A Self-Organizing Neural Network for Detecting Novelties

Marcelo Keese Albertini
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
albertini@icmc.usp.br

Rodrigo Fernandes de Mello
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
mello@icmc.usp.br

ABSTRACT

In order to detect new events, a system must support on-line learning, adapting to pattern dynamic characteristics. Studies of such adaptation have originated the novelty detection area, which aims at identifying unexpected or unknown patterns. These researches have motivated this work to propose the on-line and unsupervised Self-Organizing Novelty Detection (*SONDE*) neural network. In this network, the creation of new neurons points out novelties. Experiments evaluated the influence of *SONDE* parameters and their capability to detect novelty events. These evaluations considered the datasets Biomed, ALL-AML Leukemia and DLBCL. Results are compared to others from *GWR*.

Categories and Subject Descriptors

I.2 [**Artificial Inteligence**]: Connectionism and neural nets; I.5 [**Pattern Recognition**]: Neural nets

Keywords

novelty detection, self-organizing neural networks.

1. INTRODUCTION

In stable systems only expected events occur. These systems do not require continuous learning techniques, because they can be trained based on previously known patterns. Unstable systems require techniques capable of on-line learning and adapting to the behavior of input patterns. In such situations it is necessary to detect divergences between known and unknown data.

Several studies have been conduced to design techniques which are adapted to environment conditions [7]. This motivated the adoption of classifiers to detect deviations from training data. After training a classifier, any unusual event is detected as novelty.

The most common methods used in novelty detection are based on statistics and artificial neural networks [7]. The statistics methods estimate the expected data behavior using probability distribution functions. The ones based on neural networks generalize data knowledge. In both cases, any event divergent from the expected behavior is classified as novelty.

Among the statistical techniques we may mention the Parzen windows [10], which can be parametric (they need previous knowledge on data) and non-parametric (they are able to adapt based on a data subset) and support vector machines (*SVM*), based on the statistical learning theory [11].

The neural network capability of knowledge generalization and continuous learning have motivated studies on novelty detection [7]. These studies do not consider traditional classifiers, using supervised learning, as they do not learn new behavior during execution. For efficient novelty detection, they have considered unsupervised and adaptive neural networks such as *SOM* (*Self-Organizing Maps*) [12], *ART* (*Adaptive Resonance Theory*) [3] and *GWR* (*Grow When Required*) [8].

The features of these artificial neural networks have motivated the design of the Self-Organizing Novelty Detection (*SONDE*). This network classifies, in a unsupervised manner, similar input patterns in the same neuron. After classifying a input pattern in a neuron, this unit is stimulated to represent historical input data. All neurons have maximum action radii to classify patterns. The action radius and centroid of a neuron adapt as new input patterns are classified.

When no neuron is able to classify a pattern, a new one is created, indicating a novelty. Although, as input patterns change, neurons adapt, forgetting past information. The neuron forgetness and adaptation rates are parameterized by users.

Experiments using a linear function were considered to evaluate the *SONDE* parameters. Further experiments using the datasets Biomed [4], ALL-AML Leukemia [6] and Lymphoma [1] were conduced and results compared to *GWR*.

This paper is organized as follows: 2) related work; 3) Self-Organizing Novelty Detection; 4) *SONDE* parameter analysis; 5) experiments and results; 7) conclusions.

2. RELATED WORK

Marsland [8] proposes a neural network named *GWR* (*Grow When Required*) which modifies *SOM*, adding the feature to create neurons during execution. *GWR* is composed of two stages: initialization and interaction. In the first stage, two neurons are created with random weight vectors. In the

SAC'07 March 11-15, 2007, Seoul, Korea
Copyright 2007 ACM 1-59593-480-4 /07/0003 ...$5.00.

interaction stage, the best matching unit (*BMU* – neuron with the highest activation) for each input pattern is obtained. When the best matching unit activation is below an activation threshold, a new neuron is created to represent the input pattern.

Besides that, this technique considers a habituation metric to detect novelties based on new input patterns. The habituation is measured by means of the reply (synaptic efficiency) of a neuron under stimuli. *GWR* creates a new neuron when the *BMU* is trained (high habituation) and its activation is low.

Spinosa and Carvalho [9] present classification results of one-class *SVMs* to detect novelties in bioinformatics datasets. *SVM* is a machine learning technique based on the statistical learning theory by Vapnik [11]. This technique optimizes a convex function, avoiding local minimum problems. *SVM* is less susceptible to overfitting when compared to other techniques, reaching good results when working on new samples. It also presents good results when classifying patterns with many dimensions. Experiments using the datasets Biomed [4], ALL-AML Leukemia [6] and DLBCL [1] confirm good results.

