ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

ESNigma: Efficient feature selection for Echo State Networks

Davide Bacciu and Filippo Benedetti and Alessio Micheli *

Dipartimento di Informatica - Università di Pisa - Italy

Abstract. The paper introduces a feature selection wrapper designed specifically for Echo State Networks. It defines a feature scoring heuristics, applicable to generic subset search algorithms, which allows to reduce the need for model retraining with respect to wrappers in literature. The experimental assessment on real-word noisy sequential data shows that the proposed method can identify a compact set of relevant, highly predictive features with as little as 60% of the time required by the original wrapper.

1 Introduction

Feature selection for sequential data is a key challenge in application scenarios, such as that of pervasive computing, where a considerable number of data sources is available, producing streams of heterogenous, noisy and potentially redundant measurements. Typical pervasive computing applications, for instance, comprise learning models receiving flows of measurements from a network of sensors pervasively deployed in the environment and providing real-time predictions, e.g. on ongoing events. Few approaches are specifically tailored to feature selection from multivariate time-series: [1], for instance, discusses a filter-based unsupervised approach to reduce feature redundancy; wrapper approaches [2], on the other hand, seek a feature subset that optimizes the performance of a specific learning model on a specific learning task. They typically employ a search procedure that recursively eliminates/adds features to a candidate subset [2], evaluating its fitness in terms of the validation performance of the learning model trained on the current feature subset. Clearly, such an iterative retraining approach poses strong computational requirements. This work proposes an efficient feature selection wrapper specific for Echo State Networks (ESNs) [3], that are recurrent neural networks well suited to deal with sequential data, being characterized by a good trade-off between accuracy and computational efficiency. For instance, in the RUBICON project [4], ESNs have been embedded on low-power sensor devices distributed in the environment and used to perform real-time predictive tasks based on the sensor data streams. In such a scenario, the use of feature selection approaches becomes fundamental even when the number of input features is small, as it allows to save critical resources. Computational efficiency of feature selection is also central as it is performed online within the distributed system. The wrapper approach introduced here addresses these two issue by proposing a feature ranking score that considerably reduces the need for multiple model retraining, yielding to a considerable reduction in the computational

*This work is partially supported by the EU FP7 RUBICON project (contract n. 269914).

load with respect to wrapper approaches in literature. Further, this appears to be the first feature selection algorithm designed specifically for the ESN model.

2 Feature Selection Wrapper for ESN

We introduce the *Echo State Network Input Gain Measurement Approximation* (ESNigma) score for efficient wrapper-based feature selection in ESN models. The intuition underlying the ESNigma score, inspired by the work in [5], is that the weights of a trained ESN readout can be exploited to determine the contribution of each input feature to ESN output. The magnitude of such contribution can then be used to rank the input features by relevance, guiding the selection of the feature subsets in the wrapper search procedure, without requiring to retrain the ESN model for each candidate input subset.

The ESNnigma score for the i-th feature, with respect to the k-th output, is

$$ESNigma_{ik} = \frac{LG_{ik}}{\max_i\{LG_{ik}\}} \cdot 100 \tag{1}$$

that is the normalized *local gain* LG_{ik} of the k-th output with respect to the i-th input. The local gain is defined as

$$LG_{ik} = \left|\frac{\partial y_k(n)}{\partial u_i(n)}\right| = \sum_{j=1}^{N_R} \left| W_{kj}^{out} \frac{\partial x_j(n)}{\partial u_i(n)} \right| \tag{2}$$

where $u_i(n)$ is the i-th input at time n and $y_k(n)$ is the corresponding k-th linear output. The rationale underlying (2) is that input signals that are noisy or irrelevant to the output will impact ESN performance if their gain is large in magnitude (in modulo). Local gain computation is based on a *backpropagation-like* derivation of (2), whose first step is shown as the rightmost element of the equality in (2), where W_{kj}^{out} is the k-th readout weight from the j-th reservoir neuron, having output $x_j(n)$ at time n. The derivation in (2), and those in the remainder of the section, follow from the standard equations of the leaky-integrator ESN model: these are omitted due to space limitations, but the reader is referred to [3] for a standard introduction to the model. The output of the j-th ESN reservoir neuron at time n can be rewritten as a function of the i-th input feature as

