Quantitative Model Selection for Heterogeneous Time Series Learning

William H. Hsu and Sylvian R. Ray

Department of Computer Science
University of Illinois at Urbana-Champaign
1304 West Springfield Avenue, Urbana, IL 61801
{bhsu | ray} @cs.uiuc.edu http://anncbt.ai.uiuc.edu

Abstract

A key benefit of modular learning is the ability to apply different algorithms to suit the characteristics of each subtask. This approach requires methods for task decomposition, classifier fusion, and matching of subproblems to learning techniques. In this paper, we present a new method for technique selection from a "repertoire" of statistical learning architectures (specifically, artificial neural networks and Bayesian networks) and methods (Bayesian learning, mixture models, and gradient learning). We first discuss the problem of learning *heterogeneous* time series, such as sensor data from multiple modalities. We then explain how to construct *composite learning* systems by selecting model components. Finally, we outline the design of a composite learning system for geospatial monitoring problems and present an application (precision agriculture) that demonstrates its potential benefits.

Keywords: **metric-based technique selection, time series learning, probabilistic networks, modular task decomposition, data fusion**

Introduction

Decomposition of statistical machine learning tasks can reduce both complexity and variance [JJ94]. The *mixture-of-experts model*, or mixture model, is a divide-and-conquer approach that integrates multiple sources of knowledge (including committees of experts or agents) [JJB91, JJ94, Bi95]. *Aggregation* mixtures reduce variance by replicating training data across the mixture components [Wo92, Br96]; *partitioning* mixtures use interaction among these components (at the level of *data fusion*) to force specialization among them [JJ94, FS96].

In this paper, we discuss a third function of mixture models in concept learning: to combine classifiers that are specialized to different projections, or partitions, of the training data. Such multistrategy learning approaches, where the "right tool" is analytically identified for each subtask, are useful in problems that exhibit *heterogeneity*. An example is spatiotemporal sequence learning for *time series classification.* We develop a collection of probabilistic (artificial neural and Bayesian) network architectures with complementary learning methods for inductive supervised learning, a new metric-based model selection system, and an algorithm for selecting the learning components indicated by the data set characteristics.

The key novel contributions of our system are:

1. Recombinable and reusable learning components for time series
2. Metrics for *temporal* characteristics that prescribe learning techniques
3. A framework for task decomposition and fusion of classifiers learned by different techniques

Heterogeneous Time Series Learning

In *heterogeneous* time series, the embedded temporal patterns belong to different categories of statistical models, such as *moving average* (MA) and *autoregressive* (AR or *exponential trace*) models [MMR97]. A multichannel time series learning problem can be decomposed into homogeneous 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 [Pe88, He96]; or implicitly by preprocessing transforms [HR98].

This section presents an analogy between concept learning in heterogeneous time series and compression of heterogeneous files. The significance of this analogy is that, given a system that selects the most appropriate compression algorithm for *segments* of a file, we can construct a similar learning system. This system selects the most appropriate inductive learning techniques for an *attribute subset* of a given time series. The subsets are evaluated based on available learning architectures (cf [KJ97]) and intermediate target concepts are formed (cf [FB85, CKS+88]) to achieve task decomposition.

We briefly present a successful heterogeneous compressor that employs metric-based file analysis and extend our analogy to the design of a modular, probabilistic network-based learning system.

Network Type	Architectural Metric
Simple recurrent network (SRN)	Exponential trace (AR) autocorrelation
Time delay neural network (TDNN)	Moving average (MA) autocorrelation
Gamma network	Autoregressive integrated moving average (ARIMA) autocorrelation
Temporal naïve Bayesian network	Knowledge map score (relevant attribute count)
Hidden Markov model (HMM)	Test set perplexity

Table 1. Network types 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

Compression of Heterogeneous Files

Heterogeneous files are those that contain multiple types of data such as text, image, or audio. We have developed an experimental data compressor for that outperforms several commercial, general-purpose compressors on heterogeneous files. The interested reader is referred to [HZ95] for details on the comparative experiments, which are beyond the scope of this paper.

