Algorithms for Time Series Comparison

Jiří Fejfar, Arnošt Motyčka, Štěpán Filípek

Abstract – In this paper we compare results of four chosen algorithms suitable for the time series clustering. The comparison is made in sense of agreement between clustering results represented in a special form of confusion matrix (matching matrix). We use a musical data, specifically music excerpts volume development, as a time series instances. This musical characteristic is used in computer-aided tool for musical analysis developed by our group.

Keywords – Clustering, Time series, Music analysis, Self-organizing map, Expectation-maximization, k-means

1 Introduction

There are many music descriptors used for music analysis. We can divide them according to the domain they are defined on. In the time domain we can gain volume level and indicators based on beats such as rhythm and tempo. In the frequency domain we can gain characteristics of melody and harmony and characteristics of timbre. These characteristics can be used for musical objects recognition by analogy to [4].

We are developing tool for computer aided music analysis. We decided to start by using volume level characteristic for this tool, because it is simple and clear method to describe music events in the perspective of the whole musical piece. Also it is common practice, that composers take dynamics (volume) into account and put the highest (or lowest) dynamic level to the most important parts of composition.

1.1 Dynamics development description

In figure 1 we can see diagram describing development of dynamics in the piece "Per Slava" from Krzysztof Penderecki. It is made from the score dynamic signs. We can see the signs on the Y axis. On the X axis is the time and under the axis is described formal structure of the piece. The formal parts are equidistant, so it doesn't reflect real time axis, but it is enough for orientation in the piece.

Figure 2 describes the interpretation of the piece "Per Slava" by Steven Honigberg (available from last.fm . We can express dynamic level by changing standard deviation of the signal [1].

We can observe several differences between a theoretical model made from the score and it's interpretation. One is in the length of recording changing from 4:35 min (275s) to 5:45 min (345s). Another differences are in the dynamics development. One dynamic level is defined for each formal part, but interpret makes naturally dynamic changes more often. Despite of this differences, there are also many common features. Both diagrams start and finish in low dynamics. There is lowest point in the middle followed by the highest peak.

1.2 How to find similar musical excerpt?

Now we would like to find some other recordings in our database, which have similar dynamic properties. We can use some clustering algorithm to find similar groups of recordings and after that we can chose recordings in the same cluster, where "Per Slava" is located, as similar ones. We have to set up, what is similar and what is no more similar already. We have to define the bound. This will result in the number of gained clusters and number of (similar) recordings in that clusters. Other way round we can set this border by defining the number of clusters. In this paper we are focusing to the way, how to set number of clusters in SOM. The problem is formulated in chapter 2 and our results are presented in chapter 3.3.

2 Problem Formulation – how to compare clustering results?

We want to compare results of several clustering algorithms in the task described in chapter 1.2. It is common to use Confusion matrix in classification (or prediction [7]) to compare classification results with real situation. If we have set of instances belonging to two sets, positives and negatives, there are four possibilities of results:

- True positives – instances correctly classified as positives.
- True negatives – instances correctly classified as negatives.
- False positives – instances wrongly classified as positives.

Manuscript received July 30, 2011. This work was supported by IGA project 39/2011.

Jiří Fejfar is with Faculty of Business and Economics, Mendel University in Brno, Department of Informatics, Zemědělská 1, 613 00, Brno, Czech Republic, e-mail: jiri.fejfar@mendelu.cz

Arnošt Motyčka is with Faculty of Business and Economics, Mendel University in Brno, Department of Informatics, Zemědělská 1, 613 00, Brno, Czech Republic, e-mail: arnost.motycka@mendelu.cz

Štěpán Filípek is with Faculty of Music, Janáček Academy of Music and Performing Arts, Komenskeho nam. 6, 616 69 Brno, Czech Republic, e-mail: stepan.filipek@hf.jamu.cz

