Visual Anomaly Detection in Educational Data

Jan Géryk, Luboš Popelínský(✉), and Jozef Triščík

Knowledge Discovery Lab, Faculty of Informatics,
Masaryk University, Brno, Czech Republic
popel@fi.muni.cz

Abstract. This paper is dedicated to finding anomalies in short multivariate time series and focus on analysis of educational data. We present ODEXEDAIME, a new method for automated finding and visualising anomalies that can be applied to different types of short multivariate time series. The method was implemented as an extension of EDAIME, a tool for visual data mining in temporal data that has been successfully used for various academic analytics tasks, namely its Motion Charts module. We demonstrate a use of ODEXEDAIME on analysis of computer science study fields.

Keywords: Visual analytics · Academic analytics · Anomaly detection · Temporal data · Educational data mining

1 Introduction

Visual analytics [3,9,10,12,14] by means of animations is an amazing area of temporal data analysis. Animations allows us to detect temporal patterns, or better to say, patterns changing in time in much more comprehensive way than classical data mining or static graphs.

Motion Charts (MC) is a dynamic and interactive visualization method which enable analyst to display complex quantitative data in an intelligible way. The adjective dynamic refers to the animation of rich multidimensional data through time. Interactive refers to dynamic interactive features which allow analysts to explore, interpret, and analyze information hidden in complex data.

MC are very useful in analyzing multidimensional time-dependent data as it allows the visualization of high dimensional datasets. Motion Charts displays changes of element appearances over time by showing animations within a two-dimensional space. An element is basically a two-dimensional shape, e.g. a circle that represents one object from the dataset. The third dimension is time. Other dimension can be displayed inside circles e.g. in form of sectors or rings. The basic concept was introduced by Hans Rosling who popularized the Motion Charts visualization in a TED Talk[1]. MC enables exploring long-term trends which represent the subject of high-level analysis as well as the elements that form the

[1] http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html.

C. Dichev and G. Agre (Eds.): AIMSA 2016, LNAI 9883, pp. 99–108, 2016.
DOI: 10.1007/978-3-319-44748-3_10

patterns which represent the target analysis. The dynamic nature of MC allows a better identification of trends in the longitudinal multivariate data and enables the visualization of more element characteristics simultaneously [2]. E.g. in feature selection or mapping, it is visual analytics, and for time-dependent data even more animations, that can be helpful as a user is free to choose the feature selection according to his or her intentions and can see the results immediately.

Quite often we need not only to detect typical trends in time-dependent data but also to discover processes that differs from them the most significantly to find anomalous trends [1]. Naturally, a good feature selection significantly affect not only a detection of relationship but also of anomalies, the task that we try to solve here in collaboration of classical anomaly detection and visual analytics. In this paper we present a new tool ODEXEDAIME for anomaly detection in short series of time-dependent data. Its main advantage if compared with common anomaly detection methods is their comprehensibility and also their easy combination with visual analytics tool.

The paper is structured as follows. Section 2 contains a description of visual data mining tool EDAIME focusing on Motion Charts module. In Sect. 3 we gives an overview of the methods that we employed for outlier detection in time-dependent data focusing on short series. Section 4 describes ODEXEDAIME, a tool for outlier detection in short time series. Description of CS study fields dataset can be found in Sect. 5 and the results of experiments in Sect. 6. Discussion, conclusion and future work are presented at the end of the paper in Sect. 7.

2 Motion Charts in EDAIME

EDAIME [5–7], the tool for visual analytics in different kind of data has been addressed two main challenges. This tool enables visualization of multivariate data and the interactive exploration of data with temporal characteristics, actually, not only motion charts. EDAIME has been used not only for research purposes but also by FI MU management as it is optimised to process academic analytics (AA) [11]. For more information on properties and methods of EDAIME, see the demos

http://www.fi.muni.cz/~xgeryk/framework/video/clustering_of_elements.webm
http://www.fi.muni.cz/~xgeryk/framework/video/groups_of_elements.webm
http://www.fi.muni.cz/~xgeryk/framework/video/extending_animations.webm

X axis displays an average grade for each field (from 1.0 as Excellent to 4.0 as Failed), Y axis is an average number of the credits obtained (typically, 2 h course finished with exam is for 4 credits), the number in the bottom-right corner is the order of a semester. Green sectors means a fraction of successfully finished studies, red ones are for unsuccessful ones.

