INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

Interaction Selection and Complexity Control for Learning in Binarized Domains

Gerald Fahner

TR-96-001

May 1996

Abstract

We empirically investigate the potential of a novel, greatly simplified classifier design for binarized data. The generic model allocates a sparse, "digital" hidden layer comprised of interaction nodes that compute **Parity** of selected submasks of input bits, followed by a sigmoidal output node with adjustable weights. Model identification incorporates user-assigned complexity preferences. We discuss the situations: a) when the input space obeys a metrics, and b) when the inputs are discrete attributes. We propose a family of respective model priors that make search through the combinatorial space of multi-input interactions feasible. Model capacity and smoothness of the approximation are controlled by two complexity parameters. Model comparison over the parameter plane discovers models with excellent performance. In some cases interpretable structures are achieved. We point out the significance of our novel data mining tool for overcoming scaling problems, impacts on real-time systems, and possible contributions to the development of non-standard computing devices for inductive inference.

1 Introduction

In traditional Exploratory Data Analysis, model identification is often performed manually, by including a modest number of candidate variables and reasonable functions therof. Screening for interactions between variables is often required to enrich the model with reasonable interaction terms. Parameters are then estimated from the data within a fixed model structure. Eventually, the process is iterated with a modified or augmented model which takes into account additional interactions. Besides the principal difficulty of discovering unexpected dependencies during this biased way of modelling, huge databases with many variables that interact in complex, unpredictable ways can render the manual design cycle unmanageable (Elder and Pregibon 1996).
Vapnik emphasizes a complementary approach to data modelling, which is largely followed by the Neural Network community: *"Real-life problems are such that there exist a large number of "weak features" whose "smart" linear combination approximates the unknown dependency well. Therefore, it is not very important what kind of "weak feature" one uses, it is more important to form "smart" linear combinations."* (Vapnik 1995). Data mining tools operating accordingly must support automatic identification of the relevant interactions. The present paper focuses on three issues of model selection:

- **Sparseness**: how many interaction terms should a reasonable model include?
- **Preference**: which interaction terms are likely candidates for the problem at hand?
- **Simplicity**: can the computation of interactions be greatly simplified?

The first question is answered by an application of "Structural Risk Minimization" (Vapnik 1992). For the sparse multinomial model family discussed in this paper, a nested set of models of increasing size is searched and the optimum compromise between low training error and tight worst case bound for the test error is determined.
A good answer to the second question assigns a priori preferences to individual interactions in order to speed up the search process dramatically and to improve model performance over the worst case bounds. In the spirit of Bayesian inference, interactions that seem natural are added to the model with higher probability. Bayesian networks are special examples of this kind of reasoning. Of course, less obvious interactions should still be explored such that unexpected dependencies can also be identified. If no domain knowledge is available at all, "uninformed" priors may be assigned, e.g. by punishing high order or rapidly oscillating interactions.
The issue of computational complexity is often neglected in statistics but gains importance for mining huge amounts of data. It is also of much interest for the development of novel computing technology such as optical or biomolecular devices. Here, we focus on the valuable goal to simplify the computation of interaction terms greatly.
In the following, we discuss heuristic methods for the identification of sparse multinomial logistic models for binary data. Such data arise from a variety of circumstances, such as indicating presence or absence of certain attributes, the logical values of relational expressions, indicating if some real number exceeds a certain threshold, or representing numbers by their binary codes. In order to discover knowledge we may estimate joint probabilities

or perform soft classification of binary vectors $\underline{x} \in \{-1,1\}^N$ [1].

