Clustering of Context Data Using K-Means with an Integrate and Fire Type Neuron Model

John A. Flanagan
Nokia Research Center
PO Box 407, FIN-00045, NOKIA GROUP, Finland
adrian.flanagan@nokia.com

Abstract - *K-Means and the Self-Organizing Map (SOM) are two important unsupervised learning algorithms. The SOM is more robust in its convergence than K-Means but generally requires the input data to be independent and identically distributed (iid). In both cases, for sequential input, a time varying learning rate needs to be defined. In the context recognition problem, iid data samples and varying learning rates are difficult to obtain. A K-Means type algorithm referred to as the K-SCM is described. Using an integrate and fire type neuron model the K-SCM with a constant learning rate can cluster non iid symbol string data. Using real measured context data the clustering abilities of the K-SCM are demonstrated.*

Key words - Unsupervised Clustering, SOM, Symbol Strings, Integrate and Fire, K-Means, K-SCM, Context Recognition

1 Introduction

The K-Means algorithm [1] and its self-organizing generalization the Self-Organizing Map (SOM) [2] are two of the most commonly used, unsupervised, learning algorithms. In a typical application the K-Means is used to cluster a set of data $\{\mathbf{x}(t) \in R^n : t = 1, \ldots, \}$ while the SOM forms a topology preserving, vector quantization, of $\mathcal{S}$ the support of the probability distribution of $\mathbf{x}$ denoted by $p_{\mathbf{x}}$. Intrinsically the SOM, through the use of a neighborhood function, is less sensitive to initial conditions and hence exhibits a robust convergence compared to K-Means. The self-organization and convergence properties of the SOM have been formally analyzed, with results typically confined to the one dimensional case for example [3], [4], [5], [6]. One point in common to all of these analyzes is the requirement that the $\mathbf{x}(t)$ be independent and identically distributed (iid).
With the evolution of mobile technology and improvements in the development of various types of low cost, low power, sensors the research area of context awareness is becoming increasingly important. The aim in context awareness is to sense different characteristics of a user's environment, or state, and determine which context the user is in. Based on an accurate recognition of the user's context the idea is then to (semi-)automatically provide services, applications etc. that may be required by the user in that context without requiring any explicit input from the user. In general, part of the context recognition problem can be viewed as the extraction of features, in this case user contexts, from the fusion of multiple information sources. Feature extraction or context recognition by unsupervised clustering has many advantages in terms of personalization and the possibility of not needing any user

interaction during learning. These advantages have been discussed in more detail in Flanagan [7]. It would seem that K-Means or the SOM are potentially applicable to the context recognition problem. However, the context recognition problem has some characteristics which make their application difficult.

In the context recognition problem the information sources can be very diverse ranging from 3-axis accelerometers to location information from the GSM/GPS network, to the identities of the people in the user's immediate environment. If the output of each of these sources is represented by a real number $x_j \in R$, then the fusion of the sources is represented by the vector $\mathbf{x} = (x_1, \ldots, x_n)$ which is suitable for use in K-Means or the SOM. However this first requires some form of normalization etc of the individual components x_i in order to achieve a reasonable result. Another alternative is to perform feature extraction on the signal(s) from individual information sources, interpret the features as states of the source and represent each state by a symbol. The data vector $\mathbf{x}$, representing the fusion of the sources, is now replaced by a symbol string $\mathbf{s} = (s_1, \ldots, s_n)$ with the s_i the symbolic representation of the state of source i. Kohonen and Somervuo [8] and Somervuo [9] have used symbol string data as input to the SOM, where the SOM is used in batch mode. Clustering of symbol string data using the Symbol String Clustering Map (SCM) is described in [10]. However even if it is possible to cluster or vector quantize symbol string data there is still the problem that when using the SOM or the SCM, the input data samples need to be iid. Furthermore using K-Means, SOM or SCM with sequential inputs also necessitates defining a learning rate $\alpha(t)$ such that $\alpha(t) \rightarrow \epsilon, t \rightarrow \infty$, where $\epsilon = 0$ or $\epsilon \approx 0$. Both the need for iid input data and varying learning rate present challenges in context recognition.

