Tuning Covariance Localization using Machine Learning

Azam Moosavi[1][0000−0001−9948−5244], Ahmed Attia[2][0000−0001−5940−9247], and Adrian Sandu[3][0000−0002−5380−0103]

[1] Biomedical Engineering department, Case Western Reserve University. azamosavi@vt.edu
[2] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL. attia@mcs.anl.gov
[3] Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University. asandu@cs.vt.edu

Abstract. Ensemble Kalman filter (EnKF) has proven successful in assimilating observations of large-scale dynamical systems, such as the atmosphere, into computer simulations for better predictability. Due to the fact that a limited-size ensemble of model states is used, sampling errors accumulate, and manifest themselves as long-range spurious correlations, leading to filter divergence. This effect is alleviated in practice by applying covariance localization. This work investigates the possibility of using machine learning algorithms to automatically tune the parameters of the covariance localization step of ensemble filters. Numerical experiments carried out with the Lorenz-96 model reveal the potential of the proposed machine learning approaches.

Keywords: Data assimilation · EnKF · Covariance localization · Machine learning.

1 Introduction

Data assimilation (DA) is the set of methodologies that combine multiple sources of information about a physical system, with the goal of producing an accurate description of the state of that system [27]. Statistical DA algorithms apply Bayes' theorem to describe the system state using a probability distribution conditioned by all available sources of information. A typical starting point for most of the algorithms in this approach is the Kalman filter (KF) [26], which assumes that the underlying sources of errors are normally distributed, with known means and covariances. The ensemble Kalman filter (EnKF) [19] follows a Monte-Carlo approach to propagate covariance information, which makes it a practical approach for large-scale settings.

In typical atmospheric applications the model state space has dimension $\sim 10^9 - 10^{12}$, and a huge ensemble is required to accurately approximate the corresponding covariance matrices. However, computational resources limit the number of ensemble members to $30 - 100$, leading to "under-sampling" [24] and

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

its consequences: filter divergence, inbreeding, and long-range spurious correlations [4]. Inbreeding and the filter divergence are alleviated by some form of inflation [5]. We focus here only on long-range spurious correlations which are handled in practice by covariance localization [23].

Covariance localization is implemented by multiplying the regression coefficient in the Kalman gain with a decaying distance-dependent function such as a Gaussian [4] or the Gaspari-Cohn fifth order piecewise polynomial [22]. Different localization techniques have been recently considered for different observation types, different type of state variables, or for an observation and a state variable that are separated in time. However, in general, tuning the localization parameter for big atmospheric problems is a very expensive process. Previous efforts for building adaptive algorithms for covariance localization includes the works [3, 10, 11, 4, 29, 7].

In this study we propose to adapt covariance localization parameters using machine learning algorithms. Two approaches are proposed and discussed. In the *localization-in-time* method the radius of influence is held constant in space, but it changes adaptively from one assimilation cycle to the next. In the *space-time-localization* method, the localization radius is space-dependent and is also adapted for each assimilation time instant. The learning process is conducted off-line based on historical records such as reanalysis data, and the trained model is subsequently used to predict the proper values of localization radii in future assimilation windows.

The paper is organized as follows. Background is given in Section 2. Section 3 presents the new adaptive localization algorithms. Experimental setup, and numerical results are reported in Section 4. Conclusions and future directions are highlighted in Section 5.

2 Background

2.1 Ensemble Kalman filter (EnKF)

EnKF proceeds in a prediction-correction fashion and carries out two main steps in every assimilation cycle: *forecast* and *analysis*. Assume an analysis ensemble $\{\mathbf{x}^{\mathrm{a}}_{k-1}(e) \mid e = 1, \ldots, \mathrm{N_{ens}}\}$ is available at a time instance t_{k-1}. In the forecast step, an ensemble of forecasts $\{\mathbf{x}^{\mathrm{f}}_{k}(e) \mid e = 1, \ldots, \mathrm{N_{ens}}\}$ is generated by running the numerical model forward to the next time instance t_k where observations are available:

$$\mathbf{x}^{\mathrm{f}}_{k}(e) = \mathcal{M}_{t_{k-1} \to t_k}(\mathbf{x}^{\mathrm{a}}_{k-1}(e)) + \eta_k(e),\ e = 1, \ldots, \mathrm{N_{ens}}\,, \tag{1a}$$

where $\mathcal{M}$ is a discretization of the model dynamics. To simulate the fact that the model is an imperfect representation of reality, random model error realizations $\eta_k(e)$ are added. Typical assumption is that the model error is a random variable distributed according to a Gaussian distribution $\mathcal{N}(0, \mathbf{Q}_k)$. In this paper we follow a perfect-model approach for simplicity, i.e., we set $\mathbf{Q}_k = \mathbf{0}\ \forall k$.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

