UNSUPERVISED FEATURES LEARNING FOR SAMPLED VECTOR
FIELDS

MATEUSZ JUDA

Abstract. In this paper we introduce a new approach to computing hidden features of
sampled vector fields. The basic idea is to convert the vector field data to a graph structure
and use tools designed for automatic, unsupervised analysis of graphs. Using a few data
sets we show that the collected features of the vector fields are correlated with the dynamics
known for analytic models which generates the data. In particular the method may be useful
in analysis of data sets where the analytic model is poorly understood or not known.

1. Introduction

Continuous mathematical models are useful to analyze and draw conclusions about com-
plicated physical systems, where values of a system states are assumed to be real numbers.
However, nowadays we have countless possibilities of data collection, so scientific and in-
dustrial challenges are mostly data driven. We have only finite amount of information, so
models should be well-fitted to the observed data and generalize well. Usually it means that
we have to create highly parameterized models in an automatic way.

We propose a new method for automatic modeling of vector field data sets. In particular,
our method takes on the input a collection of finite vector fields and creates a low dimensional
description of the data. The description encodes features of the observed vector field and
allows us to further analyze the physical system using smaller amount of information. We
only require a point cloud with vectors attached at each point. There is no need to manually
model the system, e.g. via differential equations. Thanks to this we can analyze data
generated by processes for which it is hard to create a traditional model, for instance magnetic
field on the Sun surface [24].

The paper is organized as follows: Section 1 contains introduction, description of related
work, and recalls theoretical model. In Section 2 we describe the problem and reduce it
to a well-known NLP problem. In Section 3 we describe details of the algorithm. Finally,
Section 4 contains examples and Section 5 finalizes the paper.

1.1. Related work. The computation of global dynamical information is a challenging
problem for applications. Especially, characterization of the global dynamical structure and
its changes are fundamental in many disciplines, e.g. computational biology, engineering.

The author is supported by the Polish National Science Center under Maestro Grant
2014/14/A/ST1/00453.
Scientific computations give only finite information, hence the information we obtain is already reduced.

Combinatorial dynamics and computational topology are powerful tools for the task. In particular, the tools are useful in classification of the qualitative properties of parameterised models. The database approach [2] helps to understand models where it is difficult to measure parameters. For sampled parameters an outer approximation of the dynamics is computed using rigorous numerical methods. The dynamics is then represented as a directed graph and classified using the Morse decomposition and Conley indices. The methods is useful when we know the parameterised dynamical system model. Then, by rigorous simulations, we can find all possible dynamics and match them with collected data. Recent application of the method is applied to understand the global dynamics of gene regulatory networks [8, 14].

In experiments very often one quantity is measured - a time series - while in numerical simulations the full state of a system is an observable. A powerful tool to reason about the unknown system using only a partial information is the Takens embedding theorem [26]. The theorem has been used in [19, 5], together with the Conley index theory of multivalued maps, to identify dynamics from sampled data.

A different approach to sampled dynamical systems is presented in [10]. The method analyze data points given by an unknown self-map. The results presented in this paper suggest that the persistent homology of eigenspaces picks up the important dynamics from small data sample.

Problems similar to our work are considered in [25] where the input vector field is converted to a piecewise constant vector field. Then the Morse decomposition is computed for trajectories obtained using geometrical rules. A main difference, when compared to our work, is that the method is limited to triangulated manifold surfaces and considers only the Morse decomposition while we show a more general framework.

1.2. A combinatorial dynamical system from a sampled vector field. In this section we recall some definitions and results from [20, 9]. As mentioned in [9] also here the presented results may be generalized to arbitrary finite $T_0$ topological spaces [1]. From the viewpoint of applications, a finite topological space may be a collection of cells of a simplicial, cubical, or general cellular complex approximating a cloud of sampled points. For the sake of this paper we use simplicial complexes only.