Context recognition is based on recognizing a user's context, however if we sample a users context every second, successive samples are not likely to be iid. Another approach used in Himberg et al [11], [12] is to time segment the context data sequence into independent segments and use samples from successive segments as inputs. This approach still has many problems one of which is significant computational and memory resources not available on a mobile device where the feature extraction is performed. Another alternative is to transfer the data to a central server where the feature extraction is performed and the results transmitted back to the user device. This option also requires significant power and communication resources. In both approaches there still remains the problem of deciding a rate of change for the gain parameter $\alpha(t)$. Ideally data samples from all possible user contexts should be used as input at some time before $\alpha(t)$ becomes very small, difficult when the number of different contexts or the time durations of different contexts cannot be known a-priori. The ideal solution would be to have a situation where α does not need to be changed and hence there is continuous learning.

In a very general sense it is highly likely that whatever data processing occurs in the brain there is not the condition that the data signals from sensory organs be iid and continuous learning is possible. It seems that Artificial Neural Network (ANN) algorithms such as the SOM, which are considered computational models of brain functions, do not respond well to these conditions. The Integrate and Fire (I&F) neural models [13] are generative neuron models, reproducing the spiking process measured in real neurons. The computational abilities of spiking neurons is considered to be some form of temporal and/or rate coding of the spikes. I&F neurons have been used for unsupervised clustering of real valued data using temporal coding by Bothe et al [14]. Their approach depends on precise spike timing and they have the problem of transforming real vector data into appropriate spike time delays.

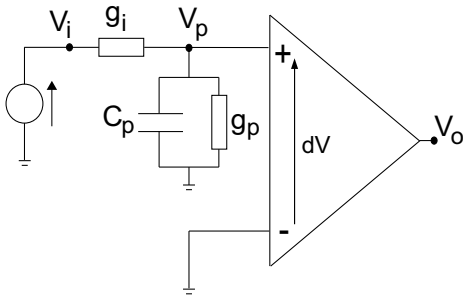


Figure 1: Basic neuron model of the K-SCM.

In what follows a K-Means algorithm using a set of I&F type neuron models is described capable of clustering symbol string data in an unsupervised manner. Unlike other spiking neuron models there is no explicit coding of information in the spike timings of the neurons. For any input string $\mathbf{s}$, which is modelled by a set of voltage sources, the winner node is the first node to fire. Winner Take All (WTA), Hebbian learning is used to update the synaptic weights such that the response of the winning neuron to the input is increased. The algorithm is now referred to as the K-SCM. Furthermore, unlike the SOM or K-Means, for the K-SCM to function the input data need not be iid and there is no learning parameters that need to be adapted and hence continuous learning can be possible.
In Sec. 2 the I&F model of a K-SCM neuron is presented and some of its properties associated with the winner determination and learning are mentioned. The means of determining the winner and adapting the synaptic weights is described in Sec. 3. An example of clustering symbol string context data with the K-SCM is given in Sec. 4 followed by the conclusion in Sec. 5.

2 K-SCM Neuron Model

In this section the basic model of a K-SCM neuron is presented and some of its properties described. Figure 1 shows an illustration of the neuron, not unlike the typical leaky integrate and fire model [13], however here we use voltages and conductances rather than currents and resistances to describe the model. The neuron has an internal voltage dV and output voltage V_o. Connected to the '+' terminal is a capacitance C_p in parallel with a conductance g_p with the second terminal of both elements grounded. Also connected to the '+' terminal of the neuron is a voltage source V_i through a second conductance g_p. Associated with the neuron is the threshold voltage V_T which for $dV = V_p > V_T$ then $V_o = 1$ and $V_o = 0$ otherwise. An important point in the K-SCM is that the V_T of the neuron is allowed to vary.
A simple analysis shows that,

$$V_p(t) \;=\; \frac{g_i}{g_i + g_p}\left(1 - e^{-t/\tau}\right) \;\;, \tag{1}$$

with $\tau \;=\; C_p/(g_i + g_p)$ the time constant. The steady state value of V_p is given by $V_p(\infty) \;=\; g_i/(g_i + g_p)$ and if $0 < V_T < V_p(\infty)$ and $V_p(0) = 0$ then the time t' for $V_p(t)$ to exceed V_T is given by,

