Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition Letters 28 (2007) 673–685

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach

Nikola Kasabov *

Knowledge Engineering and Discovery Research Institute, KEDRI Auckland University of Technology, Auckland, New Zealand

Available online 20 September 2006

Abstract

The paper is offering a comparative study of major modeling and pattern discovery approaches applicable to the area of data analysis and decision support systems in general, and to the area of Bioinformatics and Medicine – in particular. Compared are inductive versus transductive reasoning, global, local, and personalised modeling, and all these approaches are illustrated on a case study of gene expression and clinical data related to cancer outcome prognosis. While inductive modeling is used to develop a model (function) from data on the whole problem space and then to recall it on new data, transductive modeling is concerned with the creation of single model for every new input vector based on some closest vectors from the existing problem space. A new method – WWKNN (weighted distance, weighted variables *K*-nearest neighbors), and a framework for the integration of global, local and personalised models for a single input vector are proposed. Integration of data (e.g. clinical and genetic) and of models (e.g. global, local and personalised) for a better pattern discovery, adaptation and accuracy of the results, are the major points of the paper.

Keywords: Transductive reasoning; Personalised modeling; Local modeling; Model integration; Bioinformatics; Gene expression data; Medical decision support systems; Gene regulatory networks; Evolving connectionist systems; Evolutionary optimisation

1. Bioinformatics – an area of increasing data and emergence of knowledge

With the completion of the sequence draft of the human genome and the genomes of other species (more to be sequenced during this century) the task is now to be able to process this vast amount of ever growing dynamic information and to create intelligent systems for data analysis and knowledge discovery, from cells to whole organisms and species (Dow et al., 1995; Baldi and Brunak, 2001).

The central dogma of molecular biology is that the DNA (Dioxyribonucleic Acid) present in the nucleus of each cell of an organism is transcribed into RNA, which is translated into proteins (Crick, 1970). Genes are complex molecular structures that cause dynamic transformation of one substance into another during the whole life of an individual, as well as the life of the human population over many generations (Snustad and Simmons, 2003). Even the static information about a particular gene is very difficult to understand (see the GenBank database www.genebank.com). When genes are ‘‘in action’’, the dynamics of the processes in which a single gene is involved are thousand times more complex, as this gene interacts with many other genes, proteins, and is influenced by many environmental and developmental factors (D'Haeseleer et al., 2000).

Modeling these interactions and extracting meaningful patterns – knowledge, is a major goal for the area of Bioinformatics. Bioinformatics is concerned with the application and the development of the methods of information sciences for the collection, storage, analysis, modeling and knowledge discovery from biological and medical data.

The whole process of the expression of genes and the production of proteins, and back to the genes, evolves over time. Proteins have 3D structures that evolve over time governed by physical and chemical laws. Some proteins bind to the DNA and make some genes to express and

* Tel.: +64 9 91 79506; fax: +64 9 91 79501.
E-mail address: nkasabov@aut.ac.nz
URL: http://www.kedri.info.

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.08.007

other – to suppress their expression. The genes in an individual person may mutate, change slightly their code, and may therefore express differently at a next time. Genes represent both static and dynamic information that is difficult to capture as patterns (Collado-Vides and Hofestadt, 2002; Marnellos and Mjolsness, 2003).

Gene and protein expression values can be measured through micro-array equipment (Quakenbush, 2002) thus making this information available for a medical decision making, such as medical prognosis and diagnosis, and drug design.

Many challenging problems in Bioinformatics need to be addressed and new knowledge about them revealed, to name only some of them:

- Recognizing patterns from sequences of DNA, e.g. promoter recognition (Bajic et al., 2003).
- Recognizing patterns in RNA data (e.g. splice junctions between introns and exons; micro-RNA structures; non-coding regions analysis).
- Profiling gene micro-array expression data from RNA in different types of tissue (cancer versus normal), different types of cells, to identify profiles of diseases (Perou et al., 2000; Ramaswamy et al., 2001; Shipp et al., 2002a,b; Singh et al., 2002; van de Vijver et al., 2002; Veer et al., 2002).
- Predicting protein structures.
- Modeling metabolism in cells (Vides et al., 1996; Bower and Bolouri, 2001).
- Modeling entire cells (Vides et al., 1996).
- Modeling brain development and brain diseases (LeCun et al., 1990; Marnellos and Mjolsness, 2003; Kasabov and Benuskova, in press).
- Creating complex medical decision support systems that deal with a large set of variables that include both gene and clinical variables to obtain a correct diagnosis and prognosis for a patient (Kasabov et al., 2003).

A main approach to understand gene interaction and life science in general and to solve the above problems is mathematical and computational modeling (Sobral, 1999). The more new information is made available about DNA, gene expression, protein creation, metabolic pathways, etc., the more accurate their information models will become. They should be adaptive to any new information made available in a continuous way. The process of biological pattern and knowledge discovery is always evolving.

A review of problems and challenges in Bioinformatics, along with a brief introduction of the major mathematical and computational modeling techniques, is presented in (Kasabov et al., 2005).

There are three main contributions of this paper. First, it compares different modeling approaches with the emphasis not only on the accuracy of the models, but also on the type of patterns – knowledge, that these models facilitate to discover from data. Second, the paper introduces a new algorithm for personalised modeling, called WWKNN (weighted–weighted *K*-nearest neighbor), and a third contribution is the proposed general framework for an integrated modeling that combines the advantages of the different modeling approaches – global, local and personalised. Section 2 discusses briefly three generic modeling approaches – global, local and personalised. Section 3 presents one particular local modeling technique – evolving connectionist systems (ECOS). Section 4 introduces the WWKNN method. These approaches are applied in Section 5 on a case study problem of modeling and profile discovery from gene expression and clinical data related to cancer outcome prognosis. Feature and model parameter optimisation though evolutionary computation is presented in Section 6. Section 7 discusses the application of global, local and personalised models to another important problem in Bioinformatics – gene regulatory network modeling. Section 8 introduces a framework for the integration of global, local and personalised models when dealing with a single input vector. Further research directions in the area of computational modeling for Bioinformatics and medical decision support systems in general are discussed in Section 9. The main conclusion is that for a detailed research on a complex problem, different levels of knowledge need to be discovered – at global, local and personalised levels, and the integration of them may lead to better results.

2. Inductive versus transductive reasoning. Global, local and personalised modeling

2.1. Inductive versus transductive reasoning

The widely used in all fields of science *inductive reasoning* approach is concerned with the creation of a model (a function) from all available data, representing the entire problem space. The model is applied then on new data (deduction). *Transductive inference*, introduced by Vapnik (1998), is defined in contrast as a method used to estimate the value of a potential model (function) only for a single point of space (that is, a new data vector) by utilizing additional information related to that vector. While the inductive approach is useful when a global model of the problem is needed in an approximate form, the transductive approach is more appropriate for applications where the focus is not on the model, but rather on every individual case. This is very much related to clinical and medical applications where the focus needs to be centered on individual patient's conditions.

The transductive approach is related to the common sense principle (Bosnic et al., 2003) which states that to solve a given problem one should avoid solving a more general problem as an intermediate step.

In the past years, transductive reasoning has been implemented for a variety of classification tasks such as text classification (Joachims, 1999; Chen et al., 2003), heart disease diagnostics (Wu et al., 1999), synthetic data classification using graph-based approach (Li and Yuen, 2001), digit and speech recognition (Joachims, 2003), promoter recognition in bioinformatics (Kasabov and Pang, 2004), image recognition (Li and Chua, 2003) and image classification (Proedrou et al., 2002), micro-array gene expression classification (West et al., 2001; Wolf and Mukherjee, 2004) and biometric tasks such as face surveillance (Li and Wechsler, 2004). This reasoning method is also used in prediction tasks such as predicting if a given drug binds to a target site (Weston et al., 2003) and evaluating the prediction reliability in regression (Bosnic et al., 2003) and providing additional measures to determine reliability of predictions made in medical diagnosis (Kukar, 2003). Out of several research papers that utilize the transductive principals, transductive support vector machines (Joachims, 1999) and semi-supervised support vector machines (Bennett and Demiriz, 1998) are often citied (Sotiriou et al., 2003).

