Semi-supervised Learning with Data Calibration for Long-Term Time Series Forecasting

Haibin Cheng
Michigan State University
Lansing,MI,48910
chenghai@cse.msu.edu

Pang-Ning Tan
Michigan State University
Lansing,MI,48910
ptan@cse.msu.edu

ABSTRACT

Many time series prediction methods have focused on single step or short term prediction problems due to the inherent difficulty in controlling the propagation of errors from one prediction step to the next step. Yet, there is a broad range of applications such as climate impact assessments and urban growth planning that require long term forecasting capabilities for strategic decision making. Training an accurate model that produces reliable long term predictions would require an extensive amount of historical data, which are either unavailable or expensive to acquire. For some of these domains, there are alternative ways to generate potential scenarios for the future using computer-driven simulation models, such as global climate and traffic demand models. However, the data generated by these models are currently utilized in a supervised learning setting, where a predictive model trained on past observations is used to estimate the future values. In this paper, we present a semi-supervised learning framework for long-term time series forecasting based on Hidden Markov Model Regression. A covariance alignment method is also developed to deal with the issue of inconsistencies between historical and model simulation data. We evaluated our approach on data sets from a variety of domains, including climate modeling. Our experimental results demonstrate the efficacy of the approach compared to other supervised learning methods for long-term time series forecasting.

Categories and Subject Descriptors

H.m [**Information Systems**]: Miscellaneous

General Terms

Algorithms, Design, Experimentation

Keywords

Time Series Prediction and Semi-supervised Learning

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

1. INTRODUCTION

Time series prediction has long been an active area of research with applications in finance [28], weather forecasting [14][8], network monitoring [5], transportation planning [18][23], etc. Many of these applications require long-term time series forecasting capabilities for strategic decision making. For example, scientists are interested in projecting the future climate to assess their potential impacts on the ecosystem and society. Transportation planners are interested in forecasting highway traffic volumes to prevent future congestion and to reduce fuel costs and pollution. One way to perform long term forecasting is by repeatedly invoking a model that makes its prediction one step at a time. However, since the model uses predicted values from the past to infer future values, this approach may lead to an error accumulation problem, which is the propagation of errors from one prediction step to the next step [9]. Long-term forecasting also requires an extensive amount of historical data in order to train a reliable model.

Instead of predicting the future values of a time series based on its historical values alone, an alternative strategy is to employ multivariate time series prediction methods, where the time series for a set of predictor variables are fitted against the time series for the response variable. This strategy is contingent upon the availability of future values of the predictor variables. Fortunately, in some application domains, such values can be obtained from computer-driven simulation models. For example, in climate modeling, outputs from global climate models (GCMs) [14] are often used as predictor variables to determine the climate of a local region. The GCM models are developed based on the basic principles of physics, chemistry, and fluid mechanics, taking into account the coupling between land, ocean, and atmospheric processes. Although the outputs from these models sufficiently capture many of the large-scale ($\approx$ 150-300 km spatial resolution) circulation patterns of the observed climate system, they may not accurately model the response variable at the local or regional scales ($\approx$ 1-50 km) needed for climate impact assessment studies [27]. As a result, the coarse-scale GCM outputs need to be mapped into finer scale predictions, a process that is known as "downscaling" in the Earth Science literature.

Regression is a popular technique for downscaling, where the GCM outputs are used as predictor variables and the response variable corresponds to the local climate variable of interest (e.g., precipitation or temperature). However, current approaches utilize the simulation data in a supervised learning setting, where a predictive model trained on past

observations is used to estimate the future values. Such an approach fails to take advantage of information about the future data during model building. This paper presents a semi-supervised learning framework for long-term time series forecasting based on Hidden Markov Model Regression (HMMR). Our approach builds an initial HMMR model from past observations and incorporates future data to iteratively refine the model. Since the initial predictions for some of the future data may not be reliable, we need to ensure that they do not adversely affect the revised model. We developed an approach to overcome this problem by assigning weights to instances of the future data based on the consistency between their global and local predictions. This approach also helps to ensure smoothness of the target function [29]. We demonstrated the efficacy of our approach using data sets from a variety of applications domains.

One issue that requires particular consideration when applying the semi-supervised HMMR to climate modeling problems is the potential inconsistencies between the training and future data since they often come from different sources. GCM simulation runs are driven by a set of emissions scenarios, which may assume greenhouse gas concentrations that are different than those in the training data. Previous work on semi-supervised classification have shown that combining labeled and unlabeled data with different distributions may degrade the performance of a classifier [25]. We encountered a similar problem when applying the semi-supervised HMMR method to climate data. To address this problem, we developed an approach that will transform the data set in a way that aligns their covariance structure while preserving most of the neighborhood information. Experimental results for modeling climate at 40 locations in Canada showed that semi-supervised HMMR with data calibration outperforms conventional (supervised) HMMR method in more than 70% of the locations.

The remainder of this paper is organized as follows. Section 2 reviews the background materials on time series prediction and HMMR. Section 3 describes the value of unlabeled data in regression problems. The proposed semi-supervised HMMR approach with data calibration is given in Section 4. The experimental results are presented in Section 5. Finally, Section 6 presents the related work on semi-supervised learning and Section 7 conclude our study.

2. PRELIMINARIES

2.1 Problem Formulation

Let $\boldsymbol{L} = (\boldsymbol{X}_l, \boldsymbol{Y}_l)$ be a multivariate time series of length l, where $\boldsymbol{X}_l = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l]^T$ is a $(p-1)$-dimensional sequence of values for the predictor variables and $\boldsymbol{Y}_l = [y_1, y_2, ..., y_l]^T$ is the corresponding values for the response variable. The objective of multivariate time series prediction is to learn a target function f that accurately predicts the future values of the response variable, $\boldsymbol{Y}_u = [y_{l+1}, y_{l+2}, ..., y_{l+u}]^T$, given the historical data $\boldsymbol{L}$ and the unlabeled data, $\boldsymbol{X}_u = [\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, ..., \boldsymbol{x}_{l+u}]^T$. $\boldsymbol{X}_u$ can be obtained, for example, using computer-driven simulation models (e.g., GCM for climate modeling or CORSIM for traffic modeling). While the simulation runs can be used to produce coarse-scale properties of a complex system, they do not accurately capture its detailed properties. The outputs from model simulation are therefore used as inputs into regression functions that predict the fine level properties of the system.

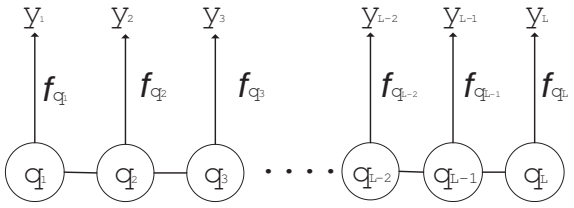


Figure 1: Hidden Markov Regression Model

There are many multivariate time series prediction techniques available, including least square regression[20], recurrent neural networks[16], Hidden Markov Model Regression [15], and support vector regression [24]. These techniques are currently employed in a supervised learning setting, and thus, may not fully utilize the value of the unlabeled data. This work presents a semi-supervised learning framework that integrates labeled and unlabeled data to improve long-term time series forecasting. The framework is implemented using Hidden Markov Model Regression [15], a stochastic model that has been successfully applied to various domains, including speech recognition [22] and climate modeling [8].