2 A Multinomial Logistic Model for Binary Data

We want to model a stochastic dependency between $\underline{x}$ and a two-valued outcome variable $y \in \{0,1\}$. The regression $p(y=1 \mid \underline{x})$ is estimated from a training database $\mathcal{T}$ of labeled examples $(\underline{x}^i; y^i)_{i=1}^{\#\mathcal{T}}$. For approximation of the regression, we use the logistic model

$$\hat{y} = \frac{1}{1+\exp\{-f(\underline{x},\underline{\theta})\}} \tag{1}$$

with the parametrized function family $f : \{-1,1\}^N \to I\!R$:

$$\begin{aligned} f(\underline{x},\underline{\theta}) &= \theta_0 + \theta_1\, x_1 + ... + \theta_N\, x_N && \text{additive terms} \\ &+ \theta_{1,2}\, x_1 x_2 + ... + \theta_{N-1,N}\, x_{N-1} x_N && 2^{nd}\,\text{order interactions} \\ &+ \theta_{1,2,3}\, x_1 x_2 x_3 + ... && \text{higher order interactions} \\ &+ ... \\ &+ \theta_{1,2,...,N}\, x_1 x_2 ... x_N \end{aligned} \tag{2}$$

subject to the constraint that

$$\underline{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ . \\ . \\ \theta_{1,2} \\ . \\ . \\ \theta_{1,2,...,N} \end{pmatrix}$$

is a *sparse* parameter vector with an *a priori fixed number* (not *set*!) of non-vanishing components.
A fixed-size model is fitted to the data by maximizing the objective function

$$log\ likelihood = \sum_{i=1}^{\#\mathcal{T}} y^i\ log\, \hat{y}(\underline{x}^i,\underline{\theta}) + (1-y^i)\, log\, (1-\hat{y}(\underline{x}^i,\underline{\theta}))\ .$$

Notice that maximization is over all models of given size and thus poses a hard combinatorial problem that can in general only be solved approximately.
The unconstrained expression (2) is known since a long time as "Walsh expansion" (Walsh 1923; Ahmed and Rao 1975). The additive and interaction terms together form the orthogonal Walsh base in function space. The expansion is universal: any function from the bipolar strings to the reals can be approximated arbitrarily close. Correspondingly, model

[1] For technical convenience, we make use of the bipolar $(-1,1)$ notation rather than the $(0,1)$ notation. The linear transformation $\xi_i = (1+x_i)/2$ switches between both representations.

(1) can represent arbitrary regression functions.
Unlike the full Walsh expansion, the sparse version is not capable of univeral function approximation, but by enlarging the number of interactions sufficiently, any probability distribution and thus any dichotomy over the input space can be approximated. This compares to corresponding theorems about universal function approximation by multilayer perceptrons including an unbounded number of hidden neurons. For both model classes, determination of a reasonable size becomes important if the model is fitted from limited, noisy data.
The second and higher order interaction terms can be considered also as hidden nodes in a sparse network. Each node basically evaluates the Parity predicate (in the $(0,1)$ representation) over some selected sub-mask of input bits, which can be done in parallel. Heuristic supervised learning algorithms for such models were proposed for classification problems of unknown order (Fahner and Eckmiller 1994), and applied for complicated robot navigation problems (Fahner 1995).

3 Searching Sparse Multinomials

Model identification requires determination of model size and set of interactions, while parameter estimation yields the respective weights $\theta_{i,k,\ldots}$. The algorithm presented in the box below is applied for simultaneous identification and estimation. The interplay between the basic modules "interaction sampling" and "model identification" is depicted in Fig.1.

1) chose model size within interval $[\frac{\#\mathcal{T}}{100}, \#\mathcal{T}]$
2) chose prior distribution $p(complexity)$ for individual interactions
3) initialize model with tentative interactions drawn according to $p(complexity)$
4) maximize *log likelihood* $(\underline{\theta} \mid \mathcal{T})$ and obtain weight vector $\underline{\theta}^*$
5) prune brittle interactions from the model
6) chose novel tentative interactions to replace the pruned ones
7) back to 4) until stopping criterium is met; output final model

