Neural Plasma

Daniel Berrar and Werner Dubitzky
Systems Biology Research Group, School of Biomedical Sciences,
University of Ulster, Northern Ireland
{dp.berrar, w.dubitzky@ulster.ac.uk,
WWW home page: http://research.bioinformatics.ulster.ac.uk/~dberrar/

Abstract. This paper presents a novel type of artificial neural network, called *neural plasma*, which is tailored for classification tasks involving few observations with a large number of variables. Neural plasma learns to adapt its classification confidence by generating artificial training data as a function of its confidence in previous decisions. In contrast to multilayer perceptrons and similar techniques, which are inspired by topological and operational aspects of biological neural networks, neural plasma is motivated by aspects of high-level behavior and reasoning in the presence of uncertainty. The basic principles of the proposed model apply to other supervised learning algorithms that provide explicit classification confidence values. The empirical evaluation of this new technique is based on benchmarking experiments involving data sets from biotechnology that are characterized by the small-*n*-large-*p* problem. The presented study exposes a comprehensive methodology and is seen as a first step in exploring different aspects of this methodology.

1 Introduction

Recent experimentation techniques in biology are probing deeper and deeper into biological phenomena. These so-called high-throughput technologies (measuring thousands of systems parameters in a single experiment) are heralding a paradigm shift (a) from traditional hypothesis-driven to data-driven research in molecular biology and (b) to a systems or systemic, as opposed to reductionistic, approach, attempting to model entire systems in order to understand study their holistic properties and dynamic properties. However, the noisy and high-dimensional data sets generated by these methods present considerable analytical and computational challenges. This study addresses this problem by analyzing high-dimensional gene expression data obtained from DNA microarray experiments investigating cancer. DNA microarrays are a high-throughput technology facilitating the simultaneous measurement of activity and interaction of thousands of genes in a single experiment [1]. This technology has led to the discovery of new biomarkers for disease diagnosis and prognosis, promoted the development of novel drugs for cancer therapy, and has provided new insights into the genesis and progression of multiple types of cancer.

Because of its importance to diagnostic and prognostic analysis, automated classification has attracted considerable interest in the context of microarray data analysis. For example, cancer types could be successfully classified based on the specific expression signatures [2,3,4]. However, microarray data classification presents substantially new challenges. First, microarray data exhibit high levels of noise due to various sources of systematic and random errors, including missing values. Second, microarray data are beset by a double 'curse' consisting of *high dimensionality* and *data set sparsity* [5]. Such data usually contain few (in the order of 10^2) observations (samples) and many (in the order of 10^4) parameters (genes). Many genes contain redundant or irrelevant information. Further, many data sets contain a relatively high number of classes but few cases per class. The curse of dimensionality in microarray data is commonly addressed by feature selection and dimension reduction techniques. However, the number of remaining genes that are significantly differently expressed in different classes can still be immense compared to the relatively small number of cases per class. This poses severe problems to an inductive learning of a classification function from such data. A desirable solution to the dimensionality problem would be to increase the number of cases. However, this is often not feasible because of (*i*) the limited number of available patients or specimens, and (*ii*) the relatively high costs of microarray experiments in terms of money and time.

Confidence values convey information about the class membership of the cases and are used in model fusion approaches such as bagging and boosting. Bagging involves a repeated random sampling (with replacement) of the original training set to generate *m* bootstrapped data sets. In noisy bagging, the bootstrapped data sets are disturbed by random noise and have shown to improve the generalization ability of ensembles of neural networks [6]. Adaptive boosting (Adaboost) creates several different models and combines their predictions using a weighted voting scheme (e.g., majority voting). Here, *k* different training set replicas are sampled adaptively (with non-uniform sampling probabilities and replacement) from the learning set. The predictions of the combined model are generated using a weighted voting scheme. The adaptive sampling procedures increase the probability of a hard-to-classify case to be sampled based on the performance of the classifier in the previous iteration. Cases that are most often misclassified are assigned an increased probability for being sampled in the next round.