Menu Controls enables to control animation playback. Apart from play, pause, and stop buttons, there is also range input field which controls five levels of the animation speed. These controls facilitate the step-by-step exploration of the animation and allow functionality for transparent exploration of the data over the entire time span. Animation playback can be interactively changed by traversing mouse over semester number localised in right bottom of the EDAIME tool. Mouse-over element events trigger tooltip with additional element-specific information. One mouse click pauses animation playback and another one starts it again. Double-click restarts the animation playback. Cross axis can be activated to enable better reading values from axes and can be well combined with dimension distortion.

Menu Data mapping allows to map data into Motion Charts variables. The variables include average number of students, average number of credits, average grade, enrolled credits, obtained credits, completed studies, and incomplete studies. Controls for data selection are also particularly useful. Univariate statistical functions can be applied on any of the aforementioned variables. Bivariate functions are also available and can be applied on pairs of variables include enrolled and obtained credits, and complete and incomplete studies.

The main technical advantages over other implementations of Motion Charts are its flexibility, the ability to manage many animations simultaneously, and the intuitive rich user interface. Optimizations of the animation process were necessary, since even tens of animated elements significantly reduced the speed and contributed to the distraction of the analyst's visual perception. The Force Layout component of D3 provides the most of the functionality behind the animations, and collisions utilized in the interactive visualization methods. Linearly interpolated values are calculated for missing and sparse data.

3 Outlier Detection in Short Time Series

3.1 Basic Approach

Time series that we are interested in has three basic properties - (1) a fixed time interval between two observations, (2) same length, and (3) shortness of a time series. For the latter, we limit the length to be smaller than 15 what covers length of study (a number of semesters) of almost all students. We found that existing tools for multivariate time series are not appropriate mainly because of shortness of a time series in tasks that we focus on. We also tested methods for sequence mining [1], namely mining frequent subsequences but none of them displayed a good result. Actually, the time series under exploration lays somewhere between time series (but are quite short) and sequences. However, relation between sequence members look less important than dependence on time and moreover, anomalies in trend are important rather then point anomalies or subpart (subsequence) anomalies.

It was the reason that we decided to (1) transform each multivariate time series into a set of univariate ones, (2) apply to each of those series outlier detection method described bellow, and then (3) join the particular outlier detection factors into one for the original multivariate time series. We observed that this approach worked well, or even better, if compared with the state-of-the-art multivariate time series outlier detection methods.

Methods for anomaly detection in time series can be usually split into distance-based, deviation-based, shape-based methods (or its variant here, trend-based), and density-based (not used here) [1]. For all the methods below we checked two variants - original (non-normalised) data and normalised one - to limit e.g. an influence of a different number of students in the study fields.

3.2 Distance-Based Method

We employed two variants, *mean-based* method - mean M of a given feature is computed as an average of its values in all time series. Outlier factor is then computed as a distance of a given time series (actually its mean value m of the feature) from the mean M. The other method, called *distance-based* in the rest of this paper, computes euclidian (or Haming for non-numeric values) distance between two time series (two vectors). Outlier factor is computed as sum of distances from k nearest time series.

3.3 Trend-Based Method

This method computes how often the trend changed from increasing to decreasing or vice versa. Outlier factor is computed as difference of this value from mean value computed for all the rest of time seties in a collection.

3.4 Deviation-Based Method

This method compares difference of a feature value in two neighboring time moments for two time series. Difference of those two differences is taken as a distance. Rest is the same as for distance-based method.

3.5 Total Outlier Factor

For each dimension (i.e. for each dependent variables in an observation), and for a given basic method from the list above we compute a vector of length n where n is a number of dependent variables. Then we use LOF [4] (see also for formal definition of a local outlier factor) for computing the outlier factor for a given observation.

