Crisis Monitoring: Methods for Heterogeneous Time Series Learning

William H. Hsu, Nathan D. Gettings, Victoria E. Lease, Yu Pan, and David C. Wilkins

Beckman Institute, University of Illinois at Urbana-Champaign
Knowledge Based Systems Laboratory, http://www-kbs.ai.uiuc.edu
405 North Mathews Avenue, Urbana, IL 61801
{w-hsu | gettings | lease | y-pan2 | dcw}@uiuc.edu

Abstract

A very important application of time series learning is online diagnosis, or *monitoring*, to detect and classify hazardous conditions in a physical system. Examples of *crisis* monitoring in the industrial, military, agricultural and environmental sciences are numerous. This paper first defines *heterogeneous* time series, those containing different types of embedded, statistical patterns. Next, it surveys basic techniques for acquiring several types of temporal models (using artificial neural networks and Bayesian networks). A new, multistrategy system for learning heterogeneous time series is then presented; it uses task decomposition and quantitative metrics to select techniques for each identifiable (and relevant) embedded subproblem. This multistrategy solution is briefly compared to some mixture models for recombining specialized classifiers. The validation experiments use two large-scale applications, shipboard damage control automation and crop monitoring in precision agriculture. This paper concludes with a report on work in progress and some early positive learning results regarding these application domains.

Keywords: **multistrategy learning, heterogeneous time series, crisis monitoring and diagnosis, agricultural applications, damage control**

Introduction

This paper discusses the problem of learning from time series data in order to predict hazardous and potentially catastrophic conditions. This prediction task is also known as *crisis monitoring*, a form of pattern recognition that is useful in decision support (or *recommender* [KV97]) systems for many time-critical applications. These include crisis control automation [Hs97, WS97], online medical diagnosis [HLB+96], simulation-based training and critiquing for crisis management [GD88, GFH92, MW96, WFH+96, GHVW98], and intelligent data visualization [HB95].

Crisis monitoring is a highly informative experimental benchmark for many time series learning systems, because it tests their predictive capability in extreme cases. Unfortunately, many time series learning methods fail to produce models that can predict imminent *catastrophic* events [GW94]; yet these predictive models are otherwise reliable. This limitation is due in part to the *heterogeneity,* or multimodality, of historical time series data [HR98b, RH98]. For example, a *drought monitoring* system for agricultural applications must account for several *modalities* or aspects of drought: climatic, hydrological, crop-specific, and economic [Pa65, Al84]. These drought-related phenomena are often tracked at varying spatial and temporal scales (e.g., due to granularity of sensors) and observed through multiple attributes (or *channels*). It is especially difficult to predict catastrophe for heterogeneous models, because of the *interaction* among such diverse components of a crisis [HR98b, HGL+98].

This paper proposes the systematic decomposition of learning tasks [FB85] as a partial solution to the problem of heterogeneity in monitoring problems. Such a process can alleviate problems that arise from having multimodal inputs and diversity in scale and structure. Equally important, it supports *technique selection* to identify the most appropriate learning architecture for each homogeneous component of a time series. Moreover, it accounts for previously known subdivision of time series learning tasks due to sensor specifications, knowledge about *relevance*, and complexity-reducing methods.

Several architectures and inductive learning algorithms that apply to artificial neural networks (ANNs) and Bayesian networks are presented, along with the types of basic temporal models they can represent and learn. The research documented here addresses how task decomposition, model selection, and data fusion can be applied *together* to handle heterogeneity in time series learning. The design of such an integrated system is described, and some preliminary results on two monitoring domains are presented.

The key novel contributions of the system are:

1. The explicit organization of learning components into recombinable and reusable classes
2. **Metrics for temporal characteristics of data sets that indicate the appropriate learning technique**
3. A framework for decomposing learning tasks and combining classifiers learned by different techniques

This paper concentrates on the second contribution: quantitative methods for model selection in time series [HR98a]. The authors' current research is primarily concerned with cases where subtasks are formed by *constructive induction* methods [DM83], which may incorporate prior knowledge (cf. [DR95]). The emphasis of this paper is on machine learning methods for crisis

monitoring (especially model selection and representation), so it touches on task decomposition and data fusion only when relevant.

Model Decomposition, Selection, and Fusion

This section first surveys three types of *linear* models [GW94, Mo94] for time series learning. It then presents a systematic approach to decomposition of time series, and an algorithm for quantitative model selection that analyzes the components thus produced to choose the most appropriate linear or Bayesian model and learning technique. Finally, a data fusion approach is given, which recombines the "piecewise" models learned using this method. The survey of precision agriculture applications later in this paper gives an example of a heterogeneous time series database containing different linear processes, and how simple model selection can improve learning for agricultural crisis monitoring.