In transductive reasoning, for every new input vector $\boldsymbol{x}_i$ that needs to be processed for a prognostic/classification task, the N_i nearest neighbors, which form a data subset D_i, are derived from an existing dataset D and a new model D_i is dynamically created from these samples to approximate the function in the locality of point $\boldsymbol{x}_i$ only. The system is then used to calculate the output value y_i for this input vector $\boldsymbol{x}_i$.

This approach has been implemented with radial basis function (Song and Kasabov, 2004) in medical decision support systems and time series prediction problem, where individual models are created for each input data vector. The results indicate that transductive inference performs better than inductive inference models mainly because it exploits the structural information of unlabeled data. However, there are a few open questions that need to be addressed while implementing transductive modeling, e.g. How many neighboring samples K are needed? What variables to use (the variable space) and how important each of them is to the input vector for which a model is being built? What type of distance measure to use when choosing the neighbors? What model to apply on the neighboring samples? These issues will be addressed in the paper.

2.2. Global, local and personalised modeling

The three main approaches investigated in the paper are:

- Global modeling – a model is created from data, that covers the whole problem space and is represented as a single function, e.g. a regression formula, a neural network of MLP (multi-layer perceptron type) etc.
- Local modeling – a set of local models are created from data, each representing a sub-space (e.g. a cluster) of the problem space, e.g. a set of rules; a set of local regressions, etc.
- Individualised (personalised) modeling – a model is created only for a single point (vector, patient record) of the problem space using transductive reasoning.

To illustrate the concepts of global, local and personalised modeling, here we use a case study problem and a publicly available data set from Bioinformatics – the DLBCL lymphoma data set for predicting survival outcome over five years period. This data set contains 58 vectors – 30 cured DLBCL lymphoma disease cases and 28 fatal (Shipp et al., 2002a,b). There are 6817 gene expression variables. Clinical data is available for 56 of the patients represented as IPI – an International Prognostic Index, which is an integrated number representing overall effect of several clinical variables (Shipp et al., 2002a,b). The task is, based on the existing data, to: (1) create a prognostic system that predicts the survival outcome of a new patient; (2) to extract profiles that can be used to provide an explanation for the prognosis; (3) to find markers (genes) that can be used for the design of new drugs to cure the disease or for an early diagnosis.

Using a global linear regression method on the 11 DLBCL prognostic genes selected in (Shipp et al., 2002a,b), denoted here as $X_1, X_2, \ldots, X_{11}$, for the 58 vectors, when data is normalised in the range $[0, 1]$, the following classification linear discrimination model is derived:

$$\begin{aligned} Y = {} & 0.36 + 0.53X_1 - 0.12X_2 - 0.41X_3 - 0.44X_4 \\ & + 0.34X_5 + 0.32X_6 - 0.07X_7 + 0.5X_8 - 0.5X_9 \\ & + 0.18X_{10} + 0.3X_{11}. \end{aligned} \tag{1}$$

Formula (1) constitutes a *global model* (i.e. it is to be used to evaluate the output for any input vector in the 11-dimensional space regardless of where it is located). It indicates to certain degree the high importance of some genes (e.g. genes X_1, X_8, X_9, X_4, X_3) and the low importance of other genes (e.g. genes X_7, X_2) on the whole problem space, but this may not be valid for a particular individual vector in the 11 dimensional space. The model, being global, gives the "big" picture, but not an individual profile. It is also difficult to adapt to new data. Despite of these problems, linear and logistic regression methods have been widely used for gene expression modeling (DeRisi et al., 1996; Furey et al., 2000) and for modeling gene regulatory networks (D'Haeseleer et al., 2000; Bower and Bolouri, 2001).

Another global statistical machine learning method, that is widely used for the creation of classification models, is the *support vector machine* (*SVM*) (Vapnik, 1998). A SVM model consists of a set of vectors described by a kernel function that "goes" on the border area between the samples that belong to different classes (the vectors are called support vectors). SVM models are very good classification models, but are difficult to adapt to new data and the knowledge extracted from them is very limited. SVM models have been used in many research papers (Vapnik, 1998; Shipp et al., 2002a,b).

In contrast to the global models, *local models* are created to evaluate the output function for only a sub-space of the problem space. Multiple local models (e.g. one for each cluster of data) can constitute together the complete model of the problem over the whole problem space. Local models are often based on clustering techniques. A cluster is a group of similar data samples, where similarity is measured predominantly as Euclidean distance in an orthogonal problem space. Clustering techniques include: k-means (Mitchell et al., 1997); Self-Organising Maps (SOM) (DeRisi et al., 1996; Kohonen, 1997), fuzzy clustering (Bezdek, 1981; Futschik, 2002; Dembele and Kastner, 2003), hierarchical clustering (Alon et al., 1999), simulated annealing (Lukashin and Fuchs, 2001). In fuzzy clustering, one sample may belong to several clusters to a certain membership degree, the sum of which is 1. Generally speaking, local models are easier to adapt to new data and can provide a better explanation for individual cases. The ECF method described in the next section, is a representative of the local modeling techniques.

A "*personalised*" *model* is created "on the fly" for every new input vector and this individual model is based on the closest data samples to the new sample taken from a data set. A simple example of personalised modeling technique is the K-NN (K-nearest neighbors) method, where for every new sample, the nearest K samples are derived from a data set using a distance measure, usually Euclidean distance, and a voting scheme is applied to define the class label for the new sample (Mitchell et al., 1997; Vapnik, 1998).

In the K-NN method, the output value y_i for a new vector $\boldsymbol{x}_i$ is calculated as the average of the output values of the k nearest samples from the data set D_i. In the weighted K-NN method (WKNN) the output y_i is calculated based not only on the output values (e.g. class label) y_j of the K NN samples, but also on a weight w_j, that depends on the distance of them to $\boldsymbol{x}_i$

$$y_i = \frac{\sum_{j=1}^{N_i} w_j y_j}{\sum_{j=1}^{N_i} w_j}, \tag{2}$$

where y_j is the output value for the sample $\boldsymbol{x}_j$ from D_i and w_j are their weights calculated based on the distance from the new input vector

$$w_j = [\max(\boldsymbol{d}) - (d_j - \min(\boldsymbol{d}))]/\max(\boldsymbol{d}). \tag{3}$$

The vector $\boldsymbol{d} = [d_1, d_2, \ldots, d_{N_i}]$ is defined as the distances between the new input vector $\boldsymbol{x}_i$ and the N_i nearest neighbors $(\boldsymbol{x}_j, y_j)$ for $j = 1$ to N_i

$$d_j = \mathrm{sqrt}[\mathrm{sum}_{l=1\text{ to }V}(x_{i,l} - x_{j,l})^2], \tag{4}$$

where V is the number of the input variables defining the dimensionality of the problem space; $x_{i,l}$ and $x_{j,l}$ are the values of variable x_l in vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively. The parameters max($\boldsymbol{d}$) and min($\boldsymbol{d}$) are the maximum and minimum values in $\boldsymbol{d}$ respectively. The weights w_j have the values between min($\boldsymbol{d}$)/max($\boldsymbol{d}$) and 1; the sample with the minimum distance to the new input vector has the weight value of 1, and it has the value min($\boldsymbol{d}$)/max($\boldsymbol{d}$) in case of maximum distance.

If WKNN is used to solve a classification problem and two classes are represented by 0 (class 1) and 1 (class 2) output class labels, the output for a new input vector $\boldsymbol{x}_i$ has the meaning of a "*personalised probability*" that the new vector $\boldsymbol{x}_i$ will belong to class 2. In order to finally classify a vector $\boldsymbol{x}_i$ into one of the (two) classes, there has to be a probability threshold selected P_{thr}, so that if $y_i \geqslant P_{\mathrm{thr}}$, then the sample $\boldsymbol{x}_i$ is classified in class 2. For different values of the threshold P_{thr}, the classification error is generally speaking, different.

