Algorithms for context prediction in Ubiquitous Systems

Prediction by self organising maps

Stephan Sigg

Institute of Distributed and Ubiquitous Systems
Technische Universität Braunschweig

January 20, 2009

Overview and Structure

- Introduction to context aware computing
- Basics of probability theory
- Algorithms
  - Simple prediction approaches: ONISI and IPAM
  - Markov prediction approaches
  - The State predictor
  - Alignment prediction
  - Prediction with self organising maps
  - Stochastic prediction approaches: ARMA and Kalman filter
  - Alternative prediction approaches
    - Dempster shafer
    - Evolutionary algorithms
    - Neural networks
    - Simulated annealing

Overview and Structure

- Introduction to context aware computing
- Basics of probability theory
- **Algorithms**
  - Simple prediction approaches: ONISI and IPAM
  - Markov prediction approaches
  - The State predictor
  - Alignment prediction
  - **Prediction with self organising maps**
  - Stochastic prediction approaches: ARMA and Kalman filter
  - Alternative prediction approaches
    - Dempster shafer
    - Evolutionary algorithms
    - Neural networks
    - Simulated annealing

Outline

SOM prediction approaches

Introduction to self organising maps

Historical remarks

- Self organising map (SOM) algorithm proposed by Teuvo Kohonen[1]
- Presented it as a model of the self-organisation of neural connections.
- Map high dimensional input data to low dimensional representation

[1] Teuvo Kohonen, *Self-Organizing Maps*, Springer, 2001.

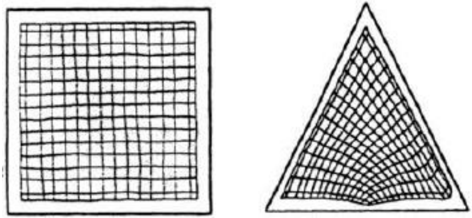
Introduction to self organising maps
Historical remarks

- Idea:
  - Given a sequence of points in a sample space,
  - Create a mapping of these pionts into a target space that respects the neighbourhood relation in the sample space
- Mapping is learned by a simple two-layer neural network

Introduction to self organising maps
Applications utilising self organising maps

- Self organising maps are commonly utilised for
  - Industrial instrumentation (Monitoring and control)
  - Medical applications (diagnostic methods, prostheses, modeling and profiling of patients)
  - Telecommmunications (allocation of resources to networks, adaptive demodulation and transmission channel equalisation)

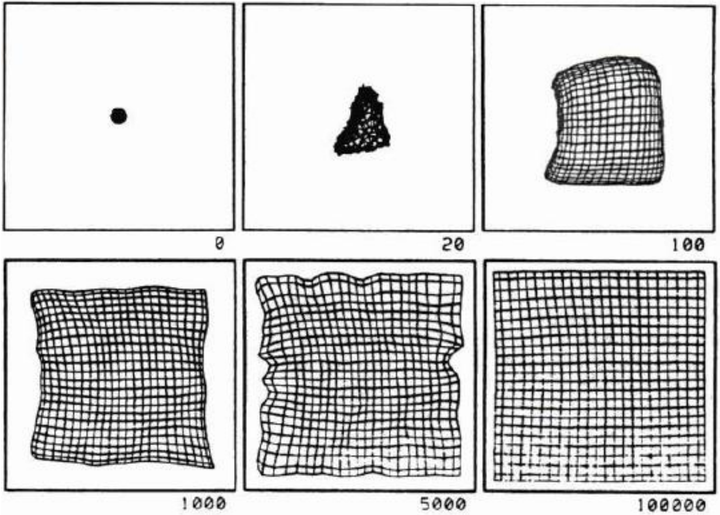
Introduction to self organising maps
Definition



- A self organising map is a topology preserving lattice of a predefined number of nodes that represents a topology of elements in the input space.
- Algorithm inherits self-organisation property
  - Able to produce organisation starting from possibly total disorder.
  - SOM algorithm defines and preserves neighbourhood structure between all nodes of the map.