2.2 Hidden Markov Model Regression (HMMR)

In Hidden Markov Model Regression, a time series is assumed to be generated by a doubly stochastic process, where the response variable is conditionally dependent on the values of the predictor variables as well as an underlying unobserved process. The unobserved process is characterized by the state space $\boldsymbol{S} = \{s_1, s_2, \cdots, s_N\}$ and is assumed to evolve according to a Markov chain, $\boldsymbol{Q}_l = [q_1, q_2, .., q_l]^T$, where $q_i \in \boldsymbol{S}$ (as shown in Figure 1). The transition between states is governed by an $N \times N$ transition probability matrix $\boldsymbol{A} = [a_{ij}]$. Each state $s_i \in \boldsymbol{S}$ is associated with an initial probability distribution π_i and a regression function $f_i(\boldsymbol{x}_t)$. For brevity, we consider a multiple linear regression model, $f_i(\boldsymbol{x}_t) = \boldsymbol{c}_i \boldsymbol{x}_t^T + \sigma_i \epsilon_t$, where $(\boldsymbol{c}_i, \sigma_i)$ are the regression parameters, and $\epsilon_t \sim N(0,1)$. Given a set of predictor variables $\boldsymbol{x}_t$ at time t, the response variable y_t is generated based on the following conditional probability:

$$p_{q_t}(y_t|\boldsymbol{x}_t) = (2\pi\sigma_{q_t}^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_t - \boldsymbol{c}_{q_t}\boldsymbol{x}_t^T)^2}{2\sigma_{q_t}^2} \right] \quad (1)$$

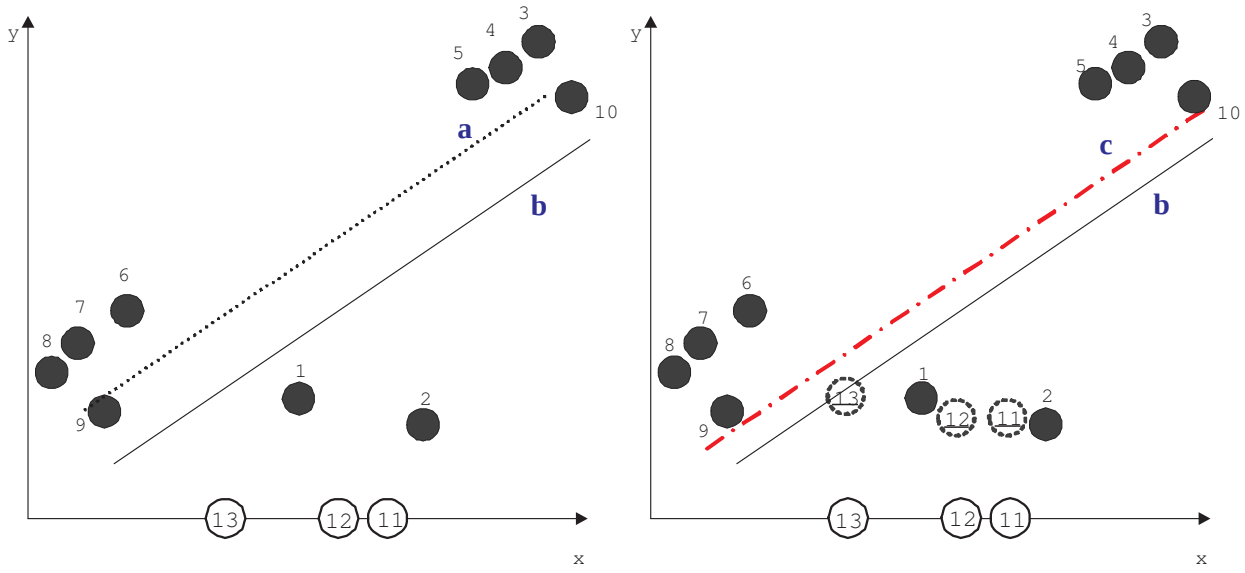
The likelihood function for the sequence of labeled observations in $\boldsymbol{L} = (\boldsymbol{X}_l, \boldsymbol{Y}_l)$ is given by

$$\begin{aligned} L &= \sum_{\boldsymbol{Q}_l} \prod_{t=1}^{l} p_{q_t}(y_t|\boldsymbol{x}_t) p(q_t|q_{t-1}) \\ &= \sum_{\boldsymbol{Q}_l} \prod_{t=1}^{l} a_{q_{t-1}q_t} (2\pi\sigma_{q_t}^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_t - \boldsymbol{c}_{q_t}\boldsymbol{x}_t^T)^2}{2\sigma_{q_t}^2} \right] \end{aligned}$$

The model parameters

$$\boldsymbol{\Lambda} = \Big(\boldsymbol{\Pi}, \boldsymbol{A}, \boldsymbol{\Sigma}, \boldsymbol{C}\Big) = \Big(\{\pi_i\}_{i=1}^N, \{a_{ij}\}_{i,j=1}^N, \{\sigma_i\}_{i=1}^N, \{\boldsymbol{c}_i\}_{i=1}^N\Big)$$

are estimated by maximizing the above likelihood function using the Baum-Welch (BW) algorithm [2]. The model parameters can be estimated iteratively using the following quantities, which are known as the forward (α) and back-



(a) Supervised learning

(b) Semi-supervised learning

Figure 2: The value of unlabeled data for regression analysis

ward (β) probabilities:

$$\alpha_t(i) = p(y_1, y_2, \cdots, y_t, q_t = i|\boldsymbol{X}_l) = \begin{cases} \pi_i p_i(y_1|\boldsymbol{x}_1), & \text{if } t = 1; \\ \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] p_i(y_t) & \text{otherwise.} \end{cases} \quad (2)$$

$$\beta_t(i) = p(y_{t+1}, y_{t+2}, \cdots, y_l|q_t = i, \boldsymbol{X}_l) = \begin{cases} 1, & \text{if } t = l; \\ \sum_{j=1}^{N} a_{ij} p_j(y_{t+1}) \beta_{t+1}(j), & \text{otherwise.} \end{cases} \quad (3)$$

Further details on the derivation of the parameter update formula using these quantities can be found in [15]. Upon convergence of the BW algorithm, the HMMR model can be applied to the unlabeled data $\boldsymbol{X}_u$ in the following manner. First, the probability of each hidden state at a given future time step is estimated as follows:

$$p(q_{l+m} = s_k|\boldsymbol{Y_l}) = \begin{cases} \frac{\sum_i \alpha_l(i) a_{ik}}{\sum_i \alpha_l(i)}, & \text{if } m = 1; \\ \sum_i a_{ik} p(q_{l+m-1} = s_i|\boldsymbol{Y_l}), & 1 < m \leq u. \end{cases}$$

