Generalized Relevance LVQ for Time Series

Marc Strickert, Thorsten Bojer, Barbara Hammer

Research Group: Learning with Neural Methods on Structured Data

University of Osnabrück, Dept. of Mathematics/Computer Science

marc@informatik.uni-osnabrueck.de

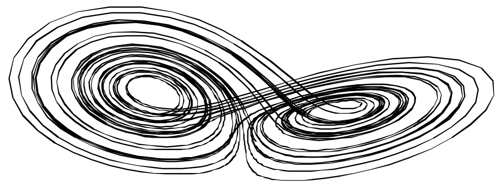
Outline of the Talk

1. Motivation

2. From LVQ to Generalized Relevance LVQ

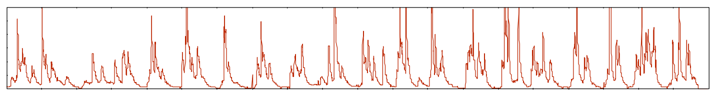
3. Temporal Analysis

   A Toy Example: The Lorenz System
   - Relevance of the Past
   - Embedding



4. Application

   Real Life Data: Runoff Series
   - Prediction



5. Conclusions

Motivation

Aim :

Analysis and Modeling of time series data.

Question :

How can we determine the relevance of data points in the past for explaining the present state?

Answer :

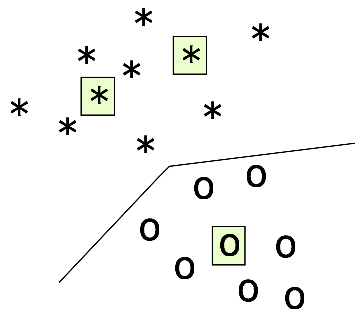
We treat time series data from a clustering perspective,

applying an extension of LVQ with component weighting

to mapping time windows on the past to the present state.

LVQ - GLVQ

Clustering :



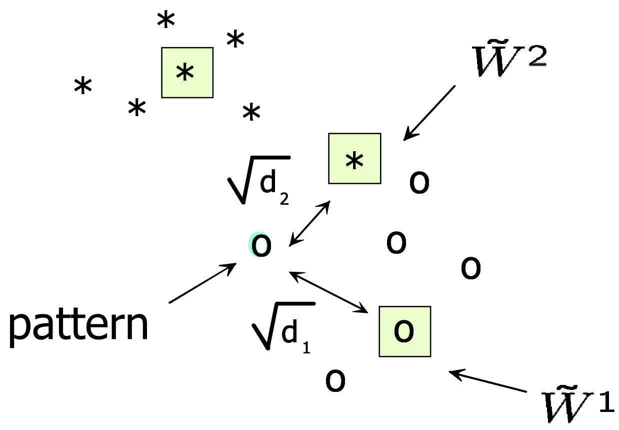
- $*$ - class 1 data
- o - class 2 data
- [*] - class 1 prototypes
- [o] - class 2 prototype(s)

Error function : (GLVQ)

$$E = \sum_x f((d_1 - d_2)/(d_1 + d_2))$$

minimized by gradient descent

Exemplary pattern presentation :



- $*$ - class 1 data
- o - class 2 data
- [*] - class 1 prototypes
- [o] - class 2 prototype(s)

GRLVQ – Weight Adaptation

Metric: *Weighted* squared Euclidean distance $d_i = d^2(x, \tilde{W}^i, \boldsymbol{\lambda})$

$$d^2(x, y, \boldsymbol{\lambda}) := \sum_{i=1}^{n} \lambda_i (x_i - y_i)^2 , \qquad \lambda_i \geq 0$$

λ - Weight Adaptation (Hebbian):

$$\lambda_l := \lambda_l - \epsilon_1 f' \cdot \left(\frac{d_2}{(d_1 + d_2)^2} (x_l^i - \tilde{W}_l^1)^2 - \frac{d_1}{(d_1 + d_2)^2} (x_l^i - \tilde{W}_l^2)^2 \right)$$

=> During the training λ becomes the *relevance vector* describing the contribution of the past.