Using global probability measures to evaluate a probability of single input vector $\boldsymbol{x}$ to belong to a class A (the Bayesian probability global inference approach) requires that some prior probabilities are available and these are not easy to obtain and often too uncertain. The Bayesian posterior probability $p(A|\boldsymbol{x})$ of a new input vector $\boldsymbol{x}$ to belong to class A is calculated with the use of the formula

$$p(A|x) = \frac{p(A) \cdot p(x|A)}{p(x)}, \tag{5}$$

where $p(A)$ and $p(x)$ are *prior* probabilities and $p(A|x)$ and $p(x|A)$ are posterior probabilities.

Calculating "personalised probability" in a transductive way, does not require any prior information.

In Section 4 a new method called WWKNN is proposed where not only the distance between a new input vector and the neighboring ones is weighted, but also variables, according to their ranking for importance in the neighborhood area.

3. Evolving connectionist systems ECOS for local modeling and cluster-based rule discovery

3.1. The ECOS architecture

Some traditional neural network models are seen as "black boxes" and are not very useful models for the discovery of new patterns from data (Arbib, 2003). A new type of neural networks, evolving connectionist systems (ECOS), is introduced in (Kasabov, 2002). They allow for structural adaptation, fast incremental, on-line learning, and rule extraction and rule adaptation. One of its simplest implementations is the evolving classifier function ECF (Kasabov, 2002; Kasabov and Song, 2002) (see Fig. 1).

The ECOS from Fig. 1 consists of five layers of neurons and four layers of connections. The first layer of neurons receives the input information. The second layer (optional) calculates the fuzzy membership degrees to which the input values belong to predefined fuzzy membership functions, e.g. Low, Medium, or High. The membership functions can be kept fixed, or can change during training. The third layer of neurons represents associations between the input and the output variables, rules. The fourth layer (optional) calculates the degree to which output membership functions are matched by the rule node activation, and the fifth layer does defuzzification and calculates values for the output variables.

3.2. The ECOS learning algorithms

ECOS in general are connectionist systems that evolve their structure and functionality in a continuous, self-organised, on-line, adaptive, interactive way from incom-

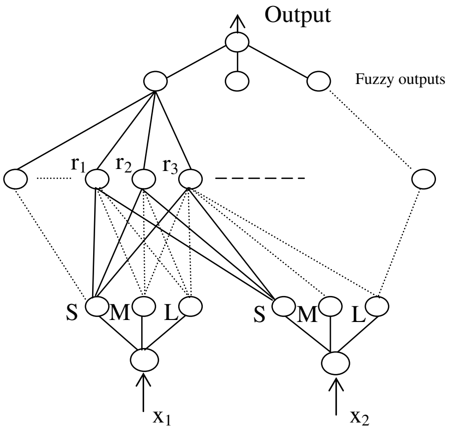


Fig. 1. A simple evolving connectionist structure – EfuNN, of two inputs and one output. In a simplified version – Evolving Classification Function (ECF), there are no fuzzy output nodes as each evolving node $r_1, r_2, \ldots$ represents a cluster centre of input vectors that belong to the same output class using a defined maximum cluster radius $R_{\max}$ with the use of Euclidean distance (hyperspherical cluster shape).

ing information. They can learn from data in a supervised or unsupervised way. Learning is based on clustering of input vectors and function estimation for the clusters in the output space. Prototype rules can be extracted to represent the clusters and the functions associated with them. The ECOS models allow for an incremental change of the number and types of inputs, outputs, nodes, connections. The algorithm to evolve a simple classification system called ECF (Evolving Classification Function) from incoming stream of data is shown in Fig. 2. The internal nodes in the ECF structure capture clusters of input data and are called rule nodes.

Different types of rules (knowledge) representation are facilitated by different ECOS architectures, i.e. Zadeh-Mamdani rules – in the evolving fuzzy neural networks EFuNN (Kasabov, 2000, 2001) – see Fig. 1, or Takagi–Sugeno rules – in the dynamic neuro-fuzzy inference systems DENFIS (Kasabov and Song, 2002). An ECOS structure grows and “shrinks” in a continuous way from an input data stream. Feed-forward and feedback connections are both used in the architecture. The ECOS are not limited in number and types of inputs, outputs, nodes, connections. Several machine learning methods are facilitated in different types of ECOS that have been already applied to Bioinformatics problems (Kasabov, 2002).

4. WWKNN: weighted–weighted *K* nearest neighbor algorithm for transductive reasoning and personalised modeling

In the WKNN the calculated output for a new input vector depends not only on the number of its neighboring vectors and their output values (class labels), but also on the distance between these vectors and the new vector

Learning algorithm of the ECF model:

1. Enter the current input vector from the data set (stream) and calculate the distances between this vector and all rule nodes already created using Euclidean distance (by default). If there is no node created, create the first one that has the coordinates of the first input vector attached as input connection weights.
2. If all calculated distances between the new input vector and the existing rule nodes are greater than a max-radius parameter Rmax, a new rule node is created. The position of the new rule node is the same as the current vector in the input data space and the radius of its receptive field is set to the min-radius parameter Rmin; the algorithm goes to step 1; otherwise it goes to the next step.
3. If there is a rule node with a distance to the current input vector less then or equal to its radius and its class is the same as the class of the new vector, nothing will be changed; go to step 1; otherwise:
4. If there is a rule node with a distance to the input vector less then or equal to its radius and its class is different from those of the input vector, its influence field should be reduced. The radius of the new field is set to the larger value from the two numbers: distance minus the min-radius; min-radius. New node is created as in 2 to represent the new data vector.
5. If there is a rule node with a distance to the input vector less than or equal to the max-radius, and its class is the same as of the input vector's, enlarge the influence field by taking the distance as a new radius if only such enlarged field does not cover any other rule nodes which belong to a different class; otherwise, create a new rule node in the same way as in step 2, and go to step 1.

Recall procedure (classification of a new input vector) in a trained ECF :

1. Enter the new input vector in the ECF trained system; If the new input vector lies within the field of one or more rule nodes associated with one class, the vector is classified in this class;
2. If the input vector lies within the fields of two or more rule nodes associated with different classes, the vector will belong to the class corresponding to the closest rule node.
3. If the input vector does not lie within any field, then take *m* highest activated by the new vector rule nodes, and calculate the average distances from the vector to the nodes with the same class; the vector will belong to the class corresponding to the smallest average distance.

Fig. 2. The training and recall algorithms of the ECF local learning model from Fig. 1.

which is represented as a weight vector (W). It is assumed that all V input variables are used and the distance is measured in an V-dimensional Euclidean space with all variables having the same impact on the output variable.

But when the variables are ranked in terms of their discriminative power of class samples over the whole V-dimensional space, we can see that different variables have different importance to separate samples from different classes, therefore – a different impact on the performance of a classification model. If we measure the discriminative power of the same variables for a sub-space (local space) of the problem space, the variables may have a different ranking. Using the ranking of the variables in terms of a discriminative power within the neighborhood of K vectors, when calculating the output for the new input vector, is the main idea behind the WWKNN algorithm, which includes one more weight vector to weigh the importance of the variables. The Euclidean distance d_j between a new vector $\boldsymbol{x}_i$ and a neighboring one $\boldsymbol{x}_j$ is calculated now as

$$d_j = \text{sqr}[\text{sum}_{l=1\text{ to }V}(c_{i,l}(x_{i,l} - x_{j,l}))^2], \tag{6}$$

where $c_{i,l}$ is the coefficient weighing variable x_l for in neighborhood of $\boldsymbol{x}_i$. It can be calculated using a Signal-to-Noise Ratio (SNR) procedure that ranks each variable across all vectors in the neighborhood set D_i of N_i vectors

$$\boldsymbol{C}_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,V}), \tag{7}$$

$$c_{i,l} = S_l/\text{sum}(S_l) \quad \text{for } l = 1, 2, \ldots, V, \text{ where} \tag{8}$$

$$S_l = \text{abs}\left(M_l^{(\text{class 1})} - M_l^{(\text{class 2})}\right)\Big/\left(\text{Std}_l^{(\text{class 1})} + \text{Std}_l^{(\text{class 2})}\right). \tag{9}$$

Here $M_l^{(\text{class 1})}$ and $\text{Std}_l^{(\text{class 1})}$ are respectively the mean value and the standard deviation of variable x_l for all vectors in D_i that belong to class 1.