The generated forecast ensemble provides estimates of the ensemble mean $\overline{\mathbf{x}}_k^{\mathrm{f}}$ and the flow-dependent background error covariance matrix $\mathbf{B}_k$ at time instance t_k:

$$\begin{aligned} \mathbf{B}_k &= \frac{1}{\mathrm{N_{ens}}-1}\,\mathbf{X}_k^{'}\,\mathbf{X}_k^{'\,T}\,; \quad \mathbf{X}_k^{'} = \left[\mathbf{x}_k^{\mathrm{f}}(e) - \overline{\mathbf{x}}_k^{\mathrm{f}}\right]_{e=1,\ldots,\mathrm{N_{ens}}}, \\ \overline{\mathbf{x}}_k^{\mathrm{f}} &= \frac{1}{\mathrm{N_{ens}}}\sum_{e=1}^{\mathrm{N_{ens}}}\mathbf{x}_k^{\mathrm{f}}(e)\,. \end{aligned} \tag{1b}$$

In the analysis step, each member of the forecast is analyzed separately using the Kalman filter formulas [16, 19]:

$$\mathbf{x}_k^{\mathrm{a}}(e) = \mathbf{x}_k^{\mathrm{f}}(e) + \mathbf{K}_k\left(\left[\mathbf{y}_k + \zeta_k(e)\right] - \mathcal{H}_k(\mathbf{x}_k^{\mathrm{f}}(e))\right), \tag{1c}$$

$$\mathbf{K}_k = \mathbf{B}_k\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{B}_k\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}, \tag{1d}$$

where $\mathbf{y}_k$ is the observation collected at time t_k. The relation between a model state $\mathbf{x}_k$ and an observation $\mathbf{y}_k$ is characterized by

$$\mathbf{y}_k = \mathcal{H}_k(\mathbf{x}_k) + \zeta_k; \quad \zeta_k \sim \mathcal{N}(0, \mathbf{R}_k)\,, \tag{2}$$

with $\mathcal{H}_k$, and $\mathbf{R}_k$ being the observation operator and the observation error covariance matrix, respectively, at time t_k. Here $\mathbf{H}_k = \mathcal{H}_k'(\overline{\mathbf{x}}_k^{\mathrm{f}})$ is the linearized observation operator, e.g. the Jacobian, at time instance t_k. Many flavors of EnKF have been developed over time. For a detailed discussion on EnKF and variants, see for example [20, 6].

2.2 Covariance localization

The small number of ensemble members may result in a poor estimation of the true correlations between state components, or between state variables and observations. In particular, spurious correlations might develop between variables that are located at large physical distances, when the true correlation between these variables is negligible. As a result, state variables are artificially affected by observations that are physically remote [2, 23]. This generally results in degradation of the quality of the analysis, and eventually leads to filter divergence. Covariance localization seeks to filter out the long range spurious correlations and enhance the estimate of forecast error covariance [23, 25]. Standard covariance localization is typically carried out by applying a Schur (Hadamard) product between a correlation matrix ρ with distance-decreasing entries and the ensemble estimated covariance matrix, resulting in the localized Kalman gain:

$$\mathbf{K}_k = \left(\rho \circ \mathbf{B}_k\right)\mathbf{H}_k^T\left(\mathbf{H}_k\left(\rho \circ \mathbf{B}_k\right)\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}. \tag{3}$$

Localization can be applied to $\mathbf{H}_k\mathbf{B}_k$, and optionally to the $\mathbf{B}_k$ projected into the observations space, that is, $\mathbf{H}_k\mathbf{B}_k\mathbf{H}_k^T$ [34]. Since the correlation matrix is a covariance matrix, the Schur product of the correlation function and the forecast background error covariance matrix is also a covariance matrix. Covariance localization has the virtue of increasing the rank of the flow-dependent

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

background error covariance matrix $\rho \circ \mathbf{B}_k$, and therefore increasing the effective sample size. A popular choice of the correlation function ρ is a Gaussian function defined by

$$\rho(z, c) = e^{-z^2/2\ell^2}, \tag{4}$$

where $z \equiv z(i, j)$ is a distance function between ith and jth grid points respectively. The value of the correlation coefficient $\rho(z, c)$ is at highest of 1 for a distance $z = 0$, and decreases as the distance increases. Depending on the implementation, z can be either the distance between an observation and grid point or the distance between grid points in the physical space. The radius of influence ℓ must be tuned for each application.