The study presented in this paper is necessarily and intentionally comprehensive as it attempts to expose and discuss various elements of a full methodology rather than only a single method. As a consequence, not all parts of the presented methodology are discussed and evaluated in detail. It is our plan to explore and investigate different aspects of this comprehensive methodology in more detail in the future. This paper focuses on how the confidence values computed in the learning phase can be used for optimization of a single classifier in the context of the small-*n*-large-*p* problem. We present a model that calibrates its confidence in classification processes. In the learning phase, the model generates artificial training data as a function of its confidence in previous decisions and uses these data for calibrating its confidence in subsequent classifications. These artificial data play a pivotal role in determining the model's form or structure and performance, and have led to the model's name. (The Greek word *plasma* means 'to be formed' or 'molded'.)

2 Confidence in Classification

In practical applications without precise definition of costs for false positives and false negative classifications, exact characterization of the reward and penalty associated with a given prediction is not possible. Information-theoretic approaches typically translate a classifier's confidence into reward and penalty scores. This is based on the following rationale: *Misclassification with high confidence is more severe than misclassification with low confidence*. Let C be the real class associated with case $\mathbf{x}$ and $\hat{p}(C|\mathbf{x})$ be the model's confidence that the case belongs to C. Then, a *reward-penalty function* $R(\hat{p})$ can be defined as follows [7].

$$R(\hat{p}) = 1 + \log_2 \hat{p}(C \mid \mathbf{x}) \tag{1}$$

Key properties of this function are that it is not symmetrical with respect to rewards and penalties, and that the discrepancy becomes larger for higher confidence values. Extreme confidences that entail a misclassification, $\hat{p}(\neg C|\mathbf{x} \in C) = 1$, lead to a penalty of $-\infty$, whereas the maximum reward for a correct classification is only 1. To avoid extreme confidences, we force the minimum and maximum confidences towards $\hat{p}_{\min} = 0.5/(N+1)$ and $\hat{p}_{\max} = (N+0.5)/(N+1)$, where N is the number of cases in the learning set [8]. For example, if a training set contains $n = 100$ cases, then the maximum confidence for a single classification is $\hat{p}_{\max} = 0.995$.

Korb *et al.* showed that if a model predicts a class with probability $\hat{p}$, and the real class will actually occur with frequency $f = \hat{p}$, then this model can be expected to obtain the highest reward [7]. Such a model is called *perfectly calibrated*. Miscalibration measures how much the probability estimates deviate from the *frequency of truth of events* [7]. Korb *et al.* proposed to measure a model's miscalibration by partitioning the range of a model's confidence values into cells, so that each cell contains at least ten confidence values and as few as possible above ten [7]. Then, the frequency of truth within the cells is compared with the confidence values that they contain. The miscalibration is defined as follows:

$$\textit{miscalibration} = \sqrt{\sum_{i=1..n} \sum_{j=1..m} \frac{\left(\sum_k f_{ik} m^{-1} - \hat{p}_{ij}\right)^2}{m-1}} \tag{2}$$

where n is the number of partitions of the range of confidence values; m is the number of confidence values in the i^{th} cell; k is the index of confidence values in the i^{th} cell; f_{ik} is 1 if the k^{th} prediction in the i^{th} cell is correct, 0 otherwise; and $\hat{p}_{ij}$ is the j^{th} confidence value in the i^{th} cell. Korb's measure of miscalibration can be used to derive a measure to quantify the model's *timidity* by considering only those confidence values $\hat{p}_{ij}$ that lead to a correct classification.

3 Jittering

Jittered data (jitter) refers to data that is deliberately corrupted by artificial noise. Several studies have demonstrated that the generalization ability of neural networks can be significantly improved by injecting jitter into the data, particularly when the size of the training set is small [9,10]. The concept of jittering has been successfully applied to tasks that are characterized by the curse of dimensionality. Van Someren *et al.* followed this strategy to model robust genetic networks from time-course gene

expression data [11]. Provided that the noise amplitude is small, jittering is equivalent to Tikhonov regularization [12]. Adding jitter can lead to an increased classification error in the training phase, but to a decreased error in the test phase. Chawla *et al.* investigated classification problems that involve imbalanced classes, i.e., data sets with classification categories that are not (approximately) equally balanced [13]. They presented the method of SMOTE, an approach for over-sampling the minority class using synthetic training cases. The generation of these synthetic cases is effectively a jittering approach that improves the classification performance in the context of skewed class distributions [13]. Empirical results have shown that SMOTE performs better than over-sampling with replacement of the minority class; it also performs better than under-sampling of the majority class [13].