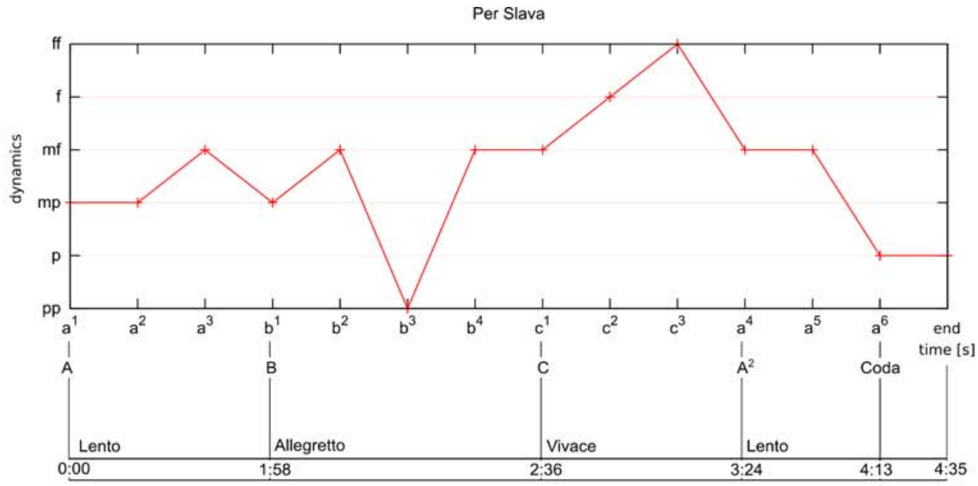

Figure 1: Dynamics development with formal structure description

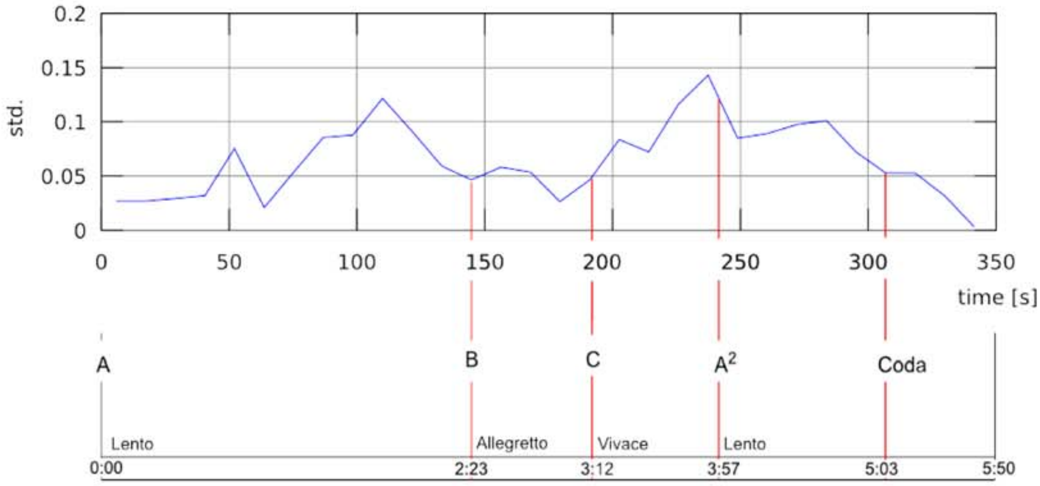

Figure 2: Standard deviation

- False negatives – instances wrongly classified as positives.

We can see confusion matrix in table 1. There are i true positive instances, j true negative, k false positive and l false negative.

In clustering (unsupervised learning [5] analogy to classification) we don't know real situation. We only have results of clustering made by different algorithms. For this comparison we modified confusion matrix to compare results of two clusterings. Let's say, we have algorithm $AlgA$ resulting in three clusters α, β, γ and algorithm $AlgB$ resulting in ψ and ω. We can make matching matrix shown in the table 2.

We can see from the matrix, that cluster α from algorithm $AlgA$ has i instances in cluster ψ and l instances in cluster ω. If $i >> l$, we can say, that cluster α from algorithm $AlgA$ is equivalent to cluster ψ from algorithm $AlgB$. From another perspective, we can say, that cluster ω from algorithm $AlgB$ has l instances in cluster α, m instances in cluster β and n instances in cluster γ of $AlgA$.

