Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 280-291

Extreme Data Mining: Inference from Small Datasets

R. Andonie

Răzvan Andonie
Computer Science Department
Central Washington University, Ellensburg, USA
and
Department of Electronics and Computers
Transylvania University of Braşov, Romania
E-mail: andonie@cwu.edu

Abstract: Neural networks have been applied successfully in many fields. However, satisfactory results can only be found under large sample conditions. When it comes to small training sets, the performance may not be so good, or the learning task can even not be accomplished. This deficiency limits the applications of neural network severely. The main reason why small datasets cannot provide enough information is that there exist gaps between samples, even the domain of samples cannot be ensured. Several computational intelligence techniques have been proposed to overcome the limits of learning from small datasets.

We have the following goals: **i.** To discuss the meaning of "small" in the context of inferring from small datasets. **ii.** To overview computational intelligence solutions for this problem. **iii.** To illustrate the introduced concepts with a real-life application.

1 Introduction

Small dataset conditions exist in many applications, such as disease diagnosis, fault diagnosis or deficiency detection in biology and biotechnology, mechanics, flexible manufacturing system scheduling, drug design, and short-term load forecasting (an activity conducted on a daily basis by electrical utilities). In this section, we describe a computational chemistry problem, review a class of neural networks to be used, and summarize our previous work in this area.

1.1 A Real-World Problem: Assist Drug Discovery

Current treatments for HIV/AIDS consist of co-administering a protease inhibitor and two reverse transcriptase inhibitors (usually referred to as combination therapy). This therapy is effective in reducing viremia to very low levels; however, in 30-50% of patients it is ineffective due to resistance development often caused by viral mutations. Due to resistance and poor bioavailability [1] profiles, as well as toxicity associated with these therapies, there is an urgent need for more efficient design of drugs.

We focus on inhibitors to the HIV-1 protease enzyme, using the IC_{50} as the target value. A detailed description of the problem, from a computational chemistry point of view, can be found in our papers [1–3]. The IC_{50} value represents the concentration of a compound that is required to reduce enzyme activity by 50%. A low IC_{50} value indicates good inhibitory activity. The available dataset consists of 196 compounds with experimentally determined IC_{50} values. Twenty of these molecules are used as an external test set after the training is completed. The remaining 176 molecules are used for training and cross-validation. Our practical goal is to predict the (unknown) IC_{50} values for 26 novel compounds which are candidates for HIV-1 protease inhibitors. We use two IC_{50} prediction accuracy measures: the RMSE (Root Mean Squared Error) and the Symmetric Mean Absolute Percentage Error (sMAPE).

[1] Bioavailability is the rate at which the drug reaches the systemic circulation.

The easiest way to represent a molecule is by a vector of features (molecular descriptors) which may be both topological indices and physico-chemical properties. The resulting features may be numerous and inter-correlated. Using the complete set of descriptors may lead to overfitting, if it is too large compared to the size of the training set. We select 35 molecular descriptors based on their contribution to molecular entity.

Although biological activity data has been obtained for many more chemical structures at various pharmaceutical companies and academic laboratories, they are not available in the public domain. Actually, most classical studies for a specific enzyme system have been performed on small datasets, due to limited experimentally determined biological activity values in the public domain. The dimensionality (the number of physico-chemical features) characterizing these molecules is relatively high. Our dataset shares these undesired characteristics: it is small, with relatively many features, and highly overlapping.

1.2 Prerequisites: FAMR for IC_{50} prediction

The FAMR is a Fuzzy ARTMAP (FAM) incremental learning system used for classification, probability estimation, and function approximation. We review the basic FAMR notation. Details can be found in [4].

A FAM consists of a pair of fuzzy ART modules, ART_a and ART_b, connected by an inter-ART module called Mapfield. The fuzzy ART_a module contains the input layer, F_1^a, and the competitive layer, F_2^a [5]. A preprocessing layer, F_0^a, is also added before F_1^a. The ART modules create stable recognition categories in response to arbitrary sequences of input patterns. The ART_a and ART_b vigilance parameters, ρ_a and ρ_b, control the matching mechanism inside the modules.