Consider the classification problem that involves the learning of the mapping from a vector $\mathbf{x}$ to a class label y, where $\mathbf{x}$ is a p-dimensional vector of gene expression data and y is a discrete variable (e.g., a cancer class). The jittered version of this vector is $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\varepsilon}$, and $\tilde{\mathbf{x}}$ has class label y. The noise vector $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p)$ has a distribution of mean $m_{\boldsymbol{\varepsilon}}$ and standard deviation $s_{\boldsymbol{\varepsilon}}$.

For cancer microarray data sets, we often observe that genes exhibit a similar expression profile in samples of the same cancer type. We propose that the magnitude of the noise level takes into account the magnitude of the actual expression levels; otherwise, the class-discriminatory effect of low-level expressed genes might vanish.

Let the i^{th} original expression profile be $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$. The jittered version of this vector, $\tilde{\mathbf{x}}_i$, is given by Equation 3 as follows:

$$\tilde{\mathbf{x}}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, ..., \tilde{x}_{ip}) \text{ , with } \tilde{x}_{ik} = (\beta_{ik}\alpha_{ik}\rho_{ik} + 1)x_{ik} \tag{3}$$

where β_{ik}, α_{ik}, and ρ_{ik} are random variables, and $\beta_{ik} \in \{1, 0\}$, $\alpha_{ik} \in \{-1, 1\}$, and $\rho_{ik} \in [\rho_{\min}, \rho_{\max}]$, with $\rho_{\min}, \rho_{\max} \in]0, 1[$. The values 1 and 0 are equally likely for β_{ik}, so that β_{ik} controls the number of variables (i.e., genes) to be jittered. If $\beta_{ik} = 0$, then the k^{th} component of the i^{th} jittered expression profile is identical to the k^{th} component of the i^{th} original profile. If $\beta_{ik} = 1$, then the k^{th} component of the i^{th} jittered expression profile is a jittered version of the k^{th} component of the i^{th} original profile. This noise is determined by both α_{ik} and ρ_{ik}.

4 Calibration Using Jittering

Equation 3 provides a general means for generating a jittered expression profile. When adding jittered duplicates to a data set, three questions need to be answered: (1) How many jittered cases should be added?, (2) Which cases are candidates for jittering?, and (3) Which distribution (type and parameters) of distortion noise should be chosen?

We can distinguish two situations: (*i*) all confidence values within a cell lead to a correct classification, and (*ii*) at least one confidence value leads to a misclassification. Consider the latter case first. If a cell contains a value that leads to a misclassification, then we decide that the respective training case should be jittered. If all confidence values in a cell lead to a correct classification, and if all cases were classified with confidence 1, then the contribution to the timidity component of the miscalibration would be zero, but such extreme confidences are

not allowed (see above). Suppose that each probability in a cell is relatively high, for instance, each confidence is $\hat{p} = 0.95$. Then this cell's contribution to (the square root of) the timidity is $10\times(1 - 0.95)^2/9 = 0.003$, which may be deemed sufficiently small. If the confidence values are all relatively small, e.g., 0.70, then the cell's contribution to (the square root of) the timidity is 0.10, which can be considered rather large. The confidence values might be too small to be judged valuable. Therefore, if the contribution to timidity in a cell is greater than a small positive threshold δ, then *all* respective training cases within this cell should be jittered.

Neural plasma is based on the probabilistic neural network (PNN) [14]. Figure 1 depicts the topology of neural plasma, illustrated for two classes of three cases each and two test cases.