$$t' \;=\; -\tau \;\log\left(1 - V_T(g_i + g_p)/g_i\right) \;\;. \tag{2}$$

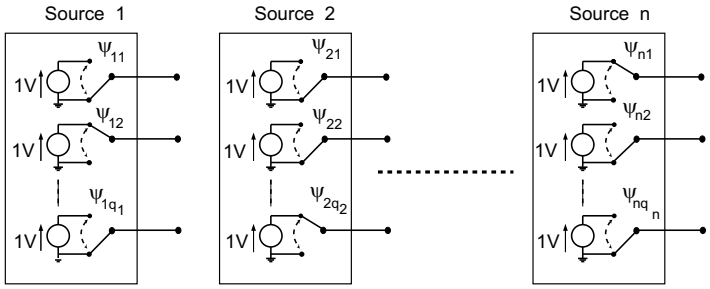


Figure 2: Voltage source model for a symbol string.

These properties of the neuron are directly related to the determination of the winner for a given input. Another important property of this model relevant to the learning stage of the K-SCM is that if g_i is increased and/or g_p is decreased then for a constant V_T the time t' is decreased.

3 Winner Determination in the K-SCM

A K-SCM has a total of N neurons and as in K-Means or the SOM for a given input the first step is to determine the best matching neuron, or winner for that input. Before discussing winner determination in the K-SCM a voltage source model of a symbol string input is presented. The symbol string $\mathbf{s} = (s_1, \ldots, s_n)$ is considered as a fusion of the discrete states of n information sources. Each source i has q_i possible states and $s_i(t) \in \{\psi_{i1}, \ldots, \psi_{iq_i}\}$ at any time t. Figure 2 shows an illustration of how the symbol string can be represented by a set of voltage sources. Information source i has a set of q_i voltage sources $\{\psi_{i1}, \ldots, \psi_{iq_i}\}$ which can be either $[0V, 1V]$ and at any one time only one of the $\psi_{ir} = 1V$ and $\psi_{is} = 0V, \forall s \neq r$. All the voltage sources ψ_{ij} of information source i are connected to the '+' node of neuron k through a conductance g_{ij}^k as illustrated in Fig. 3. The g_{ij}^k correspond to the synaptic weights and for our purposes $g_{ij}^k \in [0, 1]$. Using the fact that conductances in parallel add together we can define the equivalent g_p of Fig 1 for each neuron k as,

$$g_p^k \;=\; \tilde{g}_p^k \;+\; \sum_{l=1}^{n} \sum_{j=1}^{q_l} g_{lj}^k \; \delta(\psi_{lj}) \;\; , \tag{3}$$

where $\delta(\psi_{lj}) = 1$ if $\psi_{lj} = 0$ and $\delta(\psi_{lj}) = 0$ otherwise. The equivalent value of g_i from Fig. 1 for each neuron k is given by,

$$g_i^k \;=\; \sum_{l=1}^{n} \sum_{j=1}^{q_l} g_{lj}^k \; \delta(1 - \psi_{lj}) \;\; . \tag{4}$$

Let V_p^k be the V_p of Fig. 1 for each neuron k. For a given state of the ψ_{ij} and $V_p^k(0) = 0, \forall k$ then $V_p^k(t)$ can be calculated using a simple numerical iteration

$$V_p^k(t+1) - V_p^k(t) \;=\; \alpha(g_i^k \;-\; V_p^k(t)(g_i^k + g_p^k)) \;\; , \tag{5}$$

with $\alpha \propto 1/C_p^k$. If $V_p^k(\infty) > V_T^k$ then for some $t' < \infty$ iterations the neuron fires. The first neuron to fire is called the winner neuron. In the case that $V_p^k(\infty) \leq V_T^k, \forall k$ then no

