Using Interactive Evolution for Exploratory Data Analysis

Tomas Rehorek, Pavel Kordik

Faculty of Information Technology, Czech Technical University in Prague, Thakurova 9, Prague, 16000, CZECH REPUBLIC,
E-mails: pavel.kordik@fit.cvut.cz tomas.rehorek@fit.cvut.cz

Abstract – Multivariate data are difficult to explore. The most popular linear projection techniques mapping data to 2-dimensional space often fail to reveal the patterns of interest. Non-linear mapping techniques are both slow and inefficient. In this paper, we propose a heuristic that allows users to adjust the parameters of mapping techniques just by stating their preferences iteratively. The preliminary results on real-world dataset demonstrate the power of our approach.

Key words – Interactive Evolution, Dimensionality Reduction, Knowledge Discovery in Databases, Data Mining, Nonlinear Data Projection.

I. Introduction

As of today, there are hundreds of KDD methods available. The problem is, however, that datasets typically differ in structure, size, and complexity, from domain to domain and from problem to problem. Even though similar methods can be used in medicine, geology, meteorology, investment, marketing etc., it usually requires human expert to select appropriate methods and interpret the results. Typically, data mining expert needs to become familiar with the data by means of applying several DM methods, usually in trial-and-error manner.

A very common form of the data being analyzed is an $m \times n$ matrix, where rows code objects (i.e. patients), and columns code attributes (i.e. age, sex, height, weight, bone mass, fluids ratio etc.) sometimes referred to as *features*. When KDD is to be applied, it is a common task for expert to discover interesting relationships among attributes. While some relationships are well known in given field, entirely new facts can be easily extracted using KDD. Such facts, however, are completely unknown before the KDD procedure and hence the relationships are searched in highly investigative manner. Such investigative approach was defined in [15] as Exploratory data analysis (EDA).

Frequent method used in EDA is to visualize the data matrix. Because the examples typically lie in highly dimensional space (i.e. the number of attributes $n >> 4$), hundreds of methods have been developed to make visualization on 2 or 3-dimensional screen possible. The methods may classified as either linear and non-linear.

In the case of linear mapping, matrix $\mathbf{T} \in \mathbb{R}^{n \times 2}$ is used to do the $f : \mathbb{R}^n \to \mathbb{R}^2$ projection as $f(\mathbf{x}) = \mathbf{xT}$, resulting point in 2D-space. Example of popular linear projection technique used in EDA is Principal component analysis (PCA) [8], which maximizes variance in target space. Another linear projection approach is Linear discriminant analysis (LDA) [6]. LDA takes categorization of examples into account and builds on assumption that individual categories are normally distributed. Based on statistical analysis, LDA constructs a linear projection that maximizes distances between different categories in the target space.

In the case of non-linear projection, there is large number of possible options. There is a whole family of methods reffered to as Multidimensional scaling [2]. In MDS, the projection is build on basis of distances between all pairs of examples in the original space [14]. One such method is the Sammon projection [9], in which distances points in original space are tried to be preserved as much as possible in the target space. Another example of non-linear projection is Kernel PCA, where the original PCA operators are replaced by kernel methods in Hilbert space, producing non-linear mapping [12].

There are several important notes when considering the above-mentioned methods. While linear projections are often fast, they seldom reveal the true relationships in data. The expressivity of linear combinations is often too low to capture complex patterns in real-world data. On the other hand, non-linear projections are able to capture complex patterns, but are computationaly very expensive and quite intractable. Appropriate parameters for non-linear projection are generally difficult to find if we don't have a priori knowledge of what we are looking for – which is a very common case. Another problem is that it is very dificult to adjust the projection if, for example, new observations are to be incorporated into existing Sammon projection.

In recent years, a lot of progress has been done made the field of Interactive evolutionary computation (IEC). In IEC, an evolutionary algorithm is applied to problems that require support of human intuition. A population of candidate solutions is iteratively improved based of user feedback. In every generation, a set of solutions is presented to the user. The users evaluates the solutions based on his preferences, and the best solutions are selected, reproduced and mutated. After several generations, solutions with high value for the user will be hopefully found.