Next, the future value of the response variable is estimated using the following equation:

$$f_{q_t}(\boldsymbol{x}_t) = E[y_t] = \sum_k (\boldsymbol{c}_k \boldsymbol{x}_t^T) p(q_t = s_k|\boldsymbol{Y_l}) \quad (4)$$

3. VALUES OF UNLABELED DATA

This section provides an example to illustrate the value of unlabeled data for regression analysis. Consider the diagram shown in Figure 2, where the x-axis corresponds to a predictor variable and the y-axis corresponds to the response variable. The data set contains 10 training examples (labeled $1-10$) and 3 unlabeled examples (labeled $11-13$). The diagram on the left shows the results of applying supervised linear regression while the diagram on the right shows the results of applying semi-supervised linear regression. The solid line b indicates the true function from which the data is generated.

The dashed line a in the left diagram corresponds to the target function estimated from training examples. In the right diagram, the response values for the unlabeled data are initially computed using the values of their nearest neighbors. The target function, represented by the dashed line c, is then estimated from the combined training and previously unlabeled examples. Clearly, augmenting unlabeled data in this situation helps to produce a target function that lies closer to the true function. This example also illustrates the importance of using local prediction methods (such as nearest neighbor estimation) to compute the response values of the unlabeled examples. If the unlabeled examples were estimated using the initial regression function instead, then adding unlabeled data will not change the initial model.

It is worth noting that current research in semi-supervised learning suggests unlabeled data are valuable under two situations. First, the model assumptions should match well with the underlying data. Second, the labeled and unlabeled data must be generated from the same distribution. Otherwise, incorporating unlabeled data may actually degrade the performance of a predictive model [25][12].

4. METHODOLOGY

This section describes our proposed semi-supervised learning algorithm for HMMR and a data calibration approach to deal with inconsistencies between the distributions of labeled and unlabeled data.

4.1 Semi-Supervised HMMR for Long-Term Time Series Forecasting

The basic idea behind our proposed approach is as follows. First, an initial HMMR model is trained from the historical data using the BW algorithm described in Section 2.2. The model is then used, in conjunction with a local prediction model, to estimate the response values for future observations. The local model is used to ensure that the target function is sufficiently smooth. The estimated

future values, weighted by their confidence in estimation, are then combined with the historical data to re-train the model. This procedure is repeated to gradually refine the HMMR model.

We now describe the details of each step of our algorithm. Let $\mathbf{\Lambda}^0$ be the initial set of model parameters trained from the historical observations $\boldsymbol{L}$. We use $\mathbf{\Lambda}^0$ to compute the initial estimate of each response value in Y_u:

$$\overline{y}_t \quad = \quad \sum_i p(q_t = s_i|Y_l)\boldsymbol{c}_i\boldsymbol{x}_t^T,\ t = l+1, ..., l+u,$$

which is similar to the formula given in Equation (4).

A principled way to incorporate unlabeled data is to require that the resulting target function must be sufficiently smooth with respect to its intrinsic structure [29]. For regression problems, this requirement implies that the response values for nearby examples should be close to each other to ensure the smoothness of the target function. We obtain an estimate of the local prediction of the target variable y for a future time step t as follows:

$$\widetilde{y}_t = \frac{\sum\limits_{\boldsymbol{x}_j \in \Omega_k(\boldsymbol{x}_t)} K(\boldsymbol{x}_t, \boldsymbol{x}_j) y_j}{\sum\limits_{\boldsymbol{x}_j \in \Omega_k(\boldsymbol{x}_t)} K(\boldsymbol{x}_t, \boldsymbol{x}_j)},\ t = l+1, \ldots, l+u$$

where $\Omega_k(\boldsymbol{x}_t)$ is a subset of observations in $\boldsymbol{X}_l$ that correspond to the k nearest neighbors of the unlabeled observation $\boldsymbol{x}_t$ and $K(\boldsymbol{x}_t, \boldsymbol{x}_j)$ is the similarity measure between two observations. Based on their global and local estimations, we compute the weighted average of the response value at each future time step t as follows:

$$\hat{y}_t = \mu\overline{y}_t + (1-\mu)\widetilde{y}_t, t = l+1, \ldots, l+u \tag{5}$$

The parameter μ controls the smoothness of the target function; a smaller weight means preference will given to the local estimation.

The newly labeled observations from the future time period will be augmented to the training set in order to rebuild the model. As some of the predicted values $\hat{y}_t$ may deviate quite significantly from either the local or global estimations (depending on μ), incorporating such examples may degrade the performance of semi-supervised HMMR. To overcome this problem, we compute the confidence value for each prediction and use it to weigh the influence of the unlabeled examples during model rebuilding. Let w_t denote the weight assigned to the value of y_t:

$$w_t = \begin{cases} 1, & t = 1, 2, \cdots, l; \\ \exp[-\delta_t], & t = l+1, l+2, \cdots, l+u. \end{cases} \tag{6}$$

where $\delta_t = |\widetilde{y}_t - \overline{y}_t|/(\widetilde{y}_t + \overline{y}_t)$. Equation (6) assigns a weight of 1 to each historical observation $\boldsymbol{Y}_l$. This is based on the assumption that there is no noise in the historical data. Even if such an assumption is violated, our framework may accommodate noisy observations by applying Equation (6) to both training and future observations. The weight formula reduces the influence of future observations whose predictions remain uncertain. After the model has been revised, it is used to re-estimate the response values for the future observations. This procedure is repeated until the changes in the model parameters become insignificant.

We now describe the procedure for estimating the parameters of the semi-supervised HMMR model. Let $\hat{\boldsymbol{Y}}_{\boldsymbol{u}}$ denote the estimated response values for the unlabeled data obtained from Equation 5. The likelihood function for the combined data is:

$$\begin{aligned} L \quad &= \quad \sum_q P_{\mathbf{\Lambda}}(\boldsymbol{Y}_l, \hat{\boldsymbol{Y}}_u, \boldsymbol{Q}|\boldsymbol{X}_l, \boldsymbol{X}_u) \\ &= \quad \sum_q \left(\prod_{t=1}^{l} a_{q_{t-1}q_t} p_{q_t}(y_t|\boldsymbol{x}_t) \prod_{t=l+1}^{l+u} a_{q_{t-1}q_t} p_{q_t}(\hat{y}_t; w_t|\boldsymbol{x}_t) \right) \\ &= \quad \sum_q \left(\prod_{t=1}^{l} a_{q_{t-1}q_t} (2\pi\sigma_{q_t}^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_t - \boldsymbol{c}_{q_t}\boldsymbol{x}_t^T)^2}{2\sigma_{q_t}^2} \right] \right. \\ &\times \quad \left. \prod_{t=l+1}^{l+u} a_{q_{t-1}q_t} (2\pi\sigma_{q_t}^2)^{-\frac{1}{2}} \exp\left[-\frac{w_t(\hat{y}_t - \boldsymbol{c}_{q_t}\boldsymbol{x}_t^T)^2}{2\sigma_{q_t}^2} \right] \right) \end{aligned}$$