During learning, the Mapfield weights are updated: the strength of the weight projecting from the selected ART_a category to the correct ART_b category is increased, while the strengths of the weights to other ART_b categories are decreased. A Mapfield vigilance parameter ρ_{ab} calibrates the degree of predictive mismatch necessary to trigger the search for a different ART_a category. If the weight projecting from the active ART_a category through the Mapfield to the active ART_b category is smaller than ρ_{ab} (vigilance test), then the system responds to the unexpected outcome through the so-called *match tracking*. This triggers an ART_a search for a new input category. After choosing an ART_a category whose prediction of the correct ART_b category is strong enough, match tracking is disengaged, and the network is said to be in a resonance state. In this case, Mapfield learns by updating the weights w_{jk}^{ab} of associations between each j-th ART_a category and each k-th ART_b category.

The FAMR uses the following iterative updating scheme:

$$w_{jk}^{ab(new)} = \begin{cases} w_{jk}^{ab(old)} & \text{if } j \neq J \\ w_{JK}^{ab(old)} + \frac{q_t}{Q_J^{new}}\left(1 - w_{JK}^{ab(old)}\right) & \\ w_{Jk}^{ab(old)}\left(1 - \frac{q_t}{Q_J^{new}}\right) & \text{if } k \neq K \end{cases} \tag{1}$$

where q_t is the relevance assigned to the tth input pattern ($t = 1, 2, \ldots$) and $Q_J^{new} = Q_J^{old} + q_t$. The *relevance* q_t is a real positive finite number directly proportional to the importance of the experiment considered at step t. This w_{jk}^{ab} approximation is a correct biased estimator of the posterior probability $P(k|j)$, the probability of selecting the k-th ART_b category after having selected the j-th ART_a.

FAM (and FAMR) networks map subsets of $\mathbb{R}^n$ to $\mathbb{R}^m$ and can be used for function approximation. The FAM has been proven to be a universal function approximator [6]. We use the FAMR to predict functions that are known only at a certain number of points. More specifically, we predict IC_{50} values.

1.3 Our previous work

The present paper is based on a sequence of results, each describing new computational intelligence tools for biological activity (IC_{50}) prediction. In [7], we investigated the use of a fuzzy neural network

(FNN) for (IC_{50}) prediction. In [1] and [2], we improved this model by adding a two-stage Genetic Algorithm (GA) optimizer: the first for selecting the best subset of features and the the second for optimizing the FNN parameters. We will refer to this GA-optimized FNN as FS-GA-FNN.

In [8] we also focused on the IC_{50} prediction task, using the FAMR model. During the learning phase, each sample pair is assigned a relevance factor proportional to the importance of that pair. The prediction method consists of two stages. First, GA-optimization incorporating cross-validation is used to modify the training dataset. This modification consists of finding the best relevances for the data, according to some fitness criterion. The fitness criterion measures the FAMR IC_{50} prediction accuracy for a given training/validation dataset with given relevances. In stage two, the final FAMR is obtained by training it using the dataset with optimized relevances. In other words, stage one improves the generalization capability of the FAMR which will be obtained in stage two. We will refer to this model with GA-optimized relevances as GA-FAMR.

We compared the GA-FAMR and the Ordered FAMR (a FAMR algorithm which optimizes the order of training data presentation) in [9]. Both methods compensate for insufficient training data by additional optimizations. A trade-off between computational overhead and generalization capability is obtained.

Recently, we performed rule extraction from the trained FAMR model [10]. We post-processed the set of generated rules in order to improve generalization. We eliminated overfitting by heuristic generalization of rules and by adding new rules. This method proved to be efficient for small training sets.

The present paper results from several invited talks [9,11,12]. In Section 2, we discuss the capability of neural network to infer from rare samples. Section 3 describes two methods for neural training on small datasets. After presenting and discussing experimental results in Section 4, we conclude with our final remarks (Section 5).

2 Neural Networks Trained on Small Datasets

We aim to discuss the difficulties of inferring a Neural Network (NN) from small, or non-representative, training sets. We will look closer at the overfitting and generalization aspects of the network. But first, we need to define formally what we understand by "small training set".

2.1 What is "small"?

In many multivariable classification or regression (e.g., estimation or forecasting) problems we have a training set $T_p = (x_i, t_i)$ of p pairs of input/output vector $\mathbf{x} \in \Re^{\mathbf{n}}$ and scalar target t, and the unfortunate circumstance that T_p is small. The VC (Vapnik-Chervonenkis) dimension is a measure of the capacity of a classificator, defined as the cardinality of the largest set of points that the algorithm can shatter. According to Vapnik:

> "For estimating functions with VC dimension h, we consider the size p of data to be small if the ratio p/h is small (say $p/h < 20$)" [13].