IEC has been found useful in many areas. The applications include evolution of three-dimensional objects [5], tracks for high-end racing game [3], graphical user interfaces [7], tone mapping [4], and ant paintings [1]. In [10], a very interesting online project is introduced. The project, Picbreeder.org, is focused on online collaborative evolution of pictures. The pictures are drawn by artificial neural network and interactive version of NEAT algorithm [13] is used. In [11], Picbreeder.org project is analyzed stating that suprisingly large amounts of needles in the haystack have been found. The images depict cars, faces, insects etc.

In this paper, we propose a method of applying IEC to problems of dimenzionality reduction. We will show that our approach seems promising for problems even as hard as finding parameters of non-linear projection for real-world data.

II. Dimensionality Reduction through $\mathbb{R}^n \to \mathbb{R}^2$ Projection

As mentioned in Section I, a frequent dataset representation used in DM is $m \times n$ matrix **A**. In general case, the matrix consists of values from various domains. It is, however, quite common to transform the values into real numbers, resulting matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. In special case of task known as classification, the examples are divided into classes, i.e. there is a special attribute $\ell \in L$ assigned to each example, where L is a set of possible classes. In such a case, the matrix $\mathbf{A} \in (\mathbb{R}^n \times L)^m$. If the objects examined were patients, than the real-valued attributes might code various indicators from blood analysis, and L might equal {0,1}, coding whether given patient suffers from the investigated disease or not.

If there were only 2 real-valued attributes for each object, the dataset **A** could be easily visualized in Cartesian coordinate system: there would be m points such that x and y coordinates would map to the real-valued attributes, and the color of given point would code the class that the object belongs to. However, the number of attributes is often much larger (n > 10), making their direct mapping to coordinate axes impossible.

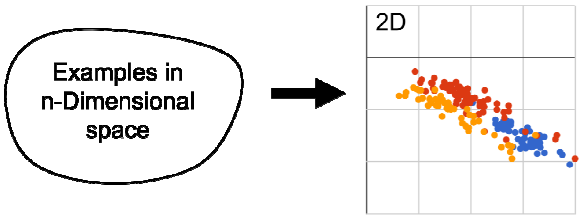

Fig.1 Projection $\mathbb{R}^n \times L \to \mathbb{R}^2 \times L$.

In Section I, we have stated that several dimensionality reduction techniques may be used, and that these can be roughly classified as the linear and the non-linear ones. In this paper, we will focus on adding interactivity to dimensionality reduction through the use of IEC approach. For the purpose of testing IEC, we will consider two different projections: one linear and one non-linear.

First projection, which we will refer to as **linear**, is the one defined as $f(\mathbf{x}) = \mathbf{xT}$, where $\mathbf{T} \in \mathbb{R}^{n \times 2}$. Values of matrix **T**, however, will be subject of evolution rather than variance maximization as in PCA.

Second projection, which we will call **sigmoidal**, is non-linear projection defined as follows. Given matrix $\mathbf{S} \in \mathbb{R}^{3 \times 2n}$ such that

$$\mathbf{S} = \left[\begin{array}{cccc|cccc} a_{1,1}, & a_{1,2}, & \ldots, & a_{1,n}, & a_{2,1}, & a_{2,2}, & \ldots, & a_{2,n} \\ b_{1,1}, & b_{1,2}, & \ldots, & b_{1,n}, & b_{2,1}, & b_{2,2}, & \ldots, & b_{2,n} \\ c_{1,1}, & c_{1,2}, & \ldots, & c_{1,n}, & c_{2,1}, & c_{2,2}, & \ldots, & c_{2,n} \end{array}\right],$$

the projection $f : \mathbb{R}^n \to \mathbb{R}^2$ is realized as follows:

$$f(\mathbf{x}) = \left[\sum_{i=1}^{n} \frac{a_{1,i}}{1+e^{b_{1,i}(x_i - c_{2,i})}}, \quad \sum_{i=1}^{n} \frac{a_{2,i}}{1+e^{b_{2,i}(x_i - c_{2,i})}}\right], \quad (1)$$

resulting a point in 2D-space. Semantics of parameters a, b and c are best depicted on Fig. 2.