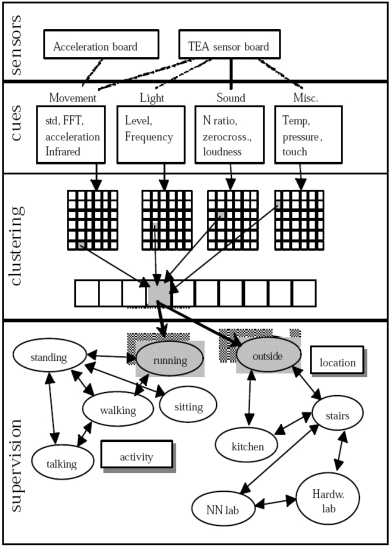
Introduction to self organising maps

Self organisation



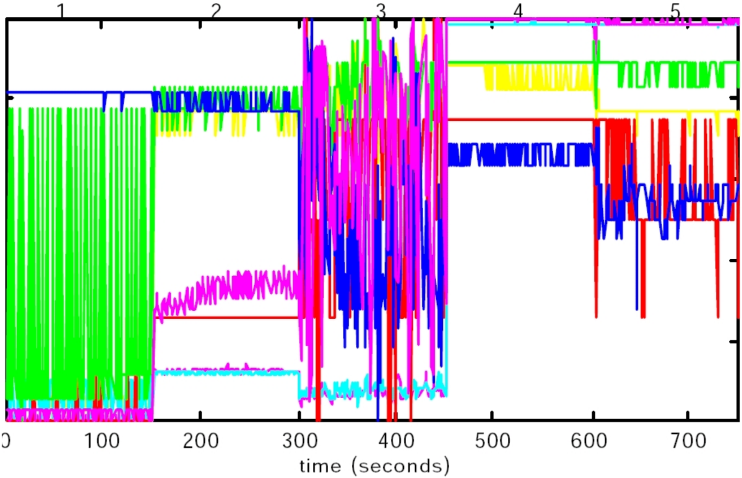
Introduction to self organising maps

Example Application – Tea



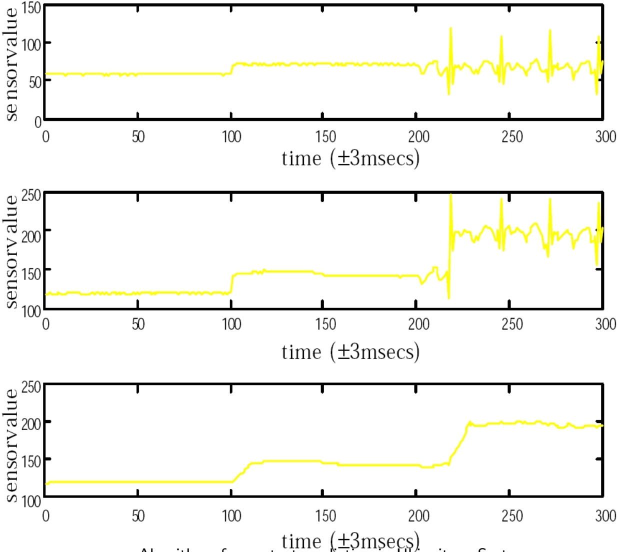
Introduction to self organising maps

Example Application – Tea



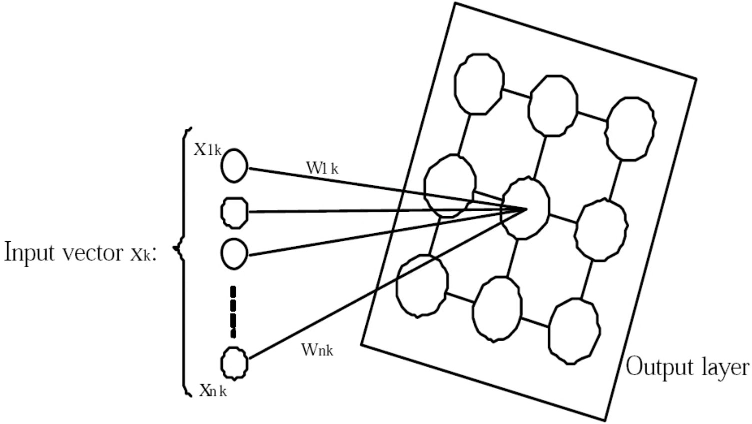
Introduction to self organising maps

Example Application – Tea

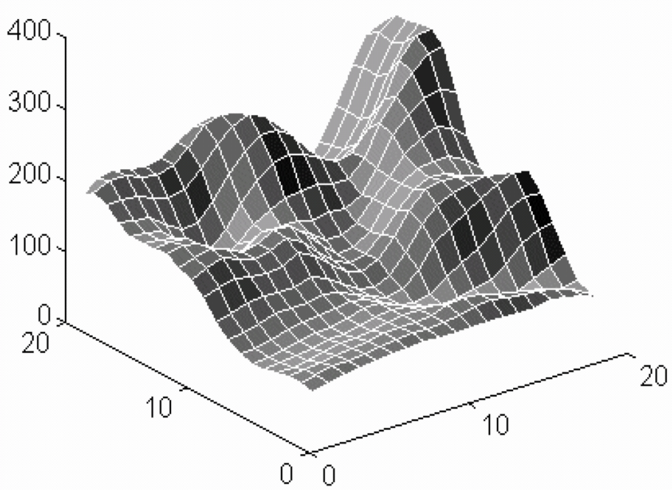


Introduction to self organising maps

Example Application – Tea

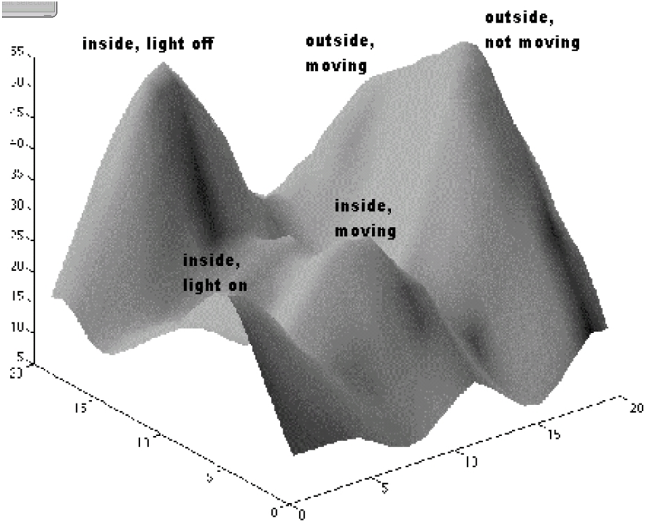


Introduction to self organising maps
Example Application – Tea



Introduction to self organising maps

Example Application – Tea



Introduction to self organising maps

Definition

- A set of elements from the input space is grouped into subsets
- Each subset represented by one node of the lattice
- Consequently, lattice defines a neighbourhood between these subsets.
- Representative or prototype can be defined for each subset

Introduction to self organising maps
SOM definition

- We recapitulate a condensed definition of the SOM algorithm that can be found in[2]

[2] M. Cottrell, J.C. Fort and G. Pages, *Theoretical aspects of the SOM algorithm*, Neurocomputing, pp. 119-138, vol 21, 1998.

Introduction to self organising maps

SOM definition

Self organising maps

- Let $I = \{\overrightarrow{\eta_1}, \dots, \overrightarrow{\eta_{|S|}}\}$ be a set of *km*-dimensional vectors that are associated with nodes in a lattice.