where

$$p_{q_t}(y_t; w_t|\boldsymbol{x}_t) = (2\pi\sigma_{q_t}^2)^{-\frac{1}{2}} \exp\left[-\frac{w_t(y_t - \boldsymbol{c}_{q_t}x_t^T)^2}{2\sigma_{q_t}^2} \right]$$

with $w_t = 1$ for historical observations. Unlike the supervised learning case, the weights are used to determine the least square error $(\hat{y}_t - \boldsymbol{c}_{q_t}\boldsymbol{x}_t^T)$ for each future observation. To maximize the likelihood function, observations with large weights will incur higher penalty if their response values disagree with the predictions made by the current HMMR model. Such observations are therefore more influential in rebuilding the HMMR model.

To determine the model parameters that maximize the likelihood function, we introduce the following auxiliary func-

---

Algorithm 1 Semi-Supervised Hidden Markov Regression for Time Series Prediction

Input: Labeled multivariate time series for the historical period, $\boldsymbol{L} = (\boldsymbol{X}_l, \boldsymbol{Y}_l)$ with $\boldsymbol{X}_l = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l]^T$ $(\boldsymbol{x}_t \in R^{1\times(p+1)})$, $\boldsymbol{Y}_l = [y_1, y_2, ..., y_l]^T$ and unlabeled multivariate time series for the future period $\boldsymbol{X}_u = [\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, ..., \boldsymbol{x}_{l+u}]^T$.
Output: $\boldsymbol{Y}_u = [y_{l+1}, y_{l+2}, ..., y_{l+u}]^T$
Method:
1. Learn the initial HMMR model $\mathbf{\Lambda}_0 = (\boldsymbol{\Pi}, \boldsymbol{A}, \boldsymbol{\Sigma}, \boldsymbol{C})$ using the training data $\boldsymbol{L}$.
2. Perform local estimation of $\boldsymbol{Y}_u$:

$$\widetilde{y}_t = \frac{\sum\limits_{\boldsymbol{x}_j \in \Omega_k(\boldsymbol{x}_t)} K(\boldsymbol{x}_t, \boldsymbol{x}_j) y_j}{\sum\limits_{\boldsymbol{x}_j \in \Omega_k(\boldsymbol{x}_t)} K(\boldsymbol{x}_t, \boldsymbol{x}_j)},\ t = l+1, \ldots, l+u$$

3. Perform global estimation of $\boldsymbol{Y}_u$ using the current parameters in $\mathbf{\Lambda}$:

$$\overline{y}_t = \sum_{i=1}^{N} p(s_i|\boldsymbol{x}_t)\boldsymbol{c}_i\boldsymbol{x}_t^T, t = l+1 \ldots l+u$$

4. Calculate the final estimation of $\boldsymbol{Y}_u$:

$$\hat{y}_t = \mu\overline{y}_t + (1-\mu)\widetilde{y}_t, t = l+1, \ldots, l+u$$

5. Calculate the confidence of the predicted values in $\boldsymbol{Y}_u$:

$$w_t = \begin{cases} 1, & t = 1, 2, \cdots, l; \\ \exp[-\delta_t], & t = l+1, l+2, \cdots, l+u. \end{cases}$$

6. Combine $(\hat{y}_t; w_t)$ estimated in steps 4 and 5 with the training data to re-train the HMMR model $\mathbf{\Lambda}' = (\boldsymbol{\Pi}', \boldsymbol{A}', \boldsymbol{\Sigma}', \boldsymbol{C})'$.
7. Repeat step $3-6$ until convergence ($\|\mathbf{\Lambda}' - \mathbf{\Lambda}\| \ll \epsilon$)

tion:

$$
\begin{aligned}
& O(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}') \\
= & \sum_{q} P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}_l, \hat{\boldsymbol{Y}}_u, Q|\boldsymbol{X}) \ln P_{\boldsymbol{\Lambda}'}(\boldsymbol{Y}_l, \hat{\boldsymbol{Y}}_u, Q|\boldsymbol{X}) \\
= & \sum_{q} \sum_{i,j=1}^{N} (\sum_{t=2}^{l+u} \mathbf{1}_{q_{t-1}=i, q_t=j}) P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}_l, \hat{\boldsymbol{Y}}_u, Q|\boldsymbol{X}) \ln a'_{ij} \\
+ & \sum_{q} \sum_{i=1}^{N} (\mathbf{1}_{q_1=i}) P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}_l, \hat{\boldsymbol{Y}}_u, Q|\boldsymbol{X}) \ln \pi'_i \qquad (7) \\
+ & \sum_{q} \sum_{i=1}^{N} \sum_{t=1}^{l+u} (\mathbf{1}_{q_t=i}) P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}_l, \hat{\boldsymbol{Y}}_u, Q|\boldsymbol{X}) \ln p_{(\sigma'_i, \boldsymbol{c}'_i)}(y'_t; w_t|\boldsymbol{x}_t)
\end{aligned}
$$

where $\boldsymbol{X} = [\boldsymbol{X}_l; \boldsymbol{X}_u]$ and

$$
y'_t = \begin{cases} y_t, & \text{if } t = 1, \cdots, l; \\ \hat{y}_t, & \text{if } t = l+1, \cdots, l+u. \end{cases}
$$

It can be shown that maximizing the auxiliary function will produce a sequence of model parameters with increasing likelihood values. Taking the derivative of $O(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}')$ in Equation (7) with respect to each model parameter in $\boldsymbol{\Lambda}'$, we obtain the following update formula:

$$
\begin{aligned}
\boldsymbol{c}'_i &= \begin{bmatrix} r_{x_0,x_0} & r_{x_0,x_1} & \cdots & r_{x_0,x_p} \\ \vdots & \vdots & & \vdots \\ r_{x_p,x_0} & r_{x_p,x_1} & \cdots & r_{x_p,x_p} \end{bmatrix} \times \begin{bmatrix} r_{x_0,\hat{y}} \\ \vdots \\ r_{x_p,\hat{y}} \end{bmatrix} \\
\sigma_i'^2 &= \frac{1}{\sum_{t=1}^{l+u} \alpha_t(i)\beta_t(i)} \Big\{ \sum_{t=1}^{l} \alpha_t(i)\beta_t(i) w_t (y_t - \boldsymbol{c}_i \boldsymbol{x}_t^T)^2 \\
&\quad + \sum_{t=l+1}^{l+u} \alpha_t(i)\beta_t(i) w_t (\hat{y}_t - \boldsymbol{c}_i \boldsymbol{x}_t^T)^2 \Big\} \\
a'_{ij} &= \frac{1}{\sum_{t=1}^{l+u-1} \alpha_t(i)\beta_t(i)} \Big\{ \sum_{t=1}^{l-1} \alpha_t(i) a_{ij} p_j(w_{t+1}, y_{t+1}) \beta_{t+1}(j) \\
&\quad + \sum_{t=l+1}^{l+u-1} \alpha_t(i) a_{ij} p_j(w_{t+1}, \hat{y}_{t+1}) \beta_{t+1}(j) \Big\} \\
\pi'_i &= \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N} \alpha_l(j)}
\end{aligned}
$$