Linear Models and Heterogeneous Time Series

To model a time series as a stochastic process, one assumes that there is some mechanism that generates a random variable at each point in time. The random variables *X(t)* can be univariate or multivariate (corresponding to single and multiple attributes or *channels* of input per exemplar) and can take discrete or continuous values, and time can be either discrete or continuous. For clarity of exposition, the experiments focus on discrete classification problems with discrete time. The classification model is *generalized linear regression* [Ne96], also known as a *1-of-C coding* [Sa98] or *local coding* [KJ97].

Following the parameter estimation literature [DH73], time series learning can be defined as finding the parameters $\Theta = \{\theta_1, \ldots, \theta_n\}$ that describe the stochastic mechanism, typically by maximizing the likelihood that a set of realized or *observable* values, $\{x(t_1), x(t_2), \ldots, x(t_k)\}$, were actually generated by that mechanism. This corresponds to the backward, or maximization, step in the *expectation-maximization (EM)* algorithm [DH73]. Forecasting with time series is accomplished by calculating the conditional density $P(X(t) \mid \{\Theta, \{X(t-1), \ldots, X(t-m)\}\})$, when the stochastic mechanism and the parameters have been identified by the observable values *{x(t)}*. The order *m* of the stochastic mechanism can, in some cases, be infinite; in this case, one can only approximate the conditional density.

Despite recent developments with nonlinear models, some of the most common stochastic models used in time series learning are parametric linear models called *autoregressive (AR), moving average (MA)*, and *autoregressive moving average (ARMA)* processes.

MA or moving average processes are the most straightforward to understand. First, let *{Z(t)}* be some fixed zero-mean, unit-variance "white noise" or "purely random" process (i.e., one for which $Cov[Z(t_i), Z(t_j)] = 1$ *iff* $t_i = t_j$, 0 otherwise). *X(t)* is an *MA(q)* process, or "moving average process of order *q*", if $X(t) = \sum_{\tau=0}^{q} \beta_\tau Z(t-\tau)$, where the β_τ are constants. It follows that $E[X(t)] = 0$ and $Var[X(t)] = \sum_{\tau=0}^{q} \beta_\tau$. Moving average processes are often used to describe stochastic mechanisms that have a finite, short-term, linear "memory" [Mo94, Ch96, MMR97, PL98].

AR or autoregressive processes are processes in which the values at time *t* depend linearly on the values at previous times. With *{Z(t)}* as defined above, *X(t)* is an *AR(p)* process, or "autoregressive process of order *p*", if $\sum_{\upsilon=0}^{p} \alpha_\upsilon X(t-\upsilon) = Z(t)$, where the α_υ are constants. In this case, $E[X(t)] = 0$, but the calculation of $Var[X(t)]$ depends upon the relationship among the α_υ; in general, if $|\alpha_\upsilon| \geq 1$, then *X(t)* will quickly diverge. Autoregressive processes are used to capture "exponential traces" in a time series; they are equivalent to infinite-length *MA* processes constants [Mo94, Ch96, MMR97, PL98].

ARMA is a straightforward combination of *AR* and *MA* processes. With the above definitions, an *ARMA(p, q)* process is a stochastic process *X(t)* in which $\sum_{\upsilon=0}^{p} \alpha_\upsilon X(t-\upsilon) = \sum_{\tau=0}^{q} \beta_\tau Z(t-\tau)$, where the $\{\alpha_\upsilon, \beta_\tau\}$ are constants [Mo94, Ch96]. Because it can be shown that *AR* and *MA* are of equal expressive power, that is, because they can both represent the same linear stochastic processes (possibly with infinite *p* or *q*) [BJR94], *ARMA* model selection and parameter fitting should be done with specific criteria in mind. For example, it is typically appropriate to balance the roles of the *AR(p)* and *MA(q)*, and to limit *p* and *q* to small constant values for tractability (empirically, 4 or 5) [BJR94, Ch96, PL98].

In *heterogeneous* time series, the embedded temporal patterns belong to different categories of statistical models, such as *MA(1)* and *AR(1)*. Examples of such embedded processes are presented in the discussion of the experimental test beds. A multichannel time series learning problem can be decomposed into homegeneous subtasks by aggregation or synthesis of attributes. *Aggregation* occurs in multimodal sensor fusion (e.g., for medical, industrial, and military monitoring), where each group of input attributes represents the bands of information available to a sensor [SM93]. In geospatial data mining, these groupings may be topographic [Hs97]. Complex attributes may be *synthesized* explicitly by constructive induction, as in causal discovery of latent (hidden) variables [He96]; or implicitly by preprocessing transforms [HR98].