- Neighbourhood structure provided by symmetrical, non-increasing neighbourhood function $d : I \times I \rightarrow \mathbb{R}$ which depends on the distance between two nodes $\overrightarrow{\eta_i}$ and $\overrightarrow{\eta_j} \in I$.
- The state of the map at time t is given by

$$\eta(t) = \left(\overrightarrow{\eta_1(t)}, \overrightarrow{\eta_2(t)}, \dots, \overrightarrow{\eta_{|S|}(t)}\right), \tag{1}$$

Introduction to self organising maps

SOM definition

Self organising map algorithm

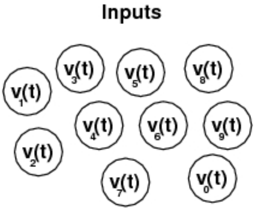
The SOM algorithm is recursively defined by

$$
\begin{aligned}
i_c\left(\overrightarrow{v(t+1)}, \overrightarrow{\eta(t)}\right) &= argmin\left\{\left\|\overrightarrow{v(t+1)} - \overrightarrow{\eta_i(t)}\right\|, \overrightarrow{\eta_i(t)} \in \eta(t)\right\}, \\
\overrightarrow{\eta_i(t+1)} &= \overrightarrow{\eta_i(t)} - \varepsilon_t d\left[i_c\left(\overrightarrow{v(t+1)}, \overrightarrow{\eta(t)}\right), \overrightarrow{\eta_i}\right] \\
&\quad \cdot \left(\overrightarrow{\eta_i(t)} - \overrightarrow{v(t+1)}\right), \forall \overrightarrow{\eta_i} \in I.
\end{aligned}
$$

- In this formula, $i_c\left(\overrightarrow{v(t+1)}, \overrightarrow{\eta(t)}\right)$ corresponds to the node in the network that is closest to the input vector.
- The parameter ε_t controls the adaptability of the self organising map.

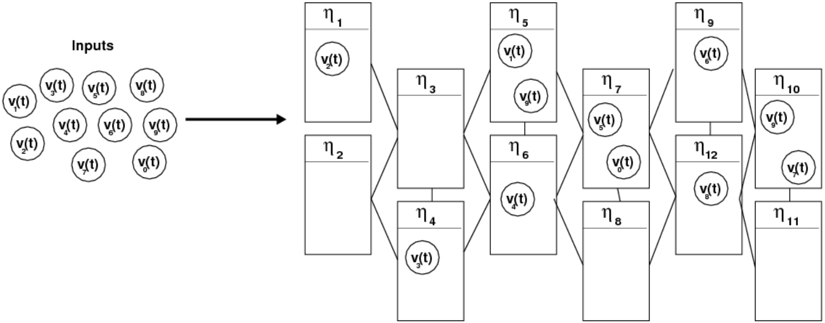
Introduction to self organising maps

Operational principle



- Input values $v_i(t)$ are to be mapped onto the target space

Introduction to self organising maps
Operational principle

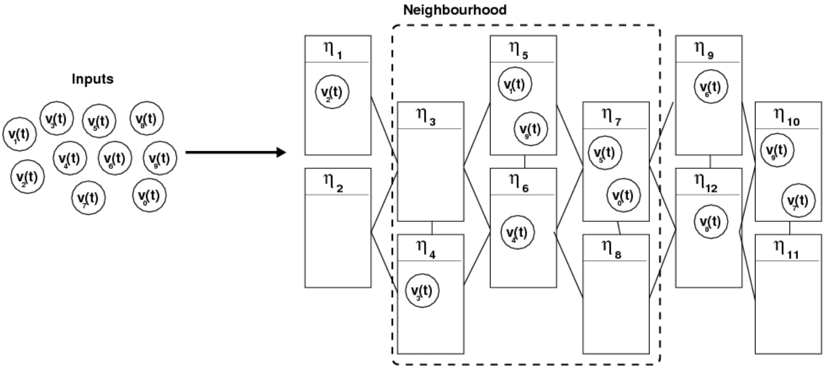


- The node with the lowest distance is associated with the input value:

$$i_c\left(\overrightarrow{v(t+1)}, \overrightarrow{\eta(t)}\right) = argmin\left\{\left\|\overrightarrow{v(t+1)} - \overrightarrow{\eta_i(t)}\right\|, \overrightarrow{\eta_i(t)} \in \eta(t)\right\}$$

Introduction to self organising maps

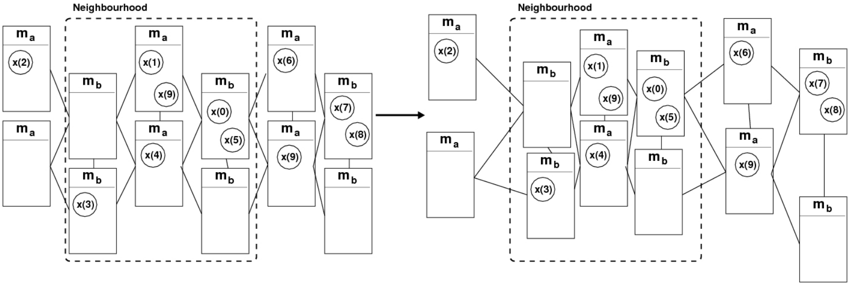
Operational principle



- Nodes in the neighbourhood of the associated node are moved closer to the input value

Introduction to self organising maps

Operational principle

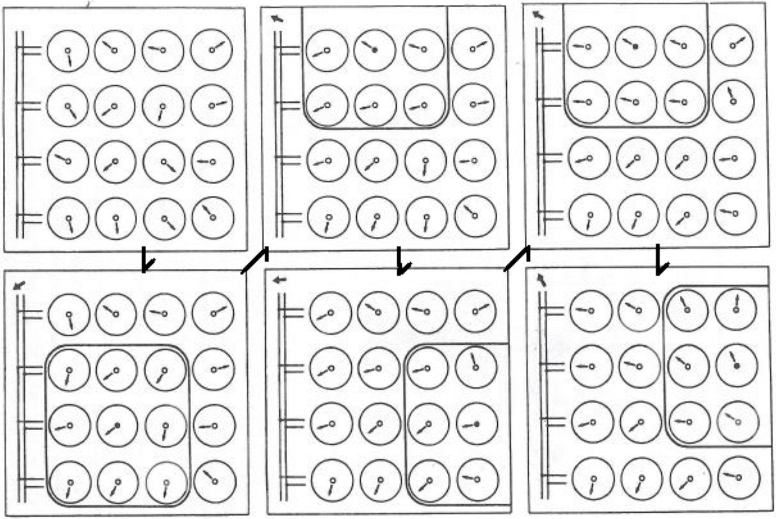


- Nodes in the neighbourhood of the associated node are moved to the input value

$$\overrightarrow{\eta_i(t+1)} = \overrightarrow{\eta_i(t)} - \varepsilon_t d\left[i_c\left(\overrightarrow{v(t+1)}, \overrightarrow{\eta(t)}\right), \overrightarrow{\eta_i}\right] \cdot \left(\overrightarrow{\eta_i(t)} - \overrightarrow{v(t+1)}\right), \forall \overrightarrow{\eta_i} \in I.$$

Introduction to self organising maps

Operational principle



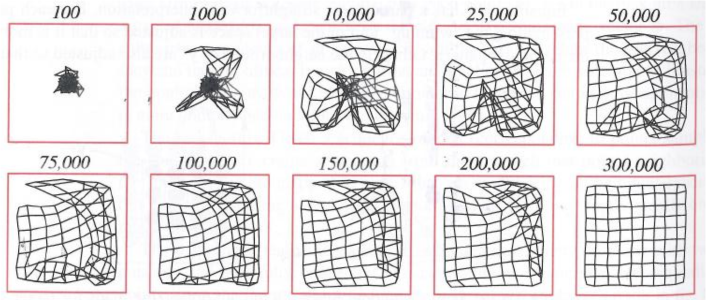
Introduction to self organising maps
Remarks

- It has been proved that the SOM algorithm always converges[3]
- Normalisation of the input vectors might improve numerical accuracy.

[3] Y. Cheng, *Neural Computation*, 9(8), 1997.