where:

$$
\begin{aligned}
\alpha_t(i) &= \begin{cases} \pi_i p_i(w_1, y_1), \text{if } t = 1 \\ \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] p_i(w_t, y_t), \text{if } t > 1 \end{cases} \\
\beta_t(i) &= \begin{cases} 1 \ \forall i, \text{if } t = 1 \\ \sum_{j=1}^{N} a_{ij} p_j(w_{t+1}, y_{t+1}) \beta_{t+1}(j), \text{if } t > 1 \end{cases} \\
r_{x_i,x_j} &= \sum_{t=1}^{l+u} w_t \alpha_t(i)\beta_t(i) \boldsymbol{x}_{ti} \boldsymbol{x}_{tj} \\
r_{x_i,y} &= \sum_{t=1}^{l} w_t \alpha_t(i)\beta_t(i) \boldsymbol{x}_{ti} y_t + \sum_{t=l+1}^{l+u} w_t \alpha_t(i)\beta_t(i) \boldsymbol{x}_{ti} \hat{y}_t \\
p_j(w_t, y_t) &= (2\pi\sigma_{q_t}^2)^{-\frac{1}{2}} \exp\left[- \frac{w_t (y_t - \boldsymbol{c}_{q_t} \boldsymbol{x}_t^T)^2}{2\sigma_{q_t}^2} \right]
\end{aligned}
$$

We omit the proof for the formula due to space restriction. The overall procedure for our proposed semi-supervised algorithm is summarized in Algorithm 4.1. After estimating the model parameters $\boldsymbol{\Lambda}$, the response values for future observations are predicted using Equation (4).

4.2 Data Calibration

The previous section described our proposed algorithm for semi-supervised time series prediction. The underlying assumption behind the algorithm is that the predictor variables for labeled and unlabeled data have the same distribution. This may not be true in real-world applications such as climate modeling or urban growth planning, where the unlabeled data are obtained from a different source (e.g., model simulations) or their distributions may have been perturbed by changes in the modeling domain (e.g., increase of population growth or greenhouse gas concentration).

Several recent studies on semi-supervised classification have suggested the negative effect of unlabeled data, especially when a classifier assumes an incorrect structure of the data [12] or when the labeled and unlabeled data have different distributions [25]. None of these studies, however, have been devoted to regression or time series problems. Our experimental results demonstrate that, while in most cases, semi-supervised HMMR indeed outperforms its supervised counterpart, for the climate modeling domain, where the historical and future observations come from different sources, semi-supervised learning do not significantly improve the performance of HMMR. To overcome this problem, we propose a data calibration technique to deal with the inconsistencies between historical and future data.

A straightforward way to calibrate $\boldsymbol{X_l}$ and $\boldsymbol{X_u}$ is to standardize the time series of each predictor variable by subtracting their means and dividing by their respective standard deviations. The drawback of this approach is that the covariance structures for $\boldsymbol{X_l}$ and $\boldsymbol{X_u}$ are not preserved by the standardization procedure. As a result, a model trained on the historical data may still not accurately predict the future data since the relationship between the predictor variables may have changed. In this paper, we propose a new data calibration approach to align the covariance structure of the historical data and future data. Our approach seeks to find a linear transformation matrix $\boldsymbol{\beta}$ that is applicable to the future unlabeled data $\boldsymbol{\bar{X}_u}$ such that the difference between their covariance matrices is minimized.

Let A denote the covariance matrix of $\boldsymbol{X_u}$ and B denote the covariance matrix of $\boldsymbol{X_l}$:

$$
\begin{aligned}
A &= E\Big[(\boldsymbol{X_l} - E(\boldsymbol{X_l}))^T (\boldsymbol{X_l} - E(\boldsymbol{X_l}) \Big] \\
B &= E\Big[(\boldsymbol{X_u} - E(\boldsymbol{X_u}))^T (\boldsymbol{X_u} - E(\boldsymbol{X_u})) \Big]
\end{aligned}
$$

The covariance matrix after transforming $\boldsymbol{X_u}$ to $\boldsymbol{X_u\beta}$ is

$$
\begin{aligned}
B' &= E\Big[(\boldsymbol{X_u\beta} - E(\boldsymbol{X_u\beta}))^T (\boldsymbol{X_u\beta} - E(\boldsymbol{X_u\beta})) \Big] \\
&= \boldsymbol{\beta}^T B \boldsymbol{\beta}
\end{aligned}
$$

The transformation matrix $\boldsymbol{\beta}$ can be estimated using a least-

square approach:

$$\begin{aligned} \arg\min_{\boldsymbol{\beta}} \boldsymbol{J} &= \arg\min_{\boldsymbol{\beta}} \|A - B'\|_F^2 \\ &= \arg\min_{\boldsymbol{\beta}} \|A - \boldsymbol{\beta}^T B \boldsymbol{\beta}\|_F^2 \end{aligned} \quad (8)$$

The optimization problem can be solved using a gradient descent algorithm:

$$\boldsymbol{\beta}^{i+1} = \boldsymbol{\beta}^i - \eta \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\beta}}$$

where:

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\beta}} = 4(B\boldsymbol{\beta}\boldsymbol{\beta}^T B^T \boldsymbol{\beta} - B\boldsymbol{\beta} A^T)$$

and $\eta > 0$ is the learning rate.

Although the preceding data calibration approach helps to align the covariance matrices of the data, our experimental results show that the transformation tends to significantly distort the neighborhood structure of the observations. Since our semi-supervised HMMR framework performs local estimation based on the nearest neighbor approach, such a transformation leads to unreliable local predictions and degrades the overall performance of the algorithm. An ideal transformation should preserve the neighborhood information while aligning the covariance structure. To accomplish this, we create a combined matrix $\boldsymbol{X} = [\boldsymbol{X_l}; \boldsymbol{X_u}]$, and compute the covariance matrix B using the matrix, i.e.,

$$B = E\left[(\boldsymbol{X} - E(\boldsymbol{X}))^T (\boldsymbol{X} - E(\boldsymbol{X}))\right]$$

After calibration using the gradient descent method described previously, both $\boldsymbol{X}_l$ and $\boldsymbol{X}_u$ will be transformed as follows:

$$\boldsymbol{X_l'} = \boldsymbol{X_l}\boldsymbol{\beta} \quad \boldsymbol{X_u'} = \boldsymbol{X_u}\boldsymbol{\beta}.$$

The transformed data $\boldsymbol{X_l'}$ and $\boldsymbol{X_l'}$ will serve as the new training and test data for the semi-supervised HMMR algorithm.