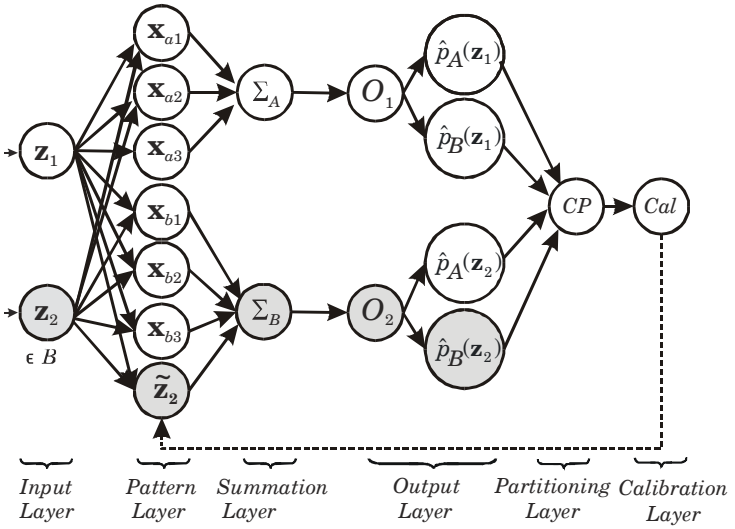

Fig 1. The topology of neural plasma.

The first part of neural plasma – input, pattern, summation, and output layer – is identical to the basic PNN. The difference consists in the *partitioning layer* and the *calibration layer*. The cell partitioning neuron *CP* receives the computed class posteriors and partitions them into cells of approximately equal size in such a way that each cell is guaranteed to contain at least ten elements and as few as possible above that number. The calibration neuron *Cal* determines the model's calibration with respect to boldness and timidity and determines which cases are candidates for jittering. Then, the calibration neuron generates jittered cases according to Equation 3 and feeds these cases back to the pattern layer. Consider the shaded parts in Figure 1. The case $\mathbf{z}_2$ is a member of class *B*. This case is assigned to one of the classes *A* or *B*, depending on which estimated class posterior is the highest. The neuron O_2 outputs these posteriors for $\mathbf{z}_2$. Suppose that $\hat{p}(B \mid \mathbf{z}_2)$ is the highest, i.e., leading to a correct classification, but $\hat{p}(B \mid \mathbf{z}_2)$ is still too small with respect to the calibration criterion. Or suppose that $\hat{p}(B \mid \mathbf{z}_2)$ is *not* the highest, leading to a misclassification of $\mathbf{z}_2$. In both cases, the calibration layer will generate a jittered duplicate of this case, $\tilde{\mathbf{z}}_2$, and add it to the pattern layer. We propose a *k*-fold sampling procedure with the sampling methodology as shown in Figure 2.

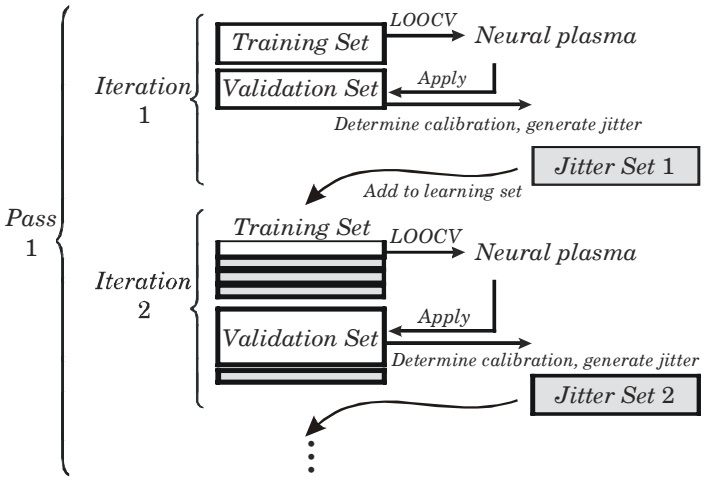

Fig. 2. One pass in the cross-validation procedure.

The learning set is randomly split in half into a training set and a validation set. Using the training set in leave-one-out cross-validation (LOOCV), the model determines the optimal kernel bandwidth. To classify the cases of the validation set, neural plasma uses that bandwidth that produces the smallest LOOCV error in the training set. Based on the performance on the validation set, the model determines its miscalibration. Based on the miscalibration, neural plasma generates jittered data. For each cell, the candidate cases for jittering are determined as follows. If at least one confidence value leads to a misclassification, then the misclassified cases are jittered. Otherwise, if all confidence values entail a correct classification, but the contribution to the timidity in a cell is greater than the threshold $\delta = 0.01$, then *all* cases in this cell are jittered.

