ORION: Online Regularized multI-task regressiON and its application to ensemble forecasting

Jianpeng Xu*, Pang-Ning Tan* and Lifeng Luo†
Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48823
Department of Geography, Michigan State University, East Lansing, MI, 48823
Email: {xujianpe,ptan,lluo}@msu.edu

***Abstract*—Ensemble forecasting is a well-known numerical prediction technique for modeling the evolution of nonlinear dynamic systems. The ensemble member forecasts are generated from multiple runs of a computer model, where each run is obtained by perturbing the starting condition or using a different model representation of the dynamic system. The ensemble mean or median is typically chosen as the consensus point estimate of the aggregated forecasts for decision making purposes. These approaches are limited in that they assume each ensemble member is equally skillful and do not consider their inherent correlations. In this paper, we cast the ensemble forecasting task as an online, multi-task regression problem and present a framework called *ORION* to estimate the optimal weights for combining the ensemble members. The weights are updated using a novel *online learning with restart* strategy as new observation data become available. Experimental results on seasonal soil moisture predictions from 12 major river basins in North America demonstrate the superiority of the proposed approach compared to the ensemble median and other baseline methods.**

***Keywords*-Online Multi-task Learning; Ensemble Forecasting**

I. Introduction

Process-based modeling refers to the use of computer models to predict the future states of complex, dynamical systems based on mathematical formulations of the physical processes governing the behavior of such systems. Since the models may not fully capture all the underlying processes as well as their parameterization accurately, their forecast errors tend to amplify with increasing lead time. Ensemble forecasting [6] aims at quantifying the range of such forecast uncertainties by combining multiple runs of the computer model to generate the final forecast. To date, this approach has been employed for various applications, including weather [5], [6], hydrological [10], [7], and species distribution [1] forecasting systems.

Figure 1 illustrates the ensemble forecasting task. A new set of forecasts is generated periodically, say, every 5 days. Each forecast corresponds to an output prediction from a single run of the computer model. Different forecasts can be generated by perturbing the initial condition or using a different model representation. The complete set of forecasts is known as the ensemble, whereas the individual forecasts within it are called ensemble members. Since the computer model is typically run for an extended duration, each run contains forecasts for T consecutive time steps. In this paper, we refer to T as the forecast duration and N as the number of forecast runs generated by each ensemble member (see Figure 1).

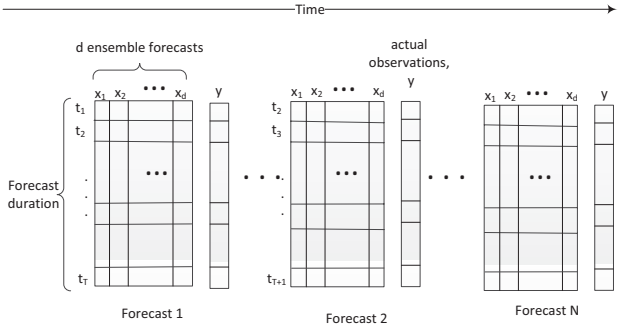


Fig. 1: A schematic illustration of ensemble forecasting task.

The forecasts need to be combined to obtain a consensus estimate of the target variable to be predicted. The ensemble mean or median is typically chosen as an unbiased estimator of the aggregated forecasts. Both approaches are reasonable if each ensemble member forecast is equally plausible. However, in reality, it might be unwise to weigh the ensemble members equally as some members might be more accurate than others. To illustrate this, consider the diagram given in Figure 2, which shows the basin-averaged soil moisture percentile forecasts of a hydrological model ensemble (with 33 members), along with the ensemble median and observation data. The ensemble predictions were made for a forecast period between September 2, 2011 and October 12, 2011. Since the models were calibrated with observed soil moisture data from September 2, 2011, all the ensemble member predictions are consistent with observation data at the beginning of their runs. However, the individual forecasts by the ensemble members (shown as thin green lines) began to diverge with increasing lead time. Some ensemble members clearly do not accurately predict the observed time series (represented by the red solid line), which in turn, affects the accuracy of the ensemble median approach (shown by the dashed line). This example motivates the need for learning an optimal set of weights to combine the ensemble member predictions to improve the aggregated forecasts.

Furthermore, as time progresses, new observations become available to verify the earlier forecasts. An online learning approach is more suitable in this setting because it can adapt the weights of the ensemble members according to their relative skills in fitting the new verification data. However, unlike conventional online learning, the ensemble forecasting

1550-4786/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDM.2014.90

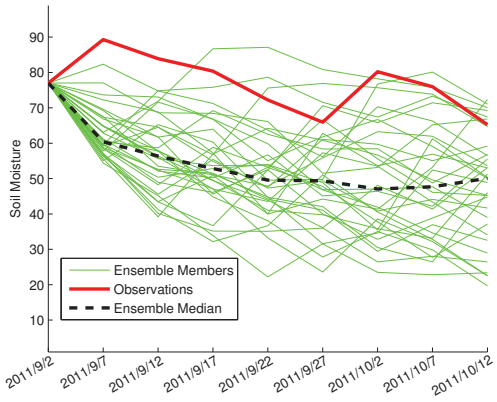


Fig. 2: Seasonable soil moisture forecasts and the observed time series at a major river basin in North America.

task requires making predictions for a time window of length T. The prediction tasks within the window are obviously not independent due to the temporal autocorrelation of the time series. Thus, it makes sense to approach the ensemble forecasting task as an online multi-task regression problem, in which the prediction steps within each time window are considered a set of related learning tasks.