Flexer *et al.* [5] apply novelty detection to retrieve songs based on the spectral similarity and gender labels. Experiments considered $2,522$ songs spread out in 22 genders. Two central minutes of each song were evaluated and novelties detected using two algorithms. The first, named ratio-reject, uses a data density distribution, obtained from the training phase, for classification. The second algorithm, named *knn-reject*, defines a neighborhood during training which is used to classify known and unknown data.

3. SELF-ORGANIZING NOVELTY DETECTION

The Self-Organizing Novelty Detection artificial neural network (*SONDE*) was designed aiming at integrating features from *SOM*, *GWR* and *ART* to detect novelties. The *SONDE* architecture (figure 1) is divided in three layers: a input and pre-processing layer – where patterns can optionally be normalized; a neuron competitive layer – where occurs the neuron activation for input patterns; and a last to choose the best matching unit *BMU* – where the best neuron (with highest activation) is stimulated to better represent the input pattern received.

The *SONDE* is based on the representation of input data in adaptive neurons. Neurons are created as novelties are detected. Each neuron c is defined by the average centroid $\vec{w}_c$ of its patterns, the average radius rad_c around these patterns and a minimum similarity degree α_c to recognize new patterns.

For each received pattern $\vec{I_t}$ (optionally normalized in the layer 1 by equation 1) at the instant t, the activation value a_c of each neuron in the competitive layer is calculated using equation 2. When no neuron is capable of representing $\vec{I_t}$, i.e. when the equation 3 is satisfied, a new neuron is created. The new neuron centroid $\vec{w}_{new}$ is equal to the pattern values responsible for its creation (resulting in maximum activation where $a_{new} = 1$), and the minimum similarity degree α_{new} is equal to a predefined value α_0. The initial average radius rad_{new} is equal to $-\ln(\alpha_0)$ what represents the radius coverage for the initial similarity degree.

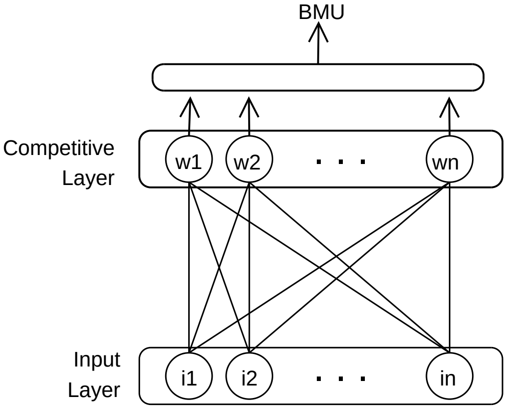

Figure 1: Self-Organizing Novelty Detection Architecture.

$$\vec{I_t} = \frac{\vec{I_t}}{\|\vec{I_t}\|} \tag{1}$$

$$a_c = \exp(-\|\vec{I_t} - \vec{w}_c\|) \tag{2}$$

$$a_c < \alpha_c, \forall c \tag{3}$$

When the equation 5 is satisfied, the winner neuron (or best-matching unit) is obtained applying the equation 4. This neuron better represents the input pattern.

$$BMU = argmax_c(\exp -\|\vec{I} - \vec{w}_c\|) \tag{4}$$

$$\|\vec{I} - \vec{w}_c\| \leq -\ln \alpha_c \tag{5}$$

The winner neuron is adapted aiming its specialization (equation 6, with t being the time step). This specialization is done in a way the neuron centroid and radius represent the average of input values following the exponential weighted moving averages (*EWMA*) presented, respectively, in the equations 6 and 7, where γ and Ω define the influence of past patterns in the current situation. The higher is the value of these parameters, the greater is the forgetness degree of *SONDE*. After updating the winner neuron centroid, its minimum similarity degree (α_c) is updated to better represent similar input patterns in next activations.After being specialized, any divergent pattern can be detected as novelty. This update is proportional to the modification degree p (equation 8) of the new average radius compared to the previous one following the equation 9.