2.3 Machine learning

Recent studies show that machine learning (ML) algorithms can be helpful in solving computational science problems, including [32, 8]. There is a plethora of ML algorithms for regression analysis. In this work, we limit ourselves to the *ensemble* approach [18] which has proven successful in enhancing the performance and results of ML algorithms. Specifically, ensemble methods work by combining several ML models into a single predictive model that can in principle overcome the limitations of the individual ML models. These limitations are generally manifested as bias and/or high-variance. Ensemble ML methods aim to decrease the bias (e.g., boosting) and the variance (e.g., bagging), and hence outperform the individual predictive models. Moreover, ML algorithms work by performing an optimization procedure that my be entrapped in a local optimum. An ensemble ML algorithm enables running the local search, carried out by each individual predictive model, from different starting points and thus enhances the predictive power. Common types of ML esnemble methods include the Bootstrap aggregation – bagging for short – [12], and Boosting [15]. In bagging, the training set is used to train an ensemble of ML models, and all trained models are equally important, i.e. the decisions made by all models are given the same weight. Each of the models is trained using a subset randomly drawn from the training dataset. A widely successful algorithm in this family of methods, is Random Forests (RF) [13]. In the boosting approach, on the other hand, the decisions made by the learners are weighted based on the performance of each model. A widely common algorithm in this approach is Gradient Boosting (GB) [14].

Random forests RFs [13] work by constructing an ensemble of decision trees, such that each tree builds a classification or regression model in the form of a tree structure. Instead of using the whole set of features available for the learning algorithm at once, each subtree uses a subset of features. The ensemble of trees is constructed using a variant of the bagging technique, thus yielding a small variance of the learning algorithm [18]. Furthermore, to ensure robustness of the ensemble-based learner, each sub-tree is assigned a subset of features selected randomly in a way that minimizes the correlation between individual learners.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

Random sampling and bootstrapping [30] can be efficiently applied to RFs to generate a parallel, robust, and very fast learner for high-dimensional data and features.

Gradient boosting GB proceeds by incrementally building the prediction model as an ensemble of weak predictors. Specifically, GB algorithm build a sequence of simple regression trees with each constructed over the prediction residual of the preceding trees [21]. This procedure gives a chance to each sub-tree to correct its predecessors, and consequently build an accurate ensemble-based model.

3 Machine Learning Approach for Adaptive Localization

This section develops two machine learning approaches for adaptive covariance localization. Specifically, we let a ML model learn, and consequently predict, the best localization radius to be used in the filtering procedure. Here, we can either allow the localization radius to vary over time only, or both in space and in time. In both cases, RF or GB, or another suitable regression, model is used to construct the learning model that takes the impactful set of features as input, and the localization radius as output.

3.1 Features and decision criteria

Under the Gaussianity assumption, the quality of the DA solution is given by the quality of its first two statistical moments. However, including the ensemble mean and correlations as a set of features can be prohibitive in large-scale applications. One idea is to select only model states with negligible correlations among them, e.g., states that are physically located at distances larger than the radius of influence. Another useful strategy to reduce model features is to select descriptive summaries such as the minimum and the maximum magnitude of state components in the ensemble. Similarly, we suggest including blocks of the correlation matrix for variables located nearby in physical space, i.e., for subsets of variables that are highly correlated.

To construct a proper objective function for the ML algorithm to optimize, we need to quantify the accuracy of the mean estimate, and ensemble-based approximation of the covariance matrix generated by the filtering algorithm. To quantify the accuracy of the ensemble mean we use the root mean-squared error (RMSE), defined as follows:

$$RMSE_k = \frac{1}{\sqrt{\mathrm{N}_{\mathrm{state}}}} \left\| \mathbf{x}_k - \mathbf{x}^{\mathrm{true}}(t_k) \right\|_2 , \tag{5}$$

where $\mathbf{x}^{\mathrm{true}}$ is the true system state, and $\|\cdot\|_2$ is the Euclidean norm. Since the true state is not known in practice, we also consider the deviation of the state from collected measurements as a useful indication of filter performance. The observation-state $RMSE$ is defined as follows:

$$RMSE_k^{\mathbf{x}|\mathbf{y}} = \frac{1}{\sqrt{\mathrm{N}_{\mathrm{obs}}}} \left\| \mathcal{H}\left(\mathbf{x}_k\right) - \mathbf{y}_k \right\|_2 . \tag{6}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

The quality of the analysis state $\mathbf{x} = \mathbf{x}^{\mathrm{a}}$ by either (5) in case of perfect problem settings, or by (6) in case of real applications. *In this work we use the observation-analysis error metric* (6)*, denoted by* $RMSE^{\mathbf{x}^{\mathrm{a}}|\mathbf{y}}$*, as the first decision criterion.*