The amount of jittered data in the i^{th} iteration represents the i^{th} *jitter set* that is added to the learning set in the $(i+1)^{\text{th}}$ iteration. Here, the learning cases are randomly mixed with the jittered cases of the previous iteration. The learning set for iteration #2 comprises now the original learning cases from iteration #1 plus the jittered cases.

In iteration #2, the model constructs the training and the validation set in such a way that they both comprise roughly the same number of cases. The jittered cases have a three times higher chance of being sampled for the training set than for the validation set. Using the training set again in LOOCV, the model optimizes the bandwidth and classifies the cases of the validation set. Again, depending on miscalibration, the model generates jittered data. The jitter set resulting from iteration #2 is mixed with the learning set and split into a training and a validation set for the next iteration. As before, jittered cases have a three times higher chance of being sampled for the training set than for the validation set. With an increasing number of iterations, both the training set and the validation set grow in size. The unequal sampling probability for jittered and original cases to be selected for the sets guarantees that the model is trained, relatively, on more artificial data and validated on more original data.

Consider Figure 2 and suppose that the depicted iterations are repeated, with the test set being the same. One *pass* encompasses n iterations with an *identical* test set. The performance – both on the test and the validation set – can vary in the iterations, because both the generation and the sampling of the jittered data are stochastic. After multiple passes have been performed, one model emerges with the smallest miscalibration. Let the number of passes be m. For example, if the model's miscalibration in the 7^{th} iteration in the 10^{th} pass is smaller than the miscalibration of the remaining $(m \times n - 1)$ models, then this model is selected. The training and the validation set – including the jittered data – of this model are merged to one set, the *best jitter-inflated set*. The entire procedure involving m passes of n iterations represents one fold in a k-fold cross-validation. Neural plasma uses the best jitter-inflated set to classify the cases of the test set of the k^{th} fold. For the present study, neural plasma uses $m = 20$ passes with $n = 10$ iterations each.

There exists a trade-off between too little and too much noise. In general, too few jittered cases will not have the desired regularization effect, whereas too many will increase the computational time and, more importantly, result in a 'blurring' of the data set, i.e., previously separated classes may become overlapping. The effect of the jittered cases will also depend on the characteristics of the data set at hand, for example, on the amount of measurement noise that the data set already contains. It has been suggested to determine the type of the noise distribution and the respective parameters using cross-validation procedures [9]. For example, ten-fold cross-validation can be repeated with different choices for these settings (e.g., uniform sampling of ρ_{ik} from (0, 0.05], (0.05, 15.0], etc.), and those parameters that provide for smallest mean classification error are considered optimal for the data set at hand. In the present study, we found that a uniform sampling of ρ_{ik} from (0.15, 0.25] provides for an acceptable trade-off between too little and too much noise for the three data sets investigated.

5 Materials and Methods

The experiments in this study comprise three well-studied, publicly available microarray data sets: (*i*) the NCI60 data set comprising gene expression profiles of 60 human cancer cell lines of various origins [2]. The data set contains 60 cases from nine cancer classes and 1,405 genes. The NCI60 data set is further pre-processed using principal component analysis and the first 23 'eigengenes' explaining over 75% of the total variance are selected. (*ii*) The ALL data set represents the expression profiles of 327 acute lymphoblastic leukemia samples [4]. This data comprises ten classes and the expression profiles of a total of 12,600 genes. (*iii*) The GCM data set contains 16,063 gene expression profiles of 198 specimens (190 primary tumors and eight metastatic samples) of predominantly solid tumors of 14 cancer types [15].

For the ALL and the GCM data set, feature selection was performed as follows. Based on the learning set L_i only, we determined the signal-to-noise (S2N) weight for each gene with respect to each class [16]. Then, we performed a permutation test involving a random permutation of the class labels and the re-computation of the S2N weights. This procedure was repeated 1,000 times to assess the significance of the signal-to-noise weights for the unpermuted class labels [17]. Based on the S2N

weights and associated *p*-values, we selected the top-ranking genes per class; all other genes were discarded from further analysis. This approach was repeated ten times to generate ten pairs, each consisting of a filtered learning set L_i and a test set T_i with the corresponding genes. Information contained in the test sets was not used in any way for feature selection.