Learning Architecture	Architectural Metric
Simple recurrent network (SRN)	Exponential trace (AR) autocorrelation
Time delay neural network (TDNN)	Moving average (MA) autocorrelation
Gamma network	Autoregressive moving average (ARMA) autocorrelation
Temporal naïve Bayesian network	Relevance score
Hidden Markov model (HMM)	Test set perplexity

Table 1. Learning architectures and their prescriptive metrics

Learning Method	Distributional Metric
HME, gradient	Modular cross entropy
HME, EM	Modular cross entropy + missing data noise
HME, MCMC	Modular cross entropy + sample complexity
Specialist-moderator, gradient	Dichotomization ratio
Specialist-moderator, EM	Dichotomization ratio + missing data noise
Specialist-moderator, MCMC	Dichotomization ratio + sample complexity

Table 2. Learning methods and their prescriptive metrics

Quantitative Model Selection

This section presents a new metric-based, multistrategy model selection system, and gives an algorithm for selecting the learning technique most strongly *indicated by the data set characteristics.*

Model selection is the problem of choosing a hypothesis class that has the appropriate complexity for the given training data [Sc97]. Quantitative, or *metric-based*, methods for model selection have previously been used to learn using highly flexible models with many degrees of freedom, but with no particular assumptions on the structure of decision surfaces (e.g., that they are linear or quadratic) [GBD92]. Learning without this characterization is known in the statistics literature as *model-free estimation* or *nonparametric statistical inference*. The premise of this paper is that for heterogeneous time series learning problems, indiscriminate use of nonparametric models such as feedforward and recurrent artificial neural networks is too unmanageable. This is especially true in crisis monitoring because decision surfaces are more sensitive to error when the target concept is a catastrophic event.

The remainder of this section describes a novel type of metric-based model selection that selects from a known, fixed "repertoire" or "toolbox" of learning techniques. This is implemented as a "lookup table" of *architectures* (rows) and *learning methods* (columns). Each architecture and learning method has a characteristic that is positively (and uniquely, or almost uniquely) correlated with its expected performance on a time series data set. For example, naïve Bayes is most useful for temporal classification when there are many discriminatory observations (or *symptoms*) all related to the hypothetical causes (or *syndromes*) that are being considered [He91, Hs98]. The strength of this characteristic is measured by an *architectural* or *distributional* metric. Each is normalized and compared against those for other (architectural or distributional) characteristics. For example, the architectural metric for temporal naïve Bayes is simply a score measuring the degree to which observed attributes are *relevant* to discrimination of every pair of hypotheses. The "winning" metrics thus identifies the dominant characteristics of a *subset* of the data (if this subset is sufficiently homogeneous to identify a single winner). These subsets are acquired by selecting *input attributes* (i.e., channels of time series data) from the original exemplar definition (cf. [KJ97]).

The next section describes a database of available learning architectures and methods (mixture models and algorithms). Based on the formal characterization of these learning techniques as time series models [GW94, Mo94, MMR97], indicator metrics can be developed for the *temporal structure* and *mixture distribution* of a *homogeneous* time series (i.e., one that *has* identifiable dominant characteristics). The highest-valued (normalized) *architectural* metric is used to select the learning architecture (Table 1); the highest for *distribution* is used to select the learning method (Table 2). The metrics are called *prescriptive* because each one provides evidence in favor of an architecture or method.

Database of Learning Components

Table 1 lists five learning architectures (the rows of a "lookup table") and the indicator metrics corresponding to their strengths [Hs97]. The principled rationale behind the design of these metrics is that each is based on an attribute chosen to *correlate positively* (and, to the extent feasible, *uniquely*) with the *characteristic memory form* of a time series. A *memory form* as defined by Mozer [Mo94] is the representation of some specific temporal pattern, such as a limited-depth buffer, exponential trace, gamma memory [PL98], or state transition model.

SRNs, TDNNs, and gamma networks are all temporal varieties of artificial neural networks (ANNs) [MMR97]. A *temporal naïve Bayesian network* is a *global knowledge map* (as defined by Heckerman [He91]) with two stipulations. The first is that some random variables may

be temporal (e.g., they may denote the durations or rates of change of original variables). The second is that the topological structure of the Bayesian network is learned by naïve Bayes. A hidden Markov model (HMM) is a stochastic state transition diagram whose transitions are also annotated with probability distributions (over output symbols) [Le89].

The prototype *architectural metrics* for temporal ANNs are average autocorrelation values for the preprocessed data. Memory forms for temporal ANNs can be characterized using a formal mathematical definition called the *kernel function*. *Convolution* of a time series with this kernel function produces a transformed representation under its memory form [Mo94, MMR97]. The design principle behind the *architectural* metrics for temporal ANNs is that a memory form is strongly *indicated* if the transformed time series has a high autocorrelation.