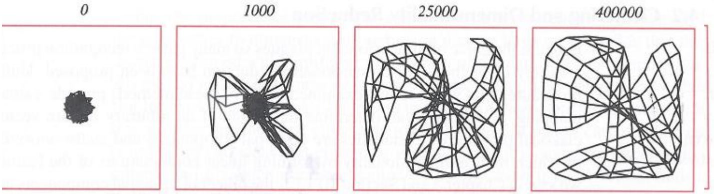
Introduction to self organising maps

Problems of SOMs



- The map created as target space might have several orientations
- It is possible that one part of the map is created following one orientation, while other parts are created following other orientations.

Introduction to self organising maps
Problems of SOMs



- The map created as target space might have several orientations
- It is possible that one part of the map is created following one orientation, while other parts are created following other orientations.

Introduction to self organising maps

Problems of SOMs

- If the neighbourhood is chosen to be too small, the map will not be ordered globally.

Outline

SOM prediction approaches

1. Introduction to self organising maps
2. Software tools for self organising maps
3. Applications of self organising maps
4. Prediction with self organising maps
5. Properties of the SOM prediction approach

Software tools for self organising maps

The SOM_PAK

SOM_PAK First public-domain software package (1990)[4]

- Released by Laboratory of Compuer and Information Science of Helsinki University of Technology
- Source code (ANSI C) and documentation completely available
- Available for UNIX or MS DOS
- Features:
  - Standard incremental-learning SOM
  - (simple) graphics programs included
  - Map size and vector dimension not restricted
  - Several neighbourhood functions available
  - Able to handle largest scale problems

[4] http://www.cis.hut.fi/research/software.shtml

Software tools for self organising maps
The SOM Toolbox

SOM Toolbox: Toolbox for MatLab (1996)[5]

- Released by Laboratory of Compuer and Information Science of Helsinki University of Technology
- MatLab version 5 or higher required
- Slower than SOM_PAK
- Features:
  - Standard incremental-learning SOM and Batch Map SOM
  - Map size and vector dimension not restricted
  - Neighbourhood and training sequences identical to SOM_PAK
  - Improved visualisation and analysis capabilities

[5] http://www.cis.hut.fi/software.shtml

Software tools for self organising maps
The Neural Networks Tool

Nenet User friendly ANN Toolbox (1997)[6]

- Released by the Neural network team of Helsinki University of Technology
- 32-bit Windows 95/NT recommended
- Suited for small scale problems only
- Features:
  - Standard incremental SOM
  - Several visualisation options:
    - Component planes with trajectories
    - U-matrix
    - 3D hit histograms
    - Display of active neuron coordinates
  - Easy to use; Good graphics programs

[6] http://www.mbnet.fi/∼phodju/nenet/Nenet/Gneral.html

Software tools for self organising maps
Viscovery SOMine

Viscovery SOMine Commercial SOM software package [7]

- Released by Eudaptics GmbH in Austria
- Windows NT and Windows NT 4.0
- Features:
    - User-friendly, flexible and powerful package
    - Interfaces for GUI, OLE, SQL and DB2
    - Batch Map algorithm
    - High computing speed
    - Unilimited map size and vector dimension
    - Several visualisation options:
        - Component planes with trajectories
        - U-matrix
        - Cluster windows
        - Iso-contours of hit density

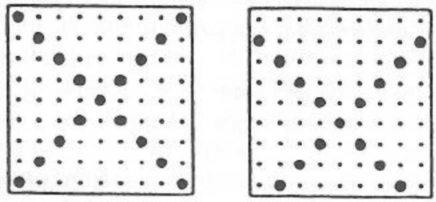
[7] http://www.eudaptics.co.at

Outline

SOM prediction approaches

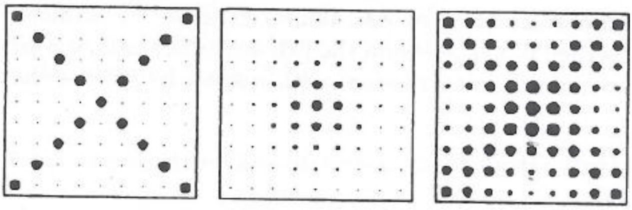
1. Introduction to self organising maps
2. Software tools for self organising maps
3. Applications of self organising maps
4. Prediction with self organising maps
5. Properties of the SOM prediction approach

Applications of self organising maps
Preprocessing of optic patterns



- Two orthogonal line figures

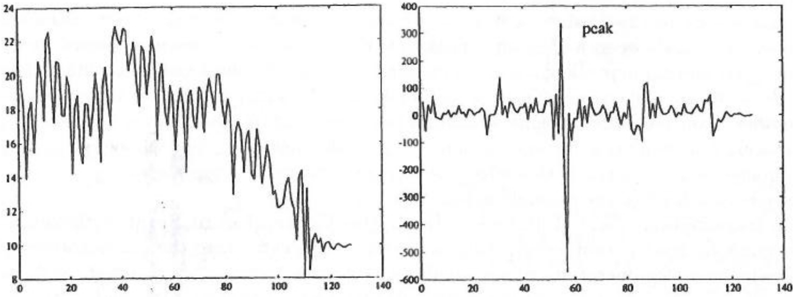
Applications of self organising maps
Preprocessing of optic patterns