Let $K$ be a finite simplicial complex, either a geometric simplicial complex in $\mathbb{R}^d$ or an abstract simplicial complex (see [21 Section 1.2, 1.3]). We consider $K$ as a poset $(K, \preceq)$ with $\sigma \preceq \tau$ if and only if $\sigma$ is a face of $\tau$ (also phrased $\tau$ is a coface of $\sigma$). The poset structure of $K$ provides, via Alexandrov Theorem [11], a $T_0$ topology on $K$. We say that $A \subseteq K$ is orderly convex if for any $\sigma_1, \sigma_2 \in A$ and $\tau \in K$ the relations $\sigma_1 \preceq \tau$ and $\tau \preceq \sigma_2$ imply $\tau \in A$. We remark that orderly convex sets in $K$ may be characterized in the language of the associated Alexandrov topology. Namely, $A \subseteq K$ is orderly convex if and only if it is locally closed (see [11] Sec. 2.7.1, pg 112]) in the Alexandrov topology $\mathcal{T}_K$. 
We define a multivector as an orderly convex subset of $K$ and a combinatorial multivector field ($\text{cmf}$ in short) on $K$ as a partition $\mathcal{V}$ of $K$ into multivectors. The definition encompass the combinatorial vector field of Forman [13, 12] as a special case.

Given a $\text{cmf} \mathcal{V}$, we denote by $|\sigma|_\mathcal{V}$ the unique $V$ in $\mathcal{V}$ such that $\sigma \in V$. We associate with $\mathcal{V}$ a combinatorial dynamical system $F_\mathcal{V} : K \rightarrow K$ given by $F_\mathcal{V}(\sigma) := \text{cl} \sigma \cup |\sigma|_\mathcal{V}$.

When the dynamics which is sampled constitutes a flow, that is when time is continuous as in the case of a differential equation, the sampled data often consists of a cloud of points with a vector attached to every point. In this case the construction of combinatorial dynamical system is done in two steps. In the first step the cloud of vectors is transformed into a combinatorial vector field in the sense of Forman [13, 12] or its generalized version of combinatorial multivector field [20]. In the second step, the combinatorial multivector field is transformed into a combinatorial dynamical system.

**Figure 1.** Left: A cloud of vectors. Middle: A possible combinatorial multivector field representation of the cloud of vectors. Right: The associated combinatorial dynamical system represented as a digraph.

Figure 1(left) recall a toy example of a cloud of vectors [9]. It consists of four vectors marked red at four points $P$, $Q$, $R$, $S$. One of possible geometric simplicial complexes with vertices at points $P$, $Q$, $R$, $S$ is the simplicial complex $K$ consisting of triangles $\{PQR\}$, $\{QRS\}$ and its faces. A possible multivector field $\mathcal{V}$ on $K$ constructed from the cloud of vectors consists of multivectors $\{P, PR\}$, $\{R, QR\}$, $\{Q, PQ\}$, $\{PQR\}$,$\{S, RS, QS, QRS\}$. It is indicated in Figure 1(middle) by blue arrows between centers of mass of simplices. Note that in order to keep the figure legible, only arrows in the direction increasing the dimension are marked. The associated combinatorial dynamical system $F_\mathcal{V}$ presented as a digraph is in Figure 1(right). Note that in general $K$ and $\mathcal{V}$ are not uniquely determined by the cloud of vectors. One possible method for constructing combinatorial multivector fields from a cloud of vectors is discussed in [9, Section 7.2]. Given a combinatorial multivector field $\mathcal{V}$, we define a combinatorial multivector field graph ($\text{cmf graph}$ in short), denoted by $G_\mathcal{V}$, as a directed graph with the set of vertices $V(G_\mathcal{V}) := \mathcal{V}$ and the set of edges $E(G_\mathcal{V}) := \{(u, v) \mid v \in F_\mathcal{V}(u)\})$. 
2. Problem description

Our goal is to explore combinatorial multivector field graphs structure. We do it, by finding features of multivectors, as fixed length sequences of real numbers, such that multivectors with a similar local structure in a cmf graph have similar features. We also want to extend the features to whole graphs, in such a way that cmf graphs representing similar combinatorial dynamical systems have similar features. At this point we skip formal definitions of the similarities mentioned above and we use experimental justifications of the presented methods. A theory of a similarity for combinatorial dynamical systems is not developed yet and the topic is beyond the scope of this paper. However, as long as a combinatorial dynamical system \( F \) reflects the dynamics of the sampled system, we can extend tools designed for graphs.

