove $|U_k| \geqslant |U_{k+1}|$. We first define

$$x_{k+1} = \frac{1}{2k+1}\Big(u_{k+1}^2 + 2 \sum_{1 \leqslant i \leqslant k} u_i u_{k+1}\Big)$$

which after a straightforward calculation, we have the following

$$|U_{k+1}| \geqslant x_{k+1} \tag{2}$$

$$|U_{k+1}| - |U_k| = \frac{2k+1}{(k+1)^2}\Big(x_{k+1} - |U_k|\Big) \tag{3}$$

and finally from Equations (2) and (3), we have

$$|U_k| \geqslant |U_{k+1}|$$

Lemma 2. *Given a resonance space $R_{|\mathcal{O}|\times|\mathcal{O}|} = WW^T$ of $\mathcal{O}$, its first eigenvalue λ, and the eigenvector $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{R}^{1\times n}$, we have*

$$\begin{aligned} &\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^{1\times n} \\ &\|R - \lambda \mathbf{u}^T\mathbf{u}\|_F \leqslant \|R - \mathbf{x}^T\mathbf{y}\|_F \end{aligned} \tag{4}$$

where $\|\bullet\|_F$ denotes the Frobenius norm of a matrix.

Proof. We denote the the first singular value of A as s (its largest absolute value), and the corresponding left and right singular vectors as $\mathbf{p}$ and $\mathbf{q}$, respectively. By the Eckart-Young theorem, any given matrix $B_{n\times n}$ that satisfies the rank is 1. Therefore, we have

$$\|A - s\mathbf{p}\mathbf{q}^T\|_F \leqslant \|A - B\|_F$$

and by the symmetric property of A, it can be proved that $s = \lambda$ and $\mathbf{p} = \mathbf{q} = \mathbf{u}$. Rewriting the inequality will give us

$$\|A - \lambda\mathbf{u}\mathbf{u}^T\|_F \leqslant \|A - B\|_F \tag{5}$$

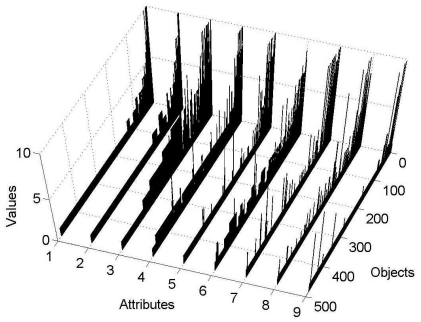
where for any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n\times 1}$, the rank of $\mathbf{x}\mathbf{y}^T$ is 1. Therefore, substituting $\mathbf{x}\mathbf{y}^T$ for B in the inequality (5) gives us Equation 4.

Theorem 3. *Given the macro-view terrain W', the average inter-resonance strength $\frac{1}{\binom{k}{2}}\sum_{1\leqslant i\neq j\leqslant k} \mathtt{r}(\mathbf{o}_i, \mathbf{o}_j)$ of the first k objects, w.r.t. the resonance strength with $\tilde{o}$, is largest for any subset with k objects.*

Proof. For linear models, i.e., $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$, $\mathbf{r} = \big(\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_1}), \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_2}), \ldots, \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o}_{|\mathcal{O}|})\big)$ converges to the first eigenvector $\mathbf{u}$ of WW^T, i.e. $\mathbf{r} = \mathbf{u}$ as shown in Theorem 2. And since the functions are linear, we can rewrite them as $WW^T = \big(\mathtt{r}(\mathbf{o}_i, \mathbf{o}_j)\big)_{|\mathcal{O}|\times|\mathcal{O}|}$. Further, since W and R are already reordered in descending order of their resonance strength $\mathbf{u}$, we have the following (together with Lemma 1 and Lemma 2)

$$|R_1| \geqslant |R_2| \ldots \geqslant |R_k| \ldots \geqslant |R_n| = |R| \tag{6}$$

and because $|R_k| = \frac{1}{\binom{k}{2}}\sum_{1\leqslant i\neq j\leqslant k} \mathtt{r}(\mathbf{o}_i, \mathbf{o}_j)$ is the average inter-resonance strength of the first k objects, we have Theorem 3.



(a) 3D *terrain* by macro-view

Top k records.	Number of malignant records (Approx. coverage)			
(% equiv.)	*terrain*	FindCBLOF	RNN	KNN
4 (1%)	4 (10%)	4 (10%)	3 (8%)	4 (10%)
16 (4%)	13 (33%)	14 (36%)	11 (28%)	9 (24%)
32 (8%)	26 (67%)	27 (69%)	25 (64%)	20 (51%)
48 (12%)	33 (85%)	35 (90%)	35 (90%)	26 (67%)
64 (16%)	37 (95%)	39 (100%)	36 (92%)	28 (72%)
100 (25%)	39 (100%)	39 (100%)	38 (97%)	28 (72%)

(b) comparison with three outlier methods

Fig. 5. Visualization of Wisconsin breast cancer data and comparison with outlier methods

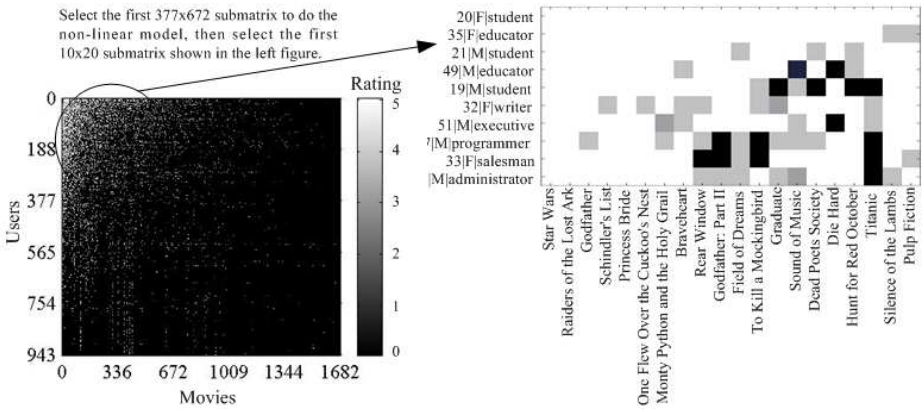
Approximation to Maximum Edge Biclique In the non-linear configuration of our model, i.e., $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$, we have another interesting property that is not available in the linear model: the approximation to the *maximum edge biclique*. Recall from the proof of Theorem 1, finding the largest σ-BLOCK can be reduced to finding the maximum edge biclique. Also, our empirical observations further confirmed this property of the non-linear model in finding the largest σ-B. Given the efficiency of our mode, it is therefore possible to find the actual terrain by running the model with different σ. We exploited this property to find the actual terrain, which we referred to as the micro-view that we shall discuss in Section 5.