The main reason why small datasets cannot provide enough information is that there exist gaps between samples, even the domain of samples cannot be ensured. For a small training set, even a simple neural network can have a complexity (e.g., number of connections/parameters) that is comparable to, or exceeds, the training size p. In such a case, we may expect to fit T_p very well. However, we can also expect poor generalization to new data identically distributed as the data in T_p. In effect, the VC dimension is too large relative to the size of the training set.

A completely different definition for “small” sets comes from algorithmic information theory. The Kolmogorov complexity of an object such as a string is a measure of the computational resources needed to specify the object. More formally, the complexity of a string is the length of the string's shortest description in some fixed universal description language. It can be shown that the Kolmogorov complexity of any string cannot be too much larger than the length of the string itself. A string is considered to be "random" if the length of the shortest problem that generates the string is the same as that of the string itself. Strings whose Kolmogorov complexity is small relative to the string's size are considered to have small information content [14]. Kolmogorov's complexity has been studied in the context of inductive inference [15, 16]. It is an open problem how to relate the Kolmogorov complexity of a training set and the generalization capability of the inferred NN.

We will use a simplified definition: A training set is small if p and n are comparable. In accordance with this definition, the training set for our chemistry problem is small.

There is no universally optimal solution to the problem of inferring from small datasets. We only can state some very general principles one can follow. For instance, one principle would be to extract from the training data the maximum useful information available. If not done thoroughly, this may lead to overfitting, and/or to a time-prohibitive training process. A principle for controlling the generalization capability of a NN is to design a network with much fewer connections than the size of the training set.

To overcome the limits of learning from small datasets, several general techniques have been proposed [17–26]: generate artificial training samples, feature selection, and parameter fine-tuning of the inferred model.

A special learning method designed for small training sets is *adaptive learning with domain range expansion*. In this case, additional information is used to dynamically improve training. Such an approach is, for instance, the Central Location Tracking method [25, 26]. This algorithm attempts to explore the predictive information through the generation of trend value of each datum. The extra information extracted from the data trend stabilizes the learning task and improves the derived knowledge from the occurrence of the latest data. The domain range is expanded to obtain the probable change of the small training data behavior.

The choice of specific technique is domain dependent. In computational chemistry, only feature selection and parameter fine-tuning have been used [27–29]. It is very difficult to generate artificial samples because, most probably, they will not physically exist.

2.2 Overfitting vs. generalization

Inference is based on a strong assumption: using a *representative* training set of samples to infer a model. In this case, we select a subset of the population, perform a statistical analysis on this sample, and use these results as an approximation to the desired statistical characteristics of the population as a whole. The more representative the sample, the larger our confidence that the statistical results obtained by using this sample are indeed a good approximation to the desired population statistics. We gauge the representativeness of a sample by how well its statistical characteristics reflect the statistical characteristics of the entire population. Many standard techniques may be used to select a representative sample set [30]. However, if we do not use expert knowledge, selecting the most representative training set from a given dataset was proved to be computationally difficult (NP-hard) [31]. The problem is actually more difficult, since in most applications the complete dataset is unknown or too large to be analyzed. Therefore, we have to rely on a more or less representative training set.

Another problem may arise from the training process itself. Especially in cases where learning was performed too long or where training the training samples are rare, the inferred model may adjust to very specific random features of the training data, that have no causal relation to the target function. In this process of *overfitting*, the performance on the training examples still increases while the performance on unseen data becomes worse (the generalization performance is poor).

In NN learning, overfitting generally occurs when excessive number of neurons is generated; the network overestimates the complexity of the problem and it cost more resources to train and implement.

There are three major strategies to avoid overfitting:

1. **Before learning**. Before being used, training samples are pre-processed, or new training samples are artificially created. A widely used before learning technique is to artificially extend the training set by introducing new training samples with additive noise [32–34]. It helps to enhance the generalization performance, speed up the training algorithm, and reduce the possibility of local minima entrapment [33–36].

2. **After learning**. The network is trained (with possible overfitting) and processed afterwards. Such techniques include *pruning*, *weight sharing*, *weight decay*, *ensemble neural networks*, and *complexity regularization* [35, 37, 38]. Pruning is the process of eliminating nodes and connections from the trained network. The reduced size network has to be sometimes retrained. NN pruning algorithms have practically developed for all major NN architectures [39].