This equation ensures two different adaptation behaviors to α_c (such as presented in figure 2) which follow the distance between the radius (which defines an inner boundary to the neuron specialization) and the similarity limit $-\ln(\alpha_c)$.The larger is this distance, the faster is the specialization which converges to better match the input patterns. When patterns are classified spread out over all the radius coverage area, the similarity limit may be close to the inner boundary, what implies that the neuron has found the best match for input patterns. This is similar to the SVM training, although the SVM learning phase is not on-line and self-adaptive.

$$\vec{w}_{BMU_t} = \vec{w}_{BMU_{t-1}} * (1 - \gamma) + \vec{I}_t * \gamma \tag{6}$$

$$rad_{BMU_t} = rad_{BMU_{t-1}} * (1 - \Omega) + \|\vec{I}_t - \vec{w}_{BMU_{t-1}}\| * \Omega \tag{7}$$

$$p = \|\frac{rad_{BMU_t} - rad_{BMU_{t-1}}}{\max(rad_{BMU_t}, rad_{BMU_{t-1}})}\| \tag{8}$$

$$\alpha_{BMU_t} = \min\left((1 + p) * \alpha_{BMU_{t-1}}, \exp^{(-rad_{BMU_t}*(1+p))}\right) \tag{9}$$

4. PARAMETER ANALYSIS

Experiments were conduced to: evaluate the *SONDE* forgetness rate according to neuron adaptations using weighted exponential moving averages (*EWMA*); demonstrate the *SONDE* algorithm behavior for a linear input dataset; and analyze the *SONDE* execution time complexity.

In the first experiment a generic *EWMA* (equation 10) was used to evaluate the behavior of centroid and radius adaptive equations. From this *EWMA* the inequation 11 was obtained where: fe is the forgetness threshold; ψ is the forgetness rate parameter; t is the current instant; n is the number of patterns received to forget any other. The goal of this inequation is determine a value ψ which considers a window of n elements to calculate the *EWMA*. A small fe must be chosen when a large number of past patterns has to be considered in neuron adaptation.

$$avg_t = avg_{t-1} * (1 - \psi) + \psi \tag{10}$$

$$fe \leq \psi * (1 - \psi)^{n-1},\ n \in \mathbb{Z}, n \geq 0 \tag{11}$$

After that, experiments evaluating the behavior of *SONDE* for a linear dataset were considered (using the equation $f(x) = 1$). The figure 2 presents the radius adaptation results of a single neuron in the intermediary layer with the parameters $\gamma = 0.01$, $\Omega = 0.01$ and $\alpha_0 = 0.2$. The y-axis represents the minimum activation (obtained by the equation 2) that a pattern must have to be accepted by a neuron. As the activation increases, the more specific the neuron becomes, accepting only patterns really close to its centroid.

The parameterization of *SONDE* influences the neuron specialization rate and how neuron follows pattern tendencies. The figures 3 and 4 present the number of neurons created by the linear function $f(x) = 0.001x$ using, respectively, $\alpha_0 = 0.1$ and $\alpha_0 = 0.4$. In this case, where the linear function behavior varies slowly, we consider to classify all the function in a single adaptive neuron and only great changes are detected as novelties (creating additional neurons).

By comparing the figures, we observe that in the first one the variation of Ω to low values of γ (lower than 0.2) influences less to create new neurons, being $\alpha_0 = 0.1$, what determines a vast coverage radius, generalizing patterns. By the figures, we may also observe that the parameter Ω influences more to create neurons, as there is low variation in the linear function values. Besides the low variation, when a neuron is too much specialized around a radius, there is good chances to create new ones.

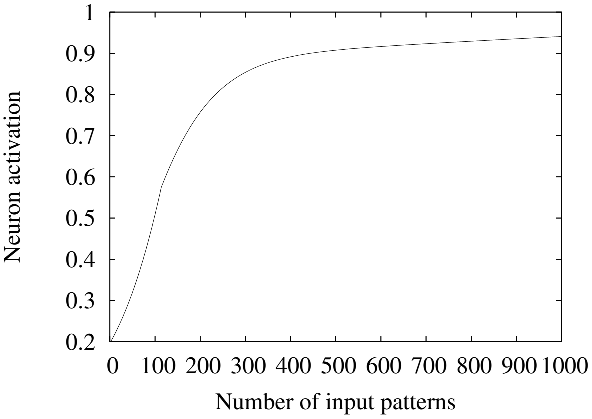

Figure 2: Neuron adaptation with input data from equation $f(x) = 1$.

