the fourth Sensor Signal Processing for Defence (SSPD 2014) Conference, Edinburgh, UK, Sept 8-9, 2014

Learning Entropy for Novelty Detection: A Cognitive Approach for Adaptive Filters

Ivo Bukovsky, Cyril Oswald, Matous Cejnek, & Peter M. Beneš

Adaptive Signal Processing and Informatics Computational Centre (ASPICC)

Dpt. of Instrumetation and Control Eng., **CZECH TECH. UNIV. IN PRAGUE**

***Abstract*—This paper recalls the practical calculation of Learning Entropy (LE) for novelty detection, extends it for various gradient techniques and discusses its use for multivariate dynamical systems with ability of distinguishing between data perturbations or system-function perturbations. LE was introduced in 2013 [6] for novelty detection in time series via supervised incremental learning of polynomial filters, i.e. higher-order neural units (HONU). This paper demonstrates LE also on enhanced gradient descent adaptation techniques that are adopted and summarized for HONU. As an aside, LE is proposed as a new performance index of adaptive filters. Then, we discuss Principal Component Analysis and Kernel PCA for HONU as a potential method to suppress detection of data-measurement perturbations and to enforce LE for system-perturbation novelties.**

Keywords—novelty detection; learning entropy; learning entropy of a model; multivariate system; higher-order neural unit; incremental learning; kernel principal component analysis

Abbreviations

AISLE … Approximated Individual Sample Learning Entropy
GD … Gradient Descent
HONU… Higher-Order Neural Unit
ISLE … Individual Sample Learning Entropy
KPCA… Kernel Principal Component Analysis
LE … Learning Entropy
LEM... Learning Entropy of a Model
LNU, LF… Linear Neural Unit, Linear (adaptive) Filter
NLMS… Normalized Least Mean Squares, (Normalized GD)
OLE … Order of Learning Entropy
QNU… Quadratic Neural Unit
SampEn …. Sample Entropy

LEARNING ENTROPY (*Sample Entropy* vs. *Entropy Learning* vs. *Learning Entropy*)

Sample Entropy (not used here): A well recognized signal complexity evaluation algorithm (probability based quantification of signal complexity, Shannon-based approach).

***Entropy Learning (not used here)*:** A well recognized Shannon inspired neural network learning algorithm based on minimizing complexity (entropy) of neural weights in a network.

Learning Entropy (*Entropy OF Learning, 2013, used here*): *A new [6] non-Shannon based novelty detection algorithm based on observation of unusual learning effort of incrementally learning systems. A relative measure of novelty (information) recognized by pre-trained learning system. Novelty detection on every individual sample of data in complex behavior using a simple adaptive filters (predictors).*

adaptive filter output: $\tilde{y}(k)=\mathbf{w}(k)\cdot\mathbf{colx}(k-p)$

parameter adaptation: $\mathbf{w}(k+1)=\mathbf{w}(k)+\Delta\mathbf{w}(k)$

$$if\ \left(\left|\Delta w_i(k)\right|>\alpha_j\,\overline{\left|\Delta w_i(k)\right|}\right)\Rightarrow$$

An unusually large learning update of i^{th} adaptable parameter for particular detection sensitivity α_j

Average recent learning update of w_i

$$\boldsymbol{\alpha}\in\boldsymbol{\alpha}=\left[\alpha_1,\alpha_2,\dots,\alpha_j,\dots,\alpha_{n_\alpha}\right]$$

$$\alpha_1>\dots>\alpha_j>\dots>\alpha_{n_\alpha}$$

k … discrete index of time, p … prediction horizon,
$\mathbf{w}$ … vector of all adaptable parameters ($n_w \times 1$)
$\mathbf{x}$ … vector of inputs (and feedback variables if NARX) (x_0=1)

Learning Entropy: $E_A(k)=\dfrac{1}{n_w\cdot n_\alpha}\sum_{j=1}^{n_\alpha}\left\{N(\alpha_j)\right\};\ E_A\in\langle 0,1\rangle$

$E_A(k)$… Approximate Individual Sample **Learning Entropy**

$N(\alpha_j)$ … quantity of unusual learning updates for given detection sensitivity over all adaptable parameters at update time k