GRLVQ - Algorithm

initialize *prototypes* $(W^j, c^j) \in \mathbb{R}^n \times \{1, \ldots, C\}$,

initialize $\lambda_1 = \ldots = \lambda_n = 1/n$.

repeat *'training'*:

choose training example (x^i, y^i),

compute $\tilde{W}^1$ *and* $\tilde{W}^2$ *using metric* $d(W^j, x^i, \boldsymbol{\lambda})$, $\forall j : c^j = y^i$,

adapt both prototypes,

adapt each $\lambda_l \neq 0$,

set $\lambda_k := 0$ *if* $0 < \lambda_k < EPS$,

normalize $\boldsymbol{\lambda}$

until classification error acceptable.

repeat *'pruning'*:

apply pruning scheme,

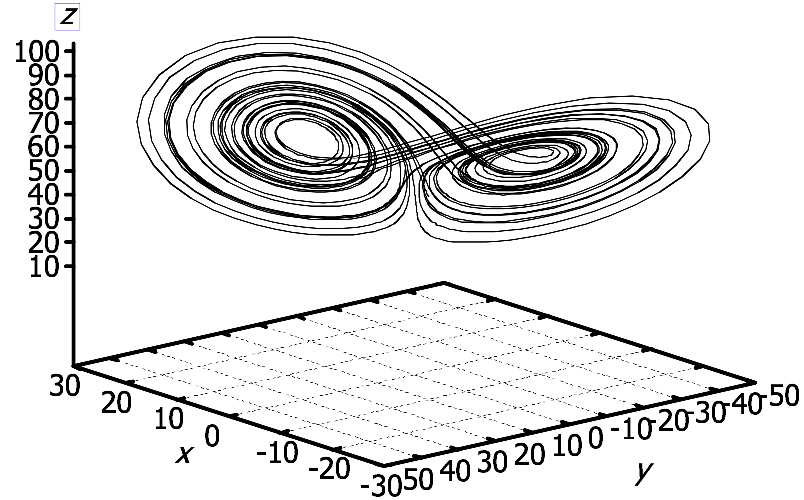
redo training

until classification error degrades significantly.

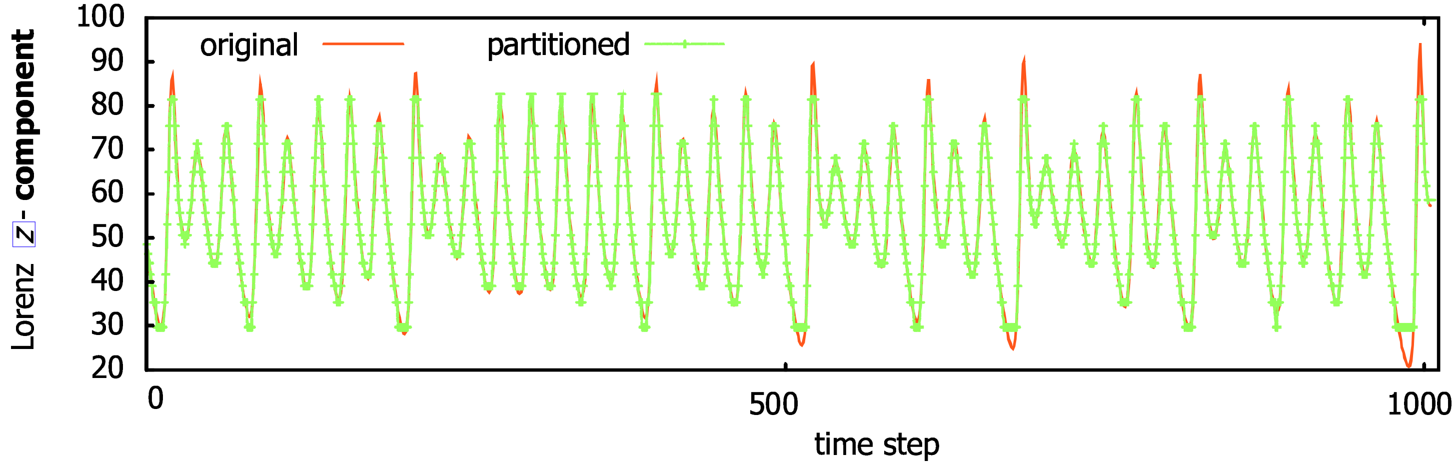
Toy Example: Lorenz System

Lorenz attractor in the regime of deterministic chaos:

Phase Space Portrait



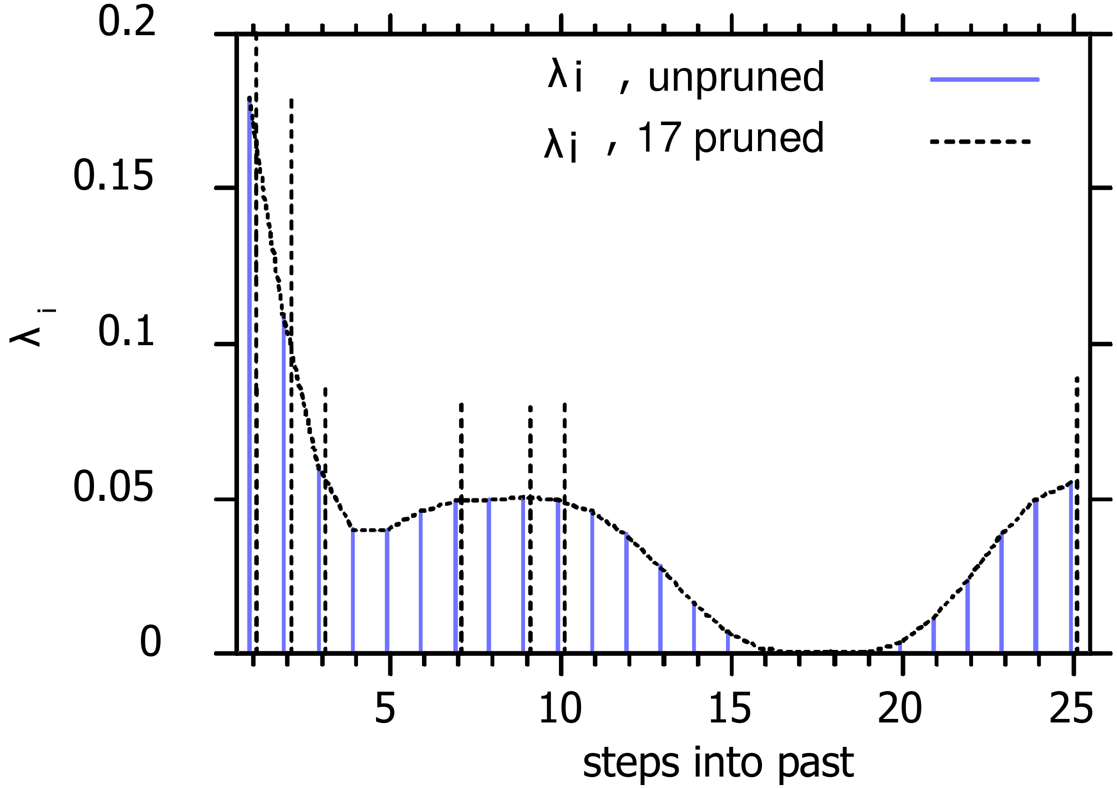
z – component partial time series plot and its *k-median partitioning* into k=17 classes:



Toy Example: Lorenz System - Relevance

Relevance profile λ :