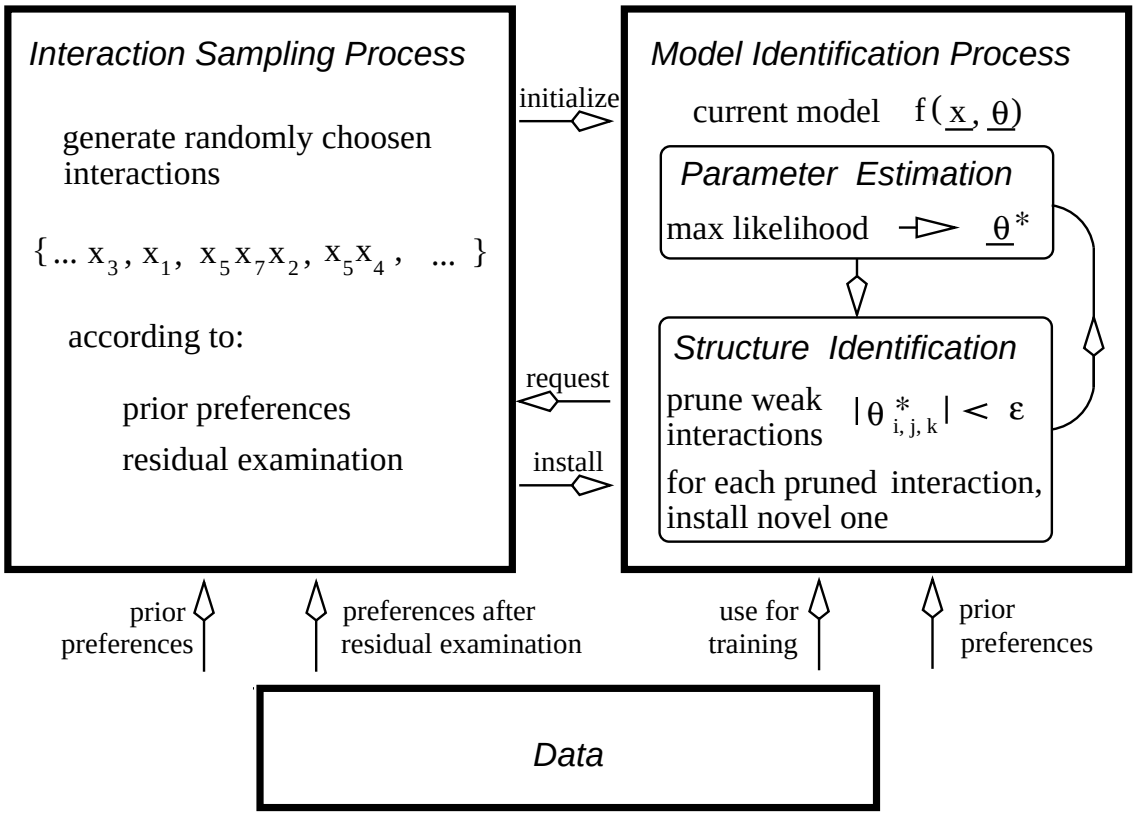

Figure 1 : Heuristic search for a sparse multinomial classifier. The left module generates randomly chosen interaction terms according to user-specified prior preferences. Also, error information is used in a greedy way to accelerate the learning process: individual interactions are tested separately; those that exhibit a significant correlation with the residual error are more likely to be installed in the model.
The right module keeps the current model, which is repeatedly modified by iterative application of parameter estimation followed by parameter inspection and structure modification. In the simplest case an interaction is pruned if its coefficient falls below some threshold. More advanced pruning mechanisms include prior preferences or statistical significance tests to reveal "brittle" interactions. For each pruned interaction, a request for novel interaction generation is sent to the sampling module.

The maximization required in step 4) of the above algorithm is over a fixed set of parameters. The likelihood function possesses a single maximum. In our implementation, we use a second order gradient method for its determination.
The stopping criterion of the algorithm varies with applications. For any preselected model size and prior distribution, the algorithm outputs a sparse multinomial expansion that approximates the training data. Search for the best model (minimum test error) is over the two-dimensional parameter plane spanned by model size and the single parameter μ which regulates the form of the prior distribution of interaction complexities (see Fig.2).