5. EXPERIMENTAL EVALUATION

We have conducted several experiments to evaluate the performance of our proposed algorithm. All the experiments were performed on a Windows XP machine with 3.0GHz CPU and 1.0GB RAM.

5.1 Experimental Setup

Table 5.1 summarizes characteristics of the time series used in our experiments. The time series are obtained from the UC Riverside time series data repository [21]. We divide each time series into 10 disjoint segments. To simulate this as a long-term forecasting problem, for the k-th run, we use segment k as training data and segments $k+1$ to $k+5$ as test data. The future period is therefore five times longer than the training period for each run. The results reported for each data set are based on the average root mean square error (rmse) for 5 different runs. We also consider three other competing algorithms for our experiments: (1) univariate autoregressive (AR) model, (2) multiple linear regression (MLR), and (3) supervised HMMR.

There are several parameters that must be determined for our semi-supervised HMMR, such as the number of nearest neighbors k, the number of hidden states N, and smoothness parameter μ. To determine the number of nearest neighbors, we perform 10-fold cross validation on the training data using the k nearest neighbor regression method [17]. There are various methods available to determine the number of hidden states N. These methods can be divided into two classes—modified cross-validation and penalized likelihood methods (such as AIC, BIC, and ICL). In this work, we employ the modified 10-fold cross validation with missing value approach [7] to select the best N. For each fold, we randomly select one-tenth of the observations to be removed from the training data. The likelihood function will be estimated from the remaining nine-tenth of the training data (while treating the removed data as missing values). The number of states N that produces the lowest root mean square error will be chosen as our parameter. To ensure smoothness in the target function, μ should be biased more towards the local prediction. Our experience shows that this can be accomplished by setting μ somewhere between 0.1 to 0.3. We fix the smoothness parameter $\mu = 0.1$ throughout our experiments.

Table 1: Description of the UCR time series data sets

Data Sets	Length	# Variables
Dryer	867	6
Foetal	2500	9
Glass	1247	9
Greatlake	984	5
Steam	9600	4
Twopat	5000	129
Logistic	1000	101
Leaf	442	151
Cl2full	4310	166

5.2 Performance Comparison

Table 5.2 compares the root mean square errors of our proposed framework (semiHMMR) against the univariate AR (UAR), multiple linear regression(MLR), and supervised Hidden Markov Model Regression (HMMR). First, observe that the performance of univariate AR is significantly worse than multivariate MLR and supervised HMMR model. This is consistent with the prevailing consensus that multivariate prediction approaches are often more effective than univariate approaches because they may utilize information in the predictor variables to improve its prediction. Second, we observe that HMMR generally performs better than MLR on most of the data sets. This result suggests the importance of learning models that take into consideration the dependencies between observations. Finally, the results also show that semi-supervised HMMR is significantly better than HMMR on the majority of the data sets. The improvements achieved using semi-supervised HMMR exceed 10% on data sets such as "Greatlake", "Steam", "Leaf" and "Cl2full".

5.3 Value of Unlabeled Data

The purpose of this experiment is to show the value of unlabeled data when there is limited training data. We have used the "Steam" time series for this experiment and vary the ratio of labeled to unlabeled data from 0.2 to 1 and report its average root mean square. As shown in Figure 3, when the ratio is 1, the rmse for MLR and supervised HMMR are

Table 2: Average root mean square error for UAR, MLR, HMMR and SemiHMMR on UCR time series data sets

Data Sets	UAR	MLR	HMMR	SemiHMMR
Dryer	12.0612	9.2975	8.5024	7.4182
Foetal	5.6943	4.7785	4.3623	4.0973
Glass	2.1215	1.0519	1.0445	1.0091
Greatlake	35.4278	24.5924	23.0445	20.0100
Steam	8.7834	7.9398	7.7537	5.7396
Twopat	1.3111	1.1426	1.2404	1.1402
Logistic	0.8732	0.5354	0.5332	0.5302
Leaf	1.0181	0.9812	0.9811	0.7712
Cl2full	0.3336	0.0889	0.0888	0.0575

nearly the same as that for semi-supervised HMMR. However, when the ratio decreases to 0.2, the performance of both MLR and supervised HMMR degrades rapidly (from 5.8093 to 7.9819 for MLR and from 5.6074 to 7.7537 for HMMR) whereas the rmse for semi-supervised HMMR increases only slightly, from 5.5370 to 5.7442. The results of this experiment show that semi-supervised HMMR can effectively utilize information in the unlabeled data to improve its prediction, especially when labeled data is scarce.

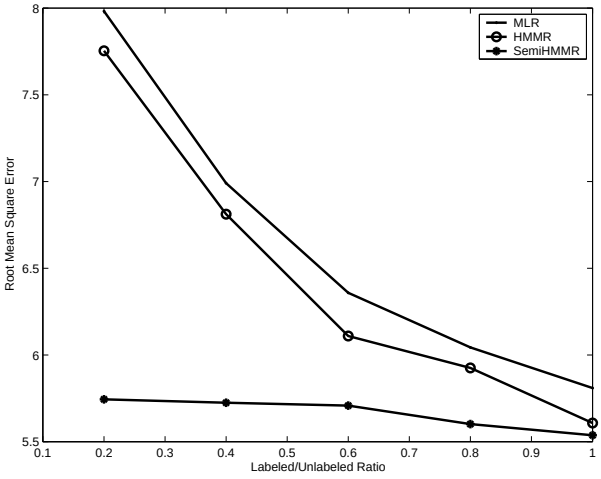


Figure 3: The performance of MLR, HMMR and SemiHMMR when varying the ratio of labeled to unlabeled data

5.4 Effect of Covariance Alignment on Semi-Supervised HMMR

This experiment investigates the effect of data calibration on the performance of semi-supervised HMMR. We show that data calibration is useful when the historical and future observations come from different sources. For this experiment, we downloaded climate data from the Canadian Climate Change Scenarios Network web site [1]. The data consists of daily observations for 26 climate predictor variables including sea-level pressure, wind direction, vorticity, humidity, etc. The response variable corresponds to the observed mean temperature at a meteorological station. In short, there are three sources of data for this experiment: (1) Mean temperature observations from 1961 to 2001 to be used as response variable, (2) Reanalysis data from NCEP (National Center for Environmental Prediction) reanalysis project from 1961 to 2001 to be used as predictor variables for training data, and (3) Simulation data from HADCM3 global climate model from 1961 to 2099 to be used as predictor variables for future data. Since the mean temperature for the future time period is unavailable, we conducted our experiment using NCEP reanalysis and the observed mean temperature data from 1961 to 1965 for training and HADCM3 simulation data from 1966 to 1991 for testing.