2.3 How to detect overfitting

Beside preventing overfitting, a major question is how to detect it. It is desirable to have a measure that can quantify underfitting or overfitting of a network on a given learning problem. We do have again two general strategies: before and after learning.

The most common after learning technique is to perform learning/validation iteratively and optimize the learning/validation generalization error by adjusting the parameters and/or architecture of the network. Several constructive/destructive algorithms were adopted to incrementally increase or decrease the parameter to be optimized [40]. During the constructive/destructive process, cross-validation is commonly used to check the network quality and the design parameter is chosen using early stopping [41]. The training data is usually divided into two independent sets: a training set and a validation or testing set. Only the training set participates in the NN learning, and the validation set is used to compute a validation error, which approximates the generalization error. The inferred NN performance during training and validation is measured, respectively, by training error E_{train} and validation error E_{valid} presented. Once the validation performance stops improving as the target parameter continues to increase, it is possible that the training has begun to fit the noise in the training data, and overfitting occurs. Therefore, the stopping criterion is set so that, when E_{valid} starts to increase, or equivalently, when E_{train} and E_{valid} start to diverge, it is assumed that the optimal value of the target parameter has been reached [36]. Cross-validation + early stopping are the common techniques used in finding optimal network structure up to date. An alternative to cross-validation is bootstrapping.

More flexible stopping criteria based on early stopping were proposed by Prechelt [41]. It helped the users to choose stopping criterion in a systematic and automatic way, based on efficiency, effectiveness, or robustness. Liu *et al.* have introduced an algorithm which, on a given NN is able to recognize the occurrence of overfitting by examining the training error without using a validation set [36]. The algorithm also shows where the recycling of the training samples can be safely stopped so that the optimal structure of the NN is found. A signal-to-noise-ratio figure (SNRF) is defined to measure the goodness-of-fit using the training error. Based on the SNRF measurement, an optimized approximation algorithm is proposed to avoid overfitting in function approximation.

An open problem is how to detect before learning the generalization capability, without even knowing the NN to be used. In this case, one should be able to determine the generalization capability of a given training set before using it! For instance, we should determine if a training set is sufficiently smooth and covers sufficiently well the input space in order to produce a reasonably good approximation of an unknown function. Such a regression problem depends on the quality of available samples. Can we determine if the training set is good enough for being used? Can we do this independent of the NN model?

3 Two Efficient Methods

We will illustrate the concepts introduced in Section 2 with two FAMR methods which work well with small training sets. Since the methods have been previously described in [9], we will only review them here.

3.1 The GA-FAMR

The relevances attached to the input data are considered as adaptive parameters to be optimized by a GA.

The GA-FAMR operates on an initial population of relevance vectors. Each relevance vector has a single relevance associated with a specific training datum in accordance with the FAMR. Because the relevance of specific data is not known beforehand, this population must be optimized using the following GA:

Step One. Initialize a population of Pop_{size} chromosomes. Each chromosome is composed of N genes, where N equals the size of the training dataset. Each gene is a real value in the range (0, 10), defining the relevance of one of the training molecules.

Step Two. For each chromosome, train and validate the FAMR using cross-validation. Compute the fitness value of each chromosome: $Fit = 1/sMAPE$.

Step Three. Establish the next generation.

1. Find Fit_{low}, which is the smallest fitness value in the population.
2. Subtract Fit_{low} from the fitness value of each chromosome.
3. Sum the fitness values of all chromosomes to calculate the total fitness, Fit_{all}, of the population.
4. Divide each chromosome's fitness value by Fit_{all}.
5. Generate Pop_{size} new chromosomes to replace the current population. Each new chromosome is created by one of two methods: breeding or elitism.
    (a) BREEDING:
        i. For each child, two parents are selected according to the concept of the survival of the fittest.
        ii. Each parent is selected by first generating a random number, $0 < s < 1$.
        iii. Iterate through the chromosomes in the population. If $Fit \geq s$, the chromosome is selected. Else, subtract Fit from s, and continue to the next chromosome. The probability that a chromosome will be selected for reproduction at any given time is given by: $(Fit - Fit_{low})/(Fit_{all} - Fit_{low} * Pop_{size})$.
        iv. When two parents have been selected for each child, perform crossover to generate the new chromosome. For each child, one of two crossover methods is chosen with equal probability:
            A. For each gene, copy the genetic material from one or the other parent; the parent copied for each gene is selected randomly.
            B. Average the genes of the two parents. Because the effect of switching specific bits in a real value can be extremely unpredictable, it may be more effective to average two real values.
        v. Before the new child is introduced into the next generation, there is a 0.25 probability that it will undergo mutation in one of its genes, by randomly generating a new real value.