The quality of the ensemble-based covariance can be inspected by investigating the spread of the ensemble around truth (or the observations), using Talagrand diagram (rank histogram) [1, 17]. A quality analysis ensemble leads to a rank histogram that is close to a uniform distribution. Conversely, U-shaped and Bell-shaped rank histograms correspond to under-dispersion and over-dispersion of the ensemble, respectively. Ensemble based methods, especially with small ensemble sizes, are generally expected to yield U-shaped rank histograms, unless they are well-designed and well-tuned. In this work we use the uniformity of the analysis rank histogram, in observation space, as the second decision criterion. To quantify the level of uniformity of the rank histogram, we follow the approach proposed in [9]. Specifically, the Kullback-Leibler (KL) divergence [28] between a Beta distribution $Beta(\alpha, \beta)$ fitted to rank histogram of an ensemble, and a uniform distribution. This measure calculated using the forecast ensemble is used as a learning feature, while the one calculated using the analysis ensemble is used as a decision criterion. To account for both accuracy and dispersion, we combine the two metrics into a single criterion, as follows:

$$\mathcal{C}_{\mathbf{r}} = w_1 \, RMSE^{\mathbf{x}^{\mathrm{a}}|\mathbf{y}} + w_2 \, D_{KL}\big(Beta(\alpha, \beta) \| Beta(1.0, 1.0)\big) \,, \tag{7}$$

where the weighting parameters realize an appropriate scaling of the two metrics. The weights w_1, w_2 can be predefined, or can be learned from the data them as part of the ML procedure. Here, we define the best set of localization radii at every assimilation cycle to be the minimizer of (7).

Adaptive-in-time localization Here, the value of this radius is fixed in space, and only varied from one assimilation cycle to the next. Specifically, at the current cycle we perform the assimilation using all localization radii from a pool of possible value, and for each case compute the cost function (7). The radius associated with the minimum cost function is selected as winner. The analysis of the current assimilation cycle is then computed using the winner radius. During the training phase, at each assimilation cycle, the ML algorithm learns the best localization radius (i.e., winner) corresponding to the selected features. During the test phase, the learned model uses the current features to estimate the proper value of the localization radius.

Space-time adaptive localization Here, the localization radii vary both in time and in space. In this case, the localization radius is a vector $\mathbf{r}$ containing a scalar localization parameter for each state variable of the system. At each assimilation cycle we collect a sample consisting of the model features as inputs and the winner vector of localization radii as output of the learning model.

Computational considerations During the training phase, the proposed methodology requires trying all possible radii from the pool, and re-do the assimilation

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

with the selected radius. This is computationally demanding, but the model can be trained off-line using historical data. The testing phase the learning model predicts a good value of the localization radius, which is then used in the assimilation; no additional costs are incurred except for the (relatively inexpensive) prediction made by the trained model.

4 Numerical Results

In order to study the performance of the proposed adaptive localization algorithm we employ he Lorenz-96 model [31], described by:

$$\frac{dX_k}{dt} = -X_{k-1}\left(X_{k-2}X_{k-1} - X_{k+1}\right) - X_k + F, \quad k = 1, 2, \cdots, K, \tag{8}$$

with $K = 40$ variables, and a forcing term $F = 8$. A vector of equidistant component values ranging from $[-2, 2]$ was integrated forward in time for 1000 steps, each of size 0.005 [units], and the final state was taken as the reference initial condition for the experiments. The background uncertainty is set to 8% of average magnitude of the reference solution. All state vector components are observed, i.e., $\mathcal{H} = \mathbf{I} \in \mathbb{R}^{K\times K}$ with $\mathbf{I}$ the identity operator. To avoid filter collapse, the analysis ensemble is inflated at the end of each assimilation cycle, with the inflation factor set to $\delta = 1.09$.

Assimilation filter All experiments are implemented in Python using the DATeS framework [9]. The performance of the proposed methodology is compared against the deterministic implementation of EnKF (DEnKF) with parameters empirically tuned as reported in [35]. The EnKF uses 25 ensemble members, with an inflation factor of 1.09 applied to the analysis ensemble.

Machine learning model Several ML regressors to model and predict the localization radii, for ensemble data assimilation algorithms, have been explored and tested. However, for brevity, we use RF and GB as the main learning tools in the numerical experiments discussed below. We use *Scikit-learn*, the machine learning library in Python [33], to construct the ML models used in this work.