$$x_j(n) = (1-a)x_j(n-1) + a \cdot f\left(W_{ji}^{in} u_i(n) + C_{ji}(n) + \sum_{z=1}^{N_R} \hat{W}_{jz} x_z(n-1)\right) \tag{3}$$

where $W_{j\cdot}^{in}$ and $\hat{W}_{j\cdot}$ are the input-to-reservoir and the recurrent reservoir weights for the j-th neuron, respectively, f is the hyperbolic tangent activation function and a is a *leaking rate* (controls the speed of state dynamics). The term $C_{ji}(n)$ holds the contribution from all ESN inputs u to the j-th reservoir unit (including bias) except that from the i-th feature: hence, it can be considered a constant when differentiating by $u_i(n)$ in (2) and thus disappears when deriving (3).

The derivation of (3) can be simplified as in [5] by considering an approximation that substitutes the nonlinear activation function f (an hyperbolic tangent) with a linear factor F. This approximation is reasonable since we are mostly interested in the contribution of the weights' magnitude to the output, rather than in the relationship between the output and the neuron dynamics; the approximation's error in computing the local gain is deemed acceptable [5] when the inputs of the neural model are all in the same range (in our case, this holds as ESN inputs are normalized). Therefore, the derivative in (3) writes as

$$\frac{\partial x_j(n)}{\partial u_i(n)} = (1-a)\frac{\partial x_j(n-1)}{\partial u_i(n)} + a \cdot F \cdot \left(W^{in}_{ji} + \sum_{z=1}^{N_R} \hat{W}_{jz} \frac{\partial x_z(n-1)}{\partial u_i(n)} \right) \tag{4}$$

given that $\frac{\partial u_i(n)}{\partial u_i(n)} = 1$ and $\frac{\partial C_{ji}}{\partial u_i(n)} = 0$. To complete calculation of (4), we need to be able to compute, for each reservoir neuron j', the recursively defined derivative

$$\frac{\partial x_{j'}(n-1)}{\partial u_i(n)} = (1-a)\frac{\partial x_{j'}(n-2)}{\partial u_i(n)} + aF \left(W^{in}_{j'i} \frac{\partial u_i(n-1)}{\partial u_i(n)} + \sum_{z=1}^{N_R} \hat{W}_{j'z} \frac{\partial x_z(n-2)}{\partial u_i(n)} \right). \tag{5}$$

The recursion base is for $n = 1$, when $\frac{\partial x_{j'}(0)}{\partial u_i(1)} = 0$. The term $\frac{\partial u_i(n-1)}{\partial u_i(n)}$ is a time-lagged derivative of the i-th input signal. Now, since we are dealing with sequential data, we should assume that there exists a certain relationship between the input signal at time n and its value at time $n-1$. Hence, it is reasonable to assume that such derivative is not null. On the other hand, for the sake of computing the local gain, we are not interested in quantifying the exact value of such relationship. Therefore, it is reasonable to assume that such a derivative exists and it is equal to some value q that is constant at each time instant and for each input feature i. As a result, (5) rewrites as

$$\frac{\partial x_{j'}(n-1)}{\partial u_i(n)} = (1-a)\frac{\partial x_{j'}(n-2)}{\partial u_i(n)} + aF \left(W^{in}_{j'i} \cdot q + \sum_{z=1}^{N_R} \hat{W}_{j'z} \frac{\partial x_z(n-2)}{\partial u_i(n)} \right). \tag{6}$$

In the remainder of the paper, we will assume $q = 1$ for the sake of simplicity. The term in (6) can be recursively computed by unfolding it in time, with base case for $\frac{\partial x_{j'}(1)}{\partial u_i(2)} = a \cdot F(W^{in}_{j'i} q)$, providing a closed form definition for the local gain LG_{ik}. Computing (6), using a simple identity function as linear approximation $F(\cdot)$, requires $2 + N_R$ multiplications for each reservoir neuron and time instant. By appropriately exploiting the common terms in (6) and its recursive definition, it can be efficiently computed if unfolded from time 1 to $n-1$: at a given time step t, we compute and store the partial derivative $\frac{\partial x_{j'}(t-1)}{\partial u_i(t)}$ for all reservoir neurons j', which is used at time $t+1$, to compute the next partial derivative. This yields to a complexity that is $O(n \cdot (2N_R + N_R^2))$ multiplications for computing partial derivatives up to time n. As regards memory occupation, at each time t, we need only to store information on the partial derivatives (6) at the previous time step (for each reservoir neuron).