Let \( \mathcal{G} \) be a collection of labeled graphs, namely each vertex \( v \) of a graph \( G \in \mathcal{G} \) has a label \( l_v \) from some set of labels \( L \subseteq \mathbb{N} \). In Section 3.1 we propose an assignment of labels which uses topological properties of the multivectors. Our main goal is to learn latent representations of labels of a cmf graph \( G_V \). Namely, we want to find an encoding function \( \Phi : L \rightarrow \mathbb{R}^D \), for fixed dimension \( D \), such that the codes \( \Phi(l_u) \) and \( \Phi(l_v) \) are close whenever there is a similar local structure in \( G_V \) around two multivectors \( u \) and \( v \). We want to extend the encoding function to graphs, namely for two cmf \( V_1 \) and \( V_2 \) the codes of \( \Phi(G_{V_1}) \) and \( \Phi(G_{V_2}) \) are close whenever the dynamical systems which generates \( V_1 \) and \( V_2 \) are similar. The values of \( \Phi \) should depend on qualitative local features of the vertices and do not depend on the vertices enumeration.

Let \( W_v = \{w_1, w_2, \ldots, w_k\} \subseteq G \) be a random walk rooted at \( v \in G \in \mathcal{G} \), i.e. \( v = w_1 \) and \( w_{i+1} \) is a randomly selected neighbor of \( w_i \). We explore the graph \( \mathcal{G} \) using the DeepWalk technique, namely short random walks, as described in [22]. The idea behind the approach is to generalize Natural Language Processing (NLP) methods to explore graphs. In this setting we treat labels of vertices as words and the random walks as sentences in an artificial language. Afterwards, we use the Continuous Skip-gram Model [17] to analyze the text documents structure and to find \( \Phi \):

\[
\minimize_{\Phi} \quad \minimize - \log P(\{l_{w_i-w}, \ldots, l_{w_i+w}\} \setminus \{l_{w_i}\} \mid \Phi(l_{w_i})).
\]

As we can see in (1) the goal is to predicts context based on a word without taking into account the order of words. A relaxation scheme described in [17, 18] provides efficient algorithms to compute the latent representation of words, the encoding \( \Phi \).

3. Algorithm

In the context of this paper we assume a family \( \mathcal{V} \) of combinatorial multivector fields is given. Having a cmf \( V \in \mathcal{V} \) we transform it to the NLP data in the following steps:

(1) assign labels (words) to multivectors,
(2) using short random walks encode the structure of \( V \) as a text document,
(3) using NLP techniques analyze the document and extract information about the multivector fields.
Bellow we present detailed description of the steps.

3.1. **Topological vocabulary.** The NLP procedure requires a vocabulary in order to assign labels to the vertices. We construct labels which grasp some local, topological properties of the vertices in the vector field. Let $h : \mathbb{L} \to \mathbb{N}$ be a hash function for lists. We need it to transform list of numbers and list of lists of numbers into a finite set of natural numbers. For a multivector $v \in \mathcal{V}$, that is a vertex in $G_{\mathcal{V}}$, we first define the *label of $v$ at level $(0,0)$*, denoted by $\mathcal{L}^{0,0}(v)$, in the following way:

$$\mathcal{L}^{0,0}(v) := h(\max_{\sigma \in v} \dim \sigma, |v|, \chi(v)),$$

where $\dim \sigma$ denotes the dimension of a cell $\sigma$, $|v|$ stands for the cardinality of $v$, and $\chi(v)$ is the Euler characteristic of $v$. We define *label of $v$ at level $(b,f)$*, denoted by $\mathcal{L}^{(b,f)}(v)$, in the following way:

$$\mathcal{L}^{(b,f)}(v) := h(\mathcal{L}^{0}(v), \text{sorted}(\{\mathcal{L}^{0}(u) | u \in N^+_f(v)\}), \text{sorted}(\{\mathcal{L}^{0}(u) | u \in N^-_b(v)\})),$$

where $N^+_f(v)$ (resp. $N^-_b(v)$) are sets of vertices in the forward (resp. backward) distance from $v$ not bigger than $f$ (resp. $b$). It is worth to note that the above construction requires to fix parameters $(b,f)$ and it is only an example of many possible labelings.

Currently we use NLP methods which do not use the words (labels) structure. It means that we can arbitrary map the tuples to numbers using a reasonable good hashing function $h$. However, there are NLP methods which operates on sub-words ($n$-grams) [7] and more sophisticated labelings may take advantage of a multivector neighborhood structure.