4 ODEXEDAIME

4.1 Algorithm

ODEXEDAIME (Outlier Detection and EXplanation with EDAIME), the tool for outlier detection in short multidimensional time series consists of four methods described above. We chose them because each of those method detect different kind of anomaly and we wanted to detect as wide spectre of anomalies as possible. The outlier detection method is unsuprevised, We do not have any

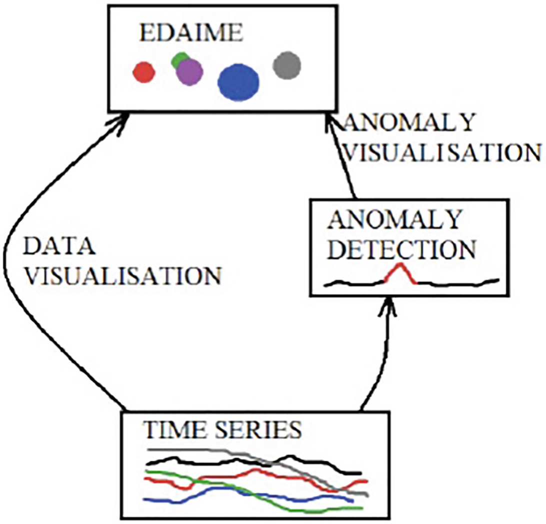


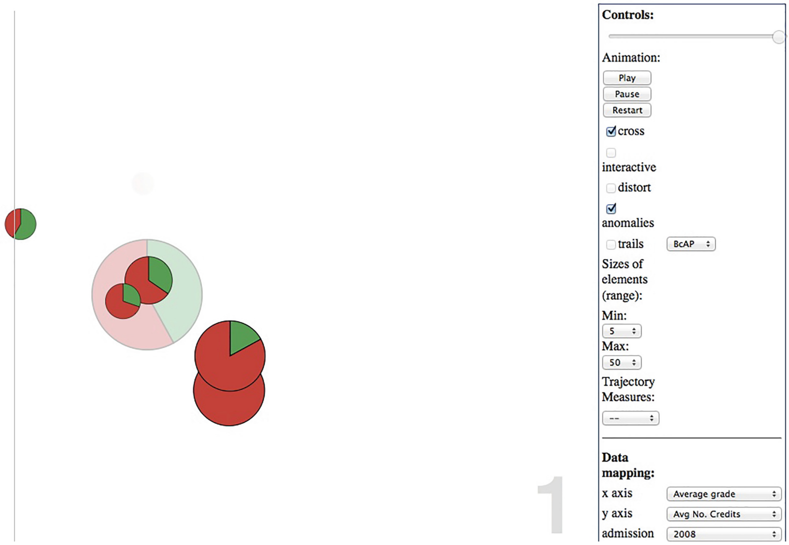
Fig. 1. ODEXEDAIME scheme

example of normal or abnormal anomalous series. The ODEXEDAIME algorithm can be split into five steps. In the first step, multivariate time series has been transformed into series of univariate, one-dimensional, time series. In the second step, an outlier factor has been computed for each univariate series and each of the four methods meanbased, distancebased, trendbased and deviation based. E.g. for our data where we analysed 7 features we obtain 28 characteristics for each multivariate time series. The outlier factors from the previous step are used for computing final outlier factor of the original multivariate time series. Local outlier factor LOF [4] has been used. The last step is visualisation. The scheme of ODEXEDAIME that has been implemented in Java can be seen in Fig. 1.

4.2 Visualisation

All the detected anomalous entities, e.g. a study field, are immediately visualised. Visualisation of anomalies is independent on features selected for visualisation. It means that features selected for anomaly detection can be different from features that has been chosen for visualisation. Layout of the ODEXEDAIME user interface can be seen on Fig. 2. The names of circles, actually CS study fields, are explained in the data section. A user select a use of EDAIME without or with anomaly detection. If the later was chosen, anomalous entities (circles) will be highlighted.

ODEXEDAIME can be found here
http://www.fi.muni.cz/~xgeryk/analyze/outlier/motion_chart_pie_anim_adv_neobfus.pl



Controls:

Animation:

☑ cross

☐ interactive

☐ distort

☑ anomalies

☐ trails BcAP

Sizes of elements (range):

Min: 5

Max: 50

Trajectory Measures: --

Data mapping:

x axis Average grade

y axis Avg No. Credits

admission 2008

Fig. 2. ODEXEDAIME

Put the button *anomalies* on, to see the anomalous data. The acronym of a study field can be displayed after a pointer is inside a bubble. The outlying time series is/are that one(a) that is/are blinking.