For example, to compute autocorrelation for an *AR(p)* model, convolution of an exponential decay window (an *AR kernel function*) is first applied [MMR97]. This estimates the predictive power of the model *if chosen as the learning architecture*. The score for temporal naïve Bayesian network is the average number of variables *relevant* to each pair of diagnosable causes (i.e., hypotheses) [He91]. This score is computed by constructing a Bayesian network by naïve Bayes [Pe88] and then averaging a *relevance measure* (cf. [KJ97]) on the conditional distribution of symptoms (input attributes) versus syndromes (hypotheses). This relevance measure may be as simple as an average of the number of relevant attributes. Finally, the indicator metric for HMMs is the empirical *perplexity* (arithmetic mean of the branch factor) for a constructed HMM [Le89].

Table 2 lists six learning methods (the columns of the "lookup table"). A *hierarchical mixture of experts* (HME) is a mixture model composed of generalized linear elements (as used in feedforward ANNs) [JJHN91, JJ94]. It can be trained by gradient learning, expectation-maximization [JJ94], or Markov chain Monte Carlo (MCMC) methods (i.e., random sampling as in the Metropolis algorithm for simulated annealing) [MMR97]. A *specialist-moderator network*, which also can combine predictions from different learning architectures, is a mixture model whose components have different input and output attributes. Specialist-moderator networks are discussed briefly in the section below on data fusion; the interested reader is referred to [HR98a, RH98]. The prototype *distributional* metrics for HME networks are based on modular cross entropy (i.e., the Kullback-Leibler distance between conditional distributions in each branch of the tree-structured mixture model) [JJ94]. The metrics for specialist-moderator networks are proportional to dichotomization ratio (the number of distinguishable equivalence classes of the overall mixture divided by the product of its components') [HR98a]. To select a learning algorithm, gradient learning is defined as a baseline, and a term is added for the gain from estimation of missing data (by EM) [JJ94] or global optimization (by MCMC) [Ne96], adjusted for the conditional sample complexity.

Composite Learning

This section defines *composites*, which are specifications of high-level extracted attributes, together with the learning architecture and method for which this alternative representation shows the strongest evidence. Composites are generated using the algorithm below.

Definition. A *composite* is a set of tuples $\mathbf{L} = ((A_1, B_1, \theta_1, \gamma_1, S_1), \ldots, (A_k, B_k, \theta_k, \gamma_k, S_k))$, where A_i and B_i are constructed input and output attributes, θ_i and γ_1 are network parameters and hyperparameters cf. [Ne96] (i.e., the learning architecture), and S_i is a learning method.

The general algorithm for composite time series learning follows.

Given:

1. A (multichannel) time series data set $D = ((\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}))$ with input attributes $\boldsymbol{A} = (a_1, \ldots, a_I)$ such that $\boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_I^{(i)})$ and output attributes $\boldsymbol{B} = (b_1, \ldots, b_O)$ such that $\boldsymbol{y}^{(i)} = (y_1^{(i)}, \ldots, y_O^{(i)})$
2. A constructive induction function F such that $F(A, B, D) = \{(A', B')\}$

Normalization formulas for metrics x_τ (τ = metric number)

t_τ: shape parameter
λ_τ: scale parameter

$$G_\tau(x_\tau) = \int_0^{x_\tau} f_\tau(x)\,\mathrm{d}\,x$$

$$f_\tau(x) = \frac{\lambda_\tau\, \mathrm{e}^{-\lambda_\tau x} (\lambda_\tau x)^{t_\tau - 1}}{\Gamma(t_\tau)}$$

$$\Gamma(t_\tau) = \int_0^{\infty} \mathrm{e}^{-y}\, y^{t_\tau - 1}\,\mathrm{d}\,y$$

Figure 1. Normalization of metrics

Algorithm **Select-Net** *(D, **A**, **B**, F)*

repeat

Generate a candidate representation (e.g, attribute subset [KJ97]) $(A', B') \in F(A, B, D)$.

Compute *architectural* metrics $\boldsymbol{x}_\tau^a = m_\tau^a(A', B')$ that prescribe the learning architecture.

Compute *distributional* metrics $\boldsymbol{x}_\tau^d = m_\tau^d(A', B')$ that prescribe the learning method.

Normalize the metrics $\boldsymbol{x}_\tau$ using a precalibrated function $\boldsymbol{G}_\tau$ – see Figure 1.