Orders of LE (OLE) and Corresponding Detection Rules (adopted from (6)):

OLE	Notation	Detection Rule of Unusual Learning Effort
0	E^0,E_A^0	$\left\|w_i(k)\right\|>\alpha\cdot\overline{\left\|w_i(k)\right\|}$
1	E^1,E_A^1	$\left\|\Delta w_i(k)\right\|>\alpha\cdot\overline{\left\|\Delta w_i(k)\right\|}$
2	E^2,E_A^2	$\left\|\Delta^2 w_i(k)\right\|=\left\|\Delta w_i(k)-\Delta w_i(k-1)\right\|>\alpha\cdot\overline{\left\|\Delta^2 w_i(k)\right\|}$
3	E^3,E_A^3	$\left\|\Delta^3 w_i(k)\right\|=\left\|\Delta^2 w_i(k)-\Delta^2 w_i(k-1)\right\|>\alpha\cdot\overline{\left\|\Delta^3 w_i(k)\right\|}$
4	E^4,E_A^4	$\left\|\Delta^4 w_i(k)\right\|=\left\|\Delta^3 w_i(k)-\Delta^3 w_i(k-1)\right\|>\alpha\cdot\overline{\left\|\Delta^4 w_i(k)\right\|}$

Order of Learning Entropy (OLE):

OLE determines the order of difference of adaptable parameters for calculating LE. (see the table on the left)

Learning Entropy Profile (LEP):

LEP is the integral of LE in time. (LEP is a cumulative graph of LE in time).

Learning Entropy of a Model (LEM):

LEM represents the total unusual learning effort of an adaptive model. (LEM is the latest point of LEP).

HONU for Learning Entropy

The advantage of HONU can be seen in customable polynomial nonlinearity and in-parameter linearity that suppress local minima issues for optimization with fundamental learning algorithms.

$$\mathbf{x}=\begin{bmatrix}x_0=1\\x_1\\x_2\\\vdots\\x_n\end{bmatrix}$$

LNU, LF $\quad\mathbf{colx}=\mathbf{x}$

QNU $\quad\mathbf{colx}=\left[\left\{x_i\cdot x_j;\ i=0\dots n,\ j=i\dots n,\ x_0=1\right\}\right]$

CNU $\quad\mathbf{colx}=\left[\left\{x_i\cdot x_j\cdot x_l;\ i=0\dots n,\ j=i\dots n,\ l=j\dots n,\ x_0=1\right\}\right]$

GD learning rule : $\Delta\mathbf{w}(k)=-\dfrac{1}{2}\mu\dfrac{\partial e(k)^2}{\partial\mathbf{w}}=\mu\cdot e(k)\cdot\mathbf{colx}(k-p)^T \qquad e(k)=y(k)-\tilde{y}(k)$

Modifications of GD Incremental Learning for HONU

Based on Normalized Least Mean Squares (or Normalized GD)	
NLMS [7]	$\Delta\mathbf{w}(k)=\dfrac{\mu}{\varepsilon+\left\|\mathbf{colx}(k-p)\right\|_2^2}\cdot e(k)\cdot\mathbf{colx}(k-p)^T$
GNGD [10]	$\varepsilon(k+1)=\varepsilon(k)-\rho\cdot\mu\dfrac{e(k)e(k-1)\mathbf{colx}(k-p)^{\mathrm{T}}\mathbf{colx}(k-p-1)}{\left(\left\|\mathbf{colx}(k-p-1)\right\|_2^2+\varepsilon(k)\right)^2}$
RR–NLMS [14]	$\varepsilon(k+1)=\max\left[\varepsilon_{\min},\ \varepsilon(k)-\rho\cdot sign\left(e(k)\cdot e(k-1)\cdot\mathbf{colx}(k-p)^T\cdot\mathbf{colx}(k-p-1)\right)\right]$
Based on Performance Index Derivative $\mu(k+1)=\mu(k)+\rho\cdot e(k)\cdot\boldsymbol{\gamma}(k)\cdot\mathbf{colx}(k-p)$	
Benveniste's [12]	$\boldsymbol{\gamma}(k)=\left[\mathbf{I}-\mu(k-1)\mathbf{colx}(k-p-1)\cdot\mathbf{colx}(k-p-1)^{\mathrm{T}}\right]\boldsymbol{\gamma}(k-1)+e(k-1)\cdot\mathbf{colx}(k-p-1)$
Farhang's & Ang's [13]	$\boldsymbol{\gamma}(k)=\eta\cdot\boldsymbol{\gamma}(k-1)+e(k-1)\cdot\mathbf{colx}(k-p-1);\ \eta\in\langle 0\ ,\ 1\rangle$
Mathew's [8]	$\mu(k+1)=\mu(k)+\rho\cdot e(k)\cdot e(k-1)\cdot\mathbf{colx}(k-p)^{\mathrm{T}}\cdot\mathbf{colx}(k-p-1)$