4.3 Real World Examples

We will first show how a macro-view of the data terrain can help the user in analysis by using two real-world datasets. We then show how the macro-view of the terrain can have applications in data mining for finding frequent itemsets and biclustering.

Data Analysis by Macro-View of the Terrain We shall analyze two datasets to demonstrate the insights uncovered through the visualization of the macro-view terrain. The first is the Wisconsin breast cancer dataset (from the UCI Repository of Machines Learning Databases) with 699 instances, 9 attributes (each rated from 1 to 10), and two classes: benign (458 instances) and malignant (241 instances) [6]. This dataset is often used in testing outlier detection algorithms. Following the setup by He *et al.* [7], we removed some of the malignant records to form a test dataset with small number of outliers. The test dataset has 39 (8%) malignant records (as outliers) and 444 (92%) benign records.

In our scenario, the users may used our model to obtain the terrain so as to decide where the outliers are. Here, we used the non-linear configuration $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$ to visualize the data terrain as shown in Figure 5(a). By inspection, we see that only a few records (within the first 100) have attributes with large values, whereas the other records don't. Immediately, the user would be able to easily recognize the outliers' distribution and property. That is, the user is able to conclude that the *outliers are records whose*



(a) gray scale image of overall *terrain* by macro-view (b) gray scale image of local *terrain*

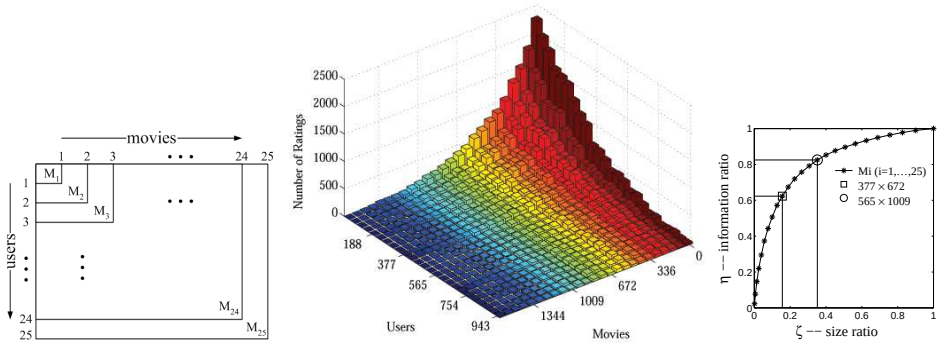
Fig. 6. Visualization of MovieLens data and analysis of a local *terrain* in detail

attributes have high values. Therefore, the user may choose to remove them by selecting the top k records in this terrain. Figure 5(b) shows the results of our approach against well-known outlier detection algorithms adapted from [7]. While the results speak for itself, it is worth to note that our approach allows the users to detect the outliers without much domain knowledge of the data.

The MovieLens dataset is our second dataset that is collected by the GroupsLen research project at the University of Minnesota (`http://www.grouplens.org`). It contains 943 users, 1,682 movies and 100,000 ratings (on the scale of 1 to 5), and each user is required to rate at least 20 movies. If the user is not interested in some movies, they will not rate them and this is recorded as a 0.

Again, we obtain the macro-view using $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ to get the terrain in Figure 6(a). Clearly, this dataset is very sparse and most of the ratings are concentrated on a small number of movies and users. If we divide the users and movies on the terrain into 25 equal intervals, as shown in Figure 6(a), we have 25 submatrices $(M_1, \ldots, M_{25})$ as shown in Figure 7(a). We then counted the number of ratings in each unit to get Figure 7(b). This 2-dimensional distribution now shows a clearer macro-view of the terrain. Upon further analysis of Figure 7(b), we obtain an exponential curve describing the percentage of size and rating in each submatrix M_i (with size $m_i \times n_i$, and number of ratings r_i), whose percentage of size is $\zeta_i = \frac{m_i \times n_i}{943 \times 1682}$, and percentage of ratings is $\eta_i = \frac{r_i}{100000}$. For example, the submatrix M_{15} in Figure 7(c) has a ratings of 82.5% while its size is surprisingly only 35.2%.

It is possible that the user is not satisfied with an overview of the dataset. Instead, the user might be interested in further analysis. From Figure 7, the user may further explore the 'crowded' but a region of the terrain, i.e., M_{10} of size 377×672. To do so, the user may simply apply $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$ on M_{10}. This allows us to 'zoom in' on a specific region on the terrain giving us some further insights to the data as shown in Figure 6(b). In this case, the user finds that the first several movies are classics (e.g., "Star Wars", "Raiders of the Lost Ark", "Godfather", etc.) with most of them being old



(a) partitions of overall *terrain* (b) rating distribution of overall *terrain* (c) size/rating percents

Fig. 7. Statistics of overall *terrain* of MovieLens data

movies, and the first several reviewers have occupations as 'students' or 'educators'. And within this group of reviewers, more than half disliked "Titanic".

Finding Frequent Itemsets In addition to visual analysis, the model also has applications in data mining. We first consider the case of finding frequent itemsets. A transaction dataset can be constructed as a matrix, where each transaction is an object, and each item is an attribute whose value w_{ij} in $W_{|\mathcal{O}|\times|\mathcal{A}|}$ is 1 if the j-th item occurs in the i-th record, and 0 otherwise. We therefore have the following that relates frequent itemsets and BLOCKs.

Theorem 4 (Frequent Itemsets and BLOCKs). *A frequent itemset is the attribute set of a 1-BLOCK, and its support is the number of objects in the BLOCK.*

Proof. It can be easily proved that when a frequent itemset C occurs in n transactions, a submatrix $W' = \mathbf{1}$ can be constructed whose attribute set is C, and object set is those n transactions. Hence, W' corresponds to a 1-BLOCK satisfying all $w'_{ij} \geqslant 1$.

To prove Theorem 4 empirically, we used the `bms-pos` dataset (`http://fimi.cs.helsinki.fi/fimi03`). The dataset can be modelled as $\mathcal{D} = (\mathcal{O}, \mathcal{A})$, where $\mathcal{O}$ are transactions and $\mathcal{A}$ are the items; we have $|\mathcal{O}|$=515,597 and $|\mathcal{A}|$=1,657. Using $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ on $\mathcal{D}$, we obtain its macro-view terrain as shown in Figure 6(a). From this terrain, we conclude that we can find a similar set of frequent itemsets from a sample with fewer transactions *and* items. We selected a sample set $\mathcal{D}' = (\mathcal{O}', \mathcal{A}')$, where $\mathcal{O}'$ contains the first $0.1|\mathcal{O}| = 51,560$ transactions and $\mathcal{A}'$ contains the first $0.1|\mathcal{A}| = 166$ items from the terrain of $\mathcal{D}$. We then compare the results using the Apriori algorithm on both $\mathcal{D}$ and $\mathcal{D}'$ using different absolute support thresholds of 516, 1032 and 5156. The results are given in Table 1.