The heterogeneous compressor divides a file into fixed-length segments and empirically analyzes each (cf. [Sa89, HM91]) for its *file type* and dominant *redundancy type*. For example, *dictionary* algorithms such as Lempel-Ziv coding are most effective with frequent repetition of strings; *run length encoding*, on long runs of bits; and *statistical* algorithms such as *Huffman coding* and *arithmetic coding*, when there is nonuniform distribution among characters. These correspond to our *redundancy metrics*: string repetition ratio, average run length, and population standard deviation of ordinal character value.

The normalization function over these metrics is calibrated using a corpus of *homogeneous* files. Using the metrics and file type, our system predicts, and applies, the most effective algorithm and update (e.g., paging) heuristic for the segment. In experiments on a second corpus of heterogeneous files, the system selected the best of the three available algorithms on about 98% of the segments, yielding significant performance wins on 95% of the test files [HZ95].

Adapting Statistical File Analysis to Heterogeneous Learning

The analogy between compression and learning [Wa72] is especially strong for technique selection from a database of components. Compression algorithms correspond to network architectures in our framework; heuristics, to applicable methods (mixture models, learning algorithms, and hyperparameters for Bayesian learning). Metric-based file analysis for compression can be adapted to technique selection for heterogeneous time series learning. To select among network architectures, we use indicators of temporal patterns typical of each; similarly, to select among learning algorithms, we use predictors of their effectiveness. The analogy is completed by the process of segmenting the file (corresponding to problem decomposition by aggregation and synthesis of attributes) and concatenation of the compressed segments (corresponding to fusion of test predictions).

A noteworthy result from the compression experiments is that using metric-based technique selection yields improvements over the brute force method of keeping the most compressed segment (i.e., trying all possibilities). This is due to the overhead of restarting the model (for both statistical and dictionary compression methods) at each segment. For details, we refer the reader to [HZ95]. We hypothesize similar consequences for model selection in machine learning. Degradation in learning quality is likely due to the nearsightedness of the brute force assumptions (namely, the *lack of a model of interaction among subtasks*). Furthermore, the number of models (even at the coarse level of granularity of this technique selection system) is large enough to be prohibitively slow when exhaustive testing is used. This loss of performance is more pronounced than in compression [Hs98].

The importance of the overall analogy is that it illustrates a useful correspondence between metric-based selection of compression techniques for heterogeneous files and metric-based selection of learning models for heterogeneous time series.

Composite Learning

Database of Learning Components

Table 1 lists the network types (the rows of a "lookup table" of learning components) and the indicator metrics corresponding to their strengths [Hs97]. 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. The last architecture listed, a hidden Markov model (HMM), is a stochastic state transition diagram whose transitions are also annotated with probability distributions (over output symbols) [Le89].

Table 2 lists the 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) [JJB91, JJ94]. It can be trained by gradient learning, expectation-maximization, or Markov chain Monte Carlo (MCMC) methods (i.e., random sampling as in the Metropolis algorithm for simulated annealing) [Ne96]. A *specialist-moderator network*, which also admits these learning algorithms, is a mixture model whose components have different input and output attributes [HR98].

Metric-Based Model Selection

In Table 1, our prototype *architectural metrics* for temporal ANNs are average conditional entropy values for the preprocessed data. For example, to compute autocorrelation for an AR model, we first apply convolution of an exponential decay window (an AR *kernel function*) [MMR97]. This estimates the predictive power of the model *if chosen as the learning architecture*. The knowledge map score for a diagnostic Bayesian network is the average number of variables *relevant* to each pair of diagnosable causes in an associative knowledge map [He91]. This knowledge map is constructed using naïve Bayes or by thresholding on the correlations between causes and observable effects. Finally, a typical indicator metric for HMMs is the *test set perplexity* (empirical mean of the branch factor) for a constructed HMM [Le89].

In Table 2, 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') [HR98]. To select a learning algorithm, we use gradient learning as a baseline and add a term for the gain from estimation of missing data (by EM) [JJ94] or global optimization (by MCMC) [Ne96], adjusted for the conditional sample complexity.

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

The general algorithm for composite time series learning follows.