Select the most strongly prescribed architecture (θ, γ) and learning method S for (A', B'), i.e., the table entry (row and column) with the highest metrics.
if the fitness (strength of prescription) of the selected model meets a predetermined threshold
then accept the proposed representation and learning technique $(A', B', \theta, \gamma, S)$
until the set of plausible representations is exhausted
Compile and train a *composite,* **L**, from the selected complex attributes and techniques.
Compose the classifiers learned by each component of **L** using data fusion.

The normalization formulas for metrics simply describe how to fit a multivariate gamma distribution f_τ based on a *corpus of homogeneous data sets* (cf. [HZ95]). Each data set is a "training point" for the metric normalization function, G_τ (i.e., the shape and scale parameters of f_τ).

Data Fusion for Decomposable Learning Tasks

The recombination of time series components (classes in the attribute subset partition) is achieved using one of two hierarchical mixture models. One is the HME network developed by Jordan *et al* [JJ94]. The other is a *specialist-moderator* network, a new hierarchical mixture model developed by Ray and Hsu [RH98, HR98a]. Specialist-moderator networks combine the predictions of differently targeted inductive generalizers (e.g., *specialist* ANNs that receive a subset of the input attributes or channels, and are trained to predict *equivalence classes* of the overall targets). A companion paper, which appears as an extended abstract in [HR98a], describes how specialist-moderator networks outperform non-modular ANNs on time series classification tasks that admit an efficient decomposition. It shows how, using the same decomposition, they can outperform *partitioning mixtures* such as *hierarchical mixtures of experts (HME)* [JJ94], given identical constraints on network complexity and convergence time. Another example of a partitioning mixture that is well studied in the machine learning literature is *boosting* [FS96]. We also considered how specialist-moderator networks can be combined with *aggregation mixtures* such as *bootstrap aggregation (bagging)* [Br96] and *stacked generalization* [Wo92].

Test Bed Domains for Time Series Learning

Crisis Monitoring in Shipboard Damage Control

Figure 2 depicts a *shipboard damage control* training system called *DC-Train*, developed at the Knowledge Based Systems Laboratory at University of Illinois [WFH+96, WS97]. The visualization shown is part of the user interface for an immersive tutoring system for *damage control assistants (DCAs)*, officers who coordinate damage control activity during shipboard crises. The purpose of *DC-Train* is to provide realistic simulation-based training (in time-constrained problem solving under stress) to DCA trainees; to offer critiques and guidance (up to and including automated control of the student's role in the simulation); and to synthesize more effective scenarios based on user (student and instructor) modeling.

Shipboard damage control crisis monitoring is also useful in *recommender systems* [KV97] for control automation [WS97, Bu98]. An intelligent problem solving system with predictive capabilities could be applied for partial automation of the damage control organization, thereby reducing manning aboard Navy vessels where such a system is deployed [WS97].

The visualized data in Figure 2 is compartment status – specifically, combustion state, flooding and rupture state, and crisis status based on reports from *damage control agents* such as investigators, firefighters, and repair personnel). We have developed systems for diagnosis of hazard level. In our preliminary studies, this is a "danger" rating, possibly measured as the number of minutes until a lethal explosion, total shipwide conflagration, instability due to flooding, or other catastrophic event. In our computer-assisted instruction application, this is known as a *kill point* that terminates the scenario.

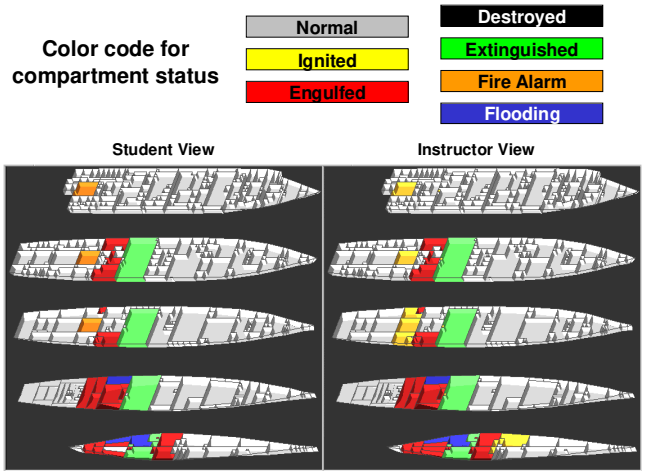

Figure 2. Shipboard damage control training simulation

Our intelligent system for time series analysis is presented with hundreds of input channels from throughout the ship, which comprises all localized temperature and gas content readings, as well as other information about doors, flooding, etc. All readings are digital, but at a high enough precision to make continuous methods desirable. Accurate fire detection is one of our primary low-level classification objectives – that is, we want to develop a system that can accurately tell us whether or not there is a fire, and if there is, what fire boundaries to establish for fire fighting purposes. Minimization of false positives is extremely important, simply for efficiency's sake. Because training data is somewhat sparse, we need analysis that can produce good

estimators with little training data and that will exhibit graceful degradation of performance when presented with novel or noisy data. Recurrent and time-lagged neural networks are particularly attractive as components of our system, since they meet our above requirements and since they can represent autoregressive and moving-average time series behavior.