- For line figures or photographs with very high contrast at their contours, one of the simplest preprocessing methods is blurring or linear convolution of the original pattern with some point spread function

Applications of self organising maps

Converting cepstra into quasiphonemes



- Each feature vector classified in one of the phonemic classes (letters)
- Speech recognition

Applications of self organising maps

Artificially generated clauses

Bob/Jim/Mary	1
horse/dog/cat	2
beer/water	3
meat/bread	4
runs/walks	5
works/speaks	6
visits/phones	7
buys/sells	8
likes/hates	9
drinks	10
eats	11
much/little	12
fast/slowly	13
often/seldom	14
well/poorly	15

Sentence Patterns:

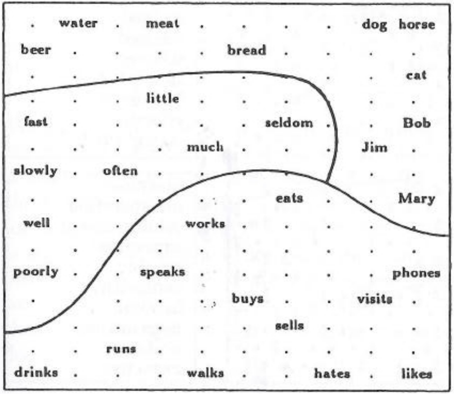
1-5-12	1-9-2	2-5-14
1-5-13	1-9-3	2-9-1
1-5-14	1-9-4	2-9-2
1-6-12	1-10-3	2-9-3
1-6-13	1-11-4	2-9-4
1-6-14	1-10-12	2-10-3
1-6-15	1-10-13	2-10-12
1-7-14	1-10-14	2-10-13
1-8-12	1-11-12	2-10-14
1-8-2	1-11-13	2-11-4
1-8-3	1-11-14	2-11-12
1-8-4	2-5-12	2-11-13
1-9-1	2-5-13	2-11-14

Mary likes meat
Jim speaks well
Mary likes Jim
Jim eats often
Mary buys meat
dog drinks fast
horse hates meat
Jim eats seldom
Bob buys meat
cat walks slowly
Jim eats bread
cat hates Jim
Bob sells beer
(etc.)

- Words are defined by artificially constructed vocabulary and classified to categories
- Words of the same category are freely interchangeable

Applications of self organising maps

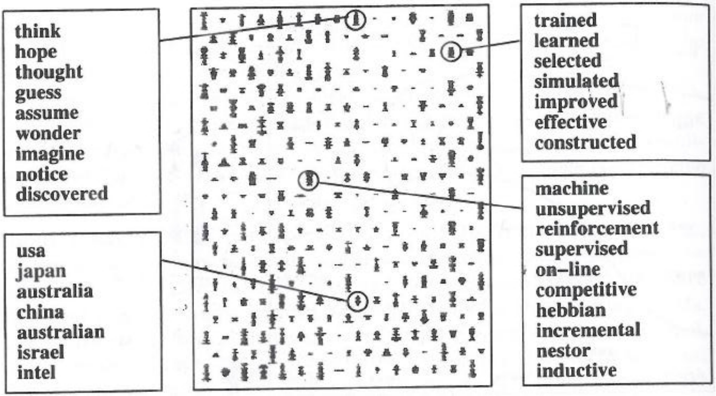
Artificially generated clauses



- Semantic map obtained on a network of 10×15 cells
- After 2000 presentations of word-context-pairs
- Derived from 10000 random sentences

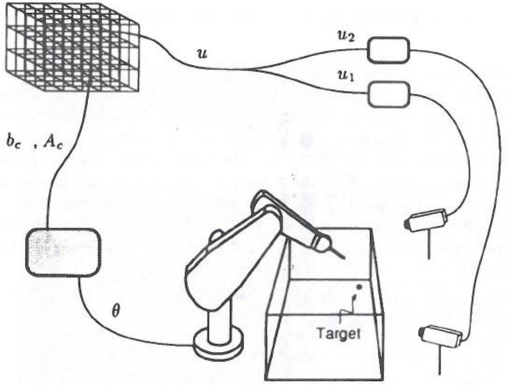
Applications of self organising maps

Processing of natural text



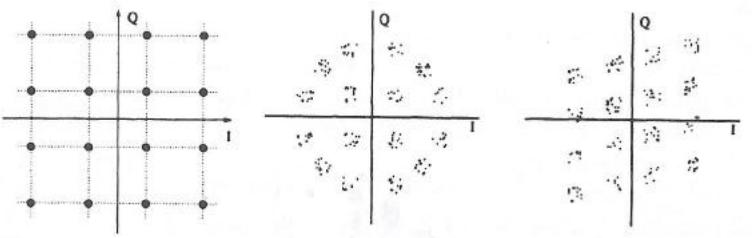
- Input of about 5000 articles during 1995 and 1996
- Non-textual information removed beforehand