An ensemble of networks is drawn from the size-μ plane and trained; the best model is chosen by crossvalidation. In our implementation, we distribute several models of varying complexity over a network of workstations. No communication is required between the individual learning processes.

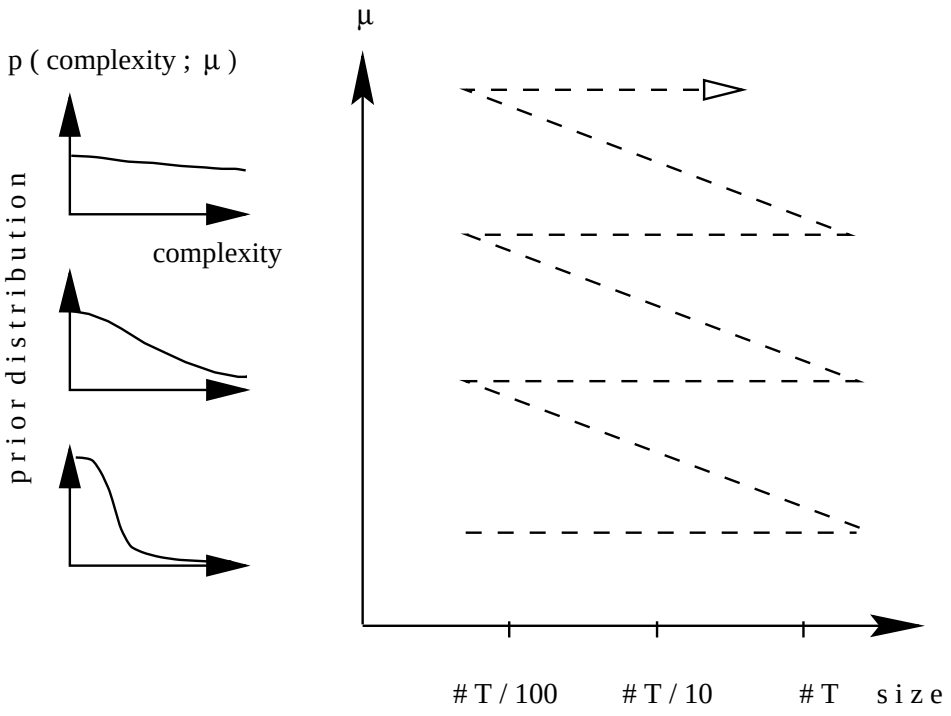

Figure 2 : Capacity control plane: with increasing size, reduction of training error is possible at the expense of overfitting. With increasing μ, the effective model space is enlarged, and an increasing number of complex interactions is likely to be present in the sparse multinomial.

4 Complexity Measures for Interaction Terms

For two types of input space semantics: a) binary representation of metrical input data, b) binary encodings of discrete attribute values, we propose respective complexity measures for individual interactions:

- **zero crossings**: maximum number of sign flips along straight line through input space (a).
- **order**: number of multiplicative factors included in the interaction (b)

Fig.3 illustrates case a) for a two-dimensional rectangular input space. Both continuous $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$ axes are uniformly discretized into 4 intervals. For each dimension, the intervals are encoded by increasing binary numbers ($-$ stands for 0, $+$ for 1), preserving the order relation of the alligned intervals. The given example generalizes to higher dimensions, to arbitrary binary resolutions which are individually assigned to each dimension and to nonuniform parcellings.