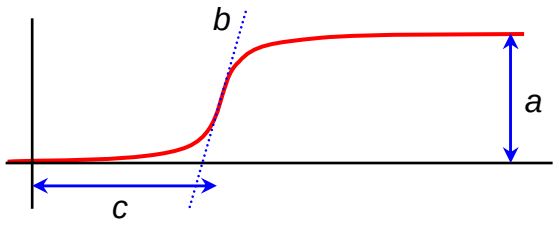

Fig. 2 Parameters of sigmoidal projection for each dimension in original space.

In case of our experiments, we are given a dataset $\mathbf{A} \in (\mathbb{R}^n \times L)^m$ for which dimensionality-reducing projection f is being evolved in order to make user satisfied with $f(\mathbf{A})$ as much as possible.

III. Interactive Evolution of $\mathbb{R}^n \to \mathbb{R}^2$ Dimensionality-Reducing Projection

We have stated that we propose the $\mathbb{R}^n \times L \to \mathbb{R}^2 \times L$ projection to be subject of human-guided evolution. Considering *linear* and *sigmoidal* projections defined in Section II, we will evolve matrices $\mathbf{T} \in \mathbb{R}^{n \times 2}$ and $\mathbf{S} \in \mathbb{R}^{3 \times 2n}$ such that the according projections will be of high usefulness for the user. The term "usefulness" is rather fuzzy, and the only options is collect feedback from the user. We propose following Real-valued genetic algorithm (RCGA) with user feedback as basic for fitness calculation.

Algorithm 1 Outline of human-guided RCGA

$g \leftarrow 0$
$P_0 = p \leftarrow$ init()
$E_0 \leftarrow$ user_evaluation(P_0)
$F_0 \leftarrow$ fitness(E_0)
while terminal condition is not met
  $g \leftarrow g + 1$
  $P'_g \leftarrow$ select(P_{g-1}, F_{g-1})
  $P_g \leftarrow$ mutate(P'_g)
  $E_g \leftarrow$ user_evaluation(P_g)
  $F_g \leftarrow$ fitness(E_g)
end while
return P_g

The algorithm works as follows. First, a random population of candidate real matrices is generated. These matrices are then subject of human evalution. According to human evaluation, the candidate matrices are assigned fitness values as in traditional RCGA.

As long as the candidate projections are not good enough, an evolutionary cycle takes step iteratively. In each step, based on user evaluations from previous generation, best projections are selected and mutated forming new population. The new population is then again subject of user evaluation.

IV. Experiment setup

There are several ways of how the user may submit evaluations. For the purposes of our experiments, we designed graphical user interface as depicted on Fig. 3. To

make the interface as ergonomic as possible, we offer three feedback options to the user: *good*, *neutral* and *bad*.

Selection procedure is done through tournament selection. We take advantage of the fact that there is an intuitive total ordering: *good* > *netral* > *bad*. Since tournament selection only requires the candidate solutions to be totally comparable, it can be employed very easily.

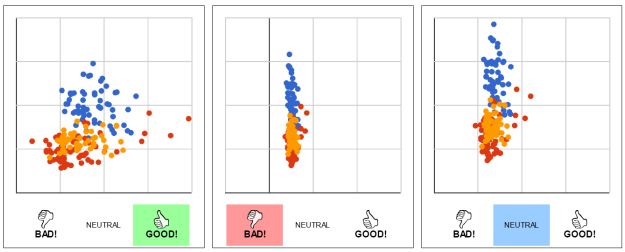

Fig. 3 User interface for evaluating candidate projections.

Because we use RCGA to evolve real-valued matrices, we need some real-valued mutation operated to be provided. We use Gaussian mutation which is very straightforward and allows the values of individual parameters to change exponentially with number of generations. Specifically, given mutation rate σ, we mutate the (i,j)-th element $(a_{i,j})$ of matrix $\mathbf{A} = \mathbb{R}^{m \times n}$ by random sampling the normal distribution $N(a_{i,j}, \sigma)$.