Preliminary results for kill point prediction (estimating the time in minutes to any catastrophic event) has indicated that it is feasible to learn from *problem-solving traces*. These traces are databases that record simulated events in *DC-Train* and actions by a human or a problem-solving, knowledge-based system [Bu98]. This approach is an ideal test bed for time series learning as applied to crisis monitoring. Early analysis for simulator development has shown that heterogeneity is indeed present (due to the multiple types of damage control crises) [WFH+96]. Mixture models (specifically, *boosting* of weak classifiers [FS96]) have also been shown to improve learning performance consistently. The gain is close to that achievable by selecting the strongest classifier, an improvement of 1-3% on test where unboosted cross validation accuracy is between 85% and 97% [VB98].

Crop Monitoring in Precision Agriculture

Figure 3 depicts an (atemporal) spatially referenced data set for diagnosis in *precision agriculture*. The inputs are: yield monitor data, crop type, elevation data and crop management records; the learning target, *cause of observed low yield* (e.g., drought) [Hs98]. Such classifiers may be used in *recommender* systems [RV97] (also called *normative* expert systems [He91]) to provide decision support for crop production planning in subsequent years. We use biweekly remote sensing images and meteorological, hydrological, and crop-specific data to learn to classify influents of *expected crop quality* (per farm) as *climatic* (drought, frost, etc.) or *non-climatic* (due to crop management decisions) [Hs98].

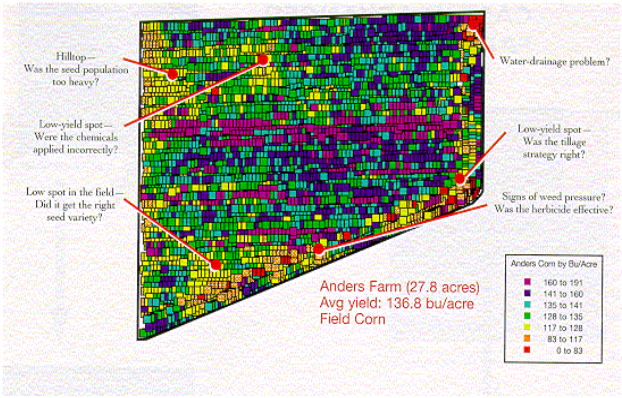

Figure 3. An agricultural diagnosis problem

Figure 4 visualizes a heterogeneous time series. The lines shown are autocorrelation plots of (subjective) weekly *crop condition* estimates, averaged from 1985-1995 for the state of Illinois. Each *point* represents the correlation between one week's mean estimate and the mean estimate for a subsequent week. Each *line* contains the correlation between values for a particular week and all subsequent weeks. The data is heterogeneous because it contains both a moving average pattern (the linear increments in autocorrelation for the first 10 weeks) and an exponential trace pattern (the larger, unevenly spaced increments from 0.4 to about 0.95 in the rightmost column). The MA pattern expresses weather "memory" (correlating early and late drought); the AR pattern, physiological damage from drought. Task decomposition can improve performance here, by isolating the MA and AR components for identification and application of the correct specialized architecture (a time delay neural network [LWH90, Ha94] or simple recurrent network [El90, PL98], respectively).