The new distance measure, that weighs all variables according to their importance as discriminating factors in the neighborhood area D_i, is the new element in the WWKNN algorithm when compared to the WKNN.

Using the WWKNN algorithm a "personalised" profile of the variable importance can be derived for any new input vector, that represents a new piece of "personalised" knowledge.

Weighting variables in personalised models is used in the TWNFI models (Transductive Weighted Neuro-Fuzzy Inference) in (Song and Kasabov, 2005).

There are several open problems, e.g. how to choose the optimal number of vectors in a neighborhood and the optimal number of variables, which for different new vectors may be different. This issue is discussed in Section 6.

5. Comparative study of global, local and personalised modeling on a case study of gene expression and clinical data modeling

5.1. Problem definition and data sets

A gene expression profile is defined here as a pattern of expression of a number of significant genes for a group (cluster) of samples of a particular output class or category. A gene expression profile is represented here as an IF–THEN inference rule:

> IF ⟨A pattern of gene expression values of selected genes is observed⟩
> THEN ⟨There is a likelihood for a certain diagnostic or prognostic outcome⟩.

Having profiles/rules for a particular disease makes it possible to set up early diagnostic tests so that a sample can be taken from a patient, data related to the sample processed, and then mapped into the existing profiles. Based on similarity between the new data and the existing profiles, the new data vector can be classified as belonging to the group of "good outcome", or "poor outcome" with a certain confidence and a good explanation can be provided for the final decision as the matched local rules/profile(s) will be the closest to the person's individual profile (Kasabov et al., 2002).

Contemporary technologies, such as gene microarrays, allow for the measurement of the level of expression of up to 30,000 genes in RNA sequences that is indicative of how much protein will be produced by each of these genes in the cell (Gollub et al., 2003). The goal of the micro-array gene expression data analysis is to identify a gene or a group of genes that are differently expressed in one state of the cell or a tissue (e.g. cancer) versus another state (normal) (Gollub et al., 1999). Generally, it is difficult to find consistent patterns of gene expression for a class of tissues.

Gene expression data is often accompanied by clinical data variables. The issue of gene and clinical variables integration for the discovery of combined patterns is addressed here as well.

5.2. Experimental results with the use of different modeling techniques

The two main reasoning approaches – inductive and transductive are used here to develop global, local and personalised models on the same data in order to compare different approaches on two main criteria – (1) accuracy of the model and (2) type of patterns discovered from data. The following classification techniques are used: multiple linear regression (MLR); SVM; ECF; WKNN; WWKNN.

Each of the models are validated through the same leave-one-out cross validation method (Vapnik, 1998). The accuracy of the different models is presented in Table 1. It can be seen that the transductive reasoning and personalised modeling is sensitive to the selection of the number of the nearest neighbors K. Its optimisation is discussed in the next section.

The WWKNN produces a balanced accuracy of 80% and 81% for each of the two classes (a balanced sensitivity and specificity values) along with an individual ranking of the importance of the variables for each individual sample.

Table 1
Experimental results in terms of model accuracy tested through leave-one-out cross validation method when using different modeling techniques on the DLBCL Lymphoma data for classification of new samples into class 1 – survival, or class 2 – fatal outcome of the disease within 5 years time (see Shipp et al., 2002a,b)

Model/ features	Induct global MLR [%]	Induct global SVM [%]	Induct local ECF [%]	Trans WKNN $K = 8$ [%], $P_{\mathrm{thr}} = 0.5$	Trans WKNN $K = 26$ [%] $P_{\mathrm{thr}} = 0.5$	Trans WW-KNN $K = 16$	Trans MLR $K = 8$ [%]	Trans MLR $k = 26$ [%]	Trans SVM $K = 8$ [%]	Trans SVM $k = 26$ [%]	Trans ECF $K = 8$ [%]	Trans ECF $k = 26$ [%]
IPI (one clinical variable)	73 (87,58)	73 (87,58)	46 (0,100)	50 (87,8)	73 (87,56)	68 (63,73)	50 (87,8)	73 (87,58)	46 (100,0)	73 (87,58)	61 (63,58)	46 (0,100)
11 genes	79 (91,65)	83 (88,78)	86 (88,84)	74 (91,54)	73 (93,47)	81 (81,81)	66 (66,65)	78 (81,73)	76 (91,58)	78 (91,62)	78 (81,73)	83 (91,73)
IPI + 11 genes	82 (83,81)	86 (90,81)	88 (83,92)	77 (90,62)	$P_{\mathrm{thr}} = 0.45$, 82% (97,65)	80 (80,81)	57 (60,54)	79 (80,77)	77 (93,58)	84 (93,73)	75 (83,65)	77 (87,65)

The table shows the overall model classification accuracy in % and the specificity and sensitivity values (accuracy for class 1 and class 2, respectively) – in brackets.

Having this knowledge, a personalised treatment can be attempted that targets the important genes and clinical variables for each patient.

The best accuracy is manifested by the local ECF model, trained on a combined feature vector of 11 gene expression variables and the clinical variable IPI. Its prognostic accuracy is 88% (83% for class 1 – cured, and 92% for class 2 – fatal). This compares favorably with the 75% accuracy of the SVM model used in Shipp et al. (2002a,b).

In addition, local rules that represent cluster gene profiles of the survival versus the fatal group of patients were extracted as graphically shown in Fig. 3. These profiles show that there is no single variable that clearly discriminates the two classes – it is a combination of the variables that discriminates different sub-groups of samples within a class and between classes.

The local profiles can be aggregated into global class profiles through averaging the variable values across all local profiles that represent one class – Fig. 4. Global profiles may not be very informative if data samples are dispersed in the problem space and each class samples are spread out in the space, but they show the "big picture" the common trends across the population of samples.

As each of the global, local and personalised profiles contains different level of information, integrating them through the integration of global, local and personalised models would facilitate a better understanding and a better accuracy of the prognosis. A framework for such integration is introduced in Section 8.

6. Model optimisation with the use of evolutionary computation

6.1. Evolutionary computation

Using a same modeling technique, but different parameter values and different input variables, may lead to different results and different information extracted from the same initial data set. One way to optimise these parameters and obtain an optimal model according to certain criteria (e.g. classification accuracy) is through e*volutionary computation* techniques (Holland, 1975; Goldberg, 1989). One of them – *genetic algorithms* (Goldberg, 1989), is an optimisation technique that generates a population of individual solutions (models) for a problem, e.g. classification systems, and trains these systems on data, so that after training, the best systems (e.g. with the highest accuracy – fitness) can be selected and some operations of "crossover" and "mutation" applied on them to obtain the next generation of models (Goldberg, 1989), etc. The process continues until a satisfactory model is obtained. Applications of GA for gene expression data modeling and for gene regulatory network (GRN) modeling are presented in (Ando et al., 2002; Fogel and Corne, 2003). The problem of the evolutionary computation techniques is that there is no guaranteed optimal solution obtained, as they are