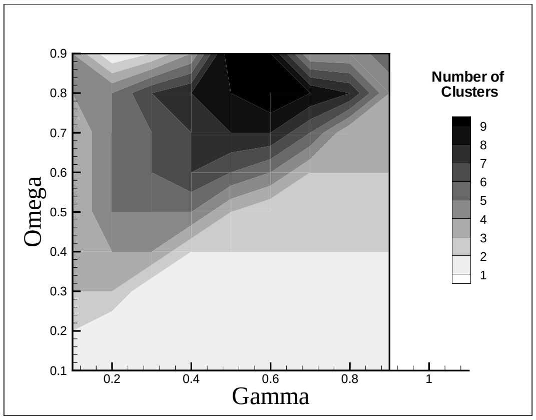

Figure 3: Linear function – alpha 0.1.

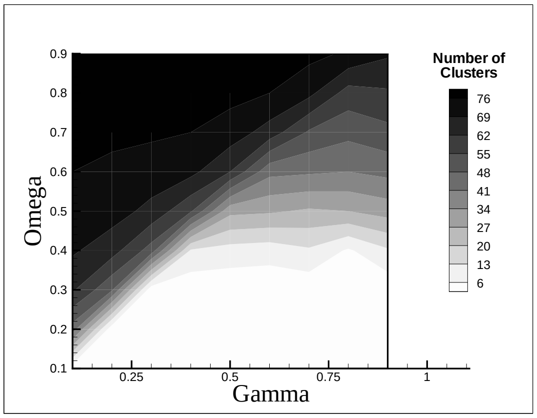

Figure 4: Linear function – alpha 0.4.

After these experiments we have analyzed the execution time complexity for the *SONDE* neural network algorithm presented in 1. This algorithm is composed of the normalization input vector phase (which is optional), the calculus of neuron activations and centroid adaptation or creation of new neurons. The complexity is measured according to the number of execution steps for input dataset dimensions (n), the number of neurons k created and the number of input patterns i.

In the normalization phase, for each input pattern $\vec{I}$, n steps are executed, where n is constant during execution and equal to the number of input dataset dimensions. In the next phase, k activations are executed. For each one a calculus of distance is done between two vectors which takes n execution steps. In this way, the activation phase executes a number of steps proportional to $k*n$ for each input $\vec{I}$. The last phase, which can update or create a new neuron, takes up to $2*n$ steps for each input because of the modification in the weight vector of the winner neuron (n steps) and the calculus of the new average distance n. Thus, by multiplying the number of steps of each input vector by the number of inputs i, we have $i*(3+k)*n$ steps, and, consequently, the algorithmic complexity is $\Theta(i*k*n)$.

Algorithm 1 *SONDE* neural network algorithm.

```
{The dimension of input vectors $\vec{I}_t$ is $n$.}
{$K$ represents the set of neuron centroids.}
{The parameters $\gamma$, $\Omega$ and $\alpha_0$ are constants.}
{A neuron $i$ is composed of: $\alpha_i$, $rad_i$ and $\vec{w}_i$.}
for all $\vec{I}_t$, from the instant $t \in \mathbb{N}$ do
  {Normalization of input vectors. It runs when $n > 1$.}
  $\vec{I}_t := \|\vec{I}_t\|$
  {Calculating each neuron activation and searching the best neuron $BMU$}
  for all $\vec{w}_c \in K$ do
    {Calculating the distance between $\vec{w}_c$ and $\vec{I}_t$.}
    $dist_c := \|\vec{w}_c - \vec{I}_t\|$
    {Calculating the activation to the current neuron $c$.}
    $act_c = \exp(-dist_c)$
    if $act_c > act_{BMU}$ and $act_c > \alpha_c$ then
      {Finding the neuron with the highest activation.}
      $BMU := c$
    end if
  end for
  if $BMU$ was found then
    {Adapting the best neuron.}
    $\vec{w}_{BMU_t} := \vec{w}_{BMU_{t-1}} * (1-\gamma) + \gamma * \vec{I}_t$
    $rad_{BMU_t} := (1-\Omega) * rad_{BMU_{t-1}} +$
        $\Omega * \|\vec{I}_t - \vec{w}_{BMU_{t-1}}\|$
    $p := \|\frac{rad_{BMU_t} - rad_{BMU_{t-1}}}{\max(rad_{BMU_t}, rad_{BMU_{t-1}})}\|$
    $\alpha_{BMU_t} := \min((1+p) * \alpha_{BMU_{t-1}},$
        $\exp^{(-rad_{BMU_t}*(1+p))})$
  else
    {Creating a new neuron $new \in K$.}
    $\vec{w}_{new} := \vec{I}_t$
    $\alpha_{new} := \alpha_0$
    $rad_{new} := -\ln \alpha_0$
  end if
end for
```