As an example we consider the multivector field $\mathcal{V}$ and its combinatorial dynamical system $F_{\mathcal{V}}$ presented in Figure 1. Figure 2 presents the associated graph on multivectors $G_{\mathcal{V}}$. Table 1 presents step by step calculations of the values of $\mathcal{L}^{(1,1)}$. 

![Figure 2. $G_{\mathcal{V}}$ graph for the example presented in Figure 1.](image-url)
Table 1. Step by step calculation of labels at level 1 for the example presented in Figure 1 and Figure 2.

| $v$ | simplices of $v$ | $\mathcal{L}^{0,1}(v)$ | $N_1^+(v)$ | $N_1^-(v)$ | $\mathcal{L}^{1,1}(v)$ |
|-----|-----------------|---------------------|-------------|-------------|---------------------|
| $v_1$ | $\{P, PR\}$ | $(1, 2, 0)$ | $\{v_2\}$ | $\{v_3, v_4\}$ | $(1, 2, 0), [(1, 2, 0), (1, 2, 0), (2, 1, 1)]$ |
| $v_2$ | $\{R, QR\}$ | $(1, 2, 0)$ | $\{v_3\}$ | $\{v_1, v_4, v_5\}$ | $(1, 2, 0), (1, 2, 0), (2, 1, 1), (2, 4, 0)$ |
| $v_3$ | $\{Q,QP\}$ | $(1, 2, 0)$ | $\{v_1\}$ | $\{v_2, v_4, v_5\}$ | $(1, 2, 0), (1, 2, 0), (2, 1, 1), (2, 4, 0)$ |
| $v_4$ | $\{P,QR\}$ | $(2, 1, 1)$ | $\{v_1, v_2, v_3\}$ | $\emptyset$ | $(2, 1, 1), (1, 2, 0), (1, 2, 0), (1, 2, 0), \emptyset$ |
| $v_5$ | $\{S, RS, QS, QRS\}$ | $(2, 4, 0)$ | $\{v_2, v_3\}$ | $\emptyset$ | $(2, 4, 0), (1, 2, 0), (1, 2, 0), \emptyset$ |

3.2. Corpus. A $d$-random multivector walk on $G_V$ from $s$, denoted by $\mathcal{W}_V^{d+}(s)$, is a stochastic process with random variables $\{W_1, W_2, \ldots, W_d\}$ such that $W_1 = s$, and $W_{i+1}$ is a vertex chosen at random from the set $N_1^+(W_i) \cup \{W_i\}$. The probability $P(W_{i+1} = u)$ is defined as $p_u/\sum_v p_v$, where

$$p_v = \begin{cases} 1, &\text{if } v \in N_1^+(W_i) \text{ and } v \neq W_i \\ 1, &\text{if } v = W_i \text{ and } W_i \text{ is critical} \\ 0, &\text{otherwise.} \end{cases}$$ (2)

In the above definition we can replace $N^+$ with $N^-$ and reverse the order of the random multivector walk. This way we define a $d$-random multivector walk on $G_V$ to $t$, denoted by $\mathcal{W}_V^{d-}(t)$.

Let $\mathcal{Y}$ be a collection of combinatorial multivector fields. A $(c,d)$-random multivector corpus of $\mathcal{Y}$ is a stochastic process:

$$\mathcal{W}_\mathcal{Y}^{c,d} := \bigoplus_{v \in \mathcal{Y}} \mathcal{W}_\mathcal{Y}^d(u),$$

where $\bigoplus$ denotes concatenation of sequences of the random variables. Note that in the above definition we take $c$ copies of each random multivector walk. We define a multivector $(f, b, c, d)$-corpus of $\mathcal{Y}$, denoted by $\mathcal{L}^{f,b,\mathcal{Y}}^{c,d}$, as $\{\mathcal{L}^{f,b}(W) \mid W \in \mathcal{W}_\mathcal{Y}^{c,d}\}$, where $\mathcal{L}^{f,b}(W)$ is a label of $W$ at level $(f, b)$ (as defined in Section 3.1). We skip the parameters $f, b, c, d$ if they are clear from the context.

We treat a multivector corpus as a text document for which the labels are words and the walks are sentences. Next we apply NLP methods to the document. We emphasize that the methods we use does not depend on the order of sentences, so we can take any order in the $\bigoplus$ notation. On the other hand, the skip-gram model generates word contexts using the sliding window technique. Hence the order of words in a sequence is important.