In this paper, we compare clustering results of these four algorithms:

- k-means (Weka 3.7.3),
- EM (Weka 3.7.3),
- SOM 1.0.1 (Weka 3.7.3) (SOM is from package manager),
- SOM own C++ implementation

3 Problem Solution

3.1 Data

We use Magnatagatune database [3] of music excerpts. It has several advantages: it is public accesible and downloadable, its licence enables scientific use of excerpts and excerpts are labeled by users. For our experiment we use recordings labeled by label "heavy" and "silence".

There are 217 music excerpts tagged by label "heavy" and 68 by tag "silence" in Magnatagatune. We processed these 285 files with GNU Octave scripts. These sound processing algorithm divide each recording into 3 parts and quantify the standard deviation of the signal for each part, resulting in 285 3-dimensional

ISBN: 978-1-61804-034-3

Table 1: Confusion matrix

	actual positive	actual negative
classified positive	i	k
classified negative	l	j

Table 2: Matching matrix

	α	β	γ
ψ	i	j	k
ω	l	m	n

feature vectors representing recordings in mean of volume. We save these vectors in csv file readable with Weka. We can see header of the file with two feature vectors:

```
X1,X2,X3,label
0.22253,0.19057,0.18924,heavy
0.21556,0.21785,0.21726,heavy
```

Acquired feature vectors are well separable as we can see in the figure 3, which is promising for good clustering results. On the X axis are values of vector components, on the Y axis are frequencies of observations. The blue color belongs to recordings labeled as "silence", the red color belongs to the "heavy recordings". We can see, that distribution of "heavy" recordings has it's peak in the middle in contrast to the "silence" vectors distribution center on the left (value 0).

3.2 Labels

Magnatagatune recordings are tagged with labels by humans. These labels has some properties rooted in the principle of tagatune game. There is effort to make label quickly as the excerpt starts playing, so it reflect the beginning of recordings more than the end.

There is no use for labels in clustering, but we can use them in quantification of clustering error – classes to clusters evaluation, which tells us how are clusters fit to humans labels.

3.3 Number of clusters

There is algorithm usable for finding the right number of clusters called v-fold clustering [2]. In our experiment we have found optimal number of clusters with E-M algorithm in Weka. This implementation has the ability of determining cluster number. This is done by minimizing loglikelihood value during cross-validation. The E-M algorithm selects optimal number of clusters to 6, executed with following configuration:

```
weka.clusterers.EM -I 100 -N ...
-1 -M 1.0E-6 -S 100
```

3.4 Comparison of results

In the figure 4 we can see cluster assignments of k-means ($k = 2$) [6] after 11 iterations. Clusters are visualised in 2-dimensional space, ignoring X3 vector component.

For results interpretation consistent with section 2, let's denominate "silence" as positive and "heavy" as negative. We can further see, that majority of "silence" recordings belongs to cluster0 (blue color) respective "heavy" recordings belongs to cluster1 (red color). We can observe:

- true positives: blue crosses (cluster0 – "silence"),
- true negatives: red crosses (cluster1 – "heavy"),
- false positives: blue squares (cluster0 – "heavy"),
- false negatives: red squares (cluster1 – "silence").

For quantified information we can see the confusion matrix for this clustering in the table 3.

There are 10 incorrectly clustered instances from 285 recordings, which results in 96,5% accuracy of k-means algorithm (when clustering into 2 clusters).

When we change number of clusters to optimal value – 6, we get confusion matrix, which is in the table 4. Here we can compare all selected algorithms according to how do they fit to human labels, supposing that clusters belong to the class, where belong most of their instances. For example: k-means cluster 4 has 25 instances in heavy class and 8 instances in silence class. We suppose, that all instances should belong to single class, so we consider those 8 instances as wrong clustered.