From the results, the 'hit rate' of finding the frequent itemset in $\mathcal{D}'$ improves quickly as the length of the itemset increases. Nearly all long frequent itemsets in $\mathcal{D}$ can be found in $\mathcal{D}'$, e.g., at minimum support of 516 and $\ell = 5, 6, 7$, we discovered over 90% of the frequent patterns; and when $\ell = 8, 9$, we find all the patterns in $\mathcal{D}'$. This result

	Min. Sup.=516			Min. Sup.=1032			Min. Sup.=5156		
l	$\#(FI^l_{D'} \cap FI^l_D)$	$\#FI^l_D$	$\frac{\#\cap}{\#FI^l_D}$	$\#(FI^l_{D'} \cap FI^l_D)$	$\#FI^l_D$	$\frac{\#\cap}{\#FI^l_D}$	$\#(FI^l_{D'} \cap FI^l_D)$	$\#FI^l_D$	$\frac{\#\cap}{\#FI^l_D}$
1	166	439	37.8%	164	333	49.3%	41	141	29.1%
2	3784	6877	55.0%	1843	3292	56.0%	174	389	44.7%
3	16885	23993	70.4%	5535	8083	68.5%	229	410	55.9%
4	31438	37649	83.5%	7414	9093	81.5%	102	151	67.6%
5	30191	32566	92.7%	4825	5268	91.6%	8	8	100%
6	15829	16197	97.7%	1391	1434	97.0%			
7	4242	4262	99.5%	137	137	100%			
8	455	455	100%	1	1	100%			
9	11	11	100%						
Conclusion	$\#(FI_{D'} \cap FI_D)$=103001 $\#FI_{D'}$=103001 $\#FI_D$=122449			$\#(FI_{D'} \cap FI_D)$=21310 $\#FI_{D'}$=21310 $\#FI_D$=27641			$\#(FI_{D'} \cap FI_D)$=554 $\#FI_{D'}$=554 $\#FI_D$=1099		

Table 1. Comparison of frequent itemsets found in D and D' with the same minimum support for the `bms-pos` dataset. l is the length of an itemset; $\#FI^l_D$ is the number of frequent itemsets with length l in D; $\#FI_D$ is the total number of frequent itemsets, i.e., including all lengths in D.

is encouraging since long frequent itemsets are often more important and interesting in real-life applications. Furthermore, we can do further analysis of $\mathcal{D}'$ terrain to gain further insights as we will discuss in Section 5.

Discovering Biclusters Another application of our model is to discover biclusters from the terrain of a dataset. To see why this can be done, we first cite a popular measure for biclusters [8]. The residue $\mathtt{H}(W)$ of given a matrix W is a δ-bicluster if $\mathtt{H}(W) \leqslant \delta$. This measure is defined as

$$\mathtt{H}(W) = \frac{1}{mn} \sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} (w_{ij} - w_{iJ} - w_{Ij} + |W|)^2 \tag{7}$$

where $w_{iJ} = \frac{1}{n}\sum_{j=1}^{n} w_{ij}$, $w_{Ij} = \frac{1}{m}\sum_{i=1}^{m} w_{ij}$, and $|W| = \frac{1}{mn}\sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} w_{ij}$.

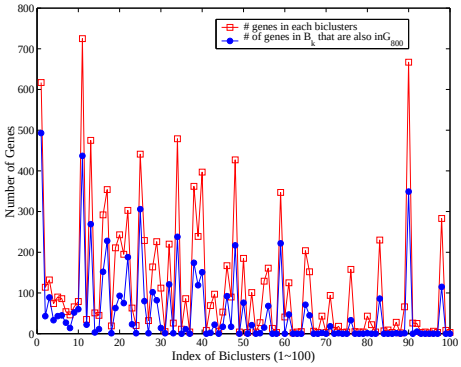
Theorem 5 (Bicluster and Average Resonance Strength of Macro-View Terrains). *Given a matrix $W = (w_{ij})_{m \times n}$, where $\mathcal{O}$ are the rows and $\mathcal{A}$ are columns, we have the inverse relation of the average inter-resonance strength and $\mathtt{H}(W)$ as follows*

$$\mathtt{H}(W) = \|W\|^2 + |W|^2 - \frac{1}{n}\overline{\mathtt{r}}(W) - \frac{1}{m}\overline{\mathtt{r}}(W^T) \tag{8}$$

where $\|W\| = \sqrt{\frac{1}{mn}\sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} w_{ij}^2}$, $\overline{\mathtt{r}}(W) = \frac{1}{\binom{m}{2}}\sum_{\substack{1 \leqslant i,j \leqslant m \\ i \neq j}} \mathtt{I}(\mathbf{w}_{i\_}, \mathbf{w}_{j\_})$ is the average inter-resonance strength among $\mathbf{w}_{i\_}$, and $\overline{\mathtt{r}}(W^T) = \frac{1}{\binom{n}{2}}\sum_{\substack{1 \leqslant i,j \leqslant n \\ i \neq j}} \mathtt{I}(\mathbf{w}_{\_i}, \mathbf{w}_{\_j})$ is the average inter-resonance strength among $\mathbf{w}_{\_j}$, and $\mathbf{w}_{i\_}$ is the i-row vector of W with $\mathbf{w}_{\_j}$ the j-column vector of W.

Proof. For simplicity, suppose

$$w_{iJ} = \frac{1}{n}\sum_{j=1}^{n} w_{ij}, \quad w_{Ij} = \frac{1}{m}\sum_{i=1}^{m} w_{ij}$$



(a) overlapping genes in B_k and G_{800}

	G_{1000}	G_{800}	G_{600}
F_1	0.66	0.70	0.74
P	0.87	0.80	0.73
R	0.53	0.62	0.75

$P = \frac{|B_1 \cup G_k|}{|B_1|},\ R = \frac{|B_1 \cup G_k|}{|G_k|},\ F_1 = \frac{2PR}{P+R}$