In our implementation, we do not use crossover operator, though there are dosend of RCGA crossover operators available. These can be subject of further research.

For the purpose of our experiments, we have chosen *Wine dataset*, which is a well-known benchmarking dataset for DM classification task. This dataset contains results of a chemical analysis of wines produced in three different cultivars. The analysis determined 13 different values (which may be considered as real numbers). Since there are three types of wines, the examples are of three different classes (labels) [Wine]. We have chosen this dataset because it is balanced dataset consisting of three classes that are moderately difficult to separate, making it suitable for unbiased testing of IEC capabilities. We have normalized the dataset, leaving attribute weighting completely to the IEC.

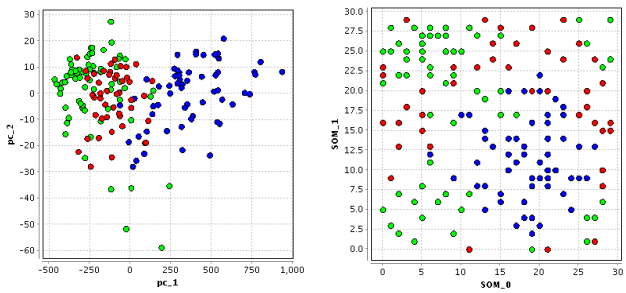

Fig. 4 Visualization of the Wine dataset using PCA (left) and SOM (right) dimensionality reduction.

The Wine dataset visualized using PCA and SOM (another popular dimensionality reduction technique) projections are shown on Fig. 4. Note that the visualization algorithms do not allow user to affect the output.

In the following sections, we will demonstrate that our approach offers much wider scale of options how to visualize the data.

V.Cluster Separation Experiment

Given an *n*-dimensional dataset of examples with labeling, natural idea where to start IEC dimensionality-reduction experiments with is to *separate* points of different labels as much as possible in the target 2D space. This idea is depicted on Fig. 5.

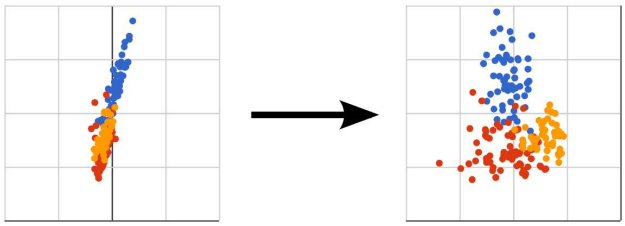

Fig. 5 Separation of points with different labels in $\mathbb{R}^2$.

This can be seen as *inverse clustering* problem. In classical clustering, we are given a set $\mathbf{X}$ of exmples from a metric space such as $\mathbb{R}^n$, and our task is to find partition $\mathcal{C}_\mathbf{X} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$ such that clusters from $\mathcal{C}_\mathbf{X}$ are homogene and easily separable from each other. In contrast, in case of our experiment, we are given a partition $\mathcal{C}_\mathbf{X} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$ and our task is to find a projection $f : \mathbb{R}^n \to \mathbb{R}^2$ such that the homogeneity and separability criterions are satisfied for corresponding clusters $\{ f(\mathbf{C}_1), \ldots, f(\mathbf{C}_k) \}$.

A. Separation Using Linear Projection

The first interactive cluster separation experiment was performed using Linear projection on the Wine dataset. The user declared the objective of separating the points with different labels. Population of size 7 was used, as we found it to be reasonable equilibrium between explorativness of the algorithm and workload for the human user.

As can be seen of Fig. 6, the initial population tends to project all the examples into single line. This is probably because in the dataset that is not normalized, there are some dominant attributes.

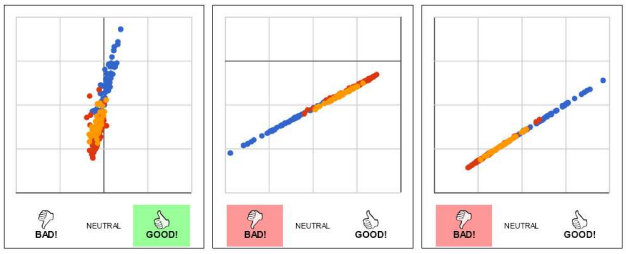

Fig. 6 Separation using linear projection, sample solutions from generation #0.