5 Data

Data contains aggregated information about bachelor study fields at Faculty of Informatics, Masaryk University Brno. BcAP denotes Applied Informatics, PSK denotes Computer Networks and Communication, UMI denotes Artificial Intelligence and Natural Language Processing, GRA denotes Computer Graphics, PSZD denotes Computer Systems and Data Processing, PDS denotes Parallel and Distributed Systems, PTS is for Embedded systems, BIO denotes Bioinformatics, and MI denotes Mathematical Informatics. A field identifier is always followed by the starting year. E.g. BcAp (2007) concerns students of Applied Informatics that began their study in the year 2007. Data contains information on

- the number of students in every term;
- the average number of credits subscribed at the beginning of a term; and
- credits obtained at the end;
- a number of students that finished their study in the term; or
- moved to some other field; or
- changed at the mode of study (e.g. temporal termination); and also
- an average rate between 1 (Excellent) and 4 (Failed) for the study field in a term

6 Experiments and Results

We used all anomaly detection methods referred in Sect. 3 and then, for presentation in this Section, chose that ones with the highest local outlier factors where the maximal LOF was at least five-times higher than the minimum LOF for the chosen anomaly detection method.

For LOF parameter $k = 5$ (for k nearest neighbours) was used in all the experiments. We also checked smaller values (1..4) but the results were not better. For $k > 5$ the difference between the maximum and minimum value of LOF did not significantly change.

All the results obtained with ODEXEDAIME has been compared with anomaly detection performed by human (referred as an expert in this section) who can use only classical two-dimensional graphs.

Table 1. Distance-based outlier detection: applied informatics

	BcAP (2007)	PDS (2007)	BIO (2007)	PTS (2008)
LOF:	**23,10**	1,28	2,67	1,09
	GRA (2008)	**BcAP (2008)**	PSK (2007)	GRA (2007)
LOF:	0,99	**21,63**	0,91	0,91
	MI (2008)	PSZD (2007)	UMI (2008)	MI (2007)
LOF:	2,55	1,11	1,11	0,96
	BcAP (2006)	PSZD (2008)	PSK (2008)	UMI (2007)
LOF:	**18,07**	0,91	2,67	0,98

In Table 1, there are results for distance-based method when the euclidian distance was used. Similar results were obtained with Manhattan distance, only the difference between the highest value of LOF and the rest of values was slightly smaller, however still a magnitude higher for BcAP then for the other fields.

Table 2. Distance-based method after normalisation

	BcAP (2007)	**PDS (2007)**	BIO (2007)	PTS (2008)
LOF:	1,97	9,38	1,01	1,18
	GRA (2008)	BcAP (2008)	PSK (2007)	GRA (2007)
LOF:	1,0	1,04	0,96	1,02
	MI (2008)	PSZD (2007)	**UMI (2008)**	MI (2007)
LOF:	1,18	0,91	**3,03**	1,92
	BcAP (2006)	PSZD (2008)	PSK (2008)	UMI (2007)
LOF:	0,99	1,00	1,07	1,18

Several fields are massive, with tens or even hundreds students. To limit the influence of it, we normalised the data and again used distance-based method. After normalisation, see Table 2, we can observe that Parallel and distributed systems differs significantly, namely because of a grade and a number of credits (both subscribed and obtained). It is surprising that the second outlying filed in Artificial intelligence UMI. This field was not chosen as anomalous by an expert. However, both field are pretty similar w.r.t grades and numbers of credits, although for UMI the difference form the other fields is not so enormous. When looking for the same field one year sooner, there is no evidence for anomaly. We can conclude that for UMI it is just a coincidence.

Using trend-based method it is again PDS (2007) followed by MI (2008) (see Table 3) although with more than twice smaller outlier factor than PDS. Neither the latter was chosen by an expert. Possible explanation can be that both fields - PDS and MI - are more theoretical fields and are being chosen by good students but the values of features for MI do not differ so much from the rest of fields and are difficult to detect from two-dimensional graphs.

Table 3. Trend-based method after normalisation

	BcAP (2007)	**PDS (2007)**	BIO (2007)	PTS (2008)
LOF:	1,0	**6,25**	1,09	1,15
	GRA (2008)	BcAP (2008)	PSK (2007)	GRA (2007)
	1,0	1,0	1,03	1,0
	MI (2008)	PSZD (2007)	UMI (2008)	MI (2007)
LOF:	**3,75**	0,88	1,66	1,09
	BcAP (2006)	PSZD (2008)	PSK (2008)	UMI (2007)
	1,66	1,03	1,0	0,99

7 Conclusion and Future Work

We proposed a novel method for anomaly detection for short time series that employes anomaly detection and visual analytics, namely motion charts. We showed how this method can be used for analysis CS study fields.