Conventional Error Criteria vs. LEMs of various OLEs for low-dimensional QNU (n=5) for MacKey-Glass chaotic time series evaluated on first 300 samples, starting from random weights:

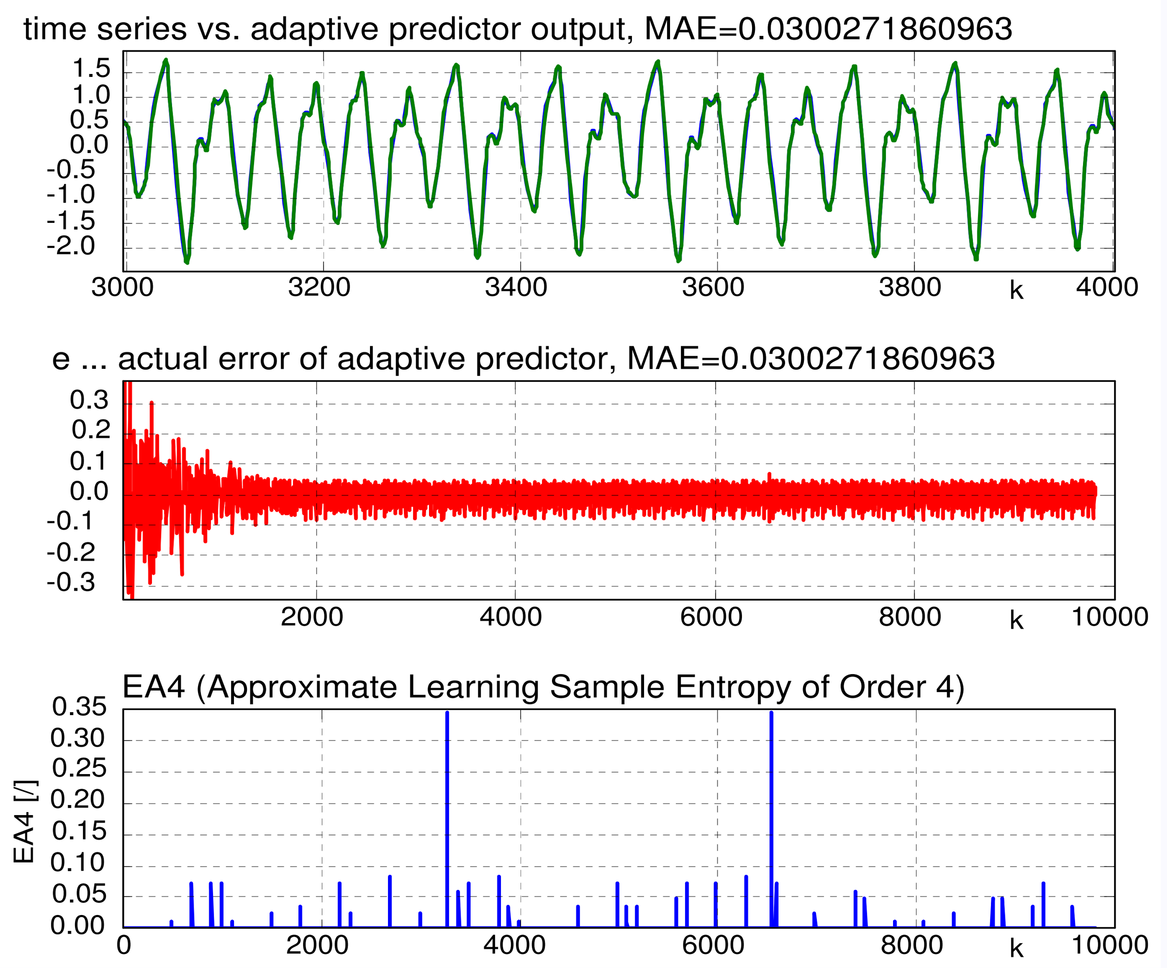
Performance Index / GD Method	MAE	RMSE	LEM₁	LEM₂	LEM₃	LEM₄
NMLS	0.099	0.13	11.8	9.39	8.73	0.31
GNGD	0.099	0.13	11.8	9.40	8.74	0.31
RR–NMLS	0.097	0.13	11.8	9.41	8.83	0.31
Benveniste's	0.237	0.35	14.9	15.0	15.7	4.2
Farhang's & Ang's	0.253	0.35	12.2	11.2	11.7	0.98
Mathew's	0.253	0.35	12.3	11.2	11.7	0.98