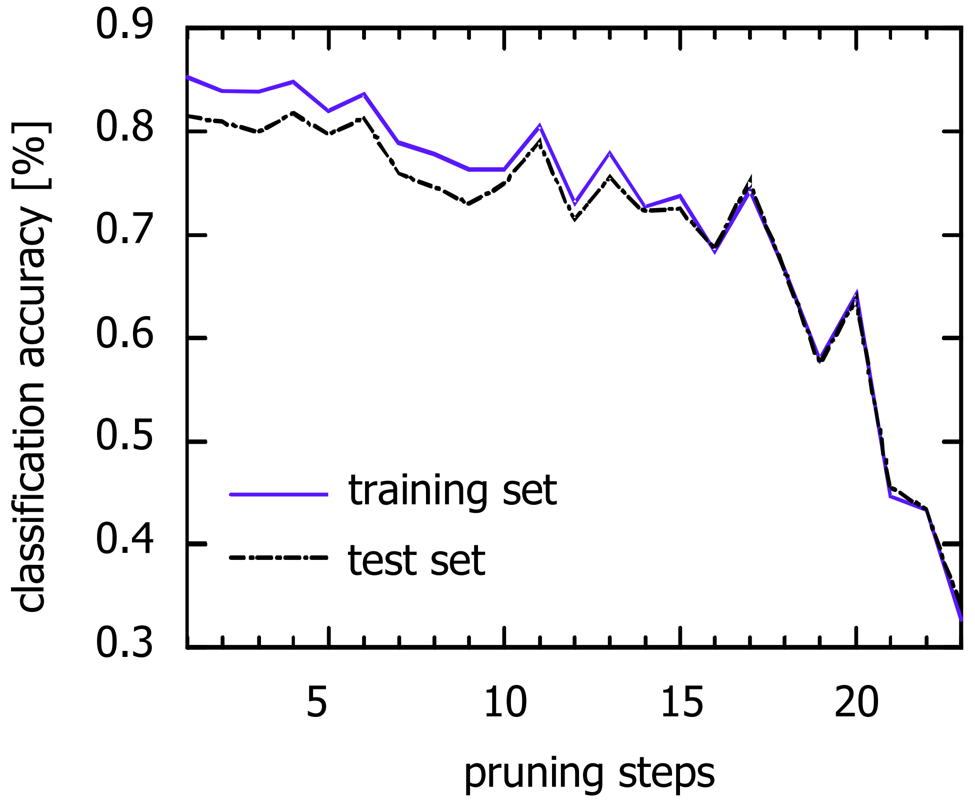
Training was performed using temporal windows of 25 steps to learn the partition class of the presence.



Toy Example: Lorenz System - Accuracy

Degrading classification accuracy

with successive pruning of the minor weights λ_i:

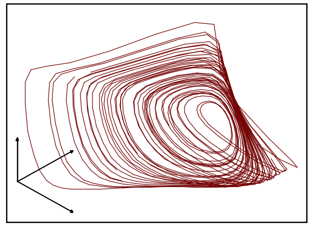


Embedding

Exemplary case : Attractor Reconstruction

Here Using : Width of temporal window = 5, embedding dimension = 3.

1. Takens' Delay Embedding $\Phi_t = (\phi_0 , \phi_1 , \phi_2)$

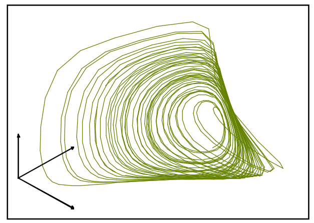


lag : 2

Time Series → $\cdots$ x_{t-5} x_{t-4} x_{t-3} x_{t-2} x_{t-1} x_t x_{t+1} $\cdots$

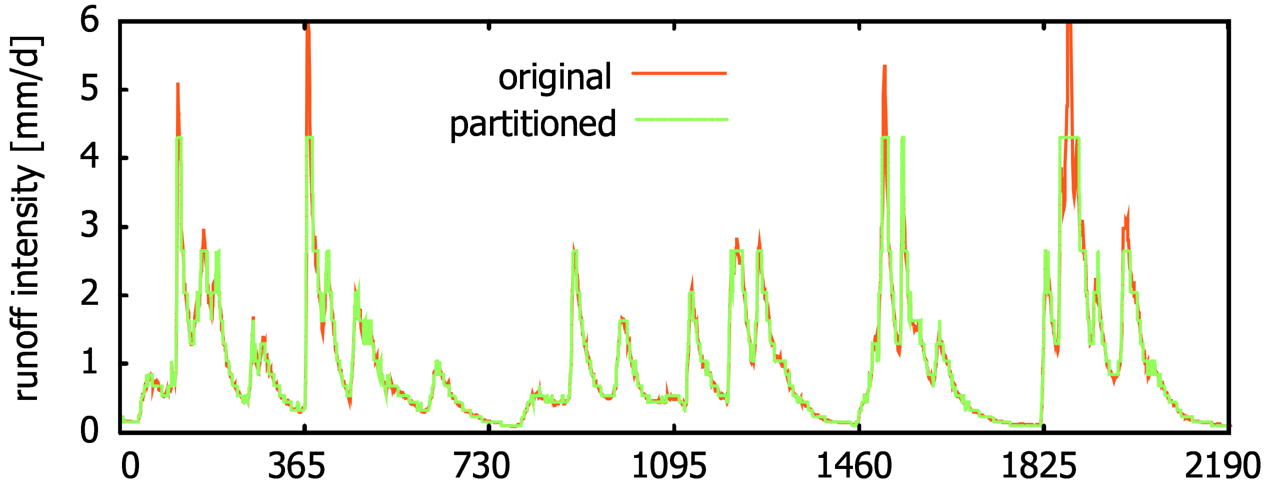
relevance vector : $\Lambda = (\lambda_4 , \lambda_3 , \lambda_2 = 0 , \lambda_1 = 0 , \lambda_0)$