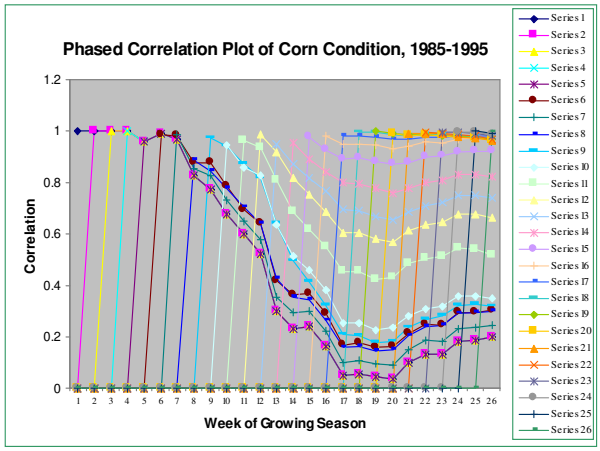

Figure 4. A heterogeneous time series

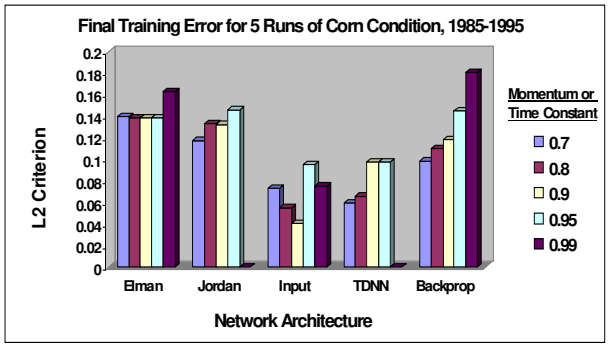

Figure 5. Results from training with temporal and feedforward ANNs

Figure 5 contains bar charts of the mean squared error from 125 training runs using ANNs of different configurations (5 architectures, 5 delay constant or momentum values for gradient learning, and 5 averaged runs per combination). On all runs, Jordan recurrent networks and time-delay neural networks failed to converge with momentum of 0.99, so the corresponding

bars are omitted. Cross validation results indicate that overtraining on this data set is minimal. As a preliminary study, we used a gamma network to select the correct classifier (if any) for each exemplar from among the two best overall networks (input recurrent with momentum of 0.9 and TDNN with momentum of 0.7). The error rate was reduced by almost half, indicating that even with identical inputs and targets, a simple mixture model could reduce variance [Hs98].

Conclusions and Future Work

This paper has presented the design of a heterogeneous time series learning system with metric-based model selection. Our study of common statistical models for time series and our experience with the (highly heterogeneous) test bed domains bears out the idea that "fitting the right tool to each job" is critical. In current research, we apply our system to specific, applied monitoring and diagnosis problems in damage control and precision agriculture, such as learning for causal queries [WS97, Hs97, Hs98]. Current research by the authors addresses the related problems of task decomposition by constructive induction (aggregation and transformation of ground attributes) and fusion of test predictions from probabilistic network classifiers [HR98a, RH98].

Acknowledgements

Support for this research was provided in part by the Office of Naval Research under grant N00014-95-1-0749 and by the Naval Research Laboratory under grant N00014-97-C-2061. The authors thank Sylvian Ray for advice and discussions on data fusion in time series analysis, and Vadim Bulitko for helpful insights and experimental discussions regarding the shipboard damage control domain.

References

[BJR94] G. E. P. Box, G. M. Jenkins, and G.C. Reinsel. *Time Series Analysis, Forecasting, and Control (3rd edition).* Holden-Day, San Fransisco, CA, 1994.

[Br96] L. Breiman. Bagging Predictors. *Machine Learning*, 1996.

[Bu98] V. V. Bulitko. *Minerva-5: A Multifunctional Dynamic Expert System.* M. S. thesis, University of Illinois. http://www-kbs.ai.uiuc.edu/minerva, 1998.

[Ch96] C. Chatfield. *The Analysis of Time Series: An Introduction (5th edition).* Chapman and Hall, London, 1996.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, NY, 1973.

[DM83] T. G. Dietterich and R. S. Michalski. A Comparative Review of Selected Methods for Learning from Examples. In *Machine Learning: An Artificial Intelligence Approach (Volume 1)*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. Tioga Publishing Company, Palo Alto, CA, 1983.

[DR95] S. K. Donoho and L. A. Rendell. Rerepresenting and Restructuring Domain Theories: A Constructive Induction Approach. *Journal of Artificial Intelligence Research*, 2:411-446, 1995.

[FB85] L.-M. Fu and B. G. Buchanan. Learning Intermediate Concepts in Constructing a Hierarchical Knowledge Base. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 659-666, Los Angeles, CA, 1985.

[FS96] T. Freund and R. E. Schapire. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, 1996.

[GD88] D. M. Gaba and A. deAnda. A Comprehensive Anesthesia Simulation Environment: Re-creating the Operating Room for Research and Training. *Anesthesia*, 69:387:394, 1988.

[GFH94] D. M. Gaba, K. J. Fish, and S. K. Howard. *Crisis Management in Anesthesiology*. Churchill Livingstone, New York, NY, 1994.

[GHVW98] E. Grois, W. H. Hsu, M. Voloshin, and D. C. Wilkins. Bayesian Network Models for Generation of Crisis Management Training Scenarios. In *Proceedings of IAAI-98*, to appear.

[GW94] N. A. Gershenfeld and A. S. Weigend. The Future of Time Series: Learning and Understanding. In *Time Series Prediction: Forecasting the Future and Understanding the Past (Santa Fe Institute Studies in the Sciences of Complexity XV),* A. S. Weigend and N. A. Gershenfeld, editors. Addison-Wesley, Reading, MA, 1994.