Algorithms were executed with following configuration:

```
weka.clusterers.SimpleKMeans -N 6 -A ...
"weka.core.EuclideanDistance ...
-R first-last" -I 500 -S 10
weka.clusterers.SelfOrganizingMap ...
-L 1.0 -O 2000 -C 1000 -H 2 -W 3 -I -S
```

, our SOM implementation with $2x3$ topology and 5 iterations:

```
./autoSom weka.csv 2 3 5 > 6-SOM-cpp.arff
```

ISBN: 978-1-61804-034-3

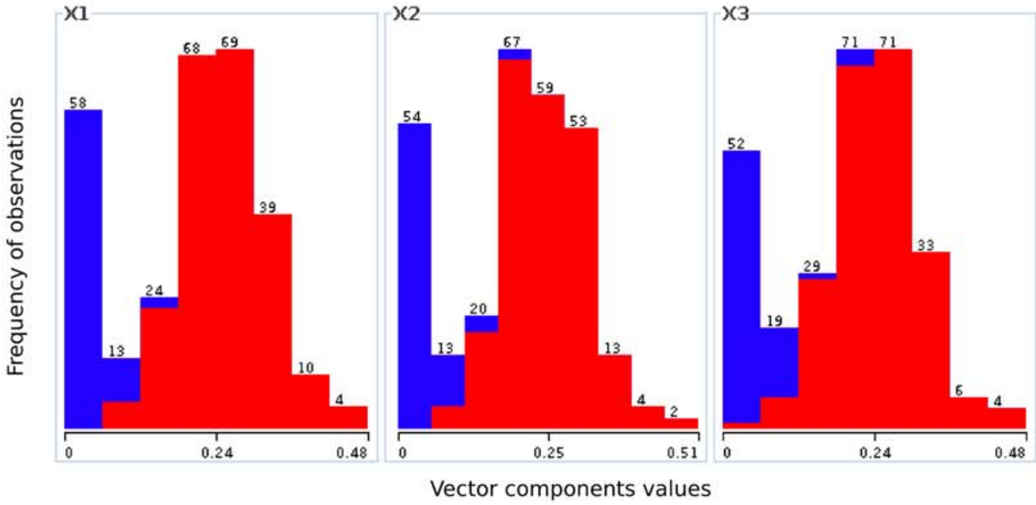

Figure 3: Vector component distribution

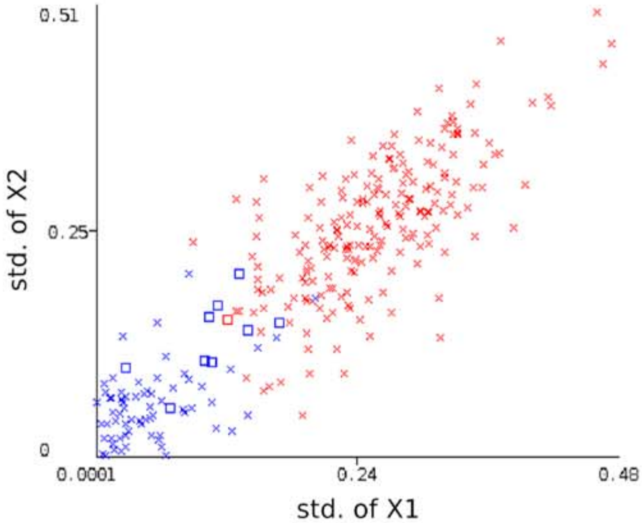

Figure 4: Vector component distribution

Table 3: k-means $k = 2$

	heavy	silence
cluster0	9	67
cluster1	208	1

We can see, that accuracy of k-means increased (from 96,49 % to 97,55 %) with incrementation of cluster number (from 2 to 6). Accuracy of EM is a bit worse and the best accuracy have SOM algorithm (both implementations have the same accuracy 97,55 %).

Now, we would like to know, how much are the different SOM algorithm implementations clustering results similar. To quantify this information we use "matching matrix" described in section 2. We can see gained matrix in the table 5.