The online multi-task learning setting considered in this study is different from those described in current literature [4], [2], [9] in that not all observation data are available in the time window when the model is updated. For example, suppose the ensemble members generate forecasts every 5 days. Let $\mathbf{X}_1$ be the set of forecasts generated on June 1 for a 40-day forecast period starting from June 6 until July 16, $\mathbf{X}_2$ be the corresponding forecasts generated on June 6 for the time window June 11 to July 21, and $\mathbf{X}_3$ be the forecasts generated on June 11 for June 16 to July 26. When the online learning model is updated with $\mathbf{X}_3$ on June 11, $\mathbf{X}_1$ has two observed values in its time window, one for June 6 and another for June 11, whereas $\mathbf{X}_2$ has observation value only for June 11. This means the observation data are not only incomplete in each time window, the number of observations also varies from one window to another. We termed this problem *online multi-task learning with partially observed data*. Due to this property of the data, instead of updating the model from its most recent time window, we need to update some of the older models from previous time windows as new verification data become available.

To overcome this challenge, we present a framework called ORION (which stands for **O**nline **R**egularized mult**I**-task regressi**ON**) to estimate the weights of the ensemble members. The framework uses an *online learning with restart* strategy to deal with the partially observed data. It also employs graph regularization constraints to ensure smoothness in the model parameters while taking into account the task relatedness within each time window. Although the ORION framework is applicable to different types of loss functions, in this paper, we demonstrate its effectiveness for the ϵ-insensitive loss function. The main contributions of this paper are summarized below:

- We introduce the problem of online regularized multi-task regression with partially observed data and demonstrate its relevance to the ensemble forecasting task.
- We present a novel framework called ORION, which uses an online learning with restart strategy to solve the problem. It also uses a graph Laplacian to capture relationships among the learning tasks along with a passive aggressive update scheme to optimize the ϵ-insensitive loss function.
- Experimental results suggest that our method reduces the forecast error of ensemble median for all major river basins datasets, and performs better than other baseline algorithms in most cases.

II. Problem Formulation

We consider a variation of the online multi-task learning process described in [4], in which the learning proceeds in a sequence of rounds. At the start of round n, where $n \in \{1, 2, \cdots, N\}$, the algorithm observes T instances, $\mathbf{x}^{(n)} = \{\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \cdots, \mathbf{x}_T^{(n)}\}$, where each instance $\mathbf{x}_j^{(n)} \in \Re^d$ is a d-dimensional vector of predictor variables. The algorithm then predicts the target value $f(\mathbf{x}_i)$ for each of the instances. We consider the prediction of each instance as a separate learning task. The algorithm subsequently observes the true values y_i for a subset of the tasks. Our goal is to learn a set of prediction functions for the T tasks such that their cumulative loss over the N rounds is minimized. Similar to previous works [4], [2], we consider only linear prediction functions of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w} \in \Re^d$ is the parameter vector.

The ensemble forecasting task can be cast into an online multi-task learning problem, where each task corresponds to a prediction for a particular time step in the given forecast window. The forecasts generated by the ensemble members form the set of predictor variables while the observation value at each time step determines the target value. The number of learning tasks is given by the forecast duration T, while the number of rounds for the online learning process is equal to the number of forecast runs N (see Figure 1). The amount of labeled observations available varies from one time window to another and increases as time progresses. Let $\mathbf{y}_m^{(n)} = \{y_1^{(n)}, y_2^{(n)}, \cdots, y_{m_n}^{(n)}\}$ denote the labeled observations available in round n for the set of forecasts generated in round m, where $m \leq n$ and $m_n \leq T$. If $m < n - T$, then $m_n = T$, which means the target values in $\mathbf{y}_m^{(n)}$ are completely observed. In contrast, if $n - T \leq m < n$, then $m_n = n - m$. Finally, $\mathbf{y}_m^{(n)}$ is an empty set if $m = n$. This partially observed data scenario distinguishes our framework from other existing works on online multi-task regression.