(b) overlapping genes in B_1 and G_k

Fig. 8. Comparison of genes in G_k and biclusters B_k ($1 \leqslant k \leqslant 100$) for gene expression data

and by the definition of "mean squared residue score" in Equation (7), we have

$$\begin{aligned}\mathbf{H}(W) &= \frac{1}{mn}\sum_{\substack{1\leqslant i\leqslant m\\ 1\leqslant j\leqslant n}}(w_{ij}-w_{iJ}-w_{Ij}+|W|)^2\\ &= \frac{1}{mn}\Bigg(\sum_{\substack{1\leqslant i\leqslant m\\ 1\leqslant j\leqslant n}} w_{ij}^2 + mn|W|^2 - n\sum_{i=1}^{m} w_{iJ}^2\\ &\quad -m\sum_{j=1}^{n} w_{Ij}^2\Bigg)\end{aligned} \tag{9}$$

and because

$$\sum_{i=1}^{m} w_{iJ}^2 = \frac{1}{n^2}\left(\sum_{\substack{1\leqslant i\leqslant m\\ 1\leqslant j\leqslant n}} w_{ij}^2 + 2\binom{n}{2}\overline{\mathtt{r}}(W^T)\right) \tag{10}$$

and

$$\sum_{j=1}^{n} w_{Ij}^2 = \frac{1}{m^2}\left(\sum_{\substack{1\leqslant i\leqslant m\\ 1\leqslant j\leqslant n}} w_{ij}^2 + 2\binom{m}{2}\overline{\mathtt{r}}(W)\right) \tag{11}$$

and by Equations (9), (10) and (11), we get

$$\begin{aligned}\mathbf{H}(W) &= \frac{(m-1)(n-1)-1}{m^2n^2}\sum_{\substack{1\leqslant i\leqslant m\\ 1\leqslant j\leqslant n}} w_{ij}^2 + |W|^2\\ &\quad -\frac{2\binom{m}{2}}{nm^2}\overline{\mathtt{r}}(W) - \frac{2\binom{n}{2}}{mn^2}\overline{\mathtt{r}}(W^T)\end{aligned} \tag{12}$$

and since we assume m and n are sufficiently large,

$$\frac{1}{mn} \approx \frac{(m-1)(n-1)-1}{m^2n^2}, \quad \binom{m}{2} \approx \frac{m^2}{2}, \quad \binom{n}{2} \approx \frac{n^2}{2}$$

we can conclude

$$\begin{aligned}\mathbf{H}(W) &= \frac{1}{mn} \sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} w_{ij}^2 + |W|^2 - \frac{1}{n}\overline{\mathtt{r}}(W) - \frac{1}{m}\overline{\mathtt{r}}(W^T) \\ &= \|W\|^2 + |W|^2 - \frac{1}{n}\overline{\mathtt{r}}(W) - \frac{1}{m}\overline{\mathtt{r}}(W^T)\end{aligned}$$

Theorem 5 can be interpreted as follows. Since $\|W\|$ and $|W|$ are sum of W in different forms, we can considered them as fixed constant. If the average inter-resonance strength of W and W^T, i.e., $\overline{\mathtt{r}}(W)$ and $\overline{\mathtt{r}}(W^T)$, is higher, then $\mathtt{H}(W)$ is lower and thus, W behaves like a bicluster. For configurations of $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ and $\mathrm{R}(\mathcal{A}, \mathcal{O}, W^T, \mathtt{I}, \mathtt{I})$, we conclude that if we select the first k rows and columns of W with large resonance strength $\mathtt{r}(\mathbf{o}_i, \tilde{\mathbf{o}})$ to form W', it is straightforward that we will have a smaller $\mathtt{H}(W')$ and thus, W a bicluster.

According to Theorem 3, it is because objects with high $\mathtt{r}(\mathbf{o}_i, \tilde{\mathbf{o}})$ are the ones that form W' that lead to high average inter-resonance strength $\overline{\mathtt{r}}(W')$, and $\overline{\mathtt{r}}(W'^T)$, we have $\mathtt{H}(W')$ that is sufficiently low to be a bicluster. Empirically, we verified this conclusion using the yeast microarray data (`http://arep.med.harvard.edu/network_discovery`) containing 2,884 genes and 17 conditions. From [8], 100 biclusers B_k ($1 \leqslant k \leqslant 100$) are found. As the number of conditions is small, we only investigate the effects of $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ on genes, where $\mathcal{O}$ are genes, $\mathcal{A}$ are conditions, and W is a row-normalized matrix. After the resonance simulation on W, we consider the subset G_k ($k \leqslant 2884$) which contains the first k genes with the largest resonance strength. We then compare how the genes in the 100 biclusters B_k ($1 \leqslant k \leqslant 100$) overlap with those in G_k. Figure 8 verified our conclusions about biclusters.

5 Micro-View of Data Terrain: Algorithms and A Case Study

Definition 2 defines the actual as the ordering of both sets $\mathcal{O}$ and $\mathcal{A}$, where a set of BLOCKs with different σ is discovered. Due to the hardness of finding a σ-B, we proposed an efficient approximation to the actual terrain. We referred to this as the macro-view terrain since the approximation serve to give an overview of the dataset. In situations where a precise single σ-B is needed (such as when the user decides to 'zoom in' on a particular region of the macro-view terrain), we need to be able to compute the micro-view terrain. To do so, we need an efficient algorithm to approximate the largest σ-B to derive a variant of the actual terrain that we call the `TERRACE`.

5.1 Finding the Largest σ-B

The hardness of finding the largest σ-B suggest that it is better to develop algorithms based on approximation techniques in favor of efficiency. Further, if the algorithm is

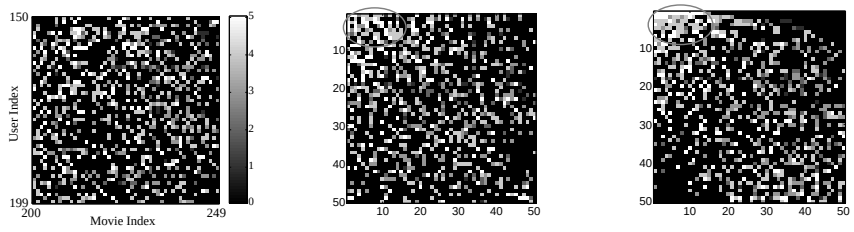
designed to run iteratively, the user can have the freedom to control the accuracy desired over the amount of computing time available, i.e., if the user has more time, more iterations can be set to achieve better accuracy. And from Theorem 1, the problem of approximating the largest σ-B can be reformulated as finding the maximum biclique G' in G whose size best approximates the exact maximum edge biclique of G. This further suggests that algorithms for bicliques may be used here. Unfortunately, most were not developed for our purpose. Hence, we further develop an algorithm to achieve the above based on our non-linear resonance model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$.

Related Works on Bicliques Biclique problems have been addressed in different fields. First, we have approximation algorithms rooted in mathematical programming relaxation [9]. Despite their polynomial runtime, their best result is 2-approximations, i.e., the subgraph discovered may not be a biclique but must contain the exact maximum edge biclique that is double in size. The other class of algorithms is to exhaustively enumerate all maximum bicliques [10] and then do a post-processing on all the maximum bicliques to obtain the desired results. Although efficient algorithms have been proposed and applied to computational biology [11], the runtime cost is too high. The third class of algorithms are developed based on some given conditions. For example, the bipartite graph G=$(\mathcal{O}, \mathcal{A}, E, W)$ must be of d-bounded degree, i.e., $|\mathcal{O}| < d$ or $|\mathcal{A}| < d$ [12] to give a complexity of $\mathbf{O}(n2^d)$ where n=$\max(|\mathcal{O}|, |\mathcal{A}|)$. While this gives the exact solution, the given conditions often do not satisfy the needs of real-world datasets and the runtime cost can be high for large d.