3.3. Encoding. The distributional hypothesis [16] in linguistics says that it is possible to state a linguistic structure in terms of patterns of co-occurrences, i.e. words with similar meaning occur in the same context. It is the main idea behind representing words as elements of a vector space. Neural network models [6, 17, 18, 7] allow us to find an encoding of words
from a large text corpus. The intuition behind the vector space elements is that the distance between similar words is small, and the norm of a word encoding is proportional to its importance in the corpus. In particular, we apply the methods to our artificial multivector corpus introduced in Section 3.2.

We use the Continuous Skip-gram Model [17] which is a shallow neural network trained to predict words within a range before and after the current word. The main parameters for the algorithm are: \textit{window size} \( w \) - number of words around current word, \textit{encoding dimension} \( D \). Intuitively the parameter \( w \) carries examined influence of a word to the meaning of a sentence. The parameter \( D \) controls the number of linguistic features learned by the model. From a trained network we extract a \( D \) dimensional representation of words, denoted by \( \Phi^D \).

For natural languages the parameters values typically are: \( w = 5 \), \( D = 300 \). In our applications usually much smaller dimension is enough. In the sequel we show, that low dimensional encoding may contain useful information.

The goal of recently developed NLP methods is to find meaningful words encoding. Currently, there is no similar method designed directly for documents. As a workaround a common trick is to use a weighted mean of words encodings as the encoding of a document. In our context, for a multivector field graph \( G_V \), we define \( \Phi^D_w(G_V) \) as \( \sum_{v \in V(G_V)} w_v \Phi^D(L(f,b)(v)) \), where \( f \) and \( b \) are fixed and \( w_v \) is a weight of \( v \), e.g. \( \frac{1}{|V(G_V)|} \) or TFIDF of \( v \) (term frequency-inverse document frequency [23]).

3.4. Implementation details. We compute the Skip-gram encoding \( \Phi \) using FastText [7] library. As we mentioned earlier, we cannot use sub-words (n-grams) in the training phase, so the parameters \texttt{minn} and \texttt{maxn} are set to 0. For the examples presented in this paper we also set: learning rate to 0.01, size of the context window to 5, size of word vectors to 2. The number of epochs used for training depends on the size of the corpus. We use 1000 and 5 epochs for examples presented in Section 4.1 and in Section 4.3.

We notice that a random multivector walk may contain a lot of repetitions of labels in adjacent steps. The intuition here is that usually a big part of a vector field contains parallel vectors in straight and unambiguous trajectories. It is worth to leave only the first occurrences of a label, e.g. a random multivector walk \( a \ b \ b \ b \ a \ a \) becomes \( a \ b \ a \). Also we skip random multivector walks build on a single label. Such transformation only slightly change the Skip-gram encoding, however it has a big impact on the running time of the algorithm because the corpus size may be an order of magnitude smaller.

We also noticed that rare words appear close to the boundary of the sampled region. It is because in that area the neighborhood of a multivector depends more on its location than on the vector field structure. We skip such multivectors by checking their distance to the region boundary. The outcome of this simplification is shown in Figure 3c, where the area close to the boundary is white, because labels of the multivectors are not in the corpus, so \( \Phi \) maps them to zero.
4. Examples

For a product of $k$ intervals $I = [I_1^-, I_1^+] \times \ldots \times [I_k^-, I_k^+] \subseteq \mathbb{R}^k$ and a set of numbers $N = \{N_1, \ldots, N_k\} \subseteq \mathbb{N}$ we define an $(I, N)$ regular grid of points, denoted by $\mathbb{R}(I, N)$, as a set of points $\{(x_1, \ldots, x_k) \in I \mid x_i = I_i^- + \frac{(I_i^+-I_i^-)}{N_i} j \text{ for } j \in [0, N_i] \subseteq \mathbb{N}\}$.