Let f_{n-1} be the model generated after round $n-1$ based on $(\mathbf{x}^{(n-1)}, \mathbf{y}_{n-1}^{(n-1)})$. It is insufficient to generate f_n in round n based on the previous model f_{n-1} alone since the latter, which was updated from f_{n-2} using partially observed data, is also outdated given the new verification data. To overcome this problem, we employ the following *online learning with restart strategy*. At each round n, we first update the labeled data from previous time windows $\mathbf{y}_{n-1}^{(n-1)}, \mathbf{y}_{n-2}^{(n-1)}, \cdots, \mathbf{y}_{n-T}^{(n-1)}$

to include the newly observed target value. The online learning algorithm is then restarted from f_{n-T} and iteratively updated until f_n is obtained. With this strategy, the algorithm needs to maintain two sets of weights, $\mathbf{w}^{(n)}$ and $\mathbf{w}^{(n-T-1)}$, to compute the prediction in the next round using $\mathbf{w}^{(n)}$ and to update the model starting from $\mathbf{w}^{(n-T)}$, which was the last set of weights estimated from complete observation data.

III. ONLINE REGULARIZED MULTI-TASK REGRESSION (ORION)

This section presents the ORION framework for the ϵ-insensitive loss function.

A. ORION for ϵ-insensitive Loss Function

Although our framework requires the online learning process to be re-started at round $n-T$ and continues until round n to deal with the partially incomplete data problem, the update formula and optimization step in each round are identical. Specifically, in round n, the ORION framework assumes that the weights are co-regularized as $\mathbf{w}_t^{(n)} = \mathbf{w}_0^{(n)} + \mathbf{v}_t^{(n)}, \ \ \forall t \in \{1, 2, \cdots, T\}$. In other words, the prediction functions for all T tasks share a common term $\mathbf{w}_0$ and a task-specific weight $\mathbf{v}_t$, which is expected to be small when the predictions are correlated. To estimate the weights, we employ the following objective function, which extends the formulation given in [3] for single-task classification to a multi-task learning setting with an ϵ-insensitive loss function:

$$\begin{aligned} \arg\min_{\mathbf{w}_0, \{\mathbf{v}_t\}} \quad & \frac{1}{2}\sum_{t=2}^{T} ||\mathbf{w}_t - \mathbf{w}_{t-1}||_2^2 + \frac{\mu}{2}\sum_{t=1}^{T} ||\mathbf{v}_t||_2^2 \\ + \quad & \frac{\lambda}{2}||\mathbf{w}_0 - \mathbf{w}_0^{(n-1)}||_2^2 + \frac{\beta}{2}\sum_{t=1}^{T} ||\mathbf{v}_t - \mathbf{v}_t^{(n-1)}||_2^2 \\ s.t. \quad & \\ & \forall t \le m_n : \ |\mathbf{w}_t^T \mathbf{x}_t^{(n)} - y_t^{(n)}| \le \epsilon \\ & \forall t \in \{1, 2, \cdots, T\} : \ \mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t \\ & \mu, \lambda, \beta \text{ and } \epsilon \ge 0 \end{aligned} \tag{1}$$

where m_n is the number of labeled observations, $\mathbf{x}_t^{(n)}$ are the predictor variables for task t in the n-th round, and $y_t^{(n)}$ is its corresponding target value. For brevity, we have omitted the superscript n in our notations for $\mathbf{v}_t$, $\mathbf{w}_t$, and $\mathbf{w}_0$. Since $\mathbf{w}_t - \mathbf{w}_{t-1} = \mathbf{v}_t - \mathbf{v}_{t-1}$, Equation (1) can be simplified as follows:

$$\begin{aligned} \arg\min_{\mathbf{w}_0, \mathbf{V}} \quad & \frac{1}{2}\,\mathrm{Tr}\left[\mathbf{V}^T(\mathbf{L} + \mu\mathbf{I}_T)\mathbf{V})\right] \\ + \quad & \frac{\lambda}{2}||\mathbf{w}_0 - \mathbf{w}_0^{(n-1)}||_2^2 + \frac{\beta}{2}||\mathbf{V} - \mathbf{V}^{(n-1)}||_F^2 \\ s.t. \quad & \\ & \forall t \le m_n, \ |\mathbf{w}_t^T \mathbf{x}_t^{(n)} - y_t^{(n)}| \le \epsilon \\ & \forall t \in \{1, 2, \cdots, T\}, \ \mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t \end{aligned} \tag{2}$$

where $\mathbf{V} = [\mathbf{v}_1^T; \mathbf{v}_2^T; \cdots; \mathbf{v}_T^T]$ is a $T \times d$-dimensional matrix, $\mathbf{I}_T$ is a $T \times T$ identity matrix, $\mathrm{Tr}[\cdot]$ denote the matrix trace operator and

$$\mathbf{L}_{i,j} = \begin{cases} 1 & \text{if } i = j = 1 \text{ or } i = j = T \\ 2 & \text{if } i = j \ne 1 \text{ and } i = j \ne T \\ -1 & \text{if } i = j + 1 \text{ or } i = j - 1 \\ 0 & \text{otherwise} \end{cases}$$

is a graph Laplacian capturing the relationships among the T tasks. The Lagrange formulation of the objective function is given by