Neural plasma and boosting are related approaches, but there exist two fundamental differences: (*i*) Neural plasma is trained on jittered duplicates, and (*ii*) boosting is a multi-model approach for generating an ensemble of classifiers. Less robust or 'brittle' classifiers such as decision trees often benefit from boosting [18]. We compare neural plasma with PNN and boosted decision trees C5.0.

The performance of the models is assessed in a 10-fold repeated random sampling procedure. In short, the procedure produces $i = 1..10$ pairs of learning sets L_i and test sets T_i with original data. L_i comprises ~70% and T_i comprises ~30% of the original cases. Notice that the learning and test cases are identical for all models, and the test sets are never used for model selection or feature selection to avoid feature selection bias [19].

6 Results

Table 1 shows the 95%-confidence intervals for the prediction accuracy of the models, averaged over the ten test sets.

Table 1. 95%-confidence intervals for the true average prediction accuracy (in %).

	NCI60	*ALL*	*GCM*
Neural plasma	79.3 ± 6.4	77.9 ± 2.4	78.9 ± 3.6
PNN	76.7 ± 6.7	77.4 ± 2.4	79.6 ± 3.6
2-fold boosted C5.0	64.3 ± 7.6	68.6 ± 2.7	64.5 ± 4.3
3-fold boosted C5.0	58.5 ± 7.8	71.0 ± 2.7	63.0 ± 4.3
4-fold boosted C5.0	62.4 ± 7.6	72.6 ± 2.6	66.5 ± 4.2
5-fold boosted C5.0	62.4 ± 7.6	72.5 ± 2.6	68.0 ± 4.2

There exist only relatively small differences between neural plasma and PNN for the ALL and GCM data sets. However, on the data set comprising the smallest number of cases, NCI60, neural plasma achieved a remarkably higher accuracy than PNN. Next, we assess whether the differences in performance between neural plasma and the best-boosted trees are statistically significant. Let p_{Ai} be the observed proportion of test cases misclassified by model A and let p_{Bi} be the observed proportion of misclassified test cases by model B during the i^{th} cross-validation fold. Assume that in each fold N cases are used for learning and M cases are used for testing. The statistic for the *variance-corrected resampled paired t-test* is then given by Equation 4 as follows [20].

$$T_c = \frac{\overline{p}}{\sqrt{(k^{-1} + M/N)s^2}} \sim t_{k-1} \tag{4}$$

Empirical results show that this corrected statistic drastically improves on the standard resampled *t*-test with respect to Type I error [20].

The difference in performance on NCI60 between neural plasma and 2-fold boosted C5.0 is significant ($P = 0.03$). The difference in performance on GCM between neural plasma and 5-fold boosted C5.0 is significant ($P = 0.003$). However, the difference in accuracy on the ALL data set ($77.9 \pm 2.4\%$ for neural plasma vs. $72.6 \pm 2.6\%$ for 4-fold boosted C5.0) is not significant ($P = 0.06$).

7 Discussion and Conclusions

Neural plasma methodology presented in this study involves several elements. Given the space limitations, not all aspects of this methodology are discussed in exhaustive detail. The neural plasma approach distinguishes itself from other neural networks with respect to two critical aspects. First, in contrast to multilayer perceptron and similar techniques, neural plasma does not attempt to mimic the topology (neurons, synapses, activation potentials, etc.) of biological neural networks. Instead, it focuses on characteristics related to intelligent behavior and reasoning, such as *timidity* (and its opposite: *boldness*). Thus, neural plasma is potentially useful for classification problems that require explicit representation of these notions in the decision process. Future work on neural plasma will concentrate on further evaluating and interpreting these concepts in the context of decision and reasoning theory.

Second, neural plasma generates artificial training cases as a function of its performance and thereby increases the learning set artificially. Within the context of high-throughput applications on biology and biotechnology, this is a novel approach to tackling the dimensionality problem in classification problems. In contrast to our approach, the SMOTE algorithm by Chawla *et al.* generates synthetic training cases only for the minority class [13].