5. EXPERIMENTS

This section presents the experiments conduced with datasets usually adopted to novelty detection. The goal of using these datasets is the *SONDE* neural network validation and the comparison to other techniques. The datasets were organized in two parts. The first, containing training sets with similar characteristics (according to the author's dataset), was submitted to *SONDE* to learn about data, the second, composed of randomly organized sets, was submitted to evaluate the neural network capacity to detect novelties. This technique considers the network in production environment, habituated to certain patterns.

Patterns were normalized in the input layer. Neurons previously created by *SONDE*, when working on the training dataset, store known information. Any neuron created when working on the randomly distributed patterns points out novelty.

The metrics precision, recall and f-measure metrics were adopted to evaluate experiments [2]. The measurements were captured varying in 0.1 units the control parameters γ, Ω and α_0. Precision is the proportion of true detected novelties over the total number of detected novelties (some events, detected as novelties, cannot really be). Recall is the proportion of true detected novelties over the total number of occurred novelties.F-measure is a harmonic average which summarizes the precision and recall results, simplifying the comparison to other techniques.

5.1 Biomed

The Biomed dataset[1], created by Larry Cox *et. al* [4], contains information and blood measurements obtained from carriers or non-carriers of a genetic disorder. Patterns with *NA* (not available) elements were removed and the columns of age and 4 blood measurements were used in experiments. Other available data are: blood measurement date; number of samples per patient and the hospital identifier.

Firstly 127 patterns of non-carriers were submitted to *SONDE*, after that, 194 randomly distributed patterns (127 from non-carriers and 67 from carriers) were submitted.

We observe that the best results were obtained for $\gamma = 0.3$, $\Omega = 0.1$ and $\alpha_0 = 0.5$ which are $P = 0.85$, $R = 0.70$ and $f-measure = 0.77$. In this case, 5 neurons were created to generalize the 127 initial patterns of non-carriers. The parameters varied in 0.1 unit, consequently, a smaller variation may reach better results.

Evaluating all results we obtained a mean f-measure of 0.59 with standard deviation equals to 0.04 (with mean precision of 0.46 and standard deviation of 0.06; with mean recall of 0.84 and standard deviation of 0.1).

5.2 DLBCL

The DLBCL dataset (Diffuse large B-cell lymphoma) [1] consists of 47 patterns with 4,026 measurements obtained from patients with tumors. This dataset was prepared using *DNA* microarrays to characterize the genetic expression in B cells. The obtained measurements show the variation of the proliferation rate, host response and the state of tumor differentiation. These data are classified in two patterns following the stage of B-cell differentiation: germinal and activated. Patients presenting the first type, answer better to the treatment.

The 47 patterns are divided into 24 of type germinal and 23 activated. Firstly the germinal patterns were submitted to *SONDE* gets habituated, after that, a set containing all patterns randomly distributed was submitted.

We observe the neural network, in many situations, presented results such as $P = 1$, $R = 1$ e $f-measure = 1$, creating at about 21 neurons to generalize 24 patterns. Apparently results are good, although there is too much specialization (almost a neuron per pattern), this generates overfitting. A better result is obtained with $\gamma = 0.2$, $\Omega = 0.1$ and $\alpha_0 = 0.3$ where $P = 0.76$, $R = 1.0$ and $f-measure = 0.86$.

In this case, 3 neurons were generated to generalize 24 patterns, what increases the neural network capability to

[1] Available at: http://lib.stat.cmu.edu/datasets/

detect novelties.Evaluating all results we obtained a mean f-measure of 0.83 with standard deviation of 0.24 (with mean precision of 0.85 and standard deviation of 0.22; with mean recall of 0.83 and standard deviation of 0.26).

5.3 ALL-AML Leukemia

The ALL-AML Leukemia dataset [6] consists of 72 patterns with 7, 129 measurements obtained from acute leukemia carriers classified in the types ALL (*acute lymphoblastic leukemia*), with 38 patterns, and AML (*acute myeloid leukemia*), with 34 patterns. Firstly 38 ALL patterns were submitted to*SONDE* gets habituated to recognize these characteristics. After that, 72 randomly organized patterns (38 of ALL and 34 of AML) were submitted. In this case an unexpected event is defined by the occurrence of an AML pattern.