In the context of this paper we assume a family $\mathcal{V}$ of combinatorial multivector fields is given. In order to present the examples we recall a possible way to construct a combinatorial multivector field from a cloud of vectors. Let $K$ be a simplicial complex with vertices in a cloud of points $\{p_i \mid i = 1, 2, \ldots, n\} \subseteq \mathbb{R}^d$ and the associated cloud of vectors $\{\vec{v}_i \mid i = 1, 2, \ldots, n\} \subseteq \mathbb{R}^d$ such that vector $\vec{v}_i$ originates from point $p_i$. To construct a cmf on $K$ one can use the algorithm CVCMF \cite{9, Table 1} (called here CVCMFv1). The algorithm requires an angular parameter $\alpha$. We do not analyze the impact of the parameter here. We also have a parameter-less version of the algorithm (called here CVCMFv2). We show results obtained with the old and the new algorithms, however the new one is not published yet.

4.1. Example: Orbit. Consider a simple dynamical system given by following equation:

\[
\frac{\partial x}{\partial t} = -y + x(4 - x^2 - y^2), \quad \frac{\partial y}{\partial t} = -x + y(4 - x^2 - y^2).
\]

The system has a Morse decomposition consisting of a repelling stationary point and an attracting periodic orbit. We want to observe this Morse decomposition in a combinatorial dynamical system constructed from a finite sample of the vector field. We build a Delaunay triangulation $K$ of a regular grid $\mathbb{R}([-4,4] \times [-4,4], \{30, 30\})$. The triangulation contains 1682 triangles and 5163 simplices. To construct a cmf $V$ of the triangulation and vectors originates from its vertices we use the algorithm CVCMFv1 \cite{9, Table 1} with the parameter $\alpha = 0$. The triangulation $K$ and two Morse sets of the system are presented in Figure \ref{fig:3a}. The number of multivectors in $V$ is 1768.

In the next step we label vertices of the graph $G_V$ using labels at level $1,1$, getting 153 distinct labels. Using the labeled graph we generate $(1,1,2,10)$-corpus for which we obtain the Skip-gram encoding $\Phi$ with parameters $w = 5$ and $D = 2$. The encodings are plotted in Figure \ref{fig:3b}, where each dot represents an unique word in the corpus (label in the graph). We use RGBA color space, where the alpha channel of a point is proportional to its norm. In Figure \ref{fig:3c} we show colored multivectors, where each multivector $v$ gets color of $\Phi(\mathcal{L}(v))$. We notice that the encoding $\Phi$ distinguish three behaviors of the vector field: close to the the orbit, around the repellng point, outside the orbit.

4.2. Example: Prey-predator. In this section we consider a prey-predator model \cite{15} given in the following form:

\[
\frac{\partial x}{\partial t} = x\left(1 - \frac{x}{\gamma}\right) - \frac{(1-c)xy}{1+\alpha\zeta+x}, \quad \frac{\partial y}{\partial t} = \frac{\beta[(1-c)x + \zeta]y}{1+\alpha\zeta+x} - \delta y
\]

where $x$ and $y$ denote the biomass of prey and predator respectively, and $\alpha, \beta, c, \delta, \gamma, \zeta$ are parameters. We investigate sampled vector spaces obtained from simulations of the model. In particular, we are interested in qualitative behaviour of the model, where $\alpha \in [0, 2]$. 

Figure 3. An example of a multivector field encoding.

\[ c \in [0, 0.45], \beta = 0.15, \delta = 0.08, \gamma = 4, \zeta = 0.2. \] For varying \( \alpha \) and \( c \) we denote the model by \( P(\alpha, c) \).

In Figure 4 we show a decomposition of the parameter plane \((\alpha, c)\) obtained analytically in [15]. Our goal is to show a correlation between the model dynamics and multivectors encoding \( \Phi \).

Figure 4. Qualitative behaviour of the model \( P \) according to [15]. The parameter plane \((\alpha, c) \subseteq [0, 2] \times [0, 0.45]\) is divided into three regions: oscillatory coexistence (yellow), stable coexistence (green), predator extinction (red).

Let \( K \) be a Delaunay triangulation of a regular grid \( \mathbb{R}([0, 6] \times [0.4, 2.4], \{100, 100\}) \). Let \( \mathcal{V}_{\alpha,c}^{\theta} \) be a cmf of \( K \) computed using the CVCMF algorithm (see Section 3) called with \( K \) and vector field sampled from the prey-predator model \( P(\alpha, c) \). The value of \( \theta \) controle the CVCMF algorithm version, namely:

- if \( \theta \in [0, 2\pi] \), then we use CVCMFv1 with its angular parameter equals to \( \theta \),
- if \( \theta = \emptyset \), then we use CVCMFv2.
Table 2. Multivector corpuses tested for the prey-predator model. Columns (left to right): total number of words in a corpus, the number of distinct words in a corpus, and the loss in the Skip-gram neural network training.