Here the natural idea is to mark line-forming projections as *bad*, and the other projections as *good*. Fig. 7 shows the population after 15 generations. At this point, it seems that sufficient attribute weighting has already been achived.

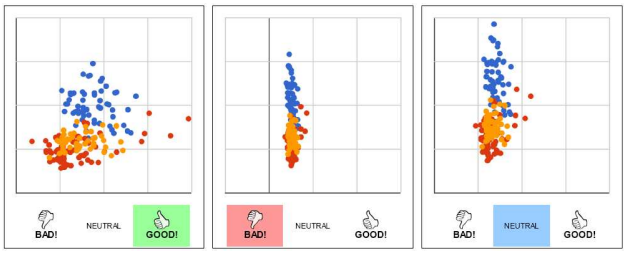

Fig. 7 Separation using linear projection, sample solutions from generation #15.

Finally, Fig. 8 shows sample solutions from generation number 30. Not only that attributes were weighted, but also good linear separation has been successfully found.

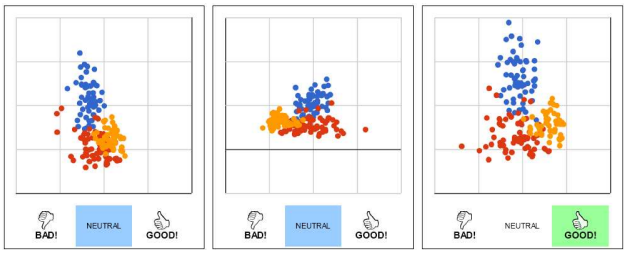

Fig. 8 Separation using linear projection, sample solutions from generation #30.

B. Separation Using Sigmoidal Projection

The second experiment was performed on the same dataset, this time using Sigmoidal projection. In this case, the search space is much more complex. For our dataset of 13 real-valued attributes, we need to find a matrix of 78 strongly-connected real numbers, that will be used to obtain projection according to Eq. (1) in Section II.

Fig. 9 shows the initial population generated. Most of the solutions are very "messy", but incidently, one of the solutions seems quite promising. Hence we marked it as *good*.

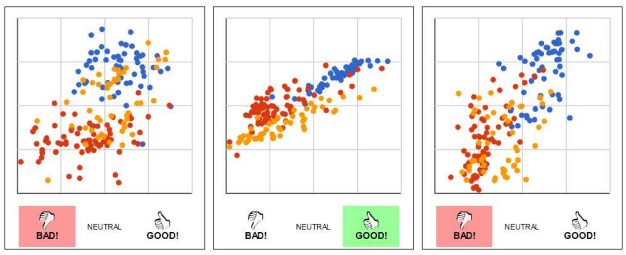

Fig. 9 Separation using sigmoidal projection, sample solutions from generation #0.

After only 5 generations, surprisingly, dispite the complexity of the search space, we have obtained much better solutions, as shown on Fig. 10.

Fig. 10 Separation using sigmoidal projection, sample solutions from generation #5.

Fig. 11 shows candidate solutions from generation number 15. It seems that our approach can easily overcome the search space complexity, because we obtained really nice projections.

Fig. 11 Separation using sigmoidal projection, sample solutions from generation #15.

Indeed, projections evolved could be further fine-tuned by decreasing mutation parameter σ in Gaussian mutation. Nevertheless, after only 15 generations of evaluating 7 candidate solutions as *bad*, *neutral*, or *good*, we have found matrix **S** that makes sigmoidal projection separate examples quite well in target 2D space.

VI. Other experiments

One important thing to note is that IEC does not force the user to follow some specified goal. In fact, the user has a wide scale of options. We will demonstate this on two following simple expariments.