Results with adaptive-in-time localization This experiment has 100 assimilation cycles, where the first 80% are dedicated to the training phase and the last 20% to the testing phase. The pool of radii for this experiment covers all possible values for the Lorenz model, i.e., $r \in [1, 40]$. We compare the performance of the adaptive localization algorithms against the best hand-tuned fixed localization radius value of 4 which is obtained by letting the localization radius ℓ take all possible integer values in the interval $[1, 40]$. Figure 1 shows the RMSE results, on a logarithmic scale, of EnKF with a fixed localization radius $r = 4$, and EnKF with adaptive covariance localization with multiple choices of the weights w_1, $w_2 = 1 - w_1$. The RMSE over the training phase is shown in Figure 1(left), and that of the testing phase is shown in Figure 1(right). We separate the results

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

into two panels here, to get a closer look at the relative performance between the different experiments during the testing phase, e.g., in Figure 1(right). The results suggest that increasing the weight of the KL distance measure, that is w_2, enhances the performance of the filter, as long as we don't completely eliminate w_1. For the best choices of the weights, the overall performance of the adaptive localization is slightly better than that of the fixed, hand-tuned radius.

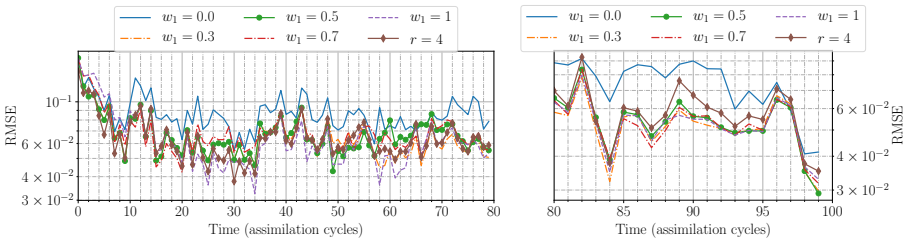


Fig. 1. EnKF results with adaptive-in-time covariance localization, using RF learning model, for different choices of the weighting factors w_1, w_2 of (7), compared to EnKF with fixed localization radius. The training phase consists of 80 assimilation cycles (left panel), followed by the testing phase with 20 assimilation cycles (right panel).

To elaborate more on the results, we pick the weights $w_1 = 0.7$ and $w_2 = 0.3$ of the adaptive localization criterion for this experiment. Figure 2 shows the variability in the tuned localization radius over time for both training and test phase. The adaptive algorithm changes the radius considerably over the simulation.

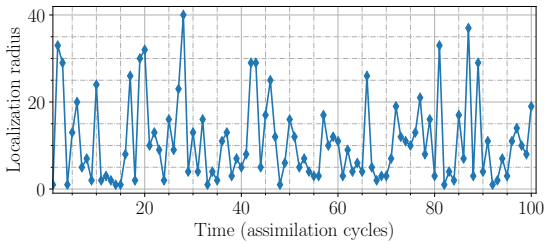


Fig. 2. EnKF results with adaptive-in-time covariance localization, using RF learning model. The evolution of the localization radius in time over all 100 assimilation cycles is shown. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

In decision trees, every node is a condition how to split values in a single feature. The criteria usually is based on Gini impurity, information gain (entropy) or variance. Upon training a tree, it is possible to compute how much each fea-

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

ture contributes to decreasing the weighted impurity. Hence, the RF model helps in recognition and selection of the the most important features affecting the target variable prediction. Figure 3 shows the 35 most important features of the Lorenz model which we included in our experiments. These results, as expected, suggest that the information about the first and second order moments are both essential for the learning algorithm.

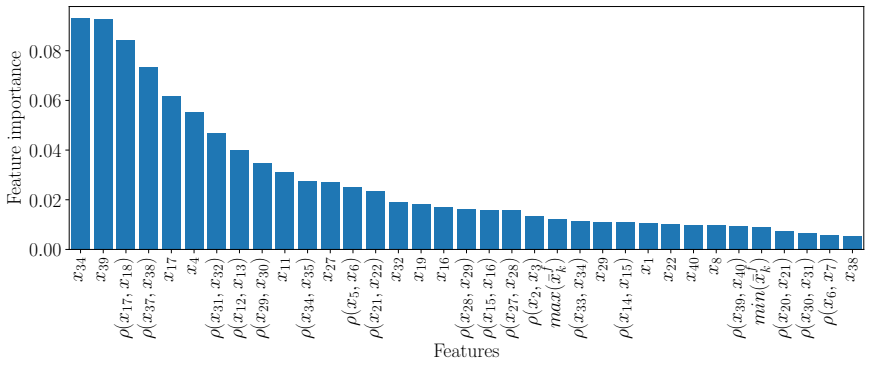


Fig. 3. EnKF results with adaptive-in-time covariance localization, using RF learning model. The plot shows the 35 most important features extracted for the DA experiment with weights $w_1 = 0.7$ and $w_2 = 0.3$.