Applications of self organising maps
Robot-Arm control



- Input pattern translates to a 'winning' vector in SOM
- Vector represents robotic control information

Applications of self organising maps
Telecommunications



- Detect distortions in practical systems using QAM modulation

Outline

SOM prediction approaches

1. Introduction to self organising maps
2. Software tools for self organising maps
3. Applications of self organising maps
4. Prediction with self organising maps
5. Properties of the SOM prediction approach

Prediction with self organising maps

SOM prediction approaches

- Process of predicting with SOMs divided into two stages.
  - Preparation and clustering phase
    - Model utilised for prediction is created
  - Prediction phase
    - Actual prediction is stated
- Recent work on the utilisation of SOM algorithms for prediction tasks can be found in [8] [9] [10]

[8] Cottrell, M., de Bodt, W., Gregoire, P.: *Simulating interest rate structure evolution on a long term horizon: A kohonen map application*. In: Proceedings of Neural Networks in the Capital Markets. (1996)

[9] Walter, J., Ritter, H., Schulten, K.: *Non-linear prediction with self-organising maps.* In: Proceedings of IJCNN. (1990)

[10] Vesanto, J.:*Using the som and local models in time series prediction.* In: Workshop on Self-Organising Maps (WSOM97). (1997)

frame

Model creation phase (1/3) :

- For any $q, i, j \in \mathbb{N}$, the context history $T_{i,j}$ is divided into the set of all time series of length q.
- For $r \in [0, \ldots, q]$ the set $T_{i+r,j-q+r}$ of time series of length $j - i + q$ is created for $T_{i,j}$.
- In addition, for $t \in [i + r, \ldots, j - q + r]$, so-called deformations $D_{t,t+r}$ are created with

$$D_{t,t+r} = T_{t+1,t+j-i+q+1} - T_{t,t+j-i+q}.$$

Prediction with self organising maps

SOM prediction approaches

Model creation phase (2/3) :

- Deformations are time series that describe modifications necessary to evolve from one time series to the time series occurring one time instant later.
- These sets of time series $T_{i+r,j-q+r}$ and $D_{t,t+r}$ are clustered with help of the vector quantisation by SOM.
- Outcome of this procedure are for $i, j \in \mathbb{N}$ prototypes $\overline{T_i}$ and $\overline{D_i}$ of time series that represent a set of similar time series and deformations

Prediction with self organising maps
SOM prediction approaches

Model creation phase (3/3) :

- Create Matrix M that describes relation between time series $T_{t,t+r}$ and $D_{t,t+r}$
- For a fixed i and $j \in [1, \ldots, \kappa]$ with κ describing the number of different prototypes $\overline{D_i}$, the row M_{ij} represents the conditional probability that $D_{t,t+r}$ belongs to $\overline{D_j}$ given that $T_{t,t+r}$ belongs to $\overline{T_i}$.

Prediction with self organising maps

SOM prediction approaches

When these preparations are made, the prediction consists of seven steps.

Prediction phase (1/3) :

- For any time t, consider a time series $T_{t-q,t}$.
- Find the associated prototype.
- Randomly choose a deformation $\overline{D_j}$ according to the probability distribution given by M.

Prediction with self organising maps

SOM prediction approaches

When these preparations are made, the prediction consists of seven steps.

Prediction phase (2/3) :

- Obtain the prediction for time $t+1$ as $T_{t-q+1,t+1} = T_{t-q,t} + \overline{D_j}$.
- Iterate these steps with the obtained predicted time series until the prediction horizon is of the desired length.

Prediction with self organising maps

SOM prediction approaches

When these preparations are made, the prediction consists of seven steps.

Prediction phase (3/3) :

- A Monte-Carlo procedure is used to repeat this process several times.
- From all obtained predictions, extract global trends of the time series as the evolution of the time series mean, its variance or confidence intervals.

Prediction with self organising maps

SOM prediction approaches

- In [11], a procedure is presented with which whole future time series can be predicted in one step instead of iterating the one-step-ahead prediction several times.
  - Utilise vectors of consecutive observations in the to be clustered time series, instead of single observations repeatedly.
  - Consequently, a vector of consecutive observations is then predicted instead of a single time series element.

[11] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort and M. Verleysen, *Time series forecasting: Obtaining long term trends with self-organising maps*, Pattern recognition letters, pp. 1795-1808, vol. 26, number 12, 2005.

Outline

Prediction with SOM approaches

Properties of the SOM prediction approach

Processing load (1/2)

- Runtime to state one prediction can be estimated as $O(|S'|)$ since for a given input vector all $|S'|$ vectors in the lattice are compared for their distance to the input vector.
- When input vectors are ordered in a tree structure, runtime can be reduced to $O(log(|S'|))$.