2. Relevance Embedding $\Phi_t = (\phi_0 , \phi_1 , \phi_2)$



Real Life Data: Lysimeter Runoff Series

Partial
time series plot :

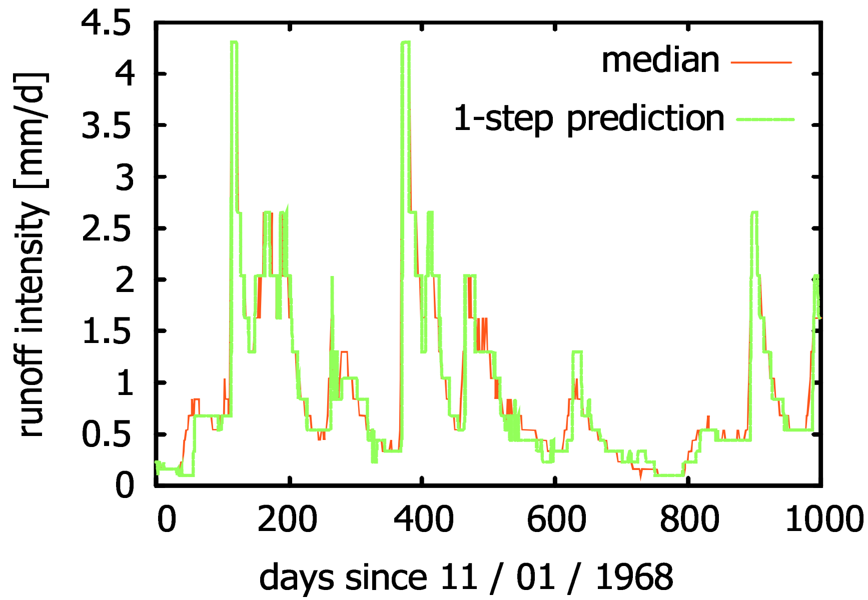


Classification of the past
=
One step ahead prediction

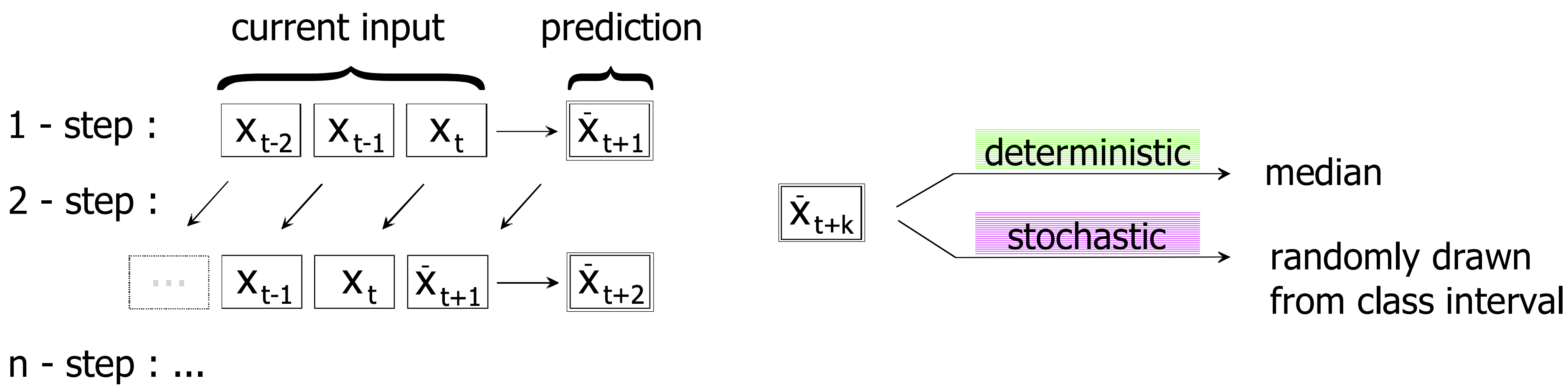
Experimental Observation:

Misclassifications are still close
to the target class median.