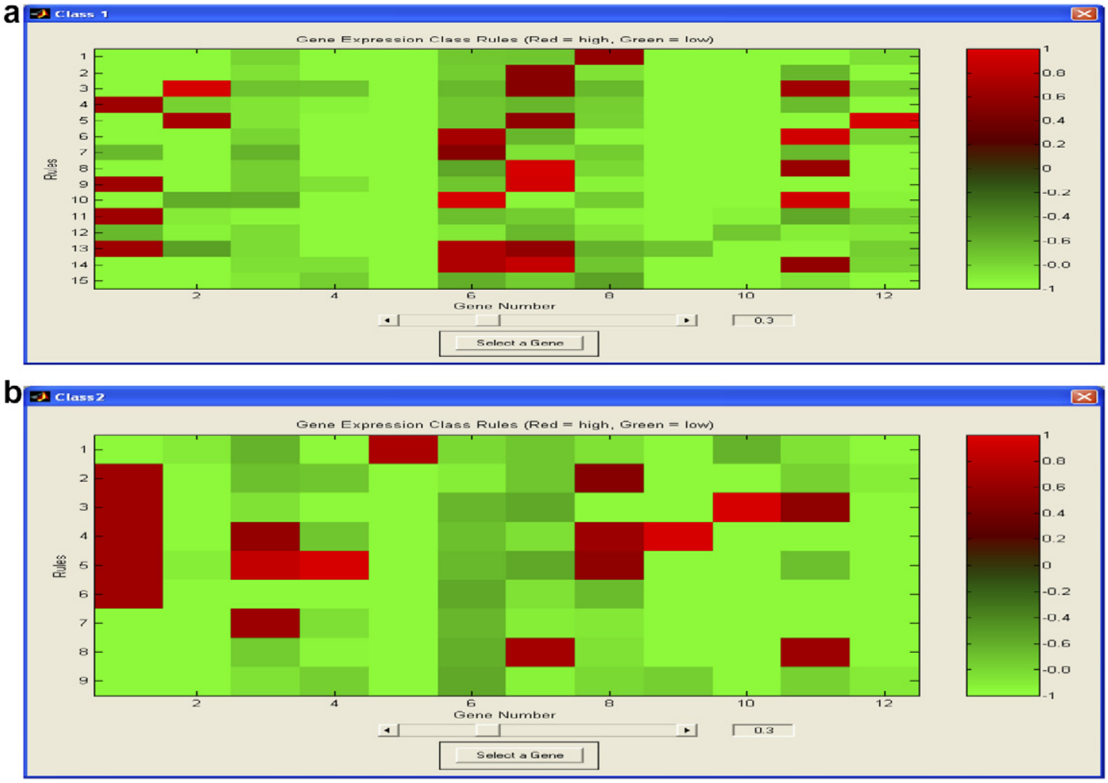
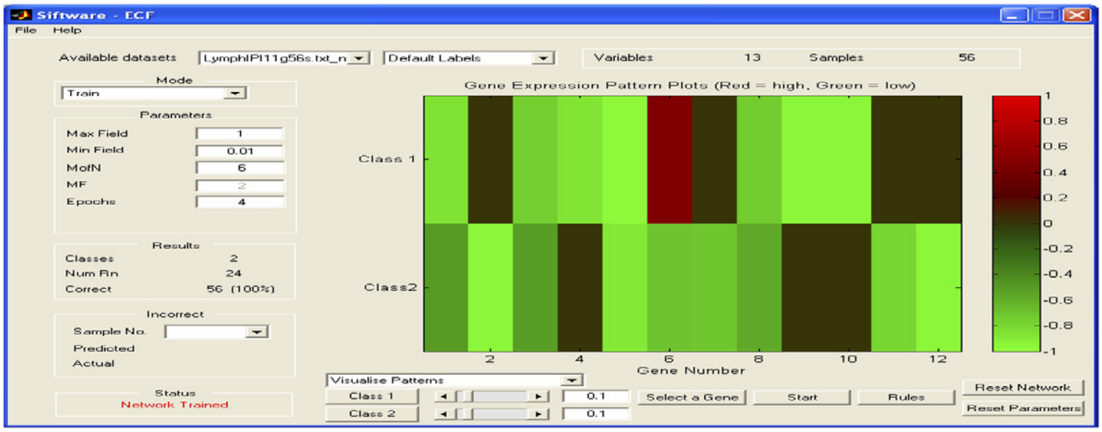


Fig. 3. Cluster-based, local patterns (rules) extracted from a trained ECF model (inductive, local training) on 11 genes expression and a clinical data of the Lymphoma outcome prediction problem. The first variable (first column) is the clinical variable IPI. The accuracy of the model measured through leave-one-out cross validation method is 88% (83% class 1 and 92% class 2). The figure shows: (a) 15 local profiles of class 1 (survive), threshold 0.3 and (b) 9 local profiles of class 2 (fatal outcome), threshold 0.3.


Available datasets LymphIPI11g56s.txt_n Default Labels Variables 13 Samples 56

Mode: Train

Parameters
Max Field 1
Min Field 0.01
MofN 6
MF 2
Epochs 4

Results
Classes 2
Num Rn 24
Correct 56 (100%)

Incorrect
Sample No.
Predicted
Actual

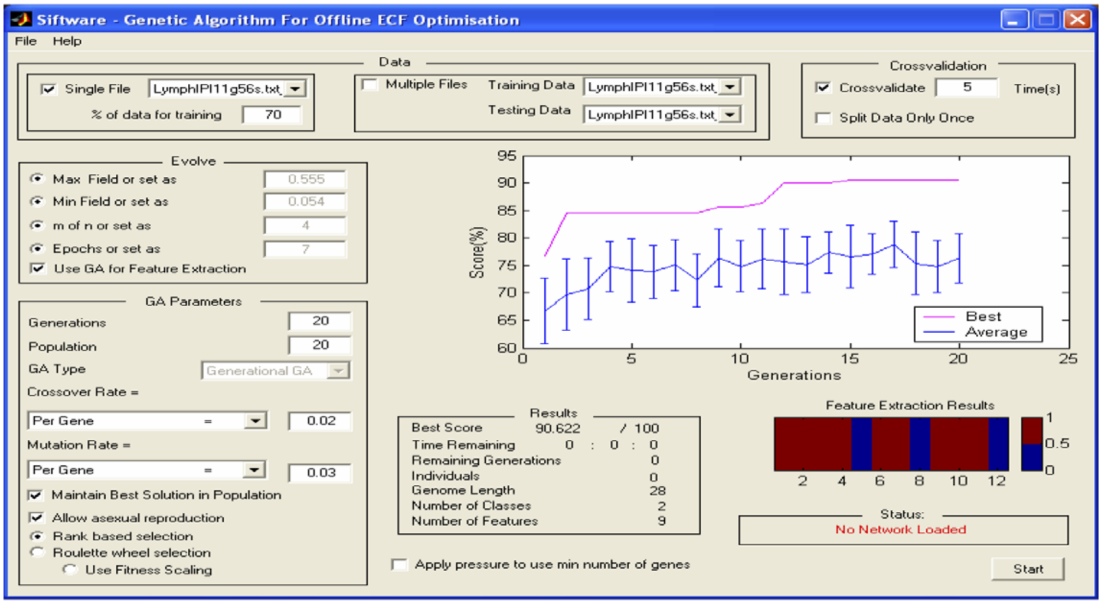
Status
Network Trained

Fig. 4. Global class profiles (rules) are derived through averaging the variable values (genes or IPI) across all local class profiles from Fig. 3 and ignoring low values (below a threshold, e.g. 0.1 as an absolute value). Combined (global) profiles for class 1 and class 2 may not be very informative as they may not manifest any variable that is significantly highly expressed in all clusters of any of the two classes if the different class samples are equally scattered in the whole problem space.

heuristic search techniques in a multi-dimensional solution space. This is in contrast to the exhaustive search technique that will guarantee an optimal solution, but the time the procedure would take may not be acceptable and practically applicable.

6.2. GA optimisation of local models

In the models explored in the previous section, neither the model parameters (such as $R_{\max}$, $R_{\min}$, m and number of membership functions in an ECF model; K in the



Siftware - Genetic Algorithm For Offline ECF Optimisation

File Help

Data

☑ Single File LymphIPI11g56s.txt

% of data for training 70

□ Multiple Files Training Data LymphIPI11g56s.txt

Testing Data LymphIPI11g56s.txt

Crossvalidation

☑ Crossvalidate 5 Time(s)

□ Split Data Only Once

Evolve

● Max Field or set as 0.555

● Min Field or set as 0.054

● m of n or set as 4

● Epochs or set as 7

☑ Use GA for Feature Extraction

GA Parameters

Generations 20

Population 20

GA Type Generational GA

Crossover Rate =

Per Gene = 0.02

Mutation Rate =

Per Gene = 0.03

☑ Maintain Best Solution in Population

☑ Allow asexual reproduction

● Rank based selection

○ Roulette wheel selection

○ Use Fitness Scaling

Results

Best Score 90.622 / 100

Time Remaining 0 : 0 : 0

Remaining Generations 0

Individuals 0

Genome Length 28

Number of Classes 2

Number of Features 9

Status:

No Network Loaded

□ Apply pressure to use min number of genes

Start

Fig. 5. A GA optimised ECF model and a feature set on the DLBCL Lymphoma data. Twenty individual models are used in a population, run for 20 generations with a fitness function – model test accuracy, where the cross validation method used is fivefold-cross validation done on every model within a population with 70% of randomly selected data for training and 30% for testing. The same data is used to test all models in a population. The best performing models are used to create a new generation of 20 individual models etc. The accuracy of the optimal model is now 90.66%, which is higher than the best model from Table 1 (no optimisation is used there). The best model does not use features 5 and 8 (genes 4 and 7).