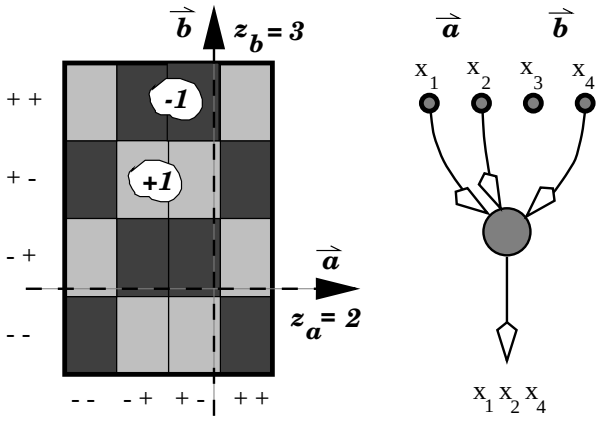

Figure 3 : Behavior of the interaction term $x_1x_2x_4$ in two dimensions. Each box in the rectangular region is encoded as a 4-tuple $x_1x_2x_3x_4$ formed by the concatenation of the discretized and binarized coordinate values of the boxes. The interaction term oscillates between -1's and 1's, undergoing zero crossings at some box borders. Along the two dashed arrows, the number of zero crossings z_a and z_b is counted separately for both coordinate axes. The maximum achievable number of zero crossings for an arbitrary direction line-sweep is given by $z_a + z_b = 5$.

5 Simulation Results

We illustrate the working of the sparse multinomial classifier for the **2-spirals** problem. In the original formulation (Lang and Witbrok 1988), the classifier has to find a decision surface to separate two continuous point sets in $I\!R^2$ that belong to one or the other of intertwined spirals. The problem is formulated for binarized inputs as follows: each point in the plane is represented by some bitstring B_xB_y which is the concatenation of the truncated binary expansions for the points x- and y-coordinates. We chose 7 bit resolution for each coordinate, which is much more than required to distinguish between any two training examples.

A particular choice of coordinate axes breaks isotropy as well as shift-invariance of the original problem, since the Walsh functions are not invariant under the translation operator. To

restore the effect of broken symmetries, we apply the binarization for a transformation set of 300 randomly shifted and rotated (around coordinate center $(0,0)$, not around center of the spirals, since we assume no a priori knowledge) versions of the original input vectors. For each coordinate system, a separate model is trained. Size and prior complexity are constant over all members of a transformation set. Generalization over the whole square $[0,1)^2$ is computed by averaging over the outputs of all members. Training is stopped as soon as all training examples are correctly classified or no further improvement can be achieved within a reasonable number of iterations.
Fig.4 shows different generalizations achieved for models of size = 50 but with prior distributions favouring increasingly complex models.

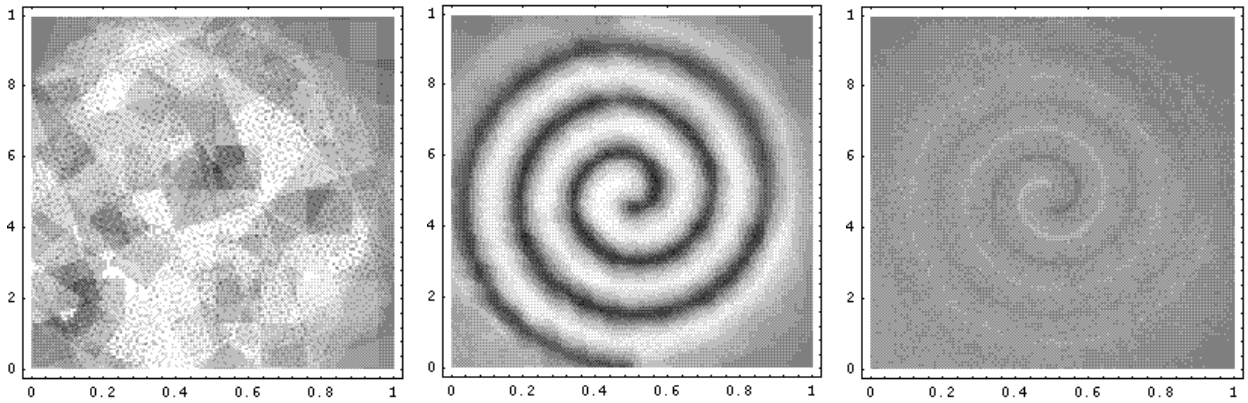

Figure 4 : Classification result on the two-spirals problem: generalization behaviour of three models of same size but with varying complexity of interactions. Model a): interactions with moderate and large numbers of zero crossings are strongly penalized. The model does not learn the oscillating spiral pattern effectively. Model b): has the right prior to learn the desired dependency quickly. Model c): is dominated by rapidly oscillating interactions that „memorize" the training examples rather than learning the concept of a spiral.