$$\begin{aligned} \mathcal{L}(\mathbf{w}_0, \mathbf{V}, \boldsymbol{\tau}) \ = \ & \frac{1}{2}\,\mathrm{Tr}(\mathbf{V}^T(\mathbf{L} + \mu\mathbf{I}_T)\mathbf{V}) \\ + \ & \frac{\lambda}{2}||\mathbf{w}_0 - \mathbf{w}_0^{(n-1)}||_2^2 + \frac{\beta}{2}||\mathbf{V} - \mathbf{V}^{(n-1)}||_F^2 \\ + \ & \sum_{t \in \mathcal{O}^{(n)}} \tau_t(|\mathbf{w}_t^T \mathbf{x}_t^{(n)} - y_t^{(n)}| - \epsilon) \end{aligned}$$

where $\mathcal{O}^{(n)} = \{t | t \le m_n \text{ and } |\mathbf{w}_t^T \mathbf{x}_t^{(n)} - y_t^{(n)}| > \epsilon\}$ is the feasible set and $\boldsymbol{\tau} = \{\tau_t\}$ is the set of Lagrangian multipliers such that $\tau_t \ge 0$ for all $t \in \mathcal{O}^{(n)}$ and $\tau_t = 0$ for all $t \notin \mathcal{O}^{(n)}$. In the next subsection, we present the solution for this optimization problem.

Notation	Definition
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product between matrices $\mathbf{A}$ and $\mathbf{B}$
$\tilde{\mathbf{X}}^{(n)}$	$\begin{bmatrix} \mathbf{x}_1^{(n)} & \mathbf{x}_2^{(n)} & \cdots & \mathbf{x}_T^{(n)} \\ \mathbf{x}_1^{(n)} & \mathbf{0}_d & \cdots & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{x}_2^{(n)} & \cdots & \mathbf{0}_d \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_d & \mathbf{0}_d & \cdots & \mathbf{x}_T^{(n)} \end{bmatrix}$
$\mathbf{y}^{(n)}$	$[y_1^{(n)}; y_2^{(n)}; \cdots; y_{m_n}^{(n)}; \mathbf{0}_{(T-m_n)}]$
$\mathbf{P}$	$\mathbf{P}_{i,j} = \begin{cases} 1 & \text{if } i = j \text{ and } i \in \mathcal{O}^{(n)} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{S}$	$\mathbf{S}_{i,j} = \begin{cases} \mathrm{sign}(\mathbf{w}_i^{(n)^T} \mathbf{x}_i^{(n)} - y_i^{(n)}) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$
$\boldsymbol{\tau}$	$[\tau_1; ...; \tau_T]$
$\mathbf{R}$	$\begin{bmatrix} \lambda\mathbf{I}_d & \mathbf{0}_{d \times Td} \\ \mathbf{0}_{Td \times d} & \beta\mathbf{I}_{Td} \end{bmatrix}$
$\mathbf{Q}$	$\begin{bmatrix} \mathbf{0}_{d \times d} & \mathbf{0}_{d \times Td} \\ \mathbf{0}_{Td \times d} & (\mathbf{L} + \mu\mathbf{I}_T) \otimes \mathbf{I}_d \end{bmatrix}$

TABLE I: Notations used in Equation (3)

B. Optimization

To simplify the notation, we first vectorize the matrix $\mathbf{V}$ and concatenate it with $\mathbf{w}_0$. Let $\mathbf{z} = [\mathbf{w}_0; \mathbf{v}_1; ..., \mathbf{v}_T]$ denote the resulting weight vector to be solved. The Lagrangian can now be written into the following form:

$$\begin{aligned} \mathcal{L}(\mathbf{z}, \boldsymbol{\tau}) \ = \ & \frac{1}{2}(\mathbf{z} - \mathbf{z}^{(n-1)})^T \mathbf{R}(\mathbf{z} - \mathbf{z}^{(n-1)}) + \frac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z} \\ + \ & \left[(\mathbf{z}^T\tilde{\mathbf{X}}^{(n)} - \mathbf{y}^{(n)^T})\mathbf{S} - \epsilon\mathbf{1}^T\right]\mathbf{P}\boldsymbol{\tau} \end{aligned} \tag{3}$$

where $\tilde{\mathbf{X}}^{(n)}$, $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{S}$ are defined in Table I.

Taking the partial derivative of $\mathcal{L}$ with respect to $\mathbf{z}$ and setting it to zero yields the following

$$\begin{aligned} \frac{\partial\mathcal{L}(\mathbf{z}, \boldsymbol{\tau})}{\partial\mathbf{z}} \ &= \ \mathbf{R}(\mathbf{z} - \mathbf{z}^{(n-1)}) + \mathbf{Q}\mathbf{z} + \tilde{\mathbf{X}}^{(n)}\mathbf{S}\mathbf{P}\boldsymbol{\tau} = 0 \\ \mathbf{z} \ &= \ \mathbf{M}(\mathbf{R}\mathbf{z}^{(n-1)} - \tilde{\mathbf{X}}^{(n)}\mathbf{S}\mathbf{P}\boldsymbol{\tau}) \end{aligned} \tag{4}$$

where $\mathbf{M} = (\mathbf{R} + \mathbf{Q})^{-1}$. It can be easily shown that $\mathbf{R} + \mathbf{Q}$ is a positive definite matrix, which means it is invertible and its inverse is also positive definite.