There are many fields where ODEXEDAIME can be used, e.g. in analysis of trends in average salary or unemployment or in analysis of financial data. The current version transforms a multivariate time series into a set of univariate ones. For our task - analysis of Computer Scinece study fields - it is no disadvantage. However, it would be necessary to overcome this limit, as in general it may be not working. Limits of LOF are well-known - a user need to be careful when compares two values of LOFs. Again, here it was not a problem. In general, a probabilistic version of LOF probably need to be used.

There are several ways that should be followed to improve ODEXEDAIME. In the recent version results of different anomaly detection methods has been evaluated and then presented to a user separately. There is also possibility to use the method in supervised manner when normal and anomalous elements are available. Challenge is to use ODEXEDAIME for class-based outliers [8,13]. Actually explored study field are grouped into two study programs - Infromatics and Applied informatics. With these methods we would be able to find e.g. a study field from Informatics study program that is more close to the Applied Informatics study fields.

Acknowledgments. We thank to the members of Knowledge Discovery Lab at FIMU for their assistance and the anonymous referees for their comments. This work has been supported by Faculty of Informatics, Masaryk University.

References

1. Aggarwal, C.C.: Outlier Analysis. Springer, New York (2013)
2. Al-Aziz, J., Christou, N., Dinov, I.D.: Socr “motion charts”: an efficient, open-source, interactive and dynamic applet for visualizing longitudinal multivariate data. J. Stat. Educ. **18**(3), 1–29 (2010)
3. Andrienko, G., Andrienko, N., Kopanakis, I., Ligtenberg, A., Wrobel, S.: Visual analytics methods for movement data. In: Giannotti, F., Pedreschi, D. (eds.) Mobility Data Mining and Privacy, pp. 375–410. Springer, Berlin (2008)
4. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (2000)
5. Géryk, J.: Using visual analytics tool for improving data comprehension. In: Proceedings for the 8th International Conference on Educational Data Mining (EDM 2015), pp. 327–334. International Educational Data Mining Society (2015)
6. Géryk, J., Popelínský, L.: Analysis of student retention and drop-out using visual analytics. In: Proceedings for the 7th International Conference on Educational Data Mining (EDM 2014), pp. 331–332. International Educational Data Mining Society (2014)
7. Géryk, J., Popelínský, L.: Towards academic analytics by means of motion charts. In: Rensing, C., Freitas, S., Ley, T., Muñoz-Merino, P.J. (eds.) EC-TEL 2014. LNCS, vol. 8719, pp. 486–489. Springer, Heidelberg (2014)
8. He, Z., Xu, X., Huang, J.Z., Deng, S.: Mining class outliers: concepts, algorithms and applications in CRM. Expert Syst. Appl. **27**(4), 681–697 (2004)
9. Keim, D.A., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: definition, process, and challenges. In: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (eds.) Information Visualization. LNCS, vol. 4950, pp. 154–175. Springer, Heidelberg (2008)
10. Keim, D.A., Mansmann, F., Schneidewind, J., Thomas, J., Ziegler, H.: Visual analytics: scope and challenges. In: Simoff, S.J., Böhlen, M.H., Mazeika, A. (eds.) Visual Data Mining. LNCS, vol. 4404, pp. 76–90. Springer, Heidelberg (2008)
11. Lauría, E.J., Moody, E.W., Jayaprakash, S.M., Jonnalagadda, N., Baron, J.D.: Open academic analytics initiative: initial research findings. In: Proceedings of the Third International Conference on Learning Analytics and Knowledge, LAK 2013, pp. 150–154, New York, NY, USA. ACM (2013)

12. Miksch, S., Aigner, W.: A matter of time: applying a data-users-tasks design triangle to visual analytics of time-oriented data. Comput. Graph. **38**, 286–290 (2014)
13. Nezvalová, L., Popelínský, L., Torgo, L., Vaculík, K.: Class-based outlier detection: staying zombies or awaiting for resurrection? In: Fromont, E., De Bie, T., van Leeuwen, M. (eds.) IDA 2015. LNCS, vol. 9385, pp. 193–204. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24465-5_17
14. Tekusova, T., Kohlhammer, J.: Applying animation to the visual analysis of financial time-dependent data. In: 11th International Conference on Information Visualisation, pp. 101–108 (2007)