Model b) in the above figure uses only the 4 most significant bits for each dimension and doesn't contain interactions with more than 15 zerocrossings. Model c) contains significant contributions of rapidly oscillating interaction terms due to rich connectivity to the least significant bits.
The second task is the **Gene** benchmark (Nordewier, Towell and Shavlik 1991) for predicting splice-junctions. From a window of 60 DNA sequence elements it is to be inferred whether the middle is an Intron-Exon (I-E) boundary, and Exon-Intron (E-I) boundary, or neither of both. The dataset contains 3175 labeled gene strings made of nucleotides C, A, G, T (for Cytosin, Adenin, Guanin and Thymin). The elements are encoded as binary tuples $(-1,-1),(-1,1),(1,-1),(1,1)$, respectively, yielding 120 inputs. Three models are trained

separately to discriminate each class against the rest classes. 900 patterns are used for training and 100 for crossvalidation, the rest for testing. Training is stopped when no further improvement can be achieved on the crossvalidation set within a reasonable number of iterations. The performance of any model evaluated during search is monitored and the best model ever produced along the search trajectory is finally used. Model sizes are $20, 40, 60$ terms for the E-I, the I-E, and the NONE-classifier, respectively, Best complexity priors are in the linear regimen. The best models contain only a few second and no significant higher order interaction. The test error is around 7%, comparable to results from literature with MLP's, and superior to experiments with $ID3$ and Nearest Neighbor (Murphy and Aha 1992).

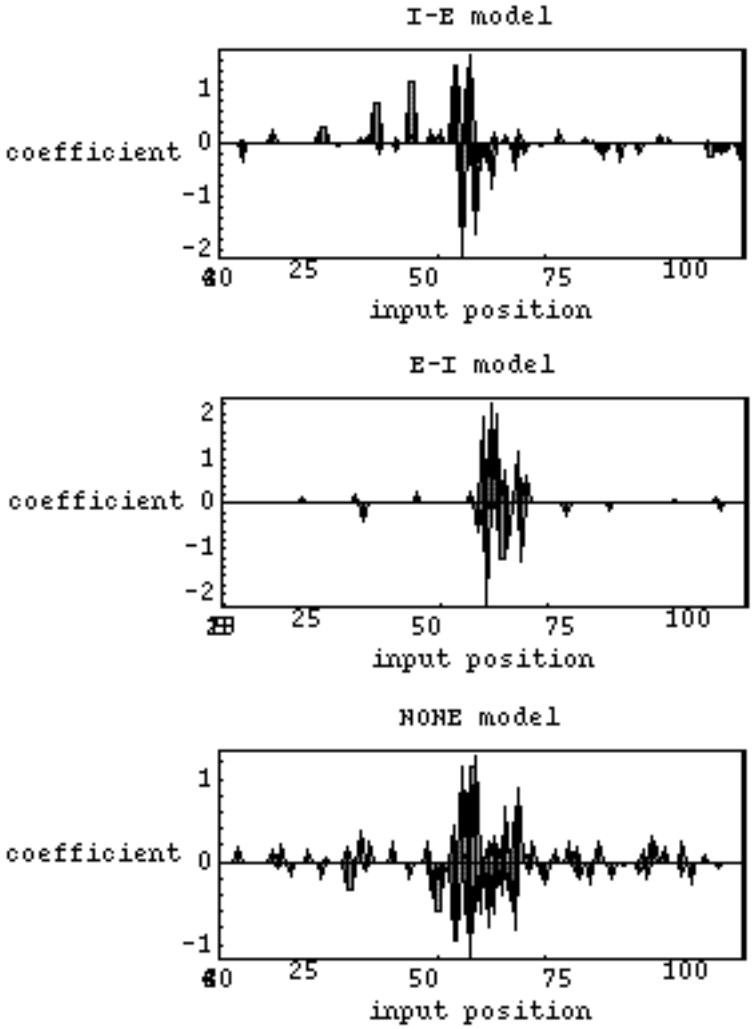

Figure 5 : Structure histograms for the splice-junction predictor: the 120 inputs are ordered along the x-axis. The vertical axis measures absolute weight coefficients assigned to each interaction. A peak at a certain location of the input string indicates that the corresponding bit contributes (in an additive way or as member of an interaction term) to the classifier decision; the height of the peak measures the strength of this contribution.