Plugging $\mathbf{z}$ in Equation (4) back into Equation (3) leads to the following equation after simplification

$$\mathcal{L}(\boldsymbol{\tau}) = -\frac{1}{2}\boldsymbol{\tau}^T \tilde{\mathbf{X}}_{PS}^{(n)T} \mathbf{M}^T \tilde{\mathbf{X}}_{PS}^{(n)} \boldsymbol{\tau} + \ell_n^T(\hat{\mathbf{z}}^{(n-1)})\boldsymbol{\tau} + \text{constant} \tag{5}$$

where

$$\begin{aligned} \tilde{\mathbf{X}}_{PS}^{(n)} &= \tilde{\mathbf{X}}^{(n)}\mathbf{S}\mathbf{P} \\ \ell_n^T(\hat{\mathbf{z}}^{(n-1)}) &= \left[(\hat{\mathbf{z}}^{(n-1)T}\tilde{\mathbf{X}}^{(n)} - \mathbf{y}^{(n)T})\mathbf{S} - \epsilon\mathbf{1}^T\right]\mathbf{P} \\ \hat{\mathbf{z}}^{(n-1)} &= \mathbf{M}^T\mathbf{R}^T\mathbf{z}^{(n-1)} \end{aligned} \tag{6}$$

Note that $\mathbf{P}$ is a diagonal matrix, whose diagonal element $\mathbf{P}_{t,t}$ is zero if $t \notin \mathcal{O}^{(n)}$. In other words, if the target value for task t is either unavailable or predicted correctly (within the ϵ-insensitive bound), all the elements in the t-th column of $\tilde{\mathbf{X}}_{PS}^{(n)}$ become 0, and the corresponding t-th element in $\ell_n^T(\hat{\mathbf{z}}^{(n-1)})$ is also 0. Thus, τ_t for $t \notin \mathcal{O}^{(n)}$ has no impact on Equation (5) and can be set to zero. In the following derivation, we assume the rows and columns corresponding to all the tasks $t \notin \mathcal{O}^{(n)}$ in τ, $\tilde{\mathbf{X}}_{PS}^{(n)}$, and $\ell_n^T(\hat{\mathbf{z}}^{(n-1)})$ have been removed.

Taking the partial derivative of the "reduced" Lagrangian with respect to $\boldsymbol{\tau}$ and setting it to zero yields

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}} &= -\tilde{\mathbf{X}}_{PS}^{(n)T}\mathbf{M}^T\tilde{\mathbf{X}}_{PS}^{(n)}\boldsymbol{\tau} + \ell_n(\hat{\mathbf{z}}^{(n-1)}) = 0 \\ \boldsymbol{\tau} &= \left[\tilde{\mathbf{X}}_{PS}^{(n)T}\mathbf{M}^T\tilde{\mathbf{X}}_{PS}^{(n)}\right]^{-1}\ell_n(\hat{\mathbf{z}}^{(n-1)}) \end{aligned} \tag{7}$$

There are several points worth noting regarding the update formula for $\mathbf{z}$ and its learning rate $\boldsymbol{\tau}$. First, note that Equation (7) is only applicable to tasks that belong to $\mathcal{O}^{(n)}$. The columns in $\tilde{\mathbf{X}}_{PS}$ for $t \notin \mathcal{O}^{(n)}$ must be removed before calculating τ. Otherwise, the matrix $\tilde{\mathbf{X}}_{PS}^{(n)T}\mathbf{M}^T\tilde{\mathbf{X}}_{PS}^{(n)}$ is not invertible. For $t \notin \mathcal{O}^{(n)}$, we set $\tau_t = 0$ before calculating $\mathbf{z}$. Second, even when $\tau_t = 0$, the corresponding weight for $\mathbf{v}_t$ may still change due to the first term of Equation (4). This distinguishes our approach from other online algorithms, where a zero learning rate implies the weights will not change in the next round. Finally, our formula for $\boldsymbol{\tau}$ has a similar form as the learning rate for the single-task learning given in [3], $\tau_n = \ell_n/||\mathbf{x}_n||^2$. The main difference is that the τ for multi-task learning must take into account the task relatedness in both ℓ_n and the inverse of $\tilde{\mathbf{X}}_{PS}^{(n)T}\mathbf{M}^T\tilde{\mathbf{X}}_{PS}^{(n)}$.

C. Algorithm

A summary of the ORION framework for ϵ-insensitive loss function is given in Algorithm 1.