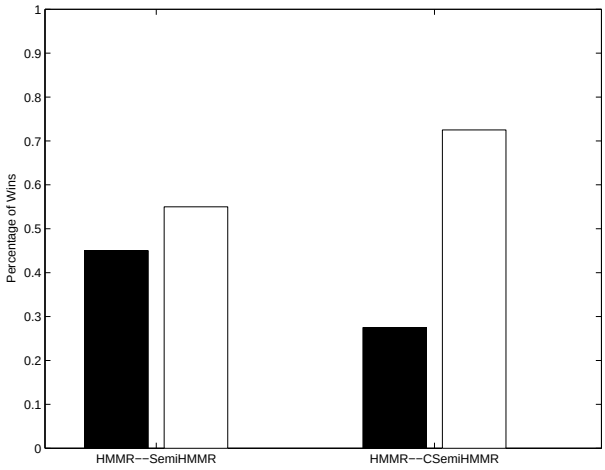


Figure 4: Performance comparison between HMMR, SemiHMRR and CSemiHMMR on Canadian climate data

To compare the relative performance of supervised HMMR, semi-supervised HMMR (SemiHMMR) and semi-supervised HMMR with data calibration (CSemiHMMR), we applied the algorithms to predict the mean daily temperature for 40 randomly selected meteorological stations[1] in Canada. The bar chart shown in Figure 4 indicates the fraction of locations in which one algorithm has a lower rmse than the other. Unlike the results reported in Section 5.2, the bar chart seems to suggest that the performance of semi-supervised HMMR is only comparable to supervised HMMR (55% versus 45%). This is actually consistent with the conclusions drawn in [12, 25] for semi-supervised classification in which it was suggested that unlabeled data with different distribution may not improve the performance of a semi-supervised algorithm. However, with the calibration method developed in Section 4.2, CSemiHMMR actually performs better than HMMR on 72% of the data sets. This result confirmed the effectiveness of incorporating the covariance alignment technique to our semi-supervised learning framework. We also illustrate the rmse values for five of the selected stations in Table 3. The latitude and longitude for each station is recorded in the first column of Table 3. Though the performance of semiHMMR appears to be worse than supervised HMMR at (52.2°N, 113.9°W) and (48.3°N 71°W), the rmse values for semi-supervised HMMR with data calibration is clearly superior for both locations.

In Section 4.2, we argued that aligning $\boldsymbol{X}_l$ with $\boldsymbol{X}_u$ (we call this calibration technique 1) may not be as effective as

[1]Another criterion for choosing a station is that the time series must be complete, i.e., it has no missing values.

Table 3: Comparing the average rmse values for MLR, HMMR, SemiHMMR, and CSemiHMMR on the Canadian climate data

(Lat°,Lon°)	MLR	HMMR	SemiHMMR	CSemiHMMR
(48.6N,123.4W)	0.855	0.825	0.758	0.764
(48.9N,54.6W)	0.832	0.812	0.752	0.730
(52.2N,113.9W)	0.762	0.710	0.713	0.684
(48.3N,71W)	0.652	0.611	0.622	0.592
(82.5N,62.3W)	0.742	0.708	0.641	0.618

Table 4: Comparing the degree of alignment and loss of neighborhood structure information using calibration techniques 1 and 2

(Lat°,Lon°)	RCovDiff1	RCovDiff2	NNLoss1	NNLoss2
(48.6N,123.4W)	0.998	0.785	0.994	0.186
(48.9N,54.6W)	0.993	0.714	0.998	0.116
(52.2N,113.9W)	0.989	0.728	0.995	0.108
(48.3N,71W)	0.992	0.703	0.996	0.086
(82.5N,62.3W)	0.993	0.702	0.991	0.096

calibrating $\boldsymbol{X}_l$ with the combined matrix $\boldsymbol{X}_C = [\boldsymbol{X}_u, \boldsymbol{X}_u]$ (we call this calibration technique 2). This is because the former may result in significant loss of nearest neighbor information, thus degrading the performance of semi-supervised HMMR[2]. To measure the degree of alignment and loss of neighborhood information using the calibration techniques, we define the following two measures: RCovDiff and NNLoss. Let $\Delta_0(\boldsymbol{X}_A, \boldsymbol{X}_B)$ denote the difference between the covariance matrices constructed from $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ before alignment, and $\Delta_1(\boldsymbol{X}_A, \boldsymbol{X}_B)$ denote the corresponding difference after alignment. RCovDiff measures the reduction of the covariance difference before and after alignment, i.e.:

$$\text{RCovDiff1} = \frac{|\Delta_0(\boldsymbol{X}_l, \boldsymbol{X}_u) - \Delta_1(\boldsymbol{X}_l, \boldsymbol{X}_u)|}{\Delta_0(\boldsymbol{X}_l, \boldsymbol{X}_u)}$$

$$\text{RCovDiff2} = \frac{|\Delta_0(\boldsymbol{X}_l, \boldsymbol{X}_C) - \Delta_1(\boldsymbol{X}_l, \boldsymbol{X}_C)|}{\Delta_0(\boldsymbol{X}_l, \boldsymbol{X}_C)}$$

A calibration technique with larger RCovDiff will produce covariance matrices that are better aligned with each other. We also measure the loss of neighborhood structure due to data calibration in the following way. Let $M_0(\boldsymbol{X}_A, \boldsymbol{X}_B)$ denote a 0/1 matrix computed based on the 1-nearest neighbor of each example in $\boldsymbol{X}_B$ to the examples in $\boldsymbol{X}_A$ before alignment and $M_1(\boldsymbol{X}_A, \boldsymbol{X}_B)$ denote the corresponding matrix after alignment. The NNLoss measure is defined as follows:

$$\text{NNLoss} = \frac{|M_0(\boldsymbol{X}_l, \boldsymbol{X}_u) - M_1(\boldsymbol{X}_l, \boldsymbol{X}_u)|}{M_0(\boldsymbol{X}_l, \boldsymbol{X}_u)}$$

Unlike RCovDiff, the same equation is applied to both calibration techniques 1 and 2. Table 4 compares the results of both calibration techniques. Although calibration technique 1 produces more well-aligned covariance matrices, it loses more information about its neighborhood structure. This explains our rationale for using calibration technique 2 for semi-supervised HMMR.

6. RELATED WORK

There have been extensive studies on the effect of incorporating unlabeled data to supervised classification problems, including those based on generative models[13], transductive SVM [19], co-training [4], self-training [31] and graph-based methods [3][32]. Some studies concluded that significant improvements in classification performance can be achieved when unlabeled examples are used, while others have indicated otherwise [4][10][12][25]. Blum and Mitchell [4] and Cozman et al. [10] suggested that unlabeled data can help to reduce variance of the estimator as long as the modeling assumptions match the ground truth data. Otherwise, unlabeled data may either improve or degrade the classification performance, depending on the complexity of the classifier compared to the training set size [12]. Tian et al. [25] showed the ill effects of using different distributions of labeled and unlabeled data on semi-supervised learning.