[HB95] E. Horvitz and M. Barry. Display of Information for Time-Critical Decision Making. In *Proceedings of UAI-95.*

[HLB+96] B. Hayes-Roth, J. E. Larsson, L. Brownston, D. Gaba, and B. Flanagan. *Guardian Project Home Page*, URL: http://www-ksl.stanford.edu/projects/guardian/index.html

[He91] D. A. Heckerman. *Probabilistic Similarity Networks*. MIT Press, Cambridge, MA, 1991.

[He96] D. A. Heckerman. *A Tutorial on Learning With Bayesian Networks*. Microsoft Research Technical Report 95-06, Revised June 1996.

[HGL+98] W. H. Hsu, N. D. Gettings, V. E. Lease, Y. Pan, and D. C. Wilkins. A Multi-strategy Approach to Time Series Learning for Crisis Monitoring. In *Proceedings of the International Workshop on Multistrategy Learning (MSL-98)*, to appear.

[HR98a] W. H. Hsu and S. R. Ray. A New Mixture Model for Concept Learning From Time Series. In *Proceedings of the 1998 Joint AAAI-ICML Workshop on Time Series Analysis*, to appear.

[HR98b] W. H. Hsu and S. R. Ray. Quantitative Model Selection for Heterogeneous Time Series Learning. In *Proceedings of the 1998 Joint AAAI-ICML Workshop on Methodology of Machine Learning*, to appear.

[Hs97] W. H. Hsu. *Spatiotemporal Sequence Learning With Probabilistic Networks*. Thesis proposal.

University of Illinois, unpublished, URL: http://anncbt.ai.uiuc.edu/prelim.doc, 1997.

[Hs98] W. H. Hsu. *Time Series Learning With Probabilistic Network Composites*. Ph.D. thesis, University of Illinois. In preparation.

[HZ95] W. H. Hsu and A. E. Zwarico. Automatic Synthesis of Compression Techniques for Heterogeneous Files. *Software: Practice and Experience*, 25(10): 1097-1116, 1995.

[JJ94] M. I. Jordan and R. A. Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6:181-214, 1994.

[JJNH91] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3:79-87, 1991.

[KJ97] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence, Special Issue on Relevance, 97*(1-2):273-324, 1997.

[Le89] K.-F. Lee. *Automatic Speech Recognition*. Kluwer Academic Publishers, Boston, MA, 1989.

[LWH90] K. J. Lang, A. H. Waibel, and G. E. Hinton. A Time-Delay Neural Network Architecture for Isolated Word Recognition. *Neural Networks* 3:23-43, 1990.

[MMR97] K. Mehrotra, C. K. Mohan, S. Ranka. *Elements of Artificial Neural Networks*. MIT Press, 1997.

[Mo94] M. C. Mozer. Neural Net Architectures for Temporal Sequence Processing. In *Time Series Prediction: Forecasting the Future and Understanding the Past (Santa Fe Institute Studies in the Sciences of Complexity XV),* A. S. Weigend and N. A. Gershenfeld, editors. Addison-Wesley, Reading, MA, 1994.

[MW96] O. J. Mengshoel and D. C. Wilkins. Recognition and Critiquing of Erroneous Student Actions. In *Proceedings of the AAAI Workshop on Agent Modeling*, 61-68. AAAI Press, 1996.

[Ne96] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, 1996.

[PL98] J. Principé, C. Lefebvre. *NeuroSolutions v3.01*, NeuroDimension, Gainesville, FL, 1998.

[RH98] S. R. Ray and W. H. Hsu. *Modular Classification for Spatiotemporal Sequence Processing*. In preparation.

[RV97] P. Resnick and H. R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56-58, 1997.

[Sarle, 1998] W. S. Sarle, ed., *Neural Network FAQ*, periodic posting to the Usenet newsgroup *comp.ai.neural-nets*, URL: ftp://ftp.sas.com/pub/neural/FAQ.html

[SM93] B. Stein and M. A. Meredith. *The Merging of the Senses*. MIT Press, Cambridge, MA, 1993.

[Wo92] D. H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241-259, 1992.

[WFH+96] D. C. Wilkins, C. Fagerlin, W. H. Hsu, E. T. Lin, and D. Kruse. *Design of a Damage Control Simulator*. Knowledge Based Systems Laboratory Technical Report UIUC-BI-KBS-96005. Beckman Institute, UIUC, 1996.

[WS97] D. C. Wilkins and J. A. Sniezek. *DC-ARM: Automation for Reduced Manning*. Knowledge Based Systems Laboratory Technical Report UIUC-BI-KBS-97012. Beckman Institute, UIUC, 1997.