WKNN; etc.), nor the set of input variables (features) were optimised to produce an optimal model. Out of 11 genes and the IPI clinical variables (features), there may by only a sub-set of them that would produce better model (if the other ones were noisy features for example).

In an experiment shown in Fig. 5 both the ECF parameters and features are optimised with the use of a GA which ran over 20 generations. There are 20 ECF models in a population, having different parameter values and feature sets, and a fitness criteria of overall highest accuracy for the smallest number of features is used. The optimal ECF parameters are given in the figure and the best model has an overall accuracy of 90.66%, which is higher than any of the non-optimised models from Table 1.

6.3. Optimisation of transductive, personalised models

The accuracy of a transductive, personalised model depends very much on some parameters that might have different values for different input vectors (new samples): number of nearest neighbor samples K; distance measure – Euclidean, Hamming, cosine etc., number of input variables – features, used for the personalised model.

Optimising these parameters during the process of the model development is crucial for the model performance and for the correct personalised knowledge derived. So, not only a personalised model is derived for every new data sample, but also an optimal one is created through an optimisation procedure.

GA can be used in a similar way as the optimisation procedure illustrated in the previous section for the parameter optimisation of local models. The fitness function is the cross validation accuracy in a leave-one-out method for all K samples in the neighborhood. The following are the general steps to optimise a personalised WWKNN model (see Section 4) for an input vector $\boldsymbol{x}_i$:

(i) Define the maximum N_i, max and the minimum N_i, min nearest neighbor vectors that can be used in the personalised model and the V_i, max and V_i, min of the max and the min number of variables to be possibly used in the model.

(ii) Select N_i, max nearest samples in a set D_i, max and rank them in terms of distance to $\boldsymbol{x}_i$.

(iii) Rank the N_v, max variables in D_i, max according to SNR (or other ranking procedures) and assign weights to them as described in the WWKNN algorithm.

(iv) For every number of samples n from N_i, min to N_i, max (starting with the closest to $\boldsymbol{x}_i$ samples), and for every number of variables v from N_v, min to N_v, max (starting with the highly ranked variables), DO

  (a) Form a temporal data set D_i, n, v and apply the WWKNN method in a cross validation mode, e.g. leave-one-out (without using the input vector $\boldsymbol{x}_i$) and evaluate the average accuracy of the personalised models built in the set D_i, n, v.

  (b) Keep (save) the set D_i, n, v as the optimal, for the moment, data set D_i, n, v, opt having the optimum number of samples N_i, opt = n and optimum number of variables V_i, opt = v, if the accuracy is higher than the previous iteration accuracy, for other values of v and n.

(v) Calculate the output y_i for the input vector $\boldsymbol{x}_i$ in the optimal data set D_i, n, v, opt using the optimal number of nearest samples N_i, opt and optimal number of variables V_i, opt.

7. Global, local and personalised modeling of gene regulatory networks

In a living cell, genes interact in a complex, dynamic way and this interaction is crucial for the cell behavior. This interaction can be represented in an approximate way as a gene regulatory network (GRN). An example is shown in Fig. 6b.

GRN models can be derived from time course gene expression data of many genes measured over a period of time. Some of these genes have similar expressions to each other as shown in Fig. 6a.

Genes that share similar biological functions usually show similar gene expression profiles and cluster together – Fig. 6a. A GRN model can be used to predict the expression of genes and proteins in a future time and to predict the development of a cell or an organism. The process of deriving GRN from data is called reverse engineering (D'Haeseleer et al., 2000).

Many global, local and personalised modeling techniques have been used so far for the problem, that include: correlation and regression analysis, Boolean networks, graph theory, differential equations, evolutionary computation, neural networks, evolving connectionist systems (ECOS), etc.

A global, regression model represents the expression level of a gene $g_i(t)$ at a time moment t, as a regression function of the expression levels of other genes and proteins at previous time moments, e.g.

$$g_i(t) = f_i(\boldsymbol{G}(t - \Delta)), \quad i = 1, 2, \ldots, n, \tag{10}$$

where $\boldsymbol{G}(t - \Delta)$ is a vector of the expressions of all (or, a sub-group) of genes at previous time moment(s).

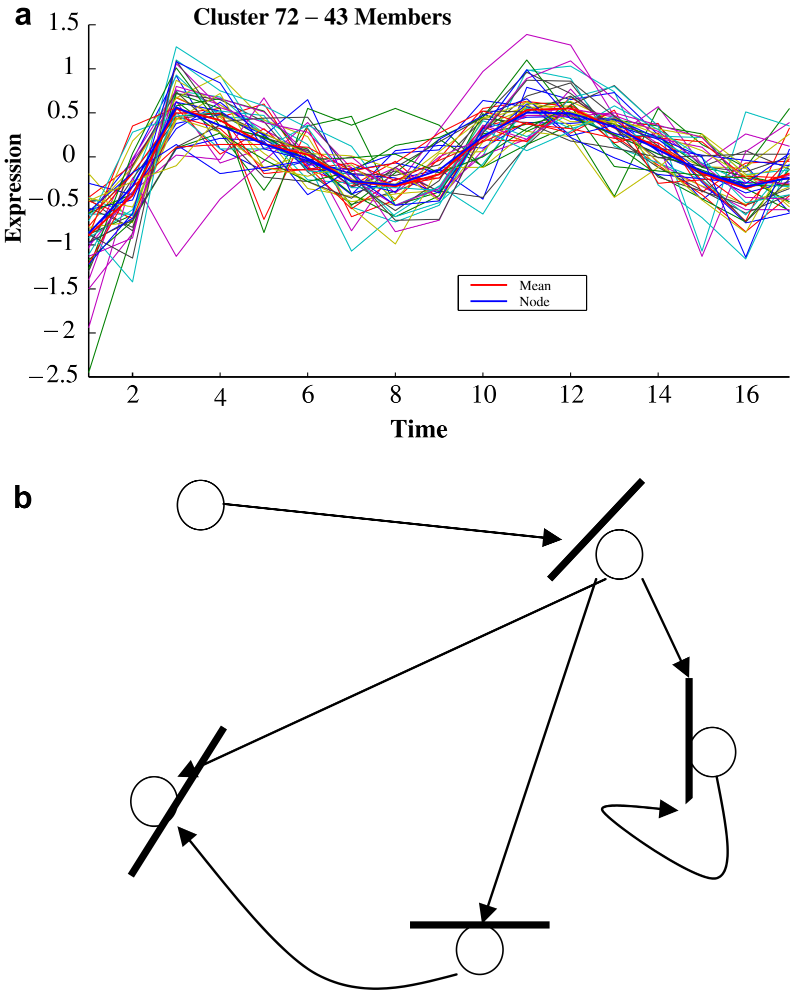


Fig. 6. (a) Time course gene expression data are clustered together based on similarity in their gene expression over time and (b) a simple gene regulatory network (GRN) of five genes and links between them that represent the interaction between the genes in consecutive time moments.

To derive the formulas f_i, n differential equations are to be solved. Kalman filter techniques have been applied as well as other traditional methods for this purpose [ ].

In (Kasabov and Dimitrov, 2002) local modeling with ECOS (EFuNN and DENFIS) was used on a small time series data set of Leukemia cell line U937 data to extract GRN and to represent it as a set of rules associating the expression of selected genes at time t to the level of their expression in the next time moment $(t + \Delta t)$.