Given:

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

```
Algorithm Select-Net
  repeat
    Generate a candidate representation (e.g, attribute subset partition [KJ97]) $(A', B') \in F(A, B, D)$.
    Compute architectural metrics that prescribe the network type.
    Compute distributional metrics that prescribe the learning method.
    Normalize the metrics using a precalibrated model (see Figure 1).
    Select the most strongly prescribed network type $(\theta, \gamma)$ 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.
```

Figure 1. Normalization of metrics x_τ

t_τ: shape parameter
λ_τ: scale parameter

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

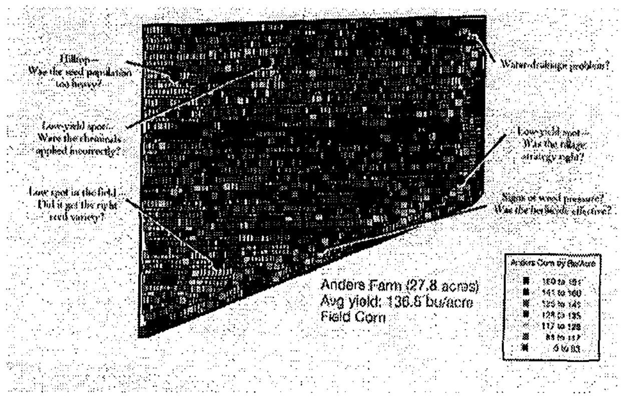
$$f_\tau(x) = \frac{\lambda_\tau \mathrm{e}^{-\lambda_\tau x} (\lambda_\tau x)^{t_\tau - 1}}{\Gamma(t_\tau)}$$

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

Preliminary Experimental Results

Figure 2 depicts an (atemporal) spatially referenced data set for diagnosis in *precision agriculture*. The inputs are:

Figure 2. A Geospatial Diagnosis Problem


Hilltop— Was the seed population too heavy?
Low-yield spot— Were the chemicals applied incorrectly?
Low spot in the field— Did it get the right seed variety?
Water-drainage problem?
Low-yield spot— Was the tillage strategy right?
Signs of weed pressure? Was the herbicide effective?
Anders Farm (27.8 acres)
Avg yield: 136.8 bu/acre
Field Corn
Anders Corn by Bu/Acre

yield monitor data, crop type, elevation data and crop management records; the learning target, *cause of observed yield* (e.g., drought) [Hs98]. Such classifiers may be used in 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.

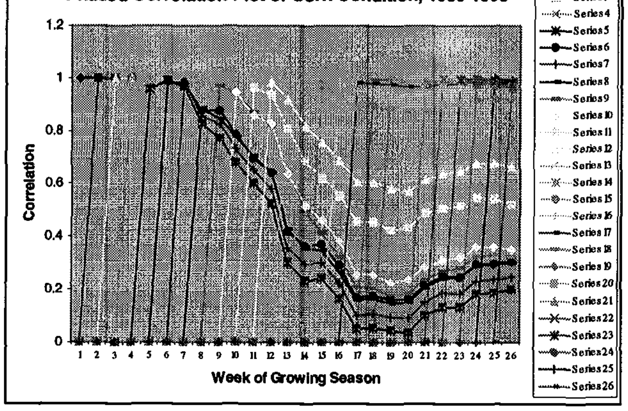


Figure 3 visualizes a heterogeneous time series. The lines shown are autocorrelation plots of (subjective) weekly *crop condition* estimates, averaged from 1985-1995 for the state of Illinois. Each *point* represents the correlation between one week's mean estimate and the mean estimate for a subsequent 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.

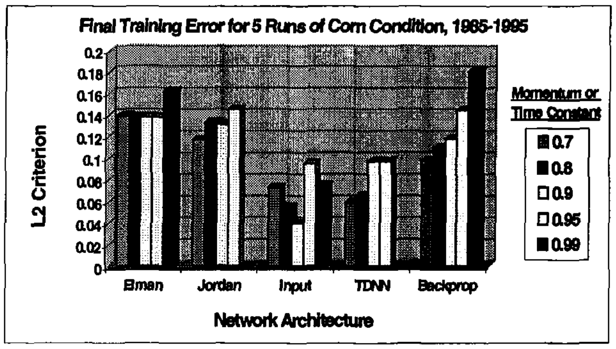


Figure 4 contains bar charts of the mean squared error from 125 training runs using ANNs of different configurations (5 architectures, 5 momentum or delay constant values for gradient learning, and 5 averaged error values per combination). On all runs, Jordan recurrent networks with a delay constant of 0.99 and time delay neural networks (TDNNs) with a momentum of 0.99 failed to converge, so the corresponding bars are omitted.