Results with space-time adaptive localization The pool of radii for this experiment consists of vectors of size 40 where each component of the vector can take any value in the interval $[1, 40]$. With the infinite number of possibilities, trying all possible permutations of of the localization radii is infeasible. One way to limit the number of trials is to test randomly selected vectors of radii in the pool. For this experiment, we set the number of trials to 30 and at each trial we randomly pick a vector of radii from the pool. The number of target variables to estimate at each assimilation cycle in the test phase is 40 and hence we need more samples for the training phase. The number of assimilation cycles for this experiment is 1000, from which 80% dedicated to the training phase, and 20% to the testing phase.

Figure 4 shows the RMSE results of EnKF with space-time adaptive localization for multiple choices of the weighting parameters w_1, $w_2 = 1 - w_1$. Figure 4(left) shows the results over the training phase, while Figure 4(right) shows the RMSE results over the last 50 assimilation cycles of the testing phase. The performance of adaptive localization is compared to EnKF with fixed localization radius $r = 4$. The RMSE results of the adaptive localization algorithm are slightly better than those of EnKF with the empirically tuned fixed radius. Of course in practice, the goal is to completely replace the empirical tuning procedure with an automated scheme. These results suggest that the proposed

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

approach, to automatically adjust the space-time covariance localization parameter can produce favorable results without the need for empirical adjustment.

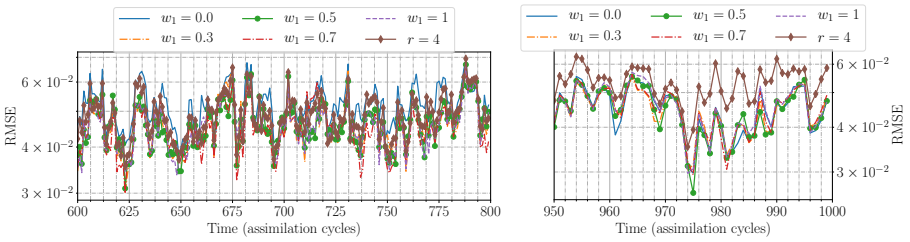
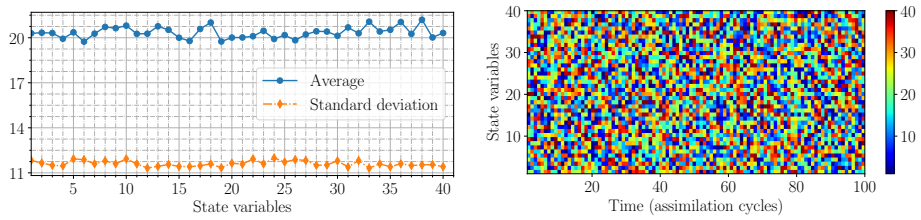


Fig. 4. EnKF results with space-time adaptive covariance localization, using RF learning model, for different choices of the weighting factors w_1, w_2, compared to EnKF results with fixed localization radius. The training phase consists of 800 assimilation cycles (left panel), followed by the testing phase with 200. For clarity, RMSE results of the last 50 assimilation cycles of the testing phase are shown in the right panel.

Figure 5a shows the average and the statistical variability of the localization radii over time, for each state variable of the Lorenz-96 model. The results are found by averaging over all 1000 assimilation cycles, with the weights $w_1 = 0.7$ and $w_2 = 0.3$. From these results, we see that the adaptive values chosen by the algorithm can vary considerably in the temporal domain of the experiment. This variability can be further seen in Figure 5b, which shows the evolution of localization radii in both time and space, over the last 100 cycles of the testing phase.



(a) Temporal statistics (b) Space-time evolution

Fig. 5. EnKF results with space-time adaptive covariance localization, using RF learning model. The weights of the adaptive localization criterion are set to $w_1 = 0.7$ and $w_2 = 0.3$. Panel (a) shows average and standard deviation results of the localization radii for the state variables of the Lorenz-96 model (8). Panel (b) shows the space-time evolution of the localization radii over the last 100 assimilation cycles of the testing phase of the experiment.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

On the choice of the learning model The work in this paper is not aimed to cover or compare all suitable ML algorithms in the context of adaptive covariance localization. In the numerical experiments presented above, we chose the RF as the main learning model, however the method proposed is not limited this choice, and can be easily extended to incorporate other suitable regression model. For example RF could be replaced with GB, however the computational cost of training the regressor, and the performance of the DA algorithm must be both accounted for.