Input: μ, λ, β, $\epsilon = 0.001$;
Initialize: $\mathbf{w}_0 = \mathbf{0}_d; \forall t \in \{1, ..., T\}, \mathbf{v}_t = \mathbf{0}_d$;
Compute $\mathbf{R}$ and $\mathbf{Q}$ using the formula in Table I ;
for $n = 2, \cdots, N$ **do**
  Receive $\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, ..., \mathbf{x}_T^{(n)}$;
  for $m = n - T, \cdots, n$ **do**
    Set $m_n = n - m$;
    for $t = 1, 2, \cdots, T$ **do**
      Predict $\hat{y}_t^{(m)} = \left[\mathbf{w}_0^{(m-1)} + \mathbf{v}_t^{(m-1)}\right]^T \mathbf{x}_t^{(m)}$;
    end
    Update $\mathbf{y}_m^{(n)} = \mathbf{y}_m^{(n-1)} \cup \{y^{(n)}\}$;
    Set $\mathcal{O}^n = \{t | t \le m_n; |\mathbf{w}_t^{(m)T}\mathbf{x}_t^{(m)} - y_{m,t}^{(n)}| > \epsilon\}$;
    Compute $\boldsymbol{\tau}$ using Equation (7) and set $\tau_t = 0$ when $t \notin \mathcal{O}^{(n)}$;
    Update $\mathbf{z}^{(m)}$ using Equation (4) ;
  end
end

Algorithm 1: Pseudocode for ORION-ϵ Algorithm

IV. Experimental Evaluation

The proposed framework was applied to the soil moisture ensemble forecasting problem. The soil moisture forecasts were obtained from a seasonal hydrological prediction system for 12 major river basins in North America. 33 ensemble member forecasts were generated by running the model multiple times with different initial conditions. The data correspond to forecasts generated every 5 days for the time period between April, 2011 and September, 2011. The forecast duration is 40 days, which is equivalent to $T = 8$ prediction tasks in our multi-task learning formulation.

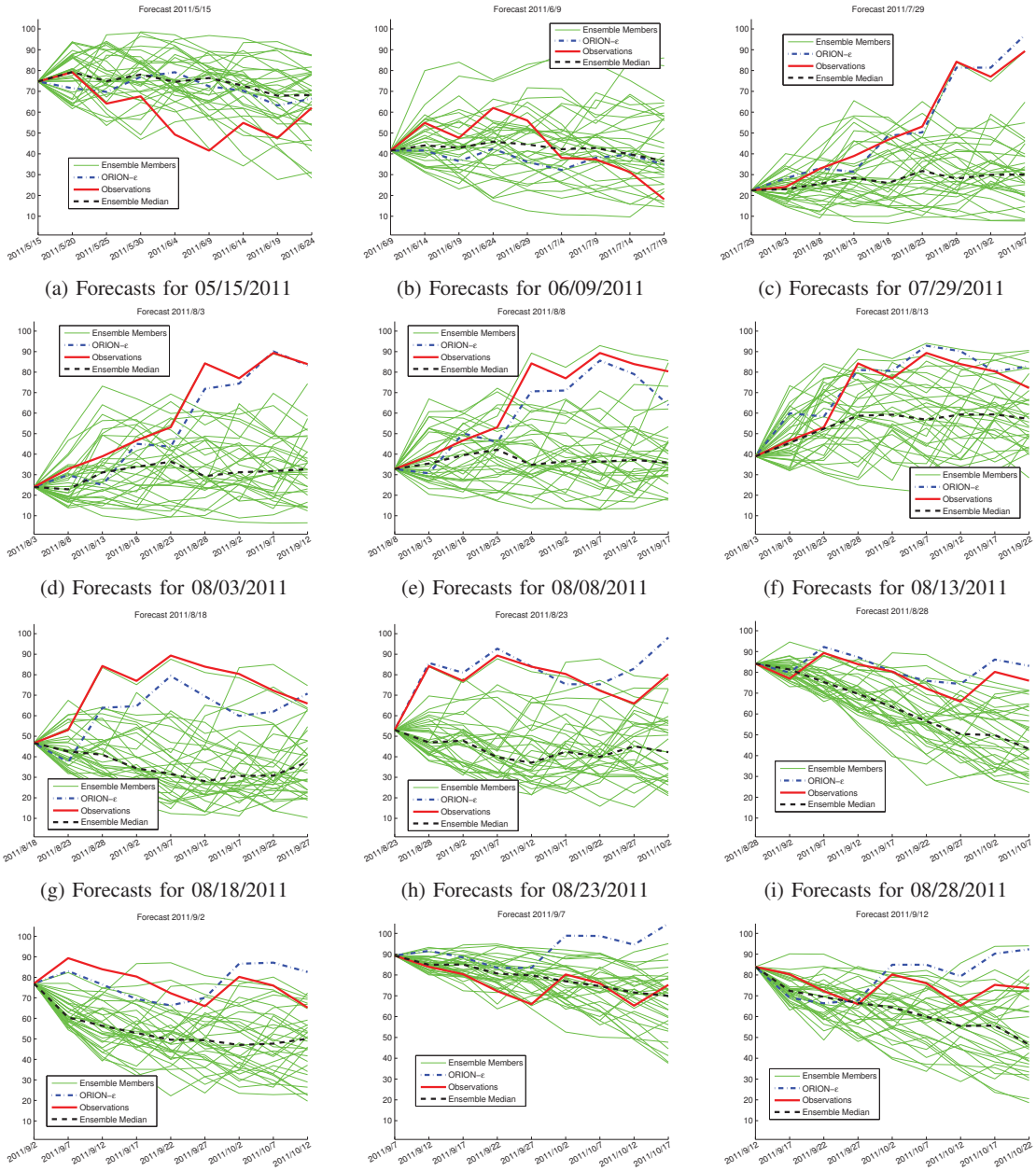
The number of forecast runs in the data set is 33, which is equal to the number of rounds the online learning model is updated. Since the model parameters were initialized to zero, the initial forecasts were poor until the model has been sufficiently trained. We use the first 23 forecast runs as "training data" and report the performance based on the predictions generated for the last 10 forecast runs ("test data"). We evaluated the performance of the different methods in terms of their mean absolute error (MAE) on the test data.