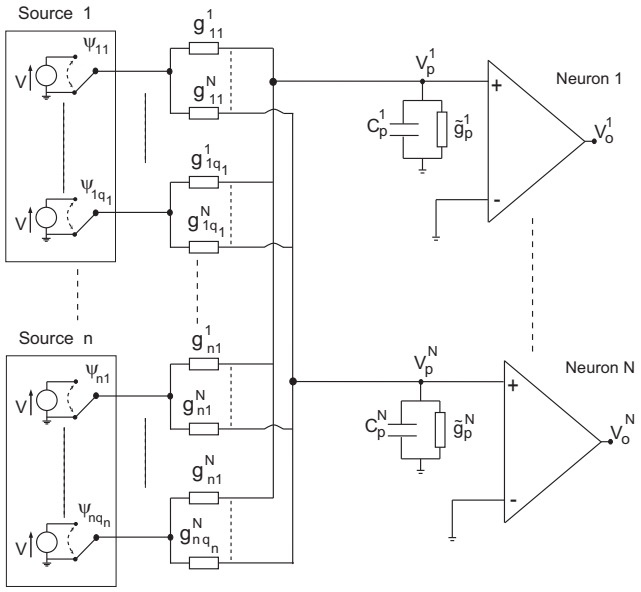


Figure 3: K-SCM circuit model with information sources and neurons.

neuron fires. For this reason the winner determination is modified so that the neuron fires if, $V_p^k(t) > \gamma_k(t)V_T^k$, where $\gamma_k(0) = 1$ and when $V_p^k(t)$ is saturated (i.e. $g_i^k - V_p^k(t)(g_i^k + g_p^k)) < \phi$, with $0 < \phi \ll 1$) then $\gamma_k(t) = \gamma_k(t-1) * \rho$, with $0 < \rho < 1$ and $\rho \approx 1$. Furthermore if there is a small random additive noise signal added to $V_p^k(t)$ it is possible to show that for every given input there is a winning neuron with probability 1.
After determining the winner neuron v for a given input the synaptic conductances g_{ij}^v are updated in a manner so as to decrease the time for V_p^v to exceed the threshold V_T^k for the same input. This is achieved by increasing the synaptic weights g_{ij}^v for which $\psi_{ij} = 1$ and decreasing the g_{ij}^v for which $\psi_{ij} = 0$ as follows,

$$g_{ij}^v = g_{ij}^v + \begin{cases} \alpha(1 - g_{ij}^v), & \text{if } \psi_{ij} = 1 \\ \alpha(1 - g_{ij}^v)(0 - g_{ij}^v), & \text{if } \psi_{ij} = 0 \end{cases} \tag{6}$$

The learning is asymmetrical and as $g_{ij}^v \to 1$ then it is more difficult to decrease it. Note that α is a constant learning factor. It turns out that in order for the K-SCM to find clusters in the data it is also necessary to adapt V_T^v as,

$$V_T^v = V_T^v + \begin{cases} \alpha_1(V_p^v(t') - V_T^v), & \text{if } V_p^v(t') < V_T^v \\ \alpha_2(V_p^v(t') - V_T^v), & \text{if } V_p^v(t') > V_T^v \end{cases} \tag{7}$$

with $\alpha_2 \ll \alpha_1$ constant gain factors. It can be shown that even if V_T^v is increased in this manner for the same input the time to fire of neuron v still decreases consistent with the update of the synaptic weights g_{ij}^v.

4 Clustering Context Data with the K-SCM

As an example of how the K-SCM can cluster symbol string data, a set of real measured data, the Nokia Context database, publicly available at [15] is used. The data set consists of a set