DA performance To compare the performance of the DA filter with localization radii predicted by RF against GB, we study the RMSE obtained by incorporating each of these two learning models. Figure 6 shows the average RMSE over the test phase resulting by replacing RF with GB. Here, the RMSE results for both cases, i.e. time-only and space-time adaptivity, resulting by incorporating RF tend to be slightly lower than that resulting when GB is used.

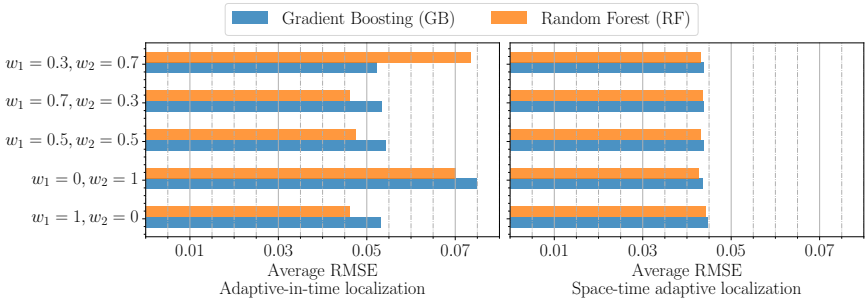


Fig. 6. RMSE results of the adaptive covariance localization approaches are shown for different choices of the weighting factors w_1, w_2. Results are shown for both adaptive-in-time (left), and space-time adaptive localization (right). RMSE is averaged the testing phase of each experiment, obtained by using both RF and GB.

Computational time Table 1 shows the CPU-time spent in fitting the training dataset or training the learning model with both RF and GB. Learning RF model

Table 1. CPU-time of the training time of the two ML algorithms, RF and GB for both time adaptivity and space-time adaptivity approaches.

CPU time (seconds)		Adaptivity type	
		Time	Space-time
ML model	GB	0.0467	16.3485
	RF	0.0308	0.7508

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

is less time consuming than GB, especially in the case of space-time adaptivity. This is mainly because RF, by construction, supports multi-target regression, while GB does not. A simple extension of GB is used for space-time adaptivity, by fitting a regressor to each of the outputs. From both Figure 6, and Table 1, we can empirically conclude that RF yields a combination of better performance and lower computational time, than GB.

5 Concluding Remarks and Future Work

This study investigates using ML models to adaptively tune the covariance localization radii for EnKF family of data assimilation methods. The learning model can be trained off-line using historical records, e.g., reanalysis data. Once it is successfully trained, the regression model is used to estimate the values of localization radii in future assimilation cycles. Numerical results carried out using two standard ML models, suggest that the proposed automatic approach performs at least as good as the traditional EnKF with empirically hand-tuned localization parameters.

One can make some empirical conclusions based on the numerical results herein. Adaptivity leads to a considerable variability of the localization radii in both time and space. Moreover, the values of state variables have a significant bearing on radius predictions. Also, the importance of all state variables is not the same, and some variables in the model have a higher impact on the prediction of localization radii. Finally, the training of the localization algorithms in both time and space with the current methodology is computationally expensive. Future research will focus on making the methodology truly practical for very large models.

In order to extend the use of ML techniques to support data assimilation, an important question that will be addressed in future research concerns the optimal choice of features in large-scale numerical models. Specifically, one has to select sufficient aspects of the model state to carry the information needed to train a ML model. In the same time, the size of the features vector needs to be relatively small, even when the model state is extremely large. Next, the computational expense of the training phase is due to the fact that the analysis needs to be repeated with multiple localization radii. Future work will seek to considerably reduce the computational effort by intelligently narrowing the pool of possible radii to test, and by devising assimilation algorithms that reuse the bulk of the calculations when computing multiple analyses with multiple localization radii.

Acknowledgments

This work was supported in part by the projects AFOSR DDDAS 15RT1037 and AFOSR Computational Mathematics FA9550-17-1-0205. NSF ACI-1709727, and NSF CCF-1613905.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