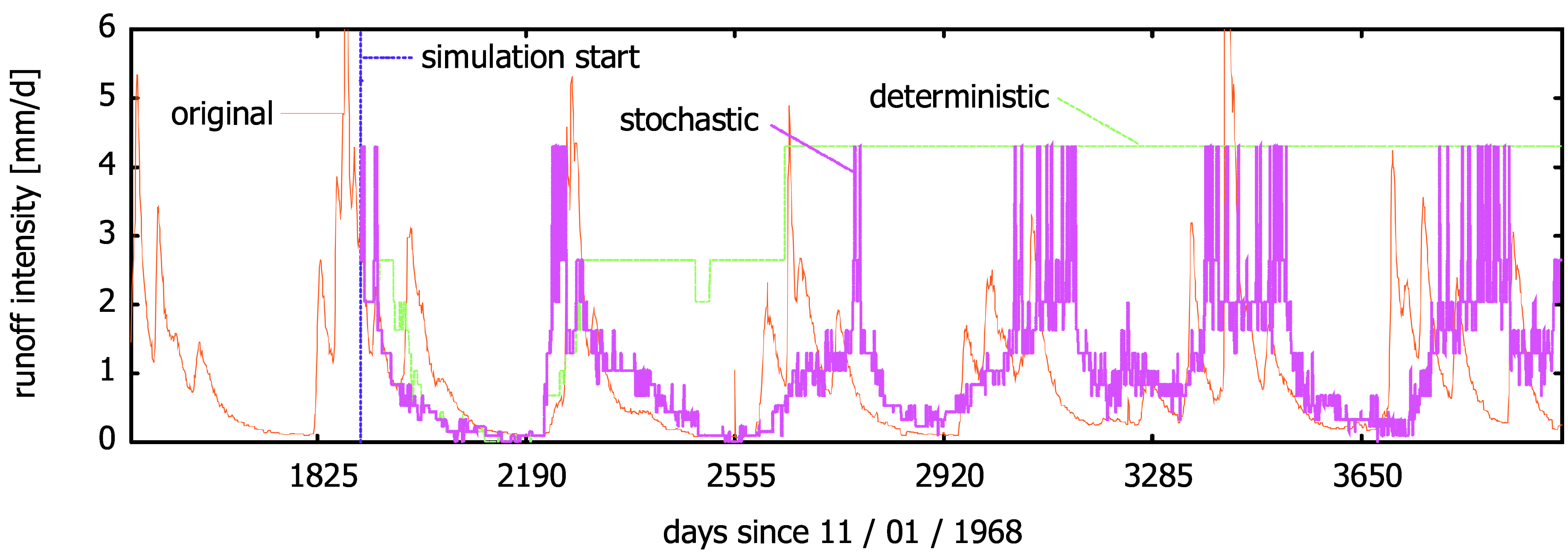
(" Neighborhood misclassification ")



Prediction



current input
prediction
1 - step :
2 - step :
n - step : ...
x_{t-2} x_{t-1} x_t → $\bar{x}_{t+1}$
... x_{t-1} x_t $\bar{x}_{t+1}$ → $\bar{x}_{t+2}$
$\bar{x}_{t+k}$
deterministic
median
stochastic
randomly drawn
from class interval



runoff intensity [mm/d]
days since 11 / 01 / 1968
simulation start
original
stochastic
deterministic
0
1
2
3
4
5
6
1825
2190
2555
2920
3285
3650

Conclusions

1. Adaptive metric yields **globally relevant** components.
   -> suitable for analysis („*Influence of the past?*“) and
   modeling (*embedding*, *prediction*) of time series data.

2. Approach leads to data driven (-> stationary ☹), compact description of the data. Only parameters:
   - Two learning rates,
   - Width of temporal window.

3. Components of the embedding vectors need not be temporally equidistant
   -> less aliasing than standard delay embedding.