(b) ELITISM: At all times, eight global best chromosomes are retained as a possible source of members of the new generation. There is a 1/500 probability that a new chromosome is generated by selecting one of these elite, rather than by crossover of two members of the current population.

3.2 Ordered FAMR

For optimizing the FAM training data ordering, Dagher *et al.* [42, 43] and Tan *et al.* [44] have introduced efficient procedures. Essentially, the training data is preprocessed to identify a fixed order of pattern presentation. We refer to this procedure as the ordering algorithm. When the training input patterns are presented to the FAMR according to this fixed order, we obtain a FAMR with improved generalization capability.

Preprocessing consists of clustering input data. Each cluster center will be a molecule in the training set. The ordering of the training data is determined by the order in which the cluster centers are obtained. It is noteworthy that this clustering is different than the formation of ART_a categories, which is also a clustering of the same input dataset.

The ordering algorithm is controlled by a pre-defined parameter, n_{clust}, which is the number of input data clusters, and consists of the following three stages:

1. Determine the first pattern to be presented. This pattern corresponds to the first cluster center of the training data.

2. Determine the next $n_{clust}-1$ patterns to be presented. These patterns correspond to the next $n_{clust}-1$ cluster centers of the training data, and are identified through the Max-Min clustering algorithm [45].

3. Determine the order of the remaining patterns. These patterns are chosen according to the minimum Euclidean distance criterion from the n_{clust} centers defined in Stages 1 and 2.

Stage 1. We start with an M-dimensional input pattern $\mathbf{a}=(a_1,\cdots,a_M)$ and obtain 2M-dimensional input pattern $\mathbf{A}=(a_1,\cdots,a_M,1-a_1,\cdots,1-a_M)$ by complement coding [5].

Input pattern **a**, which maximizes the sum in eq (2), is selected as the first pattern to be presented. This pattern is also treated as the first cluster center of the training patterns.

$$\sum_{i=1}^{M}|a_{M+i}-a_i| \tag{2}$$

Stage 2. The next $n_{clust}-1$ input patterns are identified for presentation during network training. These patterns represent the next cluster centers of the training patterns. They are determined consecutively using the Max-Min clustering algorithm. In this stage, the Euclidean distances between the remaining input patterns and the existing cluster centers $\mathbf{a}^1,\cdots,\mathbf{a}^k$ $(k\le n_{clust})$ are computed. The minimum Euclidean distance between each remaining input pattern **a** and the existing cluster centers is identified: $d^{\mathbf{a}}_{min}=min\ dist(\mathbf{a},\mathbf{a}^j)$ $(1\le j\le k)$. The input pattern which maximizes $d^{\mathbf{a}}_{min}$ is selected as the next cluster center.

Stage 3. The presentation order of the remaining input patterns is determined by finding the minimum Euclidean distances between these patterns and the n_{clust} cluster centers. The whole procedure of Stage 3 is repeated until the order of all input patterns for the network training phase has been identified.

The value of n_{clust} not only influences the input data ordering, but also has a major impact on the number of ART_a categories created. Thus, n_{clust} controls the generalization capability of the network.

Successive optimization of relevances and ordering is not a good strategy. The two optimizations can possibly cancel each other out, since they may influence each other. Therefore, we do not optimize both

Table 1: Prediction performance analysis on the training set [9].

	FS-GA-FNN	**Standard FAMR**	**GA-FAMR**	**Ordered FAMR**
sMAPE	89.28	89.99	77.65	86.04
RMSE	1132.12	1401.94	1332.53	1366.04

Table 2: Prediction performance analysis on the test set [9].

	FS-GA-FNN	**Standard FAMR**	**GA-FAMR**	**Ordered FAMR**
sMAPE	111.91	99.01	105.17	84.51
RMSE	506.08	43.45	56.99	25.49

relevances and ordering for the same network. We will refer in the following to the Ordered FAMR - a FAMR with equal (not optimized relevances) and optimized training data ordering.

4 Experimental Results

In our experiments, all networks were trained with the same set of 176 molecules, using twenty-fold cross-validation. Thus, we improve the generalization performance on this small training set by introducing some computational overhead. In all experiments we used on-line (incremental) learning: the training set is processed only once.