The ESNigma score can be used to guide different feature subset search procedures: the rationale is try to eliminate first those features with higher local gain that, if noisy or irrelevant, can have a stronger negative effect on ESN performance. Here, we focus on a simple greedy Hill Climbing (HC) search, to evaluate the sole effect of ESNigma, sparing any benefit originated by the use of refined search policies. At the same time, greedy HC search is adequate for the time and resource constrained applications, as it maintains computational complexity to the minimum. Very briefly, the ESNigma HC (EHC) algorithm has the following scheme:

1. Start with the feature subset $\mathcal{F}$ containing all the features; set iteration counter $it = 0$.
2. Use a Cross-Validation (CV) setting to train the ESN on the current subset $\mathcal{F}$ and compute the validation error at current iteration E^{it}_{val}.
3. Compute the ESNigma score for each $i \in \mathcal{F}$ at iteration it.
4. Repeat the following until a delectable feature is identified or all features in $\mathcal{F}$ have been tested:
    (a) Select from $\mathcal{F}$ the first untested feature i' with the highest ESNigma
    (b) Set $\mathcal{F}' = \mathcal{F}/i'$; train the ESN on the feature subset $\mathcal{F}'$ and compute the validation error $E_{\mathcal{F}'}$.
    (c) If $E_{\mathcal{F}'} < E^{it}_{val}$ jump to step 5, else mark i' as tested and jump to step 4(a).
5. If no features have been selected at step 4 terminate, else set $it = it + 1$, $E^{it}_{val} = E_{\mathcal{F}'}$, $\mathcal{F} = \mathcal{F}'$ and jump back to step 3.

Steps 3 shows the key advantage of using ESNnigma over performance-based wrappers in literature: the latter, in fact, would have required to retrain and assess the ESN on each subset generated by eliminating one feature from the current feature set in order to determine the feature ranking. ESNnigma, on the other hand, allows to perform feature ranking by using the ESN trained at the previous HC iteration allowing, in principle, to retrain a single ESN on the new feature subset obtained by removing the worse ESNigma feature in step 4(b). Indeed, more refined search procedures can benefit from the computational efficiency of the ESNnigma policy: [5] provides a comparative experimental analysis of different search algorithms for feature selection in feedforward neural networks.

3 Experimental Results

The experimental comparison assesses if ESNigma is advantageous in terms of trade-off between predictive accuracy, number of selected features and computational requirements (i.e. time needed to perform feature subset selection). To this end, we compare the EHC wrapper with a performance-based HC wrapper, where the latter identifies the candidate feature for elimination by training a different ESN for each feature subset, ultimately selecting the one with the best validation error. Two benchmarks from a pervasive computing application, that is localization from *Received Signal Strength Information* (RSSI), are used. The scenario includes a Wireless Sensor Network (WSN) comprising a set of stationary devices (*anchors*) that exchange radio packets with a mobile device (the *mote*). The task is to train an ESN to provide the x, y localization of the mote mounted on a mobile robot from the RSSI of the radio packets exchanged between anchors and the mote. RSSI data tends to be very noisy, with a behavior heavily influenced by changes to the operating environment, which makes this a challenging learning task [4, 6, 7]. The *Turtle* benchmark comprises 5 anchors and a mote deployed on a corridor of the Computer Science Department in Pisa. A simple Turtlebot robot is used to navigate the corridor in straight paths (one-way) of approximately $20m$ in length. Ground truth localization is collected by a localization and mapping software using an RGBD Kinect camera, for a total of 12 sequences. The *Stella* benchmark comprises 10 anchors and a mote deployed on two corridors of the Stella Maris Children Hospital, in Pisa. A commercial Robotnik AGVS robot [6] is used to perform L-shaped paths (both-ways) of approximately $40m$. Ground truth localization data is collected by an high-quality laser-based localization system, for a total of 17 sequences. Feature selection is fundamental for these tasks even if the number of input features is not large, as a reduction of ESN inputs entails the deployment of less anchors (cost saving), of less radio channels towards the mote (battery saving), and a minor use of computational resources on the mote hosting the ESN (resource saving).