Properties of the SOM prediction approach

Processing load 2/2

- Model creation phase not considered in this calculation.
- Changing environments: Model creation phase to be applied in regular intervals.
- Requires additional time of $O(|S'|^2)$ (maximum number of time series considered for the model creation phase)
- We estimate $|S'|$ of input time series by $\varsigma \cdot k$ for a suitable constant $\varsigma \in \mathbb{N}$: Overall runtime: $O(k^2)$

Properties of the SOM prediction approach
Memory requirements

- Memory requirements
  - Proportional to the number and dimension of prototype vectors utilised

Properties of the SOM prediction approach

Prediction horizon (1/2)

- Prediction horizon: 1
- Extendable by iterative prediction approach

Properties of the SOM prediction approach

Prediction horizon (2/2)

- When whole time series are predicted instead of single time series elements, size of context history is to be increased dramatically.
- For prototype vectors of length q and a desired prediction horizon of length n the context history is required to have length $n \cdot q$ at minimum.
  - In this minimum case only one prototype vector would be created
  - In order to have r prototype vectors, a context history length of $(n + r) \cdot q$ is required at least.
- Typical context patterns are seldom of extensive length in representative applications
- One step prediction approach is therefore seldom applicable to context prediction tasks

Properties of the SOM prediction approach
Adaptability

- Adaptability of the SOM-prediction-approach is low and restricted to the level of adaptability the application designer permits.
    - Number of different typical context patterns, i.e. prototype vectors restricted by fixed size of nodes in the lattice

Properties of the SOM prediction approach
Multi-dimensional time series

- Multidimensional time series are not yet considered for this approach
- Extension to multidimensional case possible
- Would furhter inflate the model.

Properties of the SOM prediction approach

Iterative prediction

- Iterative Prediction required to increase prediction horizon above 1

Properties of the SOM prediction approach

Prediction of context durations

- Possible when context duration implicitly stated by sample frequency

Properties of the SOM prediction approach

Approximate matching of patterns

- Approximate matching
  - Implicitly supported by neighbourhood function

Properties of the SOM prediction approach

Context data types

- Implicit restriction to numerical context patterns.
  - Clustering by the SOM is also applicable to contexts of nominal context data types
  - Creation of the Matrix M and the creation of the deformation vectors requires distance metric between all context elements that provides a real valued output

Properties of the SOM prediction approach

Pre-processing

- Model creation phase required as pre-processing
- Changing environments: Model creation phase to be applied in regular intervals.
- Requires additional time of $O(|S'|^2)$ (maximum number of time series considered for the model creation phase)

Aspects of prediction algorithms

Summary

	IPAM	ONISI	Markov	CRF
Numeric Contexts	yes	no	no	no
Non-numeric Contexts	yes	yes	yes	yes
Complexity	$O(k)$	$O(k^2)$	$O(C^2)$	$O(C^2)$
Learning ability	(no)	yes	yes	yes
Approximate matching	no	no	no	no
Multi-dim. TS	(no)	(no)	(no)	(no)
Discrete data	yes	yes	yes	yes
Variable length patterns	no	yes	no	(yes)
Multi-type TS	yes	no	(no)	(no)
Continuous data	no	no	no	no
Pre-processing	$O(k)$	–	$O(k)$	$O(k)$
Context durations	no	no	no	no
Continuous time	no	no	yes	yes

Aspects of prediction algorithms

Summary

	SPM	Align	SOM	PCA
Numeric Contexts	yes	yes	yes	
Non-numeric Contexts	yes	yes	yes	
Complexity	$O(1)$	$O(l \cdot k^2)$	$O(\log(\|k\|) + k^2)$	
Learning ability	(yes)	yes	yes	
Approximate matching	no	yes	yes	
Multi-dim. TS	(no)	yes	yes	
Discrete data	yes	yes	yes	
Variable length patterns	yes	yes	yes	
Multi-type TS	no	yes	yes	
Continuous data	no	no	no	
Pre-processing	$O(k)$	$O(k^2)$	yes	
Context durations	no	yes	yes	
Continuous time	no	no	no	

Properties of the SOM prediction approach
Conclusion

- Benefits
  - Implicitly groups similar time series together and represents them by a prototype
  - Automatic adaptation
  - Separate pre-processing to detect typical time series patterns not required
  - Implicit utilisation of several statistical measures
- Drawbacks
  - Method utilises predicted, possibly error prone contexts for the prediction of horizons that exceed 1
  - Prediction accuracy is expected to decrease quickly with increasing prediction horizon.

Overview and Structure

- Introduction to context aware computing
- Basics of probability theory
- **Algorithms**
  - Simple prediction approaches: ONISI and IPAM
  - Markov prediction approaches
  - The State predictor
  - Alignment prediction
  - Prediction with self organising maps
  - **Stochastic prediction approaches: ARMA and Kalman filter**
  - Alternative prediction approaches
    - Evolutionary algorithms
    - Dempster shafer
    - Simulated annealing