Fig. 12 shows the population after 5 generations in experiment that we named as "blue points up". As the name implies, our goal was to make the *y* coordinate of the blue examples as high as possible, leaving the same coordinate of other examples as small as possible.

Fig. 12 The „blue points up" experiment, sigmoidal projection after 5 generations.

Frequent task in data preprocessing state of KDD procedure is to detect so-called outliers, i.e. examples that deviate too much from the others and were probably generated by some other process. This can also be done using IEC dimensionality reduction. Fig. 13 shows suspicious examples that were detected after 8 generations of evolving linear projection.

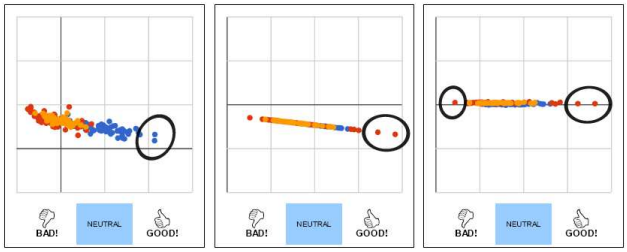

Fig. 13 Outliers detection experiment, linear projection after 8 generations.

Conclusion

We have applied the interactive evolution to explore the state space of linear and nonlinear data projection techniques. In contrast to traditional methods, evolving projections allows the user to explore the data interactively within few iterations and even change the goal during the process in order to reveal hidden patterns in the multidimensional space.

As a future work, we plan to apply the interactive evolution to more complex problems in the data mining domain.

References

[1] *S. Aupetit, V. Bordeau, N. Monmarché, M. Slimane and G. Venturini*, "Interactive Evolution of Ant Paintings". IEEE Congress on Evolutionary Computation. 2003.

[2] *I. Borg, P. Groenen*. Modern Multidimensional Scaling: theory and applications (2nd ed.). New York: Springer-Verlag. pp. 207–212. 2005.

[3] *L. Cardamone, D. Loiacono, and P. L. Lanzi*, "Interactive evolution for the procedural generation of tracks in a high-end racing game". In Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011.

[4] *S. B. Chisholm, D. V. Arnold, and S. Brooks*, "Tone mapping by interactive evolution". Genetic and Evolutionary Computation Conference, Montreal, 2009.

[5] *J. Clune*, H. Lipson. "Evolving three-dimensional objects with a generative encoding inspired by developmental biology". Proceedings of the European Conference on Artificial Life. 144-148. 2011.

[6] *R. Fischer*, "The Use of Multiple Measurements in Taxonomic Problems". Anneals of Eugenics, 7, p. 179–188. 1936.

[7] *M. Ilavsky, R. Jaksa*, "Interactive evolution of graphical user interface with GTK toolkit," Cognitive Infocommunications (CogInfoCom), 2nd International Conference, pp.1-6, 7-9 July 2011.

[8] *K. Pearson*, "On Lines and Planes of Closest Fit to Systems of Points is Space". Philosophical Magazine Series 6 **2** (11): 559–572. 1901.

[9] *J. W. Sammon*, "A nonlinear mapping for data structure analysis". IEEE Transactions on Computers **18**: 401–409. 1969.

[10] *J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, K. O. Stanley*, "Picbreeder: Evolving pictures collaboratively online". In CHI '08: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp. 1759–1768, New York, NY. ACM, 2008.

[11] *J. Secretan, N. Beato, D. B. D. Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, K. O. Stanley*. "Picbreeder: A case study in collaborative evolutionary exploration of design space". Evolutionary Computation, 2011.

[12] *B. Schölkopf, A. Smola, and K.-R. Müller*, "Nonlinear component analysis as a kernel eigenvalue problem". Neural Computation. 1998.

[13] *K. O. Stanley, R. Miikkulainen*, "Evolving Neural Networks Through Augmenting Topologie". Evolutionary Computation 10 (2), pp. 99_127. 2002.

[14] *S. Togerson*, "Multidimensional scaling: I Theory and method". Psychometrika, 17, 401–419, 1952.

[15] *J. W. Tukey*, Exploratory Data Analysis. Addison-Wesley Publishing Company. 1977.