References

1. Anderson, J.L.: A method for producing and evaluating probabilistic forecasts from ensemble model integrations. Journal of Climate **9**(7), 1518–1530 (1996)
2. Anderson, J.L.: An ensemble adjustment Kalman filter for data assimilation. Monthly weather review **129**(12), 2884–2903 (2001)
3. Anderson, J.L.: An adaptive covariance inflation error correction algorithm for ensemble filters. Tellus A **59**(2), 210–224 (2007)
4. Anderson, J.L.: Localization and sampling error correction in Ensemble Kalman Filter data assimilation. Monthly Weather Review **140**(7), 2359–2371 (2012)
5. Anderson, J.L., Anderson, S.L.: A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. Monthly Weather Review **127**(12), 2741–2758 (1999)
6. Asch, M., Bocquet, M., Nodet, M.: Data assimilation: methods, algorithms, and applications. SIAM (2016)
7. Attia, A., Constantinescu, E.: An optimal experimental design framework for adaptive inflation and covariance localization for ensemble filters. arXiv preprint arXiv:1806.10655 (2018)
8. Attia, A., Moosavi, A., Sandu, A.: Cluster sampling filters for non-gaussian data assimilation. Atmosphere **9**(6) (2018). https://doi.org/10.3390/atmos9060213, http://www.mdpi.com/2073-4433/9/6/213
9. Attia, A., Sandu, A.: DATeS: A highly-extensible data assimilation testing suite v1.0. Geoscientific Model Development (GMD) **12**, 629–2019 (2019). https://doi.org/10.5194/gmd-12-629-2019, https://www.geosci-model-dev.net/12/629/2019/
10. Bishop, C.H., Hodyss, D.: Flow-adaptive moderation of spurious ensemble correlations and its use in ensemble-based data assimilation. Quarterly Journal of the Royal Meteorological Society **133**(629), 2029–2044 (2007)
11. Bishop, C.H., Hodyss, D.: Ensemble covariances adaptively localized with eco-rap. part 2: a strategy for the atmosphere. Tellus A **61**(1), 97–111 (2009)
12. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
13. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
14. Breiman, L., et al.: Arcing classifier (with discussion and a rejoinder by the author). The annals of statistics **26**(3), 801–849 (1998)
15. Bühlmann, P., Hothorn, T.: Boosting algorithms: Regularization, prediction and model fitting. Statistical Science pp. 477–505 (2007)
16. Burgers, G., van Leeuwen, P.J., Evensen, G.: Analysis scheme in the Ensemble Kalman Filter. Monthly Weather Review **126**, 1719–1724 (1998)
17. Candille, G., Talagrand, O.: Evaluation of probabilistic prediction systems for a scalar variable. Quarterly Journal of the Royal Meteorological Society **131**(609), 2131–2150 (2005)
18. Dietterich, T.G.: Ensemble methods in machine learning. In: International workshop on multiple classifier systems. pp. 1–15. Springer (2000)
19. Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forcast error statistics . Journal of Geophysical Research **99**(C5), 10143–10162 (1994)
20. Evensen, G.: Data assimilation: the ensemble Kalman filter. Springer Science & Business Media (2009)
21. Friedman, J.H.: Stochastic gradient boosting. Computational Statistics & Data Analysis **38**(4), 367–378 (2002)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16

22. Gaspari, G., Cohn, S.E.: Construction of correlation functions in two and three dimensions. Quarterly Journal of the Royal Meteorological Society **125**, 723–757 (1999)
23. Hamill, T.M., Whitaker, J.S., Snyder, C.: Distance-dependent filtering of background error covariance estimates in an Ensemble Kalman Filter. Monthly Weather Review **129**(11), 2776–2790 (2001)
24. Houtekamer, P.L., Mitchell, H.L.: Data assimilation using an ensemble Kalman filter technique. Monthly Weather Review **126**(3), 796–811 (1998)
25. Houtekamer, P.L., Mitchell, H.L.: A sequential Ensemble Kalman Filter for atmospheric data assimilation. Monthly Weather Review **129**(1), 123–137 (2001)
26. Kalman, R.E., et al.: A new approach to linear filtering and prediction problems. Journal of basic Engineering **82**(1), 35–45 (1960)
27. Kalnay, E.: Atmospheric modeling, data assimilation and predictability. Cambridge University Press (2002)
28. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
29. Lei, L., Anderson, J.L.: Comparisons of empirical localization techniques for serial Ensemble Kalman Filter in a simple atmospheric general circulation model. Monthly Weather Review **142**(2), 739–754 (2014)
30. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. R news **2**(3), 18–22 (2002)
31. Lorenz, E.N.: Predictability: A problem partly solved. In: Proc. Seminar on predictability. vol. 1 (1996)
32. Moosavi, A., Stefanescu, R., Sandu, A.: Multivariate predictions of local reduced-order-model errors and dimensions. International Journal for Numerical Methods in Engineering pp. n/a–n/a (2017). https://doi.org/10.1002/nme.5624, http://dx.doi.org/10.1002/nme.5624, nme.5624
33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
34. Petrie, R.: Localization in the Ensemble Kalman Filter. MSc Atmosphere, Ocean and Climate University of Reading (2008)
35. Sakov, P., Oke, P.R.: A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters. Tellus A **60**(2), 361–371 (2008)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_16