Experiments have been performed by extracting 4 sequences from each dataset to constitute the out-of-sample test. The remainder of the samples has been used in a 4-fold CV procedure for both model hyperparameter selection and within the feature selection process for feature subset scoring. The tested model configurations include the reservoir size $N_R \in \{50, 100, 200, 500\}$ and the leaky parameter $a \in \{0.05, 0.1, 0.2\}$. Readout weights are trained by ridge regression with the regularization parameter λ selected from $\{0.001, 0.01, 0.1, 1\}$ also in the 4-fold CV. Table 1 shows the test results for the model selected configurations of EHC, HC and for a baseline case using all original features (also subject to model selection). On Turtle data, EHC yields to a lower localization error than HC for the same feature subset size, with 2 selected features common to both algorithms. Notably, EHC test error is not significantly different than that obtained by the baseline method using all the original 5 features, whereas EHC uses only 3 features. The EHC algorithm has also significantly lower computational requirements with respect to HC, completing the feature selection process with 66% of time required by the HC method. On the Stella data, the localization performance of EHC and HC is basically equivalent (no statistically significant difference in the test error), however EHC obtains this result using only 5 features (i.e. 50% of the original), while HC ultimately selected 7 features. Again, the computational requirements of the EHC method are considerably lower than that of HC, with a reduction of 39% of the time required to complete the feature selection process. Note that the clear difference in the localization performance between the Turtle and Stella data has to be ascribed to the quality of the ground-truth localization data that, in the latter case, is obtained through an accurate commercial laser-based system, while in the former case is obtained by an open-source research software using a low-cost RGBD camera. Nevertheless, both results are in line with, or better than, the performances of RSSI-based localization in literature, see [7] for a recent review.

Dataset	Full		ESNigma-HC			HC		
	E_{tst}	$\#\mathcal{F}$	E_{tst}	Time	$\#\mathcal{F}$	E_{tst}	Time	$\#\mathcal{F}$
Turtle	1.17 (0.26)	5	1.20 (0.54)	38.58	3	1.57 (0.41)	57.13	3
Stella	0.42 (0.08)	10	0.49 (0.08)	49.39	5	0.47 (0.03)	80.96	7

Table 1: Results on the Turtle and Stella datasets: E_{tst} is the mean Euclidean test error (in meters) with variance (in brackets), completion time (in minutes) and number of selected features. Full denotes a model using all original features.

4 Conclusion

We have introduced a wrapper algorithm designed explicitly for feature selection in Echo State Networks. In particular, we have proposed a novel score (ESNigma) that computes a feature ranking from the ESN weights, allowing to identify candidate feature subsets while reducing the need to retrain the neural model, as in wrapper approaches in literature. The experimental comparison with an hill-climbing wrapper from literature provides evidence that the ESNigma score allows to reduce the computational cost of computing feature selection of more than one-third. At the same time, ESNigma seems capable of identifying more compact feature subsets, while maintaining comparable predictive performance. The proposed ESNigma approach is general and can be applied to more refined search policies, such as those employing backtracking and stepwise elimination [5], which will be subject of future research.

References

[1] Davide Bacciu. An iterative feature filter for sensor time-series in pervasive computing applications. In *Proc. of EANN 2014*, volume 459 of *CCIS*, pages 39–48. Springer, 2014.

[2] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46(1-3):389–422, 2002.

[3] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[4] Davide Bacciu, Paolo Barsocchi, Stefano Chessa, Claudio Gallicchio, and Alessio Micheli. An experimental characterization of reservoir computing in ambient assisted living applications. *Neural Computing and Applications*, 24(6):1451–146, May 2014.

[5] Chun-Nan Hsu, Hung-Ju Huang, and Dietrich Schuschel. The ANNIGMA-wrapper approach to fast feature selection for neural nets. *IEEE Trans. Sys. Man Cyb. B*, 32(2):207–212, Apr 2002.

[6] Stefano Chessa, Claudio Gallicchio, Roberto Guzman, and Alessio Micheli. Robot localization by echo state networks using rss. In *Recent Adv. of Neural Netw. Models and App.*, volume 26 of *Smart Innovation, Sys. and Tech.*, pages 147–154. Springer, 2014.

[7] Dongliang Guo, Yudong Zhang, Qiao Xiang, and Zhonghua Li. Improved radio frequency identification indoor localization method via radial basis function neural network. *Mathematical Problems in Engineering*, 2014.