An ECOS is incrementally evolved from a series of gene expression vectors $\boldsymbol{G}(t_0), G(t_1), G(t_2), \ldots$, representing the expression values of all, or some of the genes or their clusters. Consecutive vectors $G(t)$ and $G(t + \Delta t)$ are used as input and output vectors respectively in an ECOS model, as shown in Fig. 1. After training of an ECOS on the data, rules are extracted, e.g.

$$\begin{aligned}&\text{IF} \quad g_1(t) \text{ is High } (0.87) \text{ and } g_2(t) \text{ is Low } (0.9)\\&\text{THEN} \quad g_3(t+\Delta t) \text{ is High } (0.6) \text{ and } g_5(t+\Delta t) \text{ is Low.}\end{aligned} \tag{11}$$

Each rule represents a transition between a current and a next state of the system variables – genes. All rules together form a representation of the GRN.

By modifying a threshold for rule extraction, one can extract in an incremental way stronger, or weaker patterns of relationships between the variables (Kasabov et al., 2002).

Using the DENFIS ECOS (Kasabov and Song, 2002) other types of local variable relationship rules for a GRN can be extracted, e.g.

$$\begin{aligned}&\text{IF} \quad g_1(t) \text{ is } (0.63\ 0.70\ 0.76) \text{ and } g_2(t) \text{ is } (0.71\ 0.77\ 0.84) \text{ and}\\&\quad g_3(t) \text{ is } (0.71\ 0.77\ 0.84) \text{ and } g_4(t) \text{ is } (0.59\ 0.66\ 0.72)\\&\text{THEN} \quad g_5(t+\Delta t) = 1.84 - 1.26g_1(t) - 1.22g_2(t)\\&+ 0.58g_3(t) - 0.3g_4(t),\end{aligned} \tag{12}$$

where the cluster for which the value of the gene variable g_5 is defined in the rule above, is a fuzzy cluster represented through triangular membership functions defined as triplets of values for the left-, centre-, and right-points of the triangle on a normalisation range of $[0, 1]$. The fuzzy representation allows for dealing with imprecise data. The rules extracted from the ECOS form a representation of the GRN. Rules may change with the addition of new data, thus making it possible to identify stable versus dynamic parts of the GRNs.

A personalised modeling approach is also possible for the evaluation of a vector $\boldsymbol{G}(t + \Delta t)$ of expressions of a set of genes at a time moment $(t + \Delta t)$ from a vector $\boldsymbol{G}(t)$ of the expressions of these genes in a previous moment. The following algorithm applies:

(1) Find K closest to the vector $\boldsymbol{G}(t)$ vectors of expression of the same genes from the past time data $\boldsymbol{G}_1(t_1)$, $\boldsymbol{G}_2(t_2), \ldots, \boldsymbol{G}_k(t_K)$, each of them $\boldsymbol{G}_j(t_j)$ representing the expression of the genes at a different past time moment t_j.

(2) Using the vectors $\boldsymbol{G}_1(t_1 + \Delta t), \boldsymbol{G}_2(t_2 + \Delta t), \ldots, \boldsymbol{G}_k(t_K + \Delta t)$ of expression values at the time $(\sim+\Delta t)$ as output vectors, create a function f_i for every gene output value or use the WKNN to calculate the output vector $\boldsymbol{G}(t + \Delta t)$.

In this case, there is no GRN model built in advance, and every time a new model is generated based on the closest vectors of gene expression values to the vector $\boldsymbol{G}(t)$ that is used to predict the next time moment values of the genes.

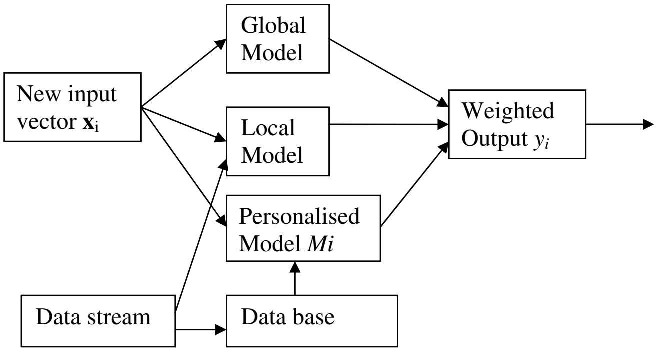


Fig. 7. An integrated multi-model system that includes: a global model (not adaptable to new data), a local model (adaptable to new data, but not to new variables), and a personalised model (derived for every new input vector and adaptable to both new data and new variables). The output value is derived as weighted outputs from all models, where the weights are personalised and optimised for every new input vector in a similar way as the parameters of the personalised model (see the text in Section 6.3).

8. An integrated modeling framework of global, local and personalised models

Global models capture trends in data that are valid for the whole problem space, and local models – capture local patterns, valid for clusters of data. Both models contain useful information and knowledge. Local models are also adaptive to new data as new clusters and new functions, that capture patterns of data in these clusters, can be incrementally created. Usually, both global and local modeling approaches assume a fixed set of variables and if new variables, along with new data, are introduced with time, the models are very difficult to modify in order to accommodate these new variables. This can be done in the personalised models, as they are created "on the fly" and can accommodate any new variables, provided that there is data for them. All the three approaches are useful for complex modeling tasks and all of them provide complementary information and knowledge, learned from the data. Integrating all of them in a single multi-model system would be an useful approach and a challenging task.

A graphical representation of an integrated multi-model system is presented in Fig. 7. For every single input vector, the outputs of the tree models are weighted. The weights can be adjusted and optimised for every new input vector in a similar way as the parameters of a personalised model – see Section 6.3.

$$y_i = w_{i,g}y_i(\boldsymbol{x}_i)^{(\text{global})} + w_{i,l}y_i(\boldsymbol{x}_i)^{(\text{local})} + w_{i,p}y_i(\boldsymbol{x}_i)^{(\text{personalised})}. \tag{13}$$

9. Conclusions and future directions

The problems in Bioinformatics are too complex to be adequately modeled with the use of a single approach. The paper compares the main existing approaches to modeling and pattern discovery, illustrating the comparison on a case study of cancer prognostic data consisting of gene expression and clinical variables. The approaches discussed are: inductive and transductive reasoning; global, local and personalised modeling.

New methods are needed in the future for personalised modeling and for data and model integration. The paper introduces a new, simple method WWKNN for personalised modeling.

As a general conclusion, for a detailed study on a given problem and for the discovery of patters that characterise different aspects of the problems in hand, all these approaches need to be applied as an integrated multi-model system as proposed in Section 8.

Applications of the above methods for personalised medicine, personalised drug design, for building embedded systems in biological environments, for computational modeling of proteins and gene regulatory networks, and for many other challenging problems in Bioinformatics, Neuroinformatics, Medicine and Social Health are to be developed. A promising direction is computational neurogenetic modeling where the model is integrating genetic and neuronal information (Kasabov et al., 2005).

Acknowledgements

The work is funded by the NERF – FRST grant AUTX0201 at the Auckland University of Technology, New Zealand (www.aut.co.nz). The data analysis in the paper was conducted with the use of two software environments – NeuCom (www.theneucom.com, or www.kedri.info/) and *SIFTWARE* (available from Pacific Edge Biotechnology Ltd.) (www.peblnz.com). I would like to thank Dr. Qun Song, Peter Hwang, Nisha Mohan, Dougal Greer and other members of KEDRI for the implementation of some of the experimental software and for the useful discussions.

References

Alon, U., Barkai, N., et al., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS 96 (12), 6745–6750.

Ando, S., Sakamoto, E., et al., 2002. Evolutionary modelling and inference of genetic networks. In: The 6th Joint Conference on Information Sciences.

Arbib, M. (Ed.), 2003. The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge, MA.

Bajic, V., Seah, S., et al., 2003. Computer model for recognition of functional transcription start sites in RNA polymerase II promoters of vertebrates. J. Mol Graphics Model (21), 323–332.

Baldi, P., Brunak, S., 2001. Bioinformatics. A Machine Learning Approach. MIT Press, Cambridge, MA.

Bennett, K.P., Demiriz, A., 1998. Semi-supervised support vector machines. In: Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II. MIT Press, Cambridge, MA, USA.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

Bosnic, Z., Kononenko, I., et al., 2003. Evaluation of prediction reliability in regression using the transduction principle. EUROCON 2003. Computer as a Tool. The IEEE Region 8 2, 99–103.