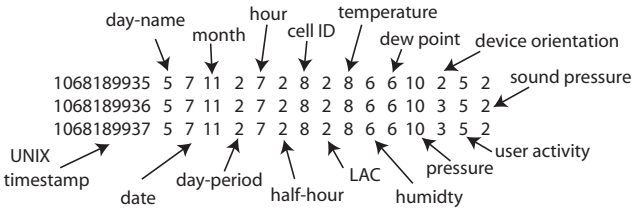
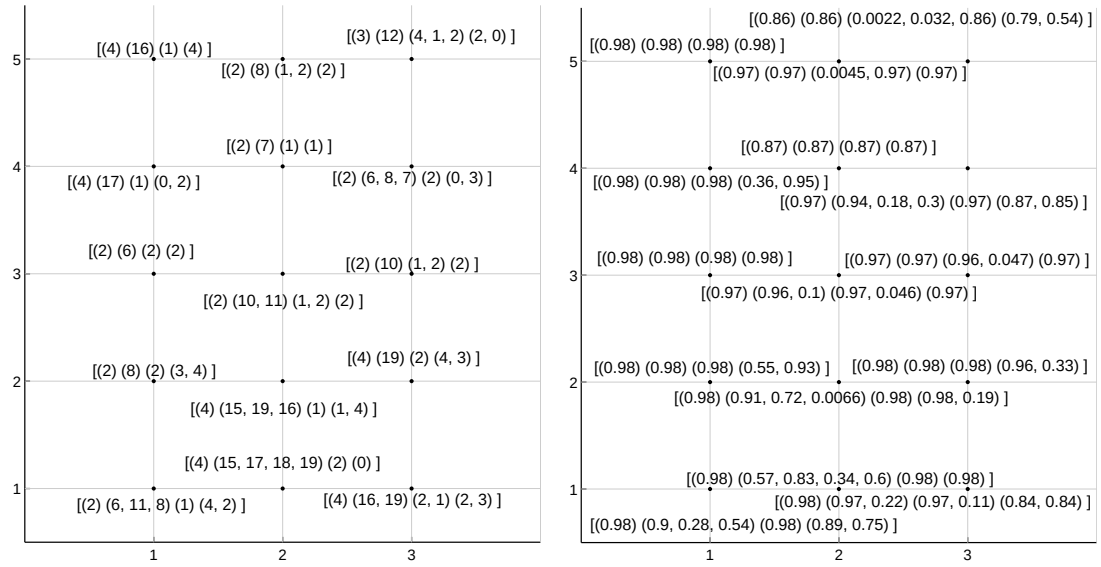


Figure 4: Sample of the symbol string data from recordings used to train the K-SCM.

of feature files for 43 different recording sessions. In each recording session the same user carried a mobile phone, sensor box and laptop PC, going from home to the workplace or vice-versa. During the journey the user walks, takes a bus and Metro and sometimes uses a car. During the session, sensors recorded 3-axis acceleration, atmospheric pressure, temperature, humidity etc. The ambient audio was recorded on the laptop using a microphone and sound card. On the mobile phone, the user's location was recorded as Cell ID and Location Area Code (LAC) as defined by the GSM network. After each recording session the signals were processed and basic features such as average sound level etc. extracted and quantized. Each quantization level of each feature was assigned an integer number denoting the "state" of the source. The state of each of the sources, including time, was recorded once a second. Figure 4 shows an example extract from one such session. In the example of the K-SCM that follows the input to the K-SCM is a symbol string using information sources, ("day-period", "hour", "half-hour", "LAC") and from Fig. 4 at time 1068189935 the symbol string is of the form (2, 7, 2, 8) where once again the integers act as symbols representing states of the information sources with their numerical value not significant. In fact each symbol represents the voltage source ψ_{ij} of information source i for which $\psi_{ij} = 1$ at that second.

During training of the K-SCM with $N = 15$ (i.e. easy to represent the result), time consecutive samples of the data were used as input to the K-SCM (i.e. samples were not iid). The results of training a randomly initialized K-SCM with this data are shown in Fig. 5. For the sake of presentation the K-SCM is presented as a 5×3 grid, with each node (x, y) of the grid representing a neuron k. Despite the fact that the K-SCM in Fig. 3 is fully connected with every voltage source of every information source connected to every neuron through the conductances g_{ij}^k, in the real implementation if $g_{ij}^k < 0.001$ then it is not represented in the results that follow. In Fig. 5 (a) beside each node (x, y) is an associated symbol string enclosed in $'[]'$s. Inside these brackets in the ()'s are the symbols, or states, of each information source for which the associated $g_{ij}^k > 0.001$. The values of the g_{ij}^k associated with the states represented in Fig. 5 (a) are shown in Fig. 5 (b). For example node $(1, 2)$ (i.e. neuron 4) in Fig. 5 (a) has associated symbol string $[(2)(8)(2)(3, 4)]$. Note the K-SCM symbol strings can have different dimension than the input. Referring to the weight string associated with node $(1, 2)$ in Fig. 5 (b) we say, ψ_{12} (i.e. info. source 1, volt source 2) is connected to neuron 4 by $g_{12}^4 = 0.98$, ψ_{28} (i.e. info. source 2, volt source 8) is connected to neuron 4 by $g_{28}^4 = 0.98$, ψ_{32} is connected to neuron 4 by $g_{32}^4 = 0.98$, ψ_{43} is connected to neuron 4 by $g_{43}^4 = 0.55$ and ψ_{44} is connected to neuron 4 by $g_{44}^3 = 0.93$ with all other $g_{ij}^4 < 0.001$. In terms of the original context data we interpret the cluster defined by node $(1, 2)$ as representing the "morning time", "between 8:15 and 8:45" in locations 3 or 4. A similar interpretation can be applied to symbol strings associated with the other nodes. It is found that the learned symbol strings correspond to the most frequently occurring symbol strings in the training



(a) Symbol strings. (b) Synaptic weights g_{ij}^k.