Conventional Error Criteria vs. LEM for MacKey-Glass chaotic time series for NLMS and various orders of HONU for adaptation samples k=8000:8300 (sampling 1 sec, n=5):

Performance Index / HONU (order)	Mean Abs. Error	Root Mean Sqrd. Error	LEM_1	LEM_2	LEM_3	LEM_4
LNU (1)	0.024	0.029	5.44	6.79	11.0	0.012
QNU (2)	0.021	0.026	10.4	9.46	11.9	0.796
CNU (3)	0.029	0.025	17.5	15.7	17.1	3.48

A typical result of detecting two small perturbations with NLMS-based modifications of GD. Notice, the predictor error (middle axes) does not indicate the two perturbations while AISLE ($EA4$) detects them uniquely:

y(3265:3267)=yo(3265:3267)+0.05; y(6530: 6532)=yo(6530: 6532)+0.05



Learning Entropy potentials for SSPD

LE was recently introduced as a cognitive signal processing algorithm that opens new potentials to novelty detection and to further research in various areas. LE displays strong potentials to instantly detect perturbation or instant changes of dynamical behavior with every new individual measured sample of data, where other floating window-based signal processing methods (e.g. SampEn) might need windows of data and longer time intervals => LE can be used as a complementary method of instant detection and time allocation of novelties including very small changes in dynamics and complex correlations of signals with the use of simple and real-time computationally effective adaptive filters (LNU, QNU). Further in our paper in the proceedings, the approach for detection of system perturbations vs. data perturbations is founded with the use of LE and KPCA. We believe this is an interesting topic for research, e.g., of real-time evaluation of data and for efficient monitoring of correct functionality of sensors. Also, LE can be used to instantly estimate actual accuracy of adaptive predictors, e.g., for synchronization purposes – we have been investigating LE for increasing the accuracy of beam targeting of radiation tracking therapy for biomedical purposes (Bukovsky et al, IEEE IJCNN, 2014), we believe there are some potentials for SSPD purposes as well. For the cognitive and nonlinear capabilities of adaptive filters and neural networks, LE might be also investigated for information processing on complex signals under the noise level. Perhaps, this might be also interesting topic for research of LE for SSPD purposes.

Acknowledgement

Ivo would like to thank prof. ***Madan M. Gupta*** *from* ***University of Saskatchewan*** *for introducing him into research of higher order neural units since 2003, and to prof.* ***Witold Kinsner*** *from* ***University of Manitoba*** *for introducing him into multi-scale analysis for signal processing since 2010.*

References:

[1] Markou, M.; Singh, S.: "Novelty detection: A review—Part 1: Statistical approaches" *Signal Process*. 2003, 83, 2481–2497.
[2] Markou, M.; Singh, S.: "Novelty detection: A review—Part 2: Neural network based approaches" *Signal Process*. 2003, 83, 2499–2521.
[3] Bukovsky, I.; Bila., J.; Gupta, M.M.; Hou Z.-G.; Homma, N.: "Foundation and Classification of Nonconventional Neural Units and Paradigm of Nonsynaptic Neural Interaction", *Discoveries and Breakthroughs in Cognitive Informatics and Natural Intelligence*; Wang, Y., Ed.; IGI Publishing: Hershey, PA, USA, 2009.
[4] Gupta, M., M., Bukovsky, I., Homma, N. , Solo M. G. A., Hou Z.-G.: "Fundamentals of Higher Order Neural Networks for Modeling and Simulation", in *Artificial Higher Order Neural Networks for Modeling and Simulation*, ed. M. Zhang, IGI Global, 2012.
[5] Bukovsky, I., Kinsner, W., Bila, J.: "Multiscale Analysis Approach for Novelty Detection in Adaptation Plot", *Sensor Signal Processing for Defence* 2012, London, UK.
[6] Bukovsky, I.: ¨Learning Entropy: Multiscale Measure for Incremental Learning¨, journal of *Entropy, special issue on Dynamical Systems*, ISSN 1099–4300, 2013, 15(10), 4159-4187; doi:10.3390/e15104159.
[7] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[8] V. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2075–2087, Jun. 1993.
[9] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction: Architectures, Learning Algorithms and Stability*. Chichester, U.K.: Wiley, 2001.
[10] D. P. Mandic, "A Generalised Normalised Gradient Descent Algorithm", *IEEE Signal Processing Letters*, Vol. 11, No. 2, pp. 115-118, 2004.
[11] D. P. Mandic and V. S. L. Goh, *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*. Wiley, 2009
[12] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation*. New York: Springer-Verlag, 1990.
[13] W.-P. Ang and B. Farhang-Boroujeny, "A new class of gradient adaptive step-size LMS algorithms," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 05–810, Apr. 2001.
[14] Young-Seok Choi, Hyun-Chool Shin, Woo-Jin Song: "Robust Regularization for Normalized LMS Algorithms", *IEEE Transactions on Circuits and Systems—II:Express Briefs*, Vol. 53, No.8, 2006.
[15] Scholkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller: "Kernel Principal Component Analysis", *Advances in Kernel Methods-Support Vector Learning,* 1999.
[16] Xueqin Liu, Uwe Kruger, Tim Littler, Lei Xie, Shuqing Wang: "Moving window kernel PCA for adaptive monitoring of nonlinear processes", *Chemometrics and Intelligent Laboratory Systems*, Volume 96, Issue 2, 15 April 2009, Pages 132–143
[17] Zhiqiang Ge E- Chunjie Yang, Zhihuan Song : "Improved kernel PCA-based monitoring approach for nonlinear processes", *Chemical Engineering Science*, Volume 64, Issue 9, 1 May 2009, Pages 2245–2255.
[18] Mingtao Ding, Zheng Tian, Haixia Xu: "Adaptive Kernel Principal Component Analysis", *Signal Processing*, Volume 90, Issue 5, May 2010, Pages 1542–1553

Code of (Approximate) Learning Entropy (http://aspicc.fs.cvut.cz/ASPICC_Software.htm)

```
#== Learning Entropy (AISLE) =======
def fcnEA(Wm,alphas,OLEs): #Wm ... recent window of weights including the very last weight updates
    OLEs=OLEs.reshape(-1)
    nw=Wm.shape[1]
    nalpha=len(alphas)
    ea=zeros(len(OLEs))
    i=0
    for ole in range(max(OLEs)+1):
        if ole==OLEs[i]: # assures the corresponding difference of Wm
            absdw=abs(Wm[-1,:]) # very last updated weights
            meanabsdw = mean(abs(Wm[0:Wm.shape[0]-1,:]),0)
            Nalpha = 0
            for alpha in alphas:
                Nalpha += sum(absdw>alpha*meanabsdw)
            ea[i] = float(Nalpha)/(nw*nalpha)
            i+=1
        Wm = Wm[1:,:]-Wm[0:(shape(Wm)[0]-1),:] #difference Wm
    return(ea)
```