A. Performance Comparison for ORION-ϵ

We compared the performance of ORION-ϵ against the baseline methods, such as *Ensemble Median*(EM), which performs an unbiased aggregation of the ensemble member forecasts, *Passive-Aggresive* (PA) Algorithm [3], which is an single-task online learning method, and some state-of-art methods, including *Tracking Climate Models* [8] (TCM) and *Online Multi-task Learning with a Shared Loss(OMTLSL)* [4].

For a fair comparison, all the baseline methods adopt the same online learning with restart strategy (similar to ORION-ϵ) to deal with the partially observed data.

Table II compares the results of the different methods. As can be seen from the table, ORION-ϵ works better than the various baseline methods on all 12 datasets. In particular, the results showed that ORION-ϵ outperforms OMTLSL, which is a state-of-the-art online multi-task learning method on all the datasets. Unlike OMTLSL, ORION-ϵ enforces the constraint $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$, which helps to improve the performance of the ensemble forecasting task. As will be shown in Table III, the



(a) Forecasts for 05/15/2011 (b) Forecasts for 06/09/2011 (c) Forecasts for 07/29/2011

(d) Forecasts for 08/03/2011 (e) Forecasts for 08/08/2011 (f) Forecasts for 08/13/2011

(g) Forecasts for 08/18/2011 (h) Forecasts for 08/23/2011 (i) Forecasts for 08/28/2011

(j) Forecasts for 09/02/2011 (k) Forecasts for 09/07/2011 (l) Forecasts for 09/12/2011

Fig. 3: Forecasts on Dataset Northeast for ORION-ϵ.

improvement is still observed even when the task relationship is removed (i.e., comparing OMTLSL against ORION-ϵ-NR).

To illustrate the effectiveness of our algorithm, Figure 3 shows an example of the predicted time series for the northeast dataset compared to Ensemble Median (EM). The first two figures are from the training set and the rest ten figures are from the test set. Initially, the time series predicted by ORION-ϵ is similar to EM (see Figure 3a and 3b). As more data becomes available, the predictions by ORION-ϵ becomes closer to observation data compared to EM (Figures 3c to 3j). In Figure 3k, there appears to be a sudden shift that causes the performance of ORION-ϵ to degrade significantly. However, after one update, ORION-ϵ recovers from the mistake and its prediction follows closely the observation data again (Figure 3l).

To summarize the difference between EM and ORION-ϵ, Figure 4 shows the absolute error of both methods during the last 15 rounds of the training data and the 10 rounds in test data for Northeast data. Although the performance of ORION-ϵ is slightly worse than EM at the beginning, after sufficient training, ORION-ϵ appears to perform significantly better than

	ORION-ϵ	EM	PA	TCM	OMTLSL
arkansusred	**2.740**	4.189	3.788	3.659	5.423
calinevada	**3.398**	4.919	4.281	4.265	4.422
colorado	**4.362**	5.934	5.741	5.634	6.068
columbia	**4.411**	6.000	6.439	6.475	6.225
lowermiss	**9.891**	12.023	11.639	10.671	14.975
midatlantic	**13.473**	24.381	25.140	20.961	23.143
missouri	**3.699**	6.029	5.470	6.575	6.913
northcentral	**6.292**	8.789	8.700	9.157	10.838
northeast	**7.422**	22.040	20.490	19.471	24.877
ohio	**14.535**	17.023	15.107	15.021	19.064
southeast	**8.229**	8.951	8.778	9.136	10.966
westgulf	**3.790**	4.697	4.490	5.689	6.150

TABLE II: Comparison of mean absolute error (MAE) for ORION-ϵ against baseline methods on soil moisture data

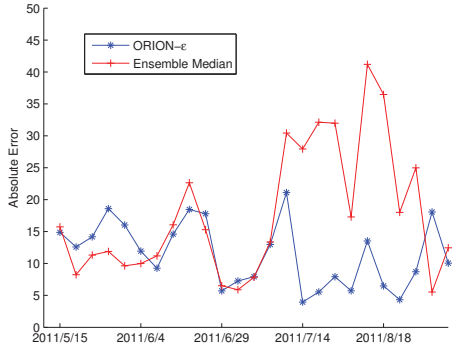


Fig. 4: Mean absolute error for ORION-ϵ and Ensemble Median on the Northeast data.