| Corpus variant | # of distinct words | # of words in the corpus |
|----------------|---------------------|--------------------------|
| $C_I = L^1,1.5_{\theta,\psi}^{360}$ | 965 | 3877M |
| $C_{II} = L^{2.2}_{\theta,\psi}^{100}$ | 7979 | 3665M |
| $C_{III} = L^{3.3}_{\theta,\psi}^{100}$ | 28124 | 3635M |
| $C_{IV} = L^{3.3}_{\theta,\psi}^{100}$ | 96589 | 4825M |

The best separation between green and yellow points is visible for $C_{IV}$, which suggests that our new non-parametrized algorithm CVCMF$^2$ finds a better partition of a given complex into multivectors. Further research using machine learning classifiers and higher dimensional encodings are in progress and beyond the scope of this paper.

4.3. Example: Time series data. In this section we present an not obvious application of the methods presented in previous sessions. Namely, we analyze time series data sets using the methods. The key idea is to transform a time series data into a sampled vector field. The approach is inspired by [27].

From now we assume that values of a time series lies in the range $[-1, 1]$. Otherwise we can re-scale the values, e.g. using min-max scaling. Let $T = \{t_i\}_{i=0}^{n-1} \subseteq [-1, 1]$ be a time series. In [27] the authors propose to transform $T$ into an 2D image, where colors of a pixel $(i, j)$ depends on the values $t_i, t_j$. We propose to use similar technique, but to create vectors. Let $K(T)$ be a Delaunay triangulation of a regular grid $\mathbb{R}([0, n] \times [0, n], \{n, n\})$, where $n$ is the length of $T$. Let $\alpha_{i,j} := \arccos(t_i) + \arccos(t_j)$ for each vertex $(i, j)$ of the grid vertex. At each vertex $(i, j)$ we attach a vector $v_{i,j} := (\cos \alpha_{i,j}, \sin \alpha_{i,j})$. In Figure 6 we show a toy example.

As a first validation of the method we use Symbols dataset from the UEA & UCR Time Series Classification Repository [3]. The dataset contains 6 classes of time series of length 398. The dataset is splitted into training set of size 25 and test set of size 995. In Figure 7 we show a few examples from the dataset.
We use the methods presented in the previous sections to obtain encodings of the vector fields generated from the Symbols time series data set. To train the Skip-gram model we use only the training set. In Figure 7 we show the encodings for the test set, where each dot represents a time series and colors represent classes. This visualization suggest that the method can be used with a classifier (like SVM), however this is a work in progress and it is beyond the scope of this paper.

5. Conclusion and future work

We introduced an efficient algorithm to compute hidden features of sampled vector fields. Our method utilizes modern machine learning techniques and provide new possibilities to study dynamical systems. By examples we show that the technique is able to find good characteristics of a given sampled vector field as well as collection of such fields.

We show applications of the method for a synthetic and experimental data sets. Our next step is to use the method to analyze bigger data sets, in particular: the magnetic field at the solar surface, time series data transformed into vector fields, velocity fields of a fluid flow measured by particle image velocimetry. Our main goal is to use the extracted features of vector fields as an input to other machine learning methods, e.g. SVM, decision trees, etc. We believe it will be a complementary description of a data set useful in an automatic analysis pipelines.
Figure 5. 2-dimensional encodings for the prey-predator multivector corpuses. Each row represents a corpus: $C_I, C_{II}, C_{III}, C_{IV}$ (top-down). First column represents encodings of labels: for each distinct label $l$ in a multivector corpus there is a dot at position $\Phi^2(l)$. Second and third columns represent encodings of graphs: for each $V = V_{\alpha,c}^\theta \in \mathcal{V}^\theta$ there is a dot at position $\Phi^2_w(G_V)$, where $w$ is the mean (second column) or TFIDF (third column) weighting. Colors of the dots correspond to the colors of $(\alpha, c)$ in Figure 4.
**Figure 6.** Left: min-max scaled time series $\mathcal{T} = \{(i - 4)^2\}_{i=0}^8$. Middle: $(i, j)$-vectors of $\mathcal{T}$ for $i, j \in \{1, \ldots, 8\}$. Right: a stream plot visualization of the $(i, j)$-vectors. It is worth to note that the time series data is rather sparse but yet the vector field looks smooth and it is possible to create an expressive stream plot.