As the DLBCL dataset, this one presented results of $P = 1$, $R = 1$ e $f - measure = 1$ with low generalization, generating approximately a neuron per pattern. A good result was obtained with $\gamma = 0.1$, $\Omega = 0.4$ and $\alpha_0 = 0.6$ where $P = 0.65$, $R = 1.0$ and $f - measure = 0.76$. In this case, 7 neurons were create to generalize 38 initial patterns. Evaluating all results, we obtained a mean f-measure of 0.62 with standard deviation of 0.24 (with mean precision of 0.56 and standard deviation of 0.26; with mean recall of 0.72 and standard deviation of 0.23).

6. RESULT ANALYSIS

There are works which propose on-line and adaptive neural networks to novelty detection such as *GWR* [8]. Although, *GWR* source code[2] considers a first training phase to *GWR* adapt to datasets, after it starts classifying. This is contradictory to the on-line and adaptive features presented in [8], as it needs a training phase before classifying. In these cases there is loss of precision and recall to detect novelties, what may restrict its application in production environments. In this way, new training can be necessary, increasing computational costs.

For allowing the comparison to *GWR* [8], experiments were conduced using the dataset *biomed* applying a training phase to *SONDE* without centroid and radii adaptation in validation phase. The obtained results with parameters $\gamma = 0.075$, $\Omega = 0.05$ and $\alpha_0 = 0.025$ allowed to detect 59 of 67 carriers and classify wrongly 8 people as carriers. Thus, it was obtained a precision of $P = 0.8805$, recall $R = 0.9516$ and $f - measure = 0.9147$. This result can be compared to the best obtained by other techniques such as *GWR* in [8] which, classifies correctly 56 of 67 carriers and wrongly 2 people as carries, being $P = 0.9655$, $R = 0.8358$ and $f - measure = 0.8959$. We observe by the harmonic average, f-measure, that this summarization confirms better values to *SONDE*.

The *SONDE* results are not compared to the ones from *SVM* in [9], because the authors used another metric (accuracy rates).

7. CONCLUSIONS

This paper presents the *SONDE* (*Self-Organizing Novelty Detection*) artificial neural network, proposed to detect unexpected events in dynamic environments, where adaptation and on-line training are necessary. Experiments were conduced to demonstrate the *SONDE* parameter influence, generalization and classification. The following datasets were adopted to compare *SONDE* to other techniques: Biomed, ALL-AML Leukemia and DLBCL. Results were compared to *GWR* [8] which have motivated the adoption of *SONDE* to detect novelties in dynamical systems.

8. ACKNOWLEDGMENTS

The authors thank Dr. André Carlos Ponce de Leon Ferreira de Carvalho and Dr. Luciano José Senger for helping reviewing this work whilst in development.

9. REFERENCES

[1] A. A. Alizadeh and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, February 2000.

[2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[3] G. A. Carpenter, S. Grossberg, and D. B. Rosen. ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition. *Neural Networks*, 4:4934–504, 1991.

[4] L. Cox, M. Johnson, and K. Kafadar. Exposition of statistical graphics technology. In *ASA Proceedings Statistical Computation Section*, pages 55–56, 1982.

[5] A. Flexer, E. Pampalk, and G. Widmer. Novelty detection based on spectral similarity of songs. In *Proceedings of 6th International Conference on Music Information Retrieval*, pages 260–263, September 2005.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999.

[7] M. Markou and S. Singh. Novelty detection: a review,part i: statistical approaches. *Signal Process*, 83(12):2481–2497, 2003.

[8] S. Marsland, J. Shapiro, and U. Nehmzow. A self-organising network that grows when required. *Neural Networks*, 15(8-9):1041–1058, October 2002.

[9] E. J. Spinosa and A. C. de Carvalho. Combining one-class classifiers for robust novelty detection in gene expression data. *Lecture Notes in Computer Science*, 3594:54–64, August 2005.

[10] L. Tarassenko. Novelty detection for the identification of masses in mammograms. In *4th IEE International Conference on Artificial Neural Networks*, volume 4, pages 442–447, Cambridge, UK, 1995.

[11] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[12] A. Ypma and R. P. W. Duin. Novelty detection using self-organizing maps. In *Progress in Connectionist-Based Information Systems*, volume 2, pages 1322–1325. Springer, London, 1997.

[2] Available at: http://www-ist.massey.ac.nz/smarsland/GWR.html.