When estimating the quality of a prediction model, the prediction accuracy obtained both for training data and new data is important. One is interested not only in how accurately the model approximates the learning data, but also how the model generalizes on new data. The test set, which is not used for training, consists of twenty molecules. This set is from a different group of molecules than the one used for training, making prediction more difficult.

We investigate the GA-FAMR and the Ordered FAMR. The results are compared to the standard FAMR model (with no optimizations and equal relevances), and to the FS-GA-FNN.

The parameters of the network are determined experimentally, and are fixed for all FAMR models considered. The ρ_a and ρ_b parameters control the number of generated FAMR categories. It is important to limit the number of categories to prevent overfitting. Maintaining constant FAMR parameters for all tested models simplifies comparison. For the standard FAMR, the number of ART_a categories is 13 and the number of ART_b categories is 8. The experimentally optimized number of ART_a categories is close to the number of scaffold subtypes, which is a significant match.

The statistical results for the test sets are in Tables 1 and 2. As expected, the optimized FAMR models adjust better than the standard FAMR to the training data (Table 1). Of the three FAMR models, the GA-FAMR adjusts best to the training data.

Does the GA-FAMR overfit? We may find the answer by analyzing the prediction performance for test data. From Table 2 we conclude that the Ordered FAMR improves the standard FAMR over the test set. The GA-FAMR appears to overfit the training data and has therefore a less performant generalization.

Overall, from Tables 1 and 2, we conclude that the Ordered-FAMR performs better than the other models.

For low IC_{50} values, all three FAMR models exhibit a similar prediction pattern and they clearly overpredict the target values of the test molecules, which is good in our particular application.

We have predicted the IC_{50} value of 26 novel potential inhibitors using all four models (see [3]). The FS-GA-FNN and the FAMR are two radically different neural paradigms. The training datasets are the same, but the number of descriptors is different: FAMR uses 35, while FS-GA-FNN uses a feature selected subset of 22 descriptors. For some of the novel molecules, all methods predicted very low IC_{50}

Table 3: GA-FAMR prediction performance analysis on the training and test sets for different number of GA generations [9].

	25 generations	**50 generations**	**100 generations**
Training sMAPE	87.78	86.48	84.66
Training RMSE	1389.54	1381.47	1360.18
Test sMAPE	95.56	101.42	106.20
Test RMSE	41.03	48.80	63.09

values. Since radically different methods indicate high inhibitory activity, these are the molecules we consider as excellent candidates for organic synthesis and further drug discovery.

It is interesting to analyze the way the GA optimization performs for different numbers of generations (Fig. 3). With an increasing number of generations, the network adjusts better to the training set, but it also reduces its generalization capability with respect to the test set. Thus, the number of generations controls overfitting. In our experiments (Tables 1 and 2), we have used 2000 generations and this explains the relatively poor generalization obtained. We could use less generations and thus improve generalization with the cost of adjusting less to the training data. To determine the optimal number of generations and establish the best trade-off between generalization and overfitting, we may use an early stopping technique, or Liu's *et al.* algorithm [36].

The generalization capability of the Ordered FAMR is good, but depends on an appropriate selection of the n_{clust} parameter, which is a weakness of this algorithm. The GA-FAMR is also a good choice, but early stopping should be used to avoid overfitting. The computational overhead of the two algorithms is insignificant when compared to the value of the results. A computationally intensive solution is acceptable because drug synthesis requires years of time and great expense. Therefore, obtaining an accurate prediction is more important than execution time.

5 Conclusions

We have discussed and illustrated how to infer from small datasets. We do not have a nice mathematical solution to the general problem of learning from small datasets. But why? Here is our answer: If the VC dimension is too large relative to the size of the training set and we do not have any information about the quality of our training data and how representative it is, then the problem is ill-posed. We only can state the general rule of thumb: From the available samples, extract maximum information, without overfitting. There is no free lunch and we have to balance overfitting and generalization.

Both presented techniques work well, but we may have a significant computational overhead, which can make our solution non-scalable. The paradox is that, in our computational chemistry problem, we do not need scalability, since we do not have enough data anyway! These methods may be used for similar application, whenever we have to infer from small training sets. This does not mean that we prefer small training sets, but that we have to adapt our methods to what is available.