Bower, J., Bolouri, H. (Eds.), 2001. Computational Modelling of Genetic and Biochemical Networks. The MIT Press, Cambridge, MA.

Chen, Y., Wang, G., et al., 2003. Learning with progressive transductive support vector machine. Pattern Recognition Lett. 24 (12), 1845–1855.

Collado-Vides, J., Hofestadt, R. (Eds.), 2002. Gene Regulation and Metabolism. Post-Genomic Computational Approaches. MIT Press, Cambridge, MA.

Crick, F., 1970. Central dogma of molecular biology. Nature 227, 561–563.

Dembele, D., Kastner, P., 2003. Fuzzy *C*-means method for clustering microarray data. Bioinformatics 19 (8), 973–980.

DeRisi, J., Penland, L., et al., 1996. Use of a cDNA microarray to analyse gene expression patterns in human cancer. Nat. Genet. 14 (4), 457–460.

D'Haeseleer, P., Liang, S., et al., 2000. Genetic network inference: From co-expression clustering to reverse engineering. Bioinformatics 16 (8), 707–726.

Dow, J., Lindsay, G., et al., 1995. Biochemistry Molecules, Cells and the Body. Addison-Wesley, Boston, MA.

Fogel, G., Corne, D., 2003. Evolutionary Computation for Bioinformatics. Morgan Kaufmann Publ.

Furey, T.S., Cristianini, N., et al., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16 (10), 906–914.

Futschik, M.E.K., N.K., 2002. Fuzzy clustering of gene expression data. Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and machine Learning. Addison-Wesley, Reading, MA.

Gollub, T.R., Slonim, D.K., et al., 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286 (5439), 531–537.

Gollub, J., Ball, C.A., et al., 2003. The Stanford microarray database: Data access and quality assessment tools. Nucl. Acids Res. 31 (1), 94–96.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI.

Joachims, T., 1999. Transductive inference for text classification using support vector machines. In: Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Joachims, T., 2003. Transductive learning via spectral graph partitioning. In: Proceedings of the Twentieth International Conference on Machine Learning, ICML-2003, Washington, DC.

Kasabov, N., 2000. Adaptive Learning method and system. University of Otago, New Zealand.

Kasabov, N., 2001. Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning. IEEE Trans. SMC – Part B, Cybernet. 31 (6), 902–918.

Kasabov, N., 2002. Evolving Connectionist Systems. Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines. Springer-Verlag, London.

Kasabov, N., Benuskova, L., in press. Computational neurogenetics. J. Comput. Theor. Nanosci. 1 (1).

Kasabov, N., Dimitrov, D., 2002. A method for gene regulatory network modelling with the use of evolving connectionist systems. In: ICONIP'2002 – International Conference on Neuro-Information Processing. IEEE Press, Singapore.

Kasabov, N., Pang, S., 2004. Transductive support vector machines and applications in bioinformatics for promoter recognition. Neural Inform. Process. – Lett. Rev. 3 (2), 31–38.

Kasabov, N., Song, Q., 2002. DENFIS: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction. IEEE Trans. Fuzzy Systems 10 (2), 144–154.

Kasabov, N., Reeve, A. et al., 2002. Medical applications of adaptive learning systems. PCT NZ03/00045. Pacific Edge Biotechnology Pte. Ltd., New Zealand.

Kasabov, N., Futschik, M.E., et al., 2003. Medical decision support systems utilizing gene expression and clinical information and methods for use. PCT/US03/25563, USA. Pacific Edge Biotechnology Pte Ltd., USA.

Kasabov, N., Benuskova, L., et al., 2005. Biologically plausible computational neurogenetic models: Modeling the interaction between genes, neurons and neural networks. J. Comput. Theor. Nanosci. 2 (4), 569–573.

Kohonen, T., 1997. Self-Organizing Maps. Springer Verlag.

Kukar, M., 2003. Transductive reliability estimation for medical diagnosis. Artif. Intell. Med. 29, 81–106.

LeCun, Y., Denker, J.S., et al., 1990. Brain damage. In: Touretzky, D.S. (Ed.), Advances in Neural Information Processing Systems. Morgan Kaufmann, San Francisco, CA, pp. 598–605.

Li, J., Chua, C.-S., 2003. Transductive inference for color-based particle filter tracking. In: Proceedings of International Conference on Image Processing, 2003. Nanyang Technol. Univ., Singapore.

Li, F., Wechsler, H., 2004. Watch list face surveillance using transductive inference. Lecture Notes in Computer Science 3072, pp. 23–29.

Li, C.H., Yuen, P.C., 2001. Transductive Learning: Learning Iris Data with Two Labeled DataICANN 2001. Springer Verlag, Heidelberg, Berlin.

Lukashin, A.V., Fuchs, R., 2001. Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters. Bioinformatics 17, 405–414.

Marnellos, G., Mjolsness, E.D., 2003. Gene network models and neural development. In: vanOoyen, A. (Ed.), Modeling Neural Development. MIT Press, Cambridge, MA, pp. 27–48.

Mitchell, M.T., Keller, R., et al., 1997. Explanation-based generalization: A unified view. Machine Learn. 1 (1), 47–80.

Perou, C., Sorlie, T., et al., 2000. Molecular portraits of human breast tumours. Nature, 406.

Proedrou, K., Nouretdinov, I., et al., 2002. Transductive confidence machine for pattern recognition. In: Proceedings of the 13th European Conference on Machine Learning. Springer-Verlag.

Quakenbush, J., 2002. Microarray data normalization and transformation. Nat. Genet. 32, 496–501.

Ramaswamy, S., Tamayo, P., et al., 2001. Multiclass cancer diagnosis using tumor gene expression signatures. Proceedings of the National Academy of Sciences of the United States of America 98 (26), 15149.

Shipp, M.A., Ross, K.N., et al., 2002a. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nat. Med. 8 (1), 68–74.

Shipp, M.A., Ross, K.N., et al., 2002b. Supplementary information for diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nat. Med. 8 (1), 68–74.

Singh, D., Febbo, P.G., et al., 2002. Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1, 203–209.

Snustad, D.P., Simmons, M.J., 2003. The Principles of Genetics. Wiley.

Sobral, B., 1999. Bioinformatics and the future role of computing in biology. From Jay Lush to Genomics: Visions for animal breeding and genetics.

Song, Q., Kasabov, N., 2004. TWRBF – Transductive RBF Neural Network with Weighted Data Normalization. Lecture Notes in Computer Science 3316, pp. 633–640.

Song, Q., Kasabov, N., 2005. NFI – Neuro-fuzzy inference method for transductive reasoning and applications for prognostic systems. IEEE Trans. Fuzzy Systems 13 (6), 799–808.

Sotiriou, C., Neo, S.-Y., et al., 2003. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. PNAS 100 (18), 10393–10398.

van de Vijver, M.J., He, Y.D., et al., 2002. A gene-expression signature as a predictor of survival in breast cancer. New Engl. J. Med. 347 (25), 1999–2009.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley Inter-Science.

Veer, L.J. v. t., Dai, H., et al., 2002. Gene expression profiling predicts clinical outcome of breast cancer. Nature 415 (6871), 530.

Vides, J., Magasanik, B., et al., 1996. Integrated Approaches to Molecular Biology. The MIT Press.

West, M., Blanchette, C., et al., 2001. Predicting the clinical status of human breast cancer by using gene expression profiles. PNAS 98 (20), 11462–11467.

Weston, J., Pérez-Cruz, F., et al., 2003. Feature selection and transduction for prediction of molecular bioactivity for drug design. Bioinformatics 19 (6), 764–771.

Wolf, L., Mukherjee, S., 2004. Transductive Learning via Model Selection. Massachusetts Institute of Technology, Cambridge, MA, The Center for Biological and Computational Learning.

Wu, D., Cristianini, N., et al., 1999. Large margin trees for induction and transduction. In: Proceedings for 16th International Conference of Machine Learning. Morgan Kaufmann, Bled, Slovenia.