EM.

B. Variations of ORION-ϵ Framework

The ORION framework leverages two types of information when updating its model parameters. First, it uses the λ and β regularizers to retain information it has acquired from previous rounds. Second, it relies on the $\mathbf{Q}$ matrix to enforce the constraint on relationships among the tasks.

In this subsection, we investigate two variations of the ORION-ϵ framework. We first consider the case when $\beta = 0$. This implies that the weight vectors $\mathbf{v}_t$ are independent of their values in the previous round.[1] We denote the approach as ORION-ϵ-β. Experimental results given in Table III showed that ORION-ϵ outperforms ORION-ϵ-β in 9 out of 12 data sets. Nevertheless the difference in their performance is not that significant except for 3 of 12 the data sets.

The second variation of our framework removes the task relationship by setting $\mathbf{Q} = \mathbf{0}$. This approach is denoted as ORION-ϵ-NR. Based on the results given in Table III, ORION-ϵ outperforms ORION-ϵ-NR in 8 out of 12 datasets, with substantial improvements in at least 4 of them. This verifies the importance of incorporating the task relationship into the ORION-ϵ framework.

[1]Setting $\lambda = 0$ makes $\mathbf{R} + \mathbf{Q}$ becomes a singular matrix. This situation is not considered in this study.

	ORION-ϵ	ORION-ϵ-NR	ORION-ϵ-β
arkansusred	**2.740**	3.937	**2.740**
calinevada	3.398	4.781	**3.390**
colorado	**4.362**	4.599	4.410
columbia	4.411	**4.278**	5.156
lowermiss	**9.891**	10.038	12.047
midatlantic	**13.473**	13.809	13.527
missouri	3.699	**3.370**	5.049
northcentral	6.292	**6.163**	6.475
northeast	**7.422**	7.814	7.427
ohio	14.535	**14.463**	14.987
southeast	**8.229**	9.583	8.232
westgulf	3.790	5.002	**3.780**

TABLE III: Comparison of mean absolute error (MAE) for different variations of ORION-ϵ framework

V. Conclusion

This paper presents a novel online regularized multi-task regression framework for modeling partially observed temporal data. The framework is readily applicable to ensemble forecasting tasks. Our framework assumes that the parameters for each task can be decomposed into a common factor $\mathbf{w}_0$ and a task-specific term $\mathbf{v}_t$. The task relationships are captured using a graph Laplacian matrix. Experimental results confirm the superiority of the proposed framework compared to several baseline methods.

VI. Acknowledgments

The research is partially supported by NOAA Climate Program office through grant NA12OAR4310081 and NASA Terrestrial Hydrology Program through grant NNX13AI44G.

References

[1] M. Araujo and M. New. Ensemble forecasting of species distributions. *Trends in ecology and evolution*, 22(1):42–47, 2007.

[2] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear algorithms for online multitask classification. *J. Mach. Learn. Res.*, 11:2901–2934, Dec. 2010.

[3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, Dec. 2006.

[4] O. Dekel, P. M. Long, and Y. Singer. Online learning of multiple tasks with a shared loss. *J. Mach. Learn. Res.*, 8:2233–2264, Dec. 2007.

[5] T. Gneiting and A. Raftery. Weather forecasting with ensemble methods. *Science*, 310:248–249, 2005.

[6] M. Leutbecher and T. Palmer. Ensemble forecasting. *J of Computational Physics*, 227:3515–3539, 2008.

[7] L. Luo and E. Wood. Use of bayesian merging techniques in a multimodel seasonal hydrologic ensemble prediction system for the eastern united states. *J of Hydrometeorology*, 9:866–884, 2008.

[8] C. Monteleoni, G. A. Schmidt, and S. Saroha. Tracking climate models. In A. N. Srivastava, N. V. Chawla, P. S. Yu, and P. Melby, editors, *CIDU*, pages 1–15. NASA Ames Research Center, 2010.

[9] A. Saha, P. Rai, H. D. III, and S. Venkatasubramanian. Online learning of multiple tasks and their relationships. In G. J. Gordon, D. B. Dunson, and M. Dudłk, editors, *AISTATS*, volume 15 of *JMLR Proceedings*, pages 643–651. JMLR.org, 2011.

[10] A. Wood, E. Maurer, A. Kumar, and D. Lettenmaier. Long-range experimental hydrologic forecasting for the eastern united states. *J of Geophysical Research*, 107(D20):doi:10.1029/2001JD000659, 2002.