**Figure 7.** Time series from the *Symbols* data set: one example of a series per class and the stream plots of their vector fields.
Figure 8. Encodings of the vector fields obtained from the Symbols time series data set.
References

[1] P. Alexandroff. Diskrete raume. 2:501–519.
[2] Zin Arai, William Kalies, Hiroshi Kokubu, Konstantin Mischaikow, Hiroe Oka, and Pawel Pilarczyk. A database schema for the analysis of global dynamics of multiparameter systems. 8(3):757–789.
[3] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 31:606–660, 2017.
[4] Jonathan A. Barmak. Algebraic Topology of Finite Topological Spaces and Applications. Lecture Notes in Mathematics. Springer-Verlag.
[5] Bogdan Batko, Konstantin Mischaikow, Marian Mrozek, and Mateusz Przybylski. Conley index approach to sampled dynamics.
[6] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senišcal, Frédéric Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In Dawn E. Holmes and Lakhmi C. Jain, editors, Innovations in Machine Learning: Theory and Applications, Studies in Fuzziness and Soft Computing, pages 137–186. Springer Berlin Heidelberg.
[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. 5:135–146.
[8] Bree Cummins, Tomas Gedeon, Shaun Harker, Konstantin Mischaikow, and Kafung Mok. Combinatorial representation of parameter space for switching networks. 15(4):2176–2212.
[9] T. Dey, M. Juda, T. Kapela, J. Kubica, M. Lipiński, and M. Mrozek. Persistent homology of Morse decompositions in combinatorial dynamics. 18(1):510–530.
[10] Herbert Edelsbrunner, Grzegorz Jablonski, and Marian Mrozek. The persistent homology of a self-map. 15(5):1213–1244.
[11] Ryszard Engelking. General topology. Heldermann Verlag, 1989.
[12] Robin Forman. Combinatorial vector fields and dynamical systems. 228(4):629–681.
[13] Robin Forman. Morse theory for cell complexes. 134(1):90–145.
[14] Tomas Gedeon, Bree Cummins, Shaun Harker, and Konstantin Mischaikow. Identifying robust hysteresis in networks. 14(4):e1006121.
[15] Joydev Ghosh, Banshidhar Sahoo, and Swarup Poria. Prey-predator dynamics with prey refuge providing additional food to predator. 96:110–119.
[16] Zellig S. Harris. Distributional structure. 10(2):146–162.
[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space.
[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc.
[19] K. Mischaikow, M. Mrozek, J. Reiss, and A. Szymczak. Construction of symbolic dynamics from experimental time series. 82(6):1144–1147.
[20] Marian Mrozek. Conley–morse–forman theory for combinatorial multivariate fields on lefschetz complexes. 17(6):1585–1633.
[21] James R. Munkres. Elements Of Algebraic Topology. CRC Press.
[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’14, pages 701–710. ACM Press.
[23] Anand Rajaraman and Jeffrey David Ullman. Mining of Massive Datasets. Cambridge University Press.
[24] P. H. Scherrer, J. Schou, R. I. Bush, A. G. Kosovichev, R. S. Bogart, J. T. Hoeksema, Y. Liu, T. L. Duvall, J. Zhao, A. M. Title, C. J. Schrijver, T. D. Tarbell, and S. Tomczyk. The helioseismic and magnetic imager (HMI) investigation for the solar dynamics observatory (SDO). 275(1):207–227.
[25] A. Szymczak and E. Zhang. Robust morse decompositions of piecewise constant vector fields. 18(6):938–951.
[26] Floris Takens. Detecting strange attractors in turbulence. In David Rand and Lai-Sang Young, editors, Dynamical Systems and Turbulence, Warwick 1980, Lecture Notes in Mathematics, pages 366–381. Springer.
[27] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In Twenty-Fourth International Joint Conference on Artificial Intelligence.

Mateusz Juda, Division of Computational Mathematics, Institute of Computer Science and Computational Mathematics, Faculty of Mathematics and Computer Science, Jagiellonian University, ul. St. Łojasiewicza 6, 30-348 Kraków, Poland.
E-mail address: mateusz.juda@ii.uj.edu.pl