Algorithm Recall from Theorem 3, the first k objects have the highest average inter-resonance strength. Therefore, we can expect a higher probability of finding the 'mountains' among these objects and hence, the chances of finding the largest σ-B (or maximum edge biclique). This has also been observed in various experiments earlier, and we note that the exponential functions in the non-linear models are better at eliciting the 'mountains' from the top k objects (compare Figure 2(c) and Figure 4(b)). A revisit of the MovieLens dataset in Section 4.3 will also illustrate this observation.

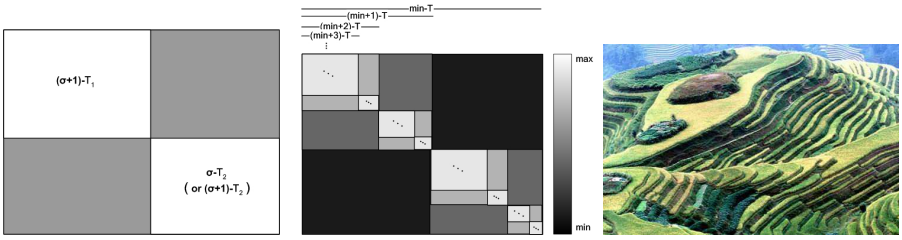
Due to over-concentration of classic movies and professional reviewers in Figure 6(a), we may select the 150th to 199th movies, and the 200th to 249th users for local analysis (i.e., we now 'zoom in' to this region of the macro-view terrain). This matrix is shown in Figure 9(a). Here, we see that ratings are scattered without any distinct clusters or concentration as Figure 6(a) shows. After reordering by using the linear and non-linear models, we see that the non-linear model in Figure 9(c) better shows the 'mountains' than that of the linear model in Figure 9(b). These observations leads to the development of the following algorithm.

Algorithm 1 presents a way to approximate the largest σ-B. While the non-linear model is capable of collecting 'mountains' to the top-left corner of the reordered matrix, a strategy is required to extend the BLOCKS found to the other parts of the matrix. The function `Find_B` is to find a BLOCK by extending a row of the reordered matrix to a biclique using the heuristic in Line 5 of `Find_B`. The loop from Line 4 to 9 in `Find_Largest_B` is needed to get the BLOCKs computed from each row. The largest BLOCK is then obtained by comparing the size $|B.L||B.R|$ among the BLOCKs found.



(a) original matrix (b) reordered by linear model (c) reordered by non-linear model

Fig. 9. Gray scale images of original and reordered matrix of the 150-199th users and the 200-249 movies from macro-view *terrain* in Figure 6(a) by different resonance models. Figure (b) and (c) can also be viewed as the micro-analysis by using its macro-views of Figure (a), like the way shown in Figure 6(b).



(a) layout of σ-T in recursive form (b) layout of TERRACE in whole (c) terrace *terrain* in real world

Fig. 10. Layout of a special *terrain* – TERRACE

The complexity of `Find_B` is $\mathbf{O}(|\mathcal{O}||\mathcal{A}|)$. Hence, the complexity of `Find_Largest_B` is $\mathbf{O}((k_1+k_2)|\mathcal{O}||\mathcal{A}|)$, where k_1 is the convergence loop number of the non-linear model, and k_2 is the loop number in the **FOR** statement of `Find_Largest_B`. If computing on all rows, k_2 is $|\mathcal{O}|$. However, because most large BLOCKs are concentrated on the left-top corner, the loop for `Find_Largest_B` is insignificant when dealing with the rows outside the top-left corner. Thus, we could set k_2 to a small value to consider only the first few rows to reduce the runtime complexity of `Find_Largest_B` to $\mathbf{O}(|\mathcal{O}||\mathcal{A}|)$.

5.2 Finding the Variant of the Actual Terrain – TERRACE

By Definition 2, the actual terrain is the reordered matrix containing the set of σ-BLOCKs with different σ. We consider a computationally effective version of this actual terrain called the TERRACE. It is defined as follows.

Definition 3 (σ-TERRACE). *A* σ-TERRACE *(or simply* σ-T*) is a* σ-B *with the structure* $\sigma\text{-T} \equiv (\sigma_1\text{-T}_1, \sigma_2\text{-T}_2)$*, where* $\sigma_1 = \sigma + 1$*,* $\sigma_2 = \sigma + 1$ *or* σ*,* T_1 *is the largest* σ_1-B *within* σ-T*, and the form* $(\mathsf{T}_1, \mathsf{T}_2)$ *represents the layout shown in Figure 10(a).*

We can consider Definition 2 as a description for a general data terrain while Definition 3 gives a special case of this generic data terrain. The recursive definition of σ-TERRACE contributes to the hierarchical structure as shown in Figure 10(b). This

Algorithm 1 $B =$ `Find_Largest_B`(G, σ), Find the largest σ-BLOCK in G

Input : $G = (\mathcal{O}, \mathcal{A}, E, W)$ and σ
Output : σ-BLOCK, $B = (L, R, E', W')$, where $L \subseteq \mathcal{O}$ and $R \subseteq \mathcal{A}$

convert $W = (w_{ij})$ to the binary matrix $W_b = (b_{ij})$, by setting b_{ij} as 1 if $w_{ij} \geqslant \sigma$ and 0 vice versa
get reordered binary matrix W_b^* by doing R($\mathcal{O}$, $\mathcal{A}$, W_b, E, E)
$maxsize = 0$ and $B = \emptyset$
for $i = 1$ to k_2 **do** {comment: i is index of row, k_2 can be set with a small fixed value by users.}
  $B =$ `Find_B`(W_b^*, i)
  if $(|B.L||B.R| > maxsize)$ **then**
    record B
  end if
end for
if $(B \neq \emptyset)$ **then**
  get $B.W'$ from W by $B.L$ and $B.R$
end if

$B =$ `Find_B`$(W_b^*, start\_row)$

set $B.L$ empty and `addset`$(B.L, start\_row)$
$B.R =$`binvec2set`$(\mathbf{b}^*_{start\_row})$ and $maxsize = |B.R|$
for $i = (start\_row + 1)$ to $|\mathcal{O}|$ **do**
  $R = B.R/$`binvec2set`$(\mathbf{b}_i^*)$
  if $((|B.L| + 1)|R| > maxsize)$ **then**
    $B.R = R$ and `addset`$(B.L, i)$
    $maxsize = |B.L||B.R|$
  end if
end for
$B =$ `Extend_B`(W_b^*, B)

$B' =$ `Extend_B`(W_b^*, B)

$start\_row =$ `min`$(B.L)$
for $i = 1$ to $(start\_row - 1)$ **do**
  $R =$ `binvec2set`$(\mathbf{b}_i^*)$
  if $(B.R \subseteq R)$ **then**
    `addset`$(B.L, i)$
  end if
end for
$B' = B$

Note on set functions:
`binvec2set` returns a set whose elements are the indices of non-zero coordinates in the binary vector.
`addset` is to add a value to a set.
`min` returns the minimum value among all elements of a set.
A/B returns a set whose elements are in A, but not in B.

layout is similar to real-world terrains such as the one shown in Figure 10(c). By this definition, we have Algorithm 2. As k depends on the distribution of σ-Bs in the matrix, the complexity of `Find_TERRACE` is $\mathbf{O}(k\mathbf{O}_{\texttt{Find\_Largest\_B}})$, which makes it suitable for generating a micro-view of terrain for in local analysis.