From this matrix we can discover, that cluster2 of SOM (C++) and cluster5 of SOM (weka) are presumably equivalent, having the same number (30) of instances. Even more interesting information is about both SOM implementations problematic cluster1. There are 5 mistakes of C++ implementation and 2 mistakes of Weka. We can see, that both algorithms agree with 34 recordings, but another 6 are placed in cluster 3 (where are 5 mistakes of SOM weka). We expect, that analysis of these differences can improve the performance, when the semi-supervised algorithm (such as co-training) will be utilised. Match matrix contains a lot of zeros, what indicates unity of results of SOM in Weka and our SOM implementation.

ISBN: 978-1-61804-034-3

Table 4: Confusion matrix (clusters – labels)

	k-means		EM		SOM (weka)		SOM (C++)	
	heavy	silence	heavy	silence	heavy	silence	heavy	silence
cluster0	81	0	37	13	0	55	87	0
cluster1	30	0	7	0	38	2	35	5
cluster2	7	0	27	0	83	0	0	30
cluster3	0	60	59	0	5	11	93	0
cluster4	25	8	0	55	61	0	2	19
cluster5	74	0	87	0	30	0	0	14
	8		13		7		7	
	97,19%		95,44%		97,55%		97,55%	

Table 5: Matching matrix (SOM weka – SOM C++)

		SOM (C++)					
		cluster0	cluster1	cluster2	cluster3	cluster4	cluster5
SOM (weka)	cluster0	0	0	30	0	11	14
	cluster1	0	34	0	6	0	0
	cluster2	57	0	0	26	0	0
	cluster3	0	6	0	0	10	0
	cluster4	0	0	0	61	0	0
	cluster5	30	0	0	0	0	0

4 Conclusion

Results of selected four algorithms are very promising. Accuracy of compared algorithms in table 4 are higher than 95 %. This is partially caused by simple situation in distinction between "heavy" and "silence" of music. We can see, that our data are well separable in figure 3. Despite of simplicity of this clustering task, we have to consider subjectivity of the bound. Proposed form of confusion matrix provides information about differences in clustering of compared algorithms. The analysis of these differences can be useful in the study of semi-supervised learning.

We will go further with this experiment to more complex tasks like instrument clustering or even birds sing clustering, where will perhaps more complex methods, like genetic algorithms [8], take place.

References

[1] Fejfar, J., Weinlichová, J., Šťastný, J., *Musical Form Retrieval*, In MENDEL 2010, 16th International Conference on Soft Computing. Brno University of Technology, 2010, s. 280-285. ISSN 1803-3814. ISBN 978-80-214-4120- 0.

[2] Hsu, Chih-Wei and Chang, Chih-Chung and Lin, Chih-Jen. *A Practical Guide to Support Vector Classification*, 2000.

[3] Law, E., Von Ahn, L., *Input-agreement: A New Mechanism for Data Collection using Human Computation Games*, 2009. Proc. of CHI, Boston, Massachusetts, USA. ACM press 978-1-60558-247-4, pp. 1197-1206.

[4] Škorpil, V., Šťastný, J. *Comparison Methods for Object Recognition*, In Proceedings of the 13th WSEAS International Conference on Systems. Rhodos, GREECE, 2009. p. 607-610. ISBN 978-960-474-097-0.

[5] Škorpil, V., Šťastný, J. *Comparison of Learning Algorithms*, In 24 th Biennial Symposium on Communications. Kingston, CANADA, 2008. p. 231-234. ISBN 978-1-4244-1945-6.

[6] Škorpil, V., Šťastný, J. *Back-Propagation and K-Means Algorithms Comparison*, In 2006 8th International Conference on SIGNAL PROCESSING Proceedings. Guilin, CHINA, IEEE Press, 2006. p. 1871-1874. ISBN 978-0-7803-9736-1.

[7] Štencl, M., Šťastný, J. *Neural network learning algorithms comparison on numerical prediction of real data*, In MENDEL 2010, 16th International Conference on Soft Computing. Brno University of Technology, 2010, p. 280-285. ISSN 1803-3814. ISBN 978-80-214-4120-0.

[8] Šťastný, J., Škorpil, V. *Genetic Algorithm and Neural Network*, In Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications, Vouliagmeni, GREECE, 2007, p. 347-351. ISSN 1790-5117. ISBN 978-960-8457-96-6.