Recently, there have been growing interests on applying semi-supervised learning to regression problems [30][6][11][33]. Some of these approaches are direct extensions from their semi-supervised classification counterparts. For example, transductive support vector regression is proposed in [11] as an extension to transductive SVM classifier. Zhou and Li developed a co-training approach for semi-supervised regression in [30]. Their algorithm employs two KNN regressors, each using a different distance metric. Another extension of co-training to regression problems was developed by Brefeld et al. [6]. Graph based semi-supervised algorithms [3][32] utilize a label propagation process to ensure that the smoothness assumption holds for both labeled and unlabeled data. An extension of the algorithm to regression problems were proposed by Wang et al. in [26]. Zhu and Goldberg [33] developed a semi-supervised regression method that incorporates additional domain knowledge to improve model performance. Since all of the previous approaches ignore the temporal dependencies between observations, they are not well-suited for time series prediction problems.

7. CONCLUSIONS

Long term time series forecasting is an important but challenging problem with many applications. In this paper, we develop and evaluate a semi-supervised time series prediction approach for Hidden Markov Model Regression (HMMR). We show that the inconsistency between training and test data may actually hurt the model's performance. To compensate for this problem, we propose a covariance aligning data calibration method that will transform the training and test data into a new space before applying the semi-supervised HMMR algorithm. Experimental results on several real-world data sets clearly demonstrate the effectiveness of the proposed algorithms.

8. ACKNOWLEDGMENTS

This work is supported by NSF grant #0712987. The authors would like to thank Dr Eamonn Keogh for providing the time series data. The authors would also like to thank Dr Julie Winkler and Dr Sharon Zhong for valuable discussion and comments.

9. REFERENCES

[1] http://www.ccsn.ca/, canadian climate change scenarios network, environment canada.

[2] P. M. Baggenstos. A modified Baum-Welch algorithm for hidden markov models with multiple observation

[2]Note that the result shown in Figure 4 is based on calibration technique 2.

spaces. *IEEE Trans. on Speech Audio Processing*, pages 411–416, 2001.

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. of the 18th Int'l Conf. on Machine Learning*, pages 19–26, 2001.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the Workshop on Computational Learning Theory*, pages 92–100, 1998.

[5] Y.-A. L. Borgne, S. Santini, and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Process*, 87(12):3010–3020, 2007.

[6] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *Proc. of the 23rd Int'l Conf. on Machine learning*, pages 137–144, 2006.

[7] G. Celeux and J. Durand. Selecting hidden markov model state number with cross-validated likelihood. In *Computational Statistics*, 2007.

[8] S. Charles, B. Bates, I. Smith, and J. Hughes. Statistical downscaling of daily precipitation from observed and modelled atmospheric fields. In *Hydrological Processes*, pages 1373–1394, 2004.

[9] H. Cheng, P.-N. Tan, J. Gao, and J. Scripps. Multistep-ahead time series prediction. In *Proc. of the Pacific-Asia Conf on Knowledge Discovery and Data Mining*, pages 765–774, 2006.

[10] I. Cohen, N. Sebe, F. G. Cozman, M. C. Cirelo, and T. S. Huang. Semi-supervised learning of classifiers: Theory and algorithms for bayesian network classifiers and applications to human-computer interaction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(12):1553–1566, Dec 2004.

[11] C. Cortes and M. Mohri. On transductive regression. In *Advances in Neural Information Processing Systems*, 2006.

[12] F. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *Proc. of the 15th Int'l Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.

[13] F. Cozman, I. Cohen, and M. Cirelo. Semi-supervised learning of mixture models. In *Proc of the 20th Int'l Conf. on Machine Learning*, 2003.

[14] W. Enke and A. Spekat. Downscaling climate model outputs into local and regional weather elements by classification and regression. In *Climate Research 8*, pages 195–207, 1997.

[15] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama. Multiple-regression hidden markov model. In *Proc. of IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, 2001.

[16] C. Giles, S. Lawrence, and A. Tsoi. Noisy time series prediction using a recurrent neural network and grammatical inference. *Machine Learning, 44(1-2)*, pages 161–183, 2001.

[17] T. Hastie and C. Loader. Local regression: Automatic kernel carpentry. In *Statistical Science*, pages 120–143, 1993.

[18] W. Hong, P. Pai, S. Yang, and R. Theng. Highway traffic forecasting by support vector regression model with tabu search algorithms. In *Proc. of Int'l Joint Conf. on Neural Networks*, pages 1617–1621, 2006.

[19] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of the 16th Int'l Conf. on Machine Learning*, pages 200–209, Bled, SL, 1999.

[20] B. Kedem and K. Fokianos. Regression models for time series analysis. *Wiley-Interscience ISBN: 0-471-36355*, 2002.

[21] E. Keogh and T. Folias. Uc riverside time series data mining archive. http://www.cs.ucr.edu/ eamonn/TSDMA/index.html.

[22] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. In *Computer Speech and Language*, pages 171–185(15). Academic Press, 1995.

[23] A. Ober-Sundermeier and H. Zackor. Prediction of congestion due to road works on freeways. In *Proc. of IEEE Intelligent Transportation Systems*, pages 240–244, 2001.

[24] A. Smola and B. Scholkopf. A tutorial on support vector regression. In *Statistics and Computing*, pages 199–222(24). Spring, 2004.

[25] Q. Tian, J. Yu, Q. Xue, and N. Sebe. A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval. In *Proc. of IEEE Int'l Conf. on Multimedia and Expo.*, pages 1019– 1022, 2004.

[26] M. Wang, X.-S. Hua, Y. Song, L.-R. Dai, and H.-J. Zhang. Semi-supervised kernel regression. In *Proc. of the 6th Int'l Conf. on Data Mining*, pages 1130–1135, Washington, DC, USA, 2006.

[27] R. Wilby, S. Charles, E. Zorita, B. Timbal, P. Whetton, and L. Mearns. Guidelines for use of climate scenarios developed from statistical downscaling methods. Available from the DDC of IPCC TGCIA, 2004.

[28] C.-C. C. Wong, M.-C. Chan, and C.-C. Lam. Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization. Technical Report 61, Society for Computational Economics, Jul 2000.

[29] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2003.

[30] Z. Zhou and M. Li. Semi-supervised regression with co-training. In *Proc. of Int'l Joint Conf. on Artificial Intelligence*, 2005.

[31] X. Zhu. Semi-supervised learning literature survey. In *Technical Report,Computer Sciences, University of Wisconsin-Madison*, 2005.

[32] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the 20th Int'l Conf. on Machine Learning*, volume 20, 2003.

[33] X. Zhu and A. Goldberg. Kernel regression with order preferences. In *Association for the Advancement of Artificial Intelligence*, page 681, 2007.