5.3 Analysis of Data Using the Micro-View Terrain

We have in Section 4.3 suggested the importance of local-analysis by 'zooming in' on a particular region of the terrain that the user is interested in. In that example, we simply obtain a macro-view of the region's terrain that actually do not allow effective analysis in some cases. Taking Figure 9(a) as an example, the macro-views of Figure 9(b) and (c) do not provide any valuable insights. But if the micro-view is now used on the MovieLens dataset, we will uncover new insights. Again, we selected the same sample as Figure 9(a) from the terrain of Figure 6(a). We do not use the sample from Figure 6(b) since their values represented the majority and hence, we are unlikely to discover more insights from further analysis.

Let the matrix W in Figure 9(a) be a bipartite graph $G = (\mathcal{O}, \mathcal{A}, E, W)$, where $\mathcal{O}$ are the 150th to 199th reviewers, and $\mathcal{A}$ are the 200th to 249th movies in Figure 6(a). As `min`(W) is 0, we run `Find_TERRACE`$(G, 0)$ to get the TERRACE terrain 0-T. The micro-view TERRACE terrain of G from 0-T is given in Figure 11(a). Here, we may find some TERRACEs appearing in 0-T, e.g., 1-T_1, 1-T_2, and 1-T_3, as Figure 11(b), (c),

Algorithm 2 $[\mathbf{l}, \mathbf{r}, W'] = \texttt{Find\_TERRACE}(G, \sigma)$, Find σ-TERRACE in G

Input : $G = (\mathcal{O}, \mathcal{A}, E, W)$ and σ
Output : sequence $\mathbf{l}$ of $\mathcal{O}$, sequence $\mathbf{r}$ of $\mathcal{A}$, and reordered matrix W'

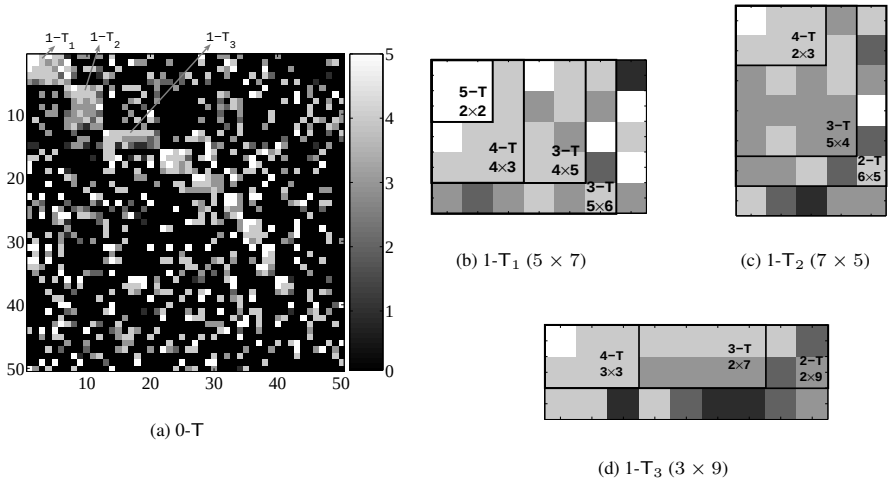
if ($|\mathcal{O}| = 1$ or $|\mathcal{A}| = 1$) **then**
  $W' = \texttt{sort}(W)$ and set $\mathbf{l}$ and $\mathbf{r}$ by the sorting result; return {comment: W is a vector, sort it in decreasing order.}
end if
if ($\sigma \geqslant \texttt{max}(W)$) **then** {comment: $\texttt{max}(W)$ returns the minimum value in a matrix.}
  $\mathbf{l}$=1:$|\mathcal{O}|$, $\mathbf{r}$=1:$|\mathcal{A}|$, W'=W and return {comment: 1:n is a sequence $(1, 2, \ldots, n)$; no need to get σ-TERRACE.}
end if
$B_1 = \texttt{Find\_Largest\_B}(G, \sigma + 1)$
if ($B_1.L = \mathcal{O}$ or $B_1.R = \mathcal{A}$) **then**
  $\mathbf{l}$=($B_1.L$,$\mathcal{O}$/$B_1.L$), $\mathbf{r}$=($B_1.R$,$\mathcal{A}$/$B_1.R$), and set W'=$W(\mathbf{l}, \mathbf{r})$; return {comment: $W(\mathbf{l}, \mathbf{r})$ is a reordered matrix of W by sequences $\mathbf{l}$ and $\mathbf{r}$.}
else if ($B = \emptyset$) **then**
  $\mathbf{l}$=1:$|\mathcal{O}|$, $\mathbf{r}$=1:$|\mathcal{A}|$, $W' = W$ and return
else
  $[\mathbf{l}_1, \mathbf{r}_1, W'_1] = \texttt{Find\_TERRACE}(B_1, \sigma + 1)$ {comment: in this case, $B_1 \neq \emptyset$.}
end if
$B_2 = (L, R, E, W_2)$, where $L = \mathcal{O}/B_1.L$, $R = \mathcal{A}/B_1.R$, and set W_2 from W by $B_2.L$ and $B_2.R$
$[\mathbf{l}_2, \mathbf{r}_2, W'_2] = \texttt{Find\_TERRACE}(B_2, \sigma + 1)$
$\mathbf{l} = (\mathbf{l}_1, \mathbf{l}_2)$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2)$, and set W'=$W(\mathbf{l}, \mathbf{r})$

and (d) shows. We select these 3 TERRACEs for further exploration. First, the structure of the 3 TERRACEs is hierarchical as we expected. More interesting is the relationship between the movies and the reviewers within these TERRACEs.

From 1-T_1 in Table 2(a), we now know that this group of reviewers like movies before the 60's such as children's films, animation, drama, and musical. The genres of these 4 movie group indicates that the reviewers of this group are potentially nostalgic, 'young at heart', and appreciates arts. On further analysis, 80% of the reviewers are middle-aged; 60% do not state their occupations, while those who stated their occupations showed that they are not mainstream careers. Thus, we can conclude the characteristics of this group of reviewers by their favorite movies.