Bibliography

[1] R. Andonie, L. Fabry-Asztalos, S. Abdul-Wahid, C. Collar, and N. Salim, "An integrated soft computing approach for predicting biological activity of potential HIV-1 protease inhibitors," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2006)*, Vancouver, BC, Canada, July 16-21 2006, pp. 7495–7502.

[2] L. Fabry-Asztalos, R. Andonie, C. Collar, S. Abdul-Wahid, and N. Salim, "A genetic algorithm optimized fuzzy neural network analysis of the affinity of inhibitors for HIV-1 protease," *Bioorganic and Medicinal Chemistry*, vol. 16, pp. 2903–2911, 2008.

[3] R. Andonie, L. Fabry-Asztalos, C. B. Abdul-Wahid, S. Abdul-Wahid, G. I. Barker, and L. C. Magill, "Fuzzy ARTMAP prediction of biological activities for potential HIV-1 protease inhibitors using a small molecular dataset," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 99, no. PrePrints, 2009.

[4] R. Andonie and L. Sasu, "Fuzzy ARTMAP with input relevances," *IEEE Transactions on Neural Networks*, vol. 17, pp. 929–941, 2006.

[5] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[6] S. Verzi, G. Heileman, M. Georgiopoulos, and G. Anagnostopoulos, "Universal approximation with fuzzy art and fuzzy ARTMAP," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '03)*, vol. 3, Portland, Oregon, 20-24 July 2003, pp. 1987–1992.

[7] R. Andonie, L. Fabry-Asztalos, C. Collar, S. Abdul-Wahid, and N. Salim, "Neuro-fuzzy prediction of biological activity and rule extraction for HIV-1 protease inhibitors," in *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'05)*, 2005, pp. 113–120.

[8] R. Andonie, L. Fabry-Asztalos, L. Magill, and S. Abdul-Wahid, "A new Fuzzy ARTMAP approach for predicting biological activity of potential HIV-1 protease inhibitors," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2007)*, I. C. S. Press, Ed., San Jose, CA, 2007, pp. 56–61.

[9] R. Andonie, "Inference from small training sets - a computational intelligence perspective," University of Ulster, Jordanstown, Nothern Ireland, United Kingdom, invited talk, June 2008.

[10] R. Andonie, L. Fabry-Asztalos, B. Crivat, S. Abdul-Wahid, and B. Abdul-Wahid, "Fuzzy ARTMAP rule extraction in computational chemistry," in *IJCNN'09: Proceedings of the 2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 2961–2967.

[11] R. Andonie, "Extreme data mining: Inference from small datasets," National University of Ireland, Maynooth, Ireland, invited talk, June 2008.

[12] ——, "How to learn from small training sets," Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno-Lugano, Switzerland, invited talk, September 2009.

[13] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 2000.

[14] J. L. Balcázar and R. V. Book, "Sets with small generalized Kolmogorov complexity," *Acta Inf.*, vol. 23, no. 6, pp. 679–688, 1986.

[15] A. Ambainis, "Application of Kolmogorov complexity to inductive inference with limited memory," in *ALT '95: Proceedings of the 6th International Conference on Algorithmic Learning Theory*. London, UK: Springer-Verlag, 1995, pp. 313–318.

[16] A. Ambainis, K. Apsitis, C. Calude, R. Freivalds, M. Karpinski, T. Larfeldt, I. Sala, and J. Smotrovs, "Effects of Kolmogorov complexity present in inductive inference as well," in *ALT '97: Proceedings of the 8th International Conference on Algorithmic Learning Theory.* London, UK: Springer-Verlag, 1997, pp. 244–259.

[17] J.-L. Yuan and T. Fine, "Neural-network design for small training sets of high dimension," *IEEE Tnansactions on Neural Networks*, vol. 9, pp. 266–280, 1998.

[18] J.-L. Yuan, "Bootstrapping nonparametric feature selection algorithms for mining small data sets," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1999, pp. 2526 – 2529.

[19] C. Huang and C. Moraga, "A diffusion-neural-network for learning from small samples," *International Journal of Approximate Reasoning*, vol. 35, pp. 137–161, 2004.

[20] R. Mao, H. Zhu, L. Zhang, and A. Chen, "A new method to assist small data set neural network learning," in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, 2006, pp. 17–22.

[21] D.-C. Li, C.-S. Wu, T. T.-I., and L. Y.-S., "Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge," *Computers and Operations Research*, vol. 34, pp. 966–982, 2007.