The simple diagrams can reveal only limited information about the structure of interaction terms, but a striking observation is that genes in the local neighborhood of the junction have the most impact on the type of junction.

6 Discussion

The paper adresses several topics of interest for data mining and contributes original research to the topics: integrated data and knowledge representation for numeric and categorical data, algorithmic complexity and scalability issues, and distributed search for the best model.
A computationally feasible automatic exploration method is presented that identifies the relevant interactions between binary variables. The foremost relevance of our novel mining tool for KDD lies in the fact that automatic exploration of possible interactions results in a much faster, less biased, and wider applicable modelling process than traditional style Explorative Data Analysis. This makes our data mining tool a good candidate for real-time and high dimensional data analysis.
A serious challenge for any automatic procedure that searches wide model classes is the problem of overfitting. We overcome these difficulties by incorporating powerful novel regularization techniques for binary data formats.
Challenged by the ongoing explosive growth of amounts of data and by shrinking time spans for real-time data analysis and decision making, we speculate that a quantum jump in device technology for KDD systems will be needed to overcome the severe scaling problems. Our findings show that expensive floating point multiplications can be entirely avoided for a large class of inductive inference problems. The bit-interactions which make up the "atomic" knowledge entities of our model seem well suited for parallel distributed manipulation and processing by quantum devices. Models similar in spirit could simplify the development of a quantum computer for large scale, high-speed inductive inference.

References

Ahmed, N.; and Rao, K. R. 1975. *Orthogonal Transforms for Digital Signal Processing.* New York: Springer.

Elder IV, J. F.; and Pregibon, D. 1996. A statistical perspective on knowledge discovery in databases. In: *Advances in Knowledge Discovery and Data Mining.* U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds. The MIT Press, Menlo Park, CA.

Fahner G.; and Eckmiller, R. 1994. Structural adaptation of parsimonious higher-order neural classifiers. Neural Networks **7**(2), pp. 279-289.

Fahner G. 1995. Vehicle guidance in dynamic environments based on pattern recognition of quasilocal spacetime embeddings. In: Proceedings of the International Conference of Artificial Neural Networks (ICANN-95). Springer-Verlag, Heidelberg.

Lang, K. J.; and Witbrok M. 1988. Learning to tell two spirals apart. In: Proceedings of the 1988 Connectionists Model Summer School, pp. 52-59. D. S. Touretzky, G. E. Hinton, T. J. Sejnowski, eds. New York: Morgan Kaufmann Publishers.

Murphy, P. M.; and Aha, D. W. 1992. UCI Repository of machine learning databases, ftp-site: ics.uci.edu, University of California, Irvine, CA.

Noordewier M. O.; Towell G. G.; and Shavlik J. W. 1991. Training knowledge-based neural networks to recognize genes in DNA sequences. In: *Advances in Neural Information Processing Systems* **3**, pp 530-536. R. P. Lippmann, J. E. Moody, D. S. Touretzky, eds. San Mateo, CA: Morgan Kauffman.

Vapnik, V. N. 1992. Principles of risk minimization for learning theory. In: *Advances in Neural Information Processing Systems* 4, pp. 831-838. J. E. Moody, S. J. Hanson, R. P. Lippmann, eds. San Mateo, CA: Morgan Kaufmann.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory.* New York: Springer.

Walsh, J. L. 1923. A closed set of orthogonal functions. American Journal of Mathematics 45, pp. 5-24.