In the case of 1-T_2 (Table 2(b)), we find that they are relatively young reviewers who primarily favor action and adventure type of movies (and a little doze of crime, science fiction, and thriller). This finding is not surprising since that would be what we expect of reviewers within this age-group. However, what is interesting is the fact that we have an outlier reviewer whose age is 47. Interestingly, he has given the highest score within this group and we could possible think of this reviewer as being 'young at heart'. In addition, we also notice at this micro-view that there is a female reviewer who gave high ratings to movies in this category. Whether this is 'interesting' would depend on the objective of analysis. In general, we know that this group of reviewers are predominantly males.

Finally, if we study 1-T_3 in Table 2(c), we will find that most of the movies in this group are recent and has either a comedy or drama genres. The reviewers in this case are young adults in modern economy jobs such as marketing, executive, or technical positions. In all likelihood, their jobs may have a higher level of stress (our guess) that motivates them to these category of movies. More importantly perhaps is that when analyzing the 4-T within this TERRACE (see Figure 11(d)), we noticed that the first 3 movies are given higher ratings than the others. Interestingly, these 3 movies are old when compared to the remaining movies. If we are marketing comedies and dramas



(a) 0-T

(b) 1-T_1 (5×7)

(c) 1-T_2 (7×5)

(d) 1-T_3 (3×9)

Fig. 11. TERRACE *terrain* (0-T) of local data in Figure 9(a) and three examples of 1-T.

based on this dataset, this insight might suggest that it is better to promote these classics instead of ignoring them altogether.

6 Related Works

Our work can be compared and discussed from a number of perspectives. From the viewpoint of visualizing high-dimensional datasets, there are several established techniques. This includes (i) geometric techniques such as scatterplots [13], landscape [14], parallel coordinates [15], projection pursuit [16], and hyperslice [17]; (ii) icon-based techniques such as stick figures [18], and color icons [19]; (iii) hierarchical techniques such as cone trees [20], and treemap [21]; and (iv) pixel-oriented techniques including recursive patterns [22], and circle segments [23].

In all cases, a certain degree of preprocessing and interaction is needed. For example, projecting the high-dimensional dataset to 2 or 3-dimensions is typically done before using the visualization techniques above. Zooming, e.g., IVEE [24] and details on demand, e.g., magiclens [25], are typical user-interaction techniques. However, the emergence of subspace clustering and biclustering in the analysis of high-dimensional datasets require more precise techniques to describe patterns embedded in the high-dimensional space. As a result, techniques like those above are inadequate. Notably, our work is designed to address this shortcoming from the perspective of visualization.

Although the terrain appear similar to the landscapes [14] used in capital markets and mountainview [26] used to navigate file systems, there are significant differences. First, our data terrain has mathematical and algorithmic roots while the landscape is simply a visualization interface. Meanwhile, the macro-view and micro-view of our terrain provides the concept of 'zoom on demand' for interactive analysis. Thus, our proposal incorporates geometric techniques and user-interactions in a natural way.

user ID	age	sex	occupation
815	**32**	M	other
897	**30**	M	other
807	**41**	F	healthcare
452	**35**	M	administrator
454	57	M	other

movie ID	title	year	genre
136	Mr. Smith Goes to Washington	1939	drama
659	Arsenic and Old Lace	1944	comedy, mystery, thriller
418	Cinderella	1950	animation, children's, musical
528	Killing Fields	1984	drama, war
404	Pinocchio	1940	animation, children's
485	My Fair Lady	1964	musical, romance
472	Dragonheart	1996	action, adventure, fantasy

times	statistics of movie genres
2	**animation, children's, drama, musical**
1	action, adventure, comedy, fantasy, mystery, romance, thriller, war

(a) 1-T_1 (5 users $\times$ 7 movies)

user ID	age	sex	occupation
87	47	M	administrator
622	**25**	M	programmer
267	**23**	M	engineer
399	**25**	M	other
280	**30**	F	librarian
886	**20**	M	student
497	**20**	M	student

movie ID	title	year	genre
62	Stargate	1994	action, adventure, Sci-Fi
233	Under Siege	1992	action
2	GoldenEye	1995	action, adventure, thriller
231	Batman Returns	1992	action, adventure, comedy, crime
94	Home Alone	1990	children's, comedy

times	statistics of movie genres
4	**action**
3	**adventure**
2	comedy
1	children's, crime, Sci-Fi, thriller

(b) 1-T_2 (7 users $\times$ 5 movies)

user ID	age	sex	occupation
244	28	M	**technician**
207	39	M	**marketing**
653	31	M	**executive**

movie ID	title	year	genre
509	My Left Foot	**1989**	drama
521	Deer Hunter	**1978**	drama, war
428	Harold and Maude	**1971**	comedy
628	Sleepers	1996	crime, drama
411	Nutty Professor	1996	comedy, fantasy, romance, Sci-Fi
410	Kingpin	1996	comedy
763	Happy Gilmore	1996	comedy
631	Crying Game	1992	action, drama, romance, war
685	Executive Decision	1996	action, thriller

times	statistics of movie genres
4	**comedy, drama**
2	action, romance
1	crime, fantasy, Sci-Fi, thriller, war

(c) 1-T_3 (3 users $\times$ 9 movies)

Table 2. Details of information of three TERRACEs, 1-T_1, 1-T_2 and 1-T_3.

We can also view our work as a form of clustering. Often, clustering in high-dimensional space is problematic [27]. Therefore, subspace clustering and biclustering were proposed to discover the clusters embedded in the subspaces of the high-dimensional space. Subspace clustering, e.g., CLIQUE, PROCLUS, ORCLUS, fascicles etc., are extensions of conventional clustering that seeks to find clusters by measuring the similarity in a subset of dimensions [28]. Biclustering was first introduced in gene expression analysis [8], and then applied in data mining and bioinformatics [29]. Biclusters are measured based on submatrices and therefore, is equivalent to the maximum edge biclique problem [29]. Under this context, a σ-B is similar to a bicluster.

As previously mentioned, the resonance model is a variant of discrete dynamical systems (DDS). To the best of our knowledge, this is the first effort to employ DDS to solve such problems. Previously, artificial neural networks like Hopfield network [30] and Bidirectional Associative Memory [31] (BAM) were represented as DDS, and have demonstrated great potential. Later, development in Web information retrieval [3], also motivated enhancements made within the framework of DDS [32–34].

Finally, we can compare our work to the study of statistics of data in 1-dimensional spaces. Well-known distributions such as the power-law (or Zipf's law) has been found to be ubiquitous in most real-world dataset [35–38], and since has been the hallmark of complex systems [38, 39]. As a matter of fact, the power-law distribution indicates data skew that can be exploited for data analysis. Interestingly, power-law has not been exploited by large in the domain of data engineering [40–42]. In this paper, the power-

law distribution actually relates to the reordering of data in one dimension [4]. In this case, our work can be seen as an extension of the power-law to reorder data in 2-dimensions.