[22] D.-C. Li, C.-W. Yeh, T.-I. Tsai, Y.-H. Fang, and S. Hu, "Acquiring knowledge with limited experience," *Expert Systems*, vol. 24, pp. 162–170, 2007.

[23] D.-C. Li, C.-S. Wu, T.-I. Tsai, and F. M. Chang, "Using mega-fuzzification and data trend estimation in small data set learning for early FMS scheduling knowledge," *Comput. Oper. Res.*, vol. 33, no. 6, pp. 1857–1869, 2006.

[24] T.-I. Tsai and D.-C. Li, "Approximate modeling for high order non-linear functions using small sample sets," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 564–569, 2008.

[25] D.-C. Li and C.-W. Yeh, "A non-parametric learning algorithm for small manufacturing data sets," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 391–398, 2008.

[26] D.-C. Li and C.-W. Liu, "A neural network weight determination model designed uniquely for small data set learning," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 9853–9858, 2009.

[27] I. V. Tetko, A. I. Luik, and G. I. Poda, "Application of neural networks in structure-activity relationships of a small number of molecules," *J. Med. Chem.*, vol. 36, pp. 811–814, 1993.

[28] D. Hecht and G. Fogel, "High-throughput ligand screening via preclustering and evolved neural networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, pp. 476–484, 2007.

[29] M. Cheung, S. Johnson, D. Hecht, and G. Fogel, "Quantitative structure-property relationships for drug solubility prediction using evolved neural networks," in *Proceedings of the IEEE World Congress on Computational Intelligence*, 2008, pp. 688–693.

[30] H. Lohr, *Sampling: Design and Analysis.* Duxbury Press, 1999.

[31] J. Gamez, F. Modave, and O. Kosheleva, "Selecting the most representative sample is NP-hard: Need for expert (fuzzy) knowledge," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, June 2008, pp. 1069–1074.

[32] L. Holmstrom and P. Koistinen, "Using additive noise in backpropagation training," *IEEE Transactions on Neural Networks*, vol. 3, pp. 24–38, 1992.

[33] C. Wang and J. C. Principe, "Training neural networks with additive noise in the desired signal," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1511–1517, 1995.

[34] K. Wang, J. Yang, G. Shi, and Q. Wang, "An expanded training set based validation method to avoid overfitting for neural network classifier," *International Conference on Natural Computation*, vol. 3, pp. 83–87, 2008.

[35] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Transactions on Neural Networks*, vol. 5, pp. 1050–1057, 2000.

[36] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 983–995, 2008.

[37] S. Bos and E. Chug, "Using weight decay to optimize the generalization ability of a perceptron," in *Proceedings of the 1996 International Conference on Neural Networks*. IEEE, 1996, pp. 241–246.

[38] K. Mahdaviani, H. Mazyar, S. Majidi, and M. H. Saraee, "A method to resolve the overfitting problem in recurrent neural networks for prediction of complex systems' behavior," in *IJCNN'08: Proceedings of the 2008 International Joint Conference on Neural Networks*, 2008, pp. 3723–3728.

[39] R. Reed, "Pruning algorithms - a survey," *IEEE Transactions on Neural Networks*, vol. 4, pp. 740–747, 1993.

[40] T.-Y. Kwok and D.-Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Transactions on Neural Networks*, vol. 8, pp. 630–645, 1997.

[41] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Networks*, vol. 11, pp. 761–767, 1998.

[42] I. Dagher, M. Georgiopoulos, G. Heileman, and G. Bebis, "Ordered Fuzzy ARTMAP: a Fuzzy ARTMAP algorithm with a fixed order of pattern presentation," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 1998), IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, 1998, pp. 1717–1722.

[43] I. Dagher, M. Georgiopoulos, G. L. Heileman, and G. Bebis, "An ordering algorithm for pattern presentation in Fuzzy ARTMAP that tends to improve generalization performance," *IEEE Transactions on Neural Networks*, vol. 10, pp. 768–778, 1999.

[44] S. Tan, M. Rao, and C. P. Lim, "A hybrid neural network classifier combining ordered Fuzzy ARTMAP and the dynamic decay adjustment algorithm," *Soft Computing*, vol. 12, pp. 765–775, 2008.

[45] J. Tou and R. Gonzales, *Pattern recognition principles*. Reading, MA: Addison-Wesley, 1976.