Tables 3 and 4 list the percentages of exemplars correctly classified after training. Cross validation results were obtained by holding out one *year* at a time. Minor overtraining was observed for suboptimal delay constants.

Network Type		Delay Constant				
		0.7	0.8	0.9	0.95	0.99
Elman	Train	78.55	78.25	78.55	78.55	77.04
	C.V.	44.44	44.44	38.89	52.78	55.56
Jordan	Train	85.50	83.08	82.48	81.57	—
	C.V.	55.56	55.56	52.78	41.67	—
Input Rec.	Train	97.58	96.68	**90.63**	77.04	77.04
	C.V.	55.56	55.56	**55.56**	47.22	52.78

Table 3. Classification accuracy (in percent) for corn condition using simple recurrent networks

Network Type		Momentum				
		0.7	0.8	0.9	0.95	0.99
10-delay TDNN	Train	**98.19**	98.49	98.79	98.79	—
	C.V.	**58.33**	58.33	55.56	50.00	—
Feed-forward	Train	86.10	86.10	87.00	89.12	89.12
	C.V.	55.56	52.78	52.78	52.78	47.22

Table 4. Classification accuracy (in percent) for corn condition using time-delay and feedforward ANNs

As a preliminary study, we used a gamma network (a type of ARMA model) to select the correct classifier (if any) for each exemplar from among the two networks with lowest mean squared error. These were the input recurrent network with delay constant of 0.9 and TDNN with momentum of 0.7, listed in boldface in Tables 3 and 4. The error rate was further reduced, as reported in Table 5. Thus, even a simple mixture model with identical inputs and targets can reduce variance [Hs98].

	Fusion (Gamma)	Best TDNN	Best SRN
Training	**99.09**	98.19	90.63
C.V.	**61.11**	58.33	55.56

Table 5. Performance boost from classifier fusion

Conclusions and Future Work

We have presented the design of a heterogeneous time series learning system with metric-based model selection, which evolved from a successful heterogeneous data compressor [HZ95, Hs98]. Our current research applies this system to a heterogeneous time series concept learning problem: monitoring and diagnosis for precision agriculture [Hs97, Hs98]. Other applications include control automation and computer-assisted instruction for crisis management [GHVW98, HGL+98]. We are addressing the related problems of task decomposition by constructive induction (aggregation and transformation of ground attributes) and fusion of test predictions from probabilistic network classifiers [HR98, RH98].

References

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

[Bi95] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, UK, 1995.

[CKS+88] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, D. Freeman. Autoclass: A Bayesian Classification System. In *Proceedings of ICML-88*, Ann Arbor, MI, 1988.

[El90] J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179-211, 1990.

[FB85] L.-M. Fu and B. G. Buchanan. Learning Intermediate Concepts in Constructing a Hierarchical Knowledge Base. In *Proceedings of IJCAI-85*, pages 659-666, 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.

[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.

[Ha94] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan, New York, 1994.

[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.

[HM91] G. Held and T. R. Marshall. *Data Compression: Techniques and Applications, 3rd ed.* Wiley, 1991.

[HGL+98] W. H. Hsu, N. D. Gettings, V. E. Lease, Y. Pan, and D. C. Wilkins. Crisis Monitoring: Test Beds and Methods for Time Series Learning. . In *Working Notes of the Joint AAAI/ICML-98 Workshop on Time Series Analysis*, to appear.

[HR98] W. H. Hsu and S. R. Ray. A New Mixture Model for Concept Learning from Time Series. In *Working Notes of the Joint AAAI/ICML-98 Workshop on Time Series Analysis*, 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.

[JJB91] R. A. Jacobs, M. I. Jordan, and A. G. Barto. Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks. *Cognitive Science*, 15:219-250, 1991.

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

[KJ97] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence* 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, Cambridge, MA, 1997.

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

[Pe88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan-Kaufmann, San Mateo, CA, 1988.

[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.

[Sa89] G. Salton. *Automatic Text Processing.* Addison Wesley, Reading, MA, 1989.

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

[Wa72] S. Watanabe. Pattern Recognition as Information Compression. In *Frontiers of Pattern Recognition*, S. Watanabe, ed. Academic Press, San Diego, CA, 1972.

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