7 Conclusions

In this paper, we proposed the data terrain as a means to visualize and analyze high-dimensional datasets. With this terrain, patterns in subspaces can be visualized and analyzed in a highly interactive fashion in macro-view or micro-view. From the perspective of information visualization and data mining, this is a novel contribution. In addition to introducing the concept of data terrain, we provided answers as to how the macro-view and micro-view of a terrain can be obtained efficiently; showed the relationships between the terrain and the typical patterns; and provided empirical evidences of our proposal by means of real-world applications.

On future works, there are a number of them. First, a theoretical analysis of the non-linear model will provide important insights to the understanding of further properties and characteristics of the model. Second, we would like to explore other types of data terrains that could potentially provide deeper insights about the data so as to assist the analyst in achieving better analytics. Third, we will be implementing a system that provides such a visualization of the data terrain. Such a system will be interactive and displays the terrain in 3D view providing mechanisms to zoom and pan on specific areas of the dataset for further analysis, or exportation to analytical algorithms.

References

1. Liu, J., Wang, W.: Op-cluster: Clustering by tendency in high dimensional space. In: Proceedings of ICDM, Melbourne, Florida (2003) 187
2. Peeters, R.: The maximum edge biclique problem is NP-complete. Disc. App. Math. **131** (2003) 651–654
3. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46** (1999) 604–632
4. Sandefur, J.T.: Discrete Dynamical Systems. Oxford: Clarendon Press (1990)
5. Golub, G., Loan, C.V.: Matrix Computations. The Johns Hopkins University Press (1996)
6. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. SIAM News **23** (1990) 1&18
7. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognition Letters **24** (2003) 1641–1650
8. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the 8th International Conference on Intelligent System for Molecular Biology. (2000)
9. Hochbaum, D.S.: Approximating clique and biclique problems. Journal of Algorithms **29** (1998) 174–200
10. Alexe, G., Alexe, S., Crama, Y., Foldes, S., Hammer, P.L., Simeone, B.: Consensus algorithms for the generation of all maximum bicliques. Technical Report 2002-52, DIMACS (2002)

[4] Joint distributions (e.g., normal distribution) are not considered as they seldom characterize the real-world dataset, and are not suitable for high-dimensional datasets.

11. Sanderson, M.J., Driskell, A.C., Ree, R.H., Eulenstein, O., Langley, S.: Obtaining maximal concatenated phylogenetic data sets from large sequence databases. Molecular Biology and Evolution **20** (2003) 1036–1042
12. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics **18** (2002) 136–144
13. Cleveland, W.S.: Visualizing Data. Hobart Press, Summit NJ (1993)
14. Wright, W.: Information animation applications in the capital markets. In: Proc. Int. Symp. on Information Visualization,, Atlanta, GA (1995) 19–25
15. Inselberg, A., Dimsdale, B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In: Proceedings of Visualization '90, San Francisco, CA (1990) 361–370
16. Huber, P.J.: Projection pursuit. The Annals of Statistics **13** (1985) 435–474
17. van Wijk, J.J., van Liere, R.D.: Hyperslice. In: Proceedings of Visualization '93, San Jose, CA (1993) 119–125
18. Pickett, R.M., Grinstein, G.G.: Iconographic displays for visualizing multidimensional data. In: Proc. IEEE Conf. on Systems, Man and Cybernetics, Piscataway, NJ (1988) 514–519
19. Keim, D.A., Kriegel, H.P.: Visdb: Database exploration using multidimensional visualization. Computer Graphics and Applications, (1994) 40–49
20. Robertson, G.G., Mackinlay, J.D., Card, S.K.: Cone trees: Animated 3d visualizations of hierarchical information. In: Proc. Human Factors in Computing Systems CHI '91 Conf., New Orleans, LA (1991) 189–194
21. Shneiderman, B.: Tree visualization with treemaps: A 2d space-filling approach. ACM Transactions on Graphics **11** (1992) 92–99
22. Keim, D.A., Kriegel, H.P., Ankerst, M.: Recursive pattern: A technique for visualizing very large amounts of data. In: Proc. Visualization '95, Atlanta, GA (1995) 279–286
23. Ankerst, M., Keim, D.A., Kriegel, H.P.: Circle segments: A technique for visually exploring large multidimensional data sets. In: Proc. Visualization '96, Hot Topic Session, San Francisco, CA (1996)
24. Ahlberg, C., Wistrand, E.: Ivee: An information visualization and exploration environment. In: Proc. Int. Symp. on Information Visualization, Atlanta, GA (1995) 66–73
25. A., E., Stone, M.C., Pier, K., Buxton, W., DeRose, T.: Toolglass and magic lenses: The see-through interface. In: Proc. SIGGRAPH '93, Anaheim, CA (1993) 73–80
26. Altom, T., Buher, M., Downey, M., Faiola, A.: Using 3d landscapes to navigate file systems: The mountainview interface. In: Proceedings of the 8th International Conference on Information Visualisation, London, England, UK (2004) 645–649
27. Beyer, K.: When is nearest neighbor meaningful? In: Proceedings of ICDT. (1999)
28. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter **6** (2004) 90–105
29. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS **1** (2004) 24–45
30. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci. USA **79** (1982) 2554–2558
31. Kosko, B.: Bidirectional associative memories. IEEE Transactions on Systems, Man and Cybernetics **18** (1988) 49–60
32. Tsaparas, P.: Using non-linear dynamical systems for web searching and ranking. In: Proc. of PODS. (2004)
33. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Finding authorities and hubs from link structures on the world wide web. In: Proc. of the 10th Int. World Wide Web Conf.,, Hong Kong (2001)
34. Ng, A.Y., Zheng, A.X., Jordan, M.I.: Stable algorithms for link analysis. In: Proceedings of the 24th SIGIR, New York (2001)

35. Li, W., Yang, Y.: Zipf's law in importance of genes for cancer classification using microarray data. Journal of Theoretical Biology **219** (2002) 539–551
36. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: Proceedings of SIGCOMM, Cambridge MA (1999) 251–262
37. Barabási, A.L., Albert, R., Jeong, H., Bianconi, G.: Power-law distribution of the world wide web. Science **287** (2000) 2115
38. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286** (1999) 509–512
39. Carlson, J.M., Doyle, J.: HOT: A mechanism for power laws in designed systems. Physical Review E **60** (1999) 1412–1427
40. Faloutsos, C., Matias, Y., Silberschatz, A.: Modeling skewed distribution using multifractals and the '80-20' law. In: Proceedings of VLDB, Mombai (Bombay) (1996) 307–317
41. Bi, Z., Faloutsos, C., Korn, F.: The "DGX" distribution for mining massive, skewed data. In: Proceedings of SIGKDD, San Francisco, California, USA (2001) 17–26
42. Palmer, C.R., Faloutsos, C.: Density biased sampling: An improved method for data mining and clustering. In: Proceedings of SIGMOD. (2000)