Figure 5: The result of training the K-SCM with a set of context data represented by symbol strings.

data, or else not so frequently occurring but quite different from the other symbol strings, for example $[(2)(7)(1)(1)]$ at node $(2, 3)$. This indicates the K-SCM is performing a clustering of the symbol string data and similar results are obtained starting from other initial conditions indicating its robustness. The same K-SCM (i.e. same parameter settings) with sensor feature input (e.g. temperature, humidity etc.) also results in a clustering despite the very different statistical nature of the data compared to the time/location example shown here.

5 Conclusions

Unsupervised clustering of data using the SOM or K-Means for context recognition, where the data samples are not independent and identically distributed poses a problem. Furthermore the sequential input mode of the SOM and K-Means are designed for real vector data rather than symbol string data. The K-SCM consisting of a set of neuron models with integrate and fire functionality has been described. The K-SCM operates with symbol string data which need not be iid and furthermore, no varying learning rate needs to be defined. The K-SCM when applied to real measured context data finds clusters in a robust, unsupervised manner. Further analysis will show in more detail the operation of the K-SCM and which functions are most important in the clustering.

Acknowledgements

This work has been performed in the framework of the IST project IST-2004-511607 MobiLife, which is partly funded by the European Union. The author would like to acknowledge the contributions of his colleagues, although the views expressed are those of the author and do not necessarily represent the project.

References

[1] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings - 5th Berkeley Symposium*, 1:281–297, 1967.

[2] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.

[3] Marie Cottrell and Jean-Claude Fort. Étude d'un processus d'auto-organisation. *Annales de l'Institut Henri Poincaré*, 23(1):1–20, 1987. (in French).

[4] Catherine Bouton and Gilles Pagès. Self-organization and a. s. convergence of the one-dimensional Kohonen algorithm with non-uniformly distributed stimuli. *Stochastic Processes and Their Applications*, 47:249–274, 1993.

[5] A.A. Sadeghi. Self-organization property of Kohonen's map with general type of stimuli distribution. *Neural Networks*, 11:1637–1643, 1998.

[6] J. A. Flanagan. Self-organization in the one-dimensional SOM with a decreasing neighborhood. *Neural Networks*, 14:1405–1417, 2001.

[7] J.A. Flanagan. Context awareness in a mobile device: Ontologies versus unsupervised/supervised learning. To appear AKRR'05, 2005.

[8] T. Kohonen and P. Somervuo. Self-organizing maps of symbol strings. *Neurocomputing*, 21:19–30, 1998.

[9] P. Somervuo. *Self-Organizing Maps for Signal and Symbol Sequences*. PhD thesis, Helsinki University of Technology, 2000. Acta Polytechnica Scandinavia, Mathematics and Computing Series No. 107.

[10] John A. Flanagan. A non-parametric approach to unsupervised learning and clustering of symbol strings and sequences. In *Proceedings of WSOM 2003 (Intelligent systems and innovational computing)*, pages pp 128–133, Kitakyushu, Japan, 2003.

[11] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H.T.T. Toivonen. Time series segmentation for context recognition in mobile devices. In *IEEE Intl' Conf. on Data Mining (ICDM2001)*, pages 203–210, 2001.

[12] J. A. Flanagan, J. Himberg, and J. Mäntyjärvi. Unsupervised clustering of symbol strings and context recognition. In *IEEE Intl' Conf. on Data Mining (ICDM02)*, pages 171–178, Maebashi City, Japan, 2002.

[13] W. Gerstner and W. Kistler. *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[14] S.M. Bothe, H. La Poutré, and J.N. Kok. Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks. *IEEE Transactions on Neural Networks*, 13(2):426–435, March 2002.

[15] R. Mayrhofer. Context database, 2004. Available at, http://www.soft.uni-linz.ac.at/-Research/Context_Database/.