How could the model's calibration be computed more effectively and efficiently? Neural plasma determines the miscalibration as a function of the frequency of truth in the cells. However, the partitioning into cells, each containing approximately ten elements, is based only on the empirical results by Korb *et al.* [7]. Which cross-validation procedure should be chosen, and which sampling procedure for the original and jittered cases should be adopted? The jittered data were sampled for the training set with a three times higher probability than the original cases, so that the model is trained on more artificial data and validated on more original data; however, other sampling ratios need to be investigated. What is considered a 'timid' classification is clearly context-dependent and can be controlled by the threshold δ, which was set to 0.01 in the present study. Future work will focus on the model's sensitivity to overfitting and on how these empirically determined parameters could be optimized.

In summary, we believe that the neural plasma methodology represents an interesting framework for exploring classification tasks in the context of faculties such as timidity and boldness, which are inherent factors of human reasoning. The evaluation presented in this study focuses on a limited set of criteria of a more comprehensive framework. As such, this study is seen as a first step in presenting and exploring this framework. Future work will explore different aspects in more detail.

8 References

[1] Brown, P.O. and Botstein, D. (1999) Exploring the new world of the genome with DNA microarrays. *Nat. Gen.* **21**(1): 33–37.

[2] Ross, D.T., Scherf, U., Eisen, M.B., *et al.* (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Gen.* **24**(3):227–235.

[3] Alizadeh, A.A., Eisen, M.B., Davis, R.E., *et al.* (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403**:503–511.

[4] Yeoh, E.J., Ross, M.E., Shurtleff, S.A., *et al.* (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* **1**:133–143.

[5] Somorjai, R.L., Dolenko, B., and Baumgartner, R. (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics* **19**(12):1484–1491.

[6] Raviv, Y., and Intrator, N. (1996) Boostrapping with noise: An effective regularization technique. *Connection Science* **8**(3–4):355–372.

[7] Korb, K.B., Hope, L.R., Hughes, M.J. (2001) The evaluation of predictive learners: some theoretical and empirical results. *Proc. 12th Europ. Conf. Machine Learning*, 276–287.

[8] Dowe, D.L., Farr, G.E., Hurst, A.J., Lentin, K.L. (1996) Information-theoretic football tipping. *Proc. 3rd Austr. Conf. Math. & Computers in Sport*, Australia, 233–241.

[9] Koistinen, P., and Holmström, L. (1992) Kernel regression and backpropagation training with noise. *Advances in Neural Inf. Proc. Sys.* **4**:1033–1039.

[10] Reed, R., Oh., S., Marks, R.J. (1992) Regularization using jittered training data. *Proc. Int. J. Conf. Neural Networks*, Baltimore MD, III147–III152.

[11] van Someren, E.P., Wessels, L.F.A., Reinders M.J.T, Backer, E. (2001) Robust genetic network modeling by adding noisy data. *Proc. IEEE Workshop on Nonlinear Signal and Image Processing*, Baltimore, Maryland.

[12] Bishop, C.M. (1994) Training with noise is equivalent to Tikhonov regularization. *Neural Computation* **7**:108–116.

[13] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002) SMOTE: Synthetic Minority Over-sampling Technique. *J. Art. Int. Res.* **16**:321–357.

[14] Specht, D.F. (1990) Probabilistic neural networks. *Neural Networks* **3**:109–118.

[15] Ramaswamy, S., Tamayo, P., Rifkin, R., *et al.* (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci. USA*. **98**(26), 15149–15154.

[16] Slonim, D., Tamayo, P., Mesirov, J., *et al.* (2000) Class prediction and discovery using gene expression data. *Proc. 4th Ann. Int. Conf. Comp. Mol. Biol.*, Tokyo, Japan, 263–272.

[17] Radmacher, M.D., McShane, L.M., Simon, R. (2002) A paradigm for class prediction using gene expression profiles. *J. Comp. Bio.* **9**(3):505–511.

[18] Duda R.O., Hart P.E., Stork D.G. (2001) *Pattern Classification*. 2nd ed., John Wiley & Sons, New York, p. 461.

[19] Ambroise, C. and McLachlan, G.J. (2002) Selection bias in gene extraction on the basis of microarray gene expression data. *Proc. Natl. Acad. Sci. USA* **98**:6562–6566.

[20] Nadeau, C., and Bengio, Y. (2003) Inference for generalization error. *Machine Learning* **52**:239–281.