Data Assimilation Methods

Learning Objectives

Following this lecture, students will be able to:

- Conceptually describe how three- and four-dimensional variational data assimilation methods merge observations with a first-guess analysis.
- Describe how the extended and ensemble Kalman filter methods allow for a flow-dependent background error covariance matrix to be obtained.
- Describe the difference between flow-independent and flow-dependent formulations for obtaining the background error covariance matrix.
- Practically implement the ensemble adjustment Kalman filter for simple one- and multi-dimensional applications.

Introduction

We developed two analogous statistical frameworks for data assimilation in one dimension in our last lecture: least-squares minimization and cost-function minimization. Both approaches involve identifying the optimal combination of background estimates with observations to produce an updated analysis that is as close to the "true" atmospheric state as possible. We also presented the multidimensional analog to the least-squares problem, showing that the same basic tenets underlie the multidimensional problem.

In this lecture, we wish to explore several of the most widely used data assimilation algorithms in greater detail. We begin with three- and four-dimensional variational data assimilation (3DVar and 4DVar), which are cost-function minimization methods that have traditionally used flow-independent background error covariance matrix specifications. We close with the extended and ensemble adjustment Kalman filters, which are least-squares minimization methods that use flow-dependent specifications for the background error covariance matrix. The variational data assimilation material is provided mostly as a continued introduction to basic data assimilation concepts; our focus through the rest of the semester lies instead with ensemble Kalman filter approaches.

The data assimilation algorithms considered in this lecture are not the only such algorithms that exist. For instance, hybrid ensemble-variational schemes use an ensemble of background estimates to provide a flow-dependent estimate of the background error covariance matrix for use in a 3DVar or 4DVar variational data assimilation scheme. Thus, the focus of this lecture is not to provide background regarding every possible algorithm but instead to demonstrate how the fundamental principles outlined in the previous lecture are used in widely used data-assimilation algorithms.

Three-Dimensional Variational Data Assimilation

Variational data assimilation algorithms use iterative methods to minimize a cost function that reflects the departure of the analysis from the background and observation(s).

Recall that the one-dimensional formulation for cost-function minimization is given by:

$$J(T_a) = J(T_o) + J(T_b) = \frac{(T_a - T_o)^2}{\sigma_o^2} + \frac{(T_a - T_b)^2}{\sigma_b^2}$$

Each term of the cost function is equal to the squared error relative to the analysis divided by its respective error variance. If we know the observation, background, and error variances of each, we can find the analysis temperature T_a that minimizes the cost function by taking the first partial derivative of J with respect to T_a, setting the result to zero, and solving for T_a.

The multidimensional formulation for the cost function takes the form:

$$J(\vec{x_a}) = J(\vec{x_b}) + J(\vec{y}) = \frac{(\vec{x_a} - \vec{x_b})(\vec{x_a} - \vec{x_b})^T}{\vec{B}} + \frac{\left(\vec{H}(\vec{x_a}) - \vec{y}\right)\left(\vec{H}(\vec{x_a}) - \vec{y}\right)^T}{\vec{R}}$$

Definitions of the terms listed above may be found in the previous lecture notes. The exponent T refers to the transpose matrix. It is hopefully apparent, however, that this formulation is identical to that for the one-dimensional problem apart from the added dimensionality: each term is equal to the squared error relative to the analysis weighted by its respective error covariance.

As in the one-dimensional problem, the cost function's minimum can be obtained by taking the gradient (across model space) of the cost function and setting it equal to zero. The analytic expression for the gradient of the cost function is given by:

$$\nabla J(\vec{x_a}) = \frac{(\vec{x_a} - \vec{x_b})}{\vec{B}} + \frac{\vec{H}(\vec{x_a})^T\left(\vec{H}(\vec{x_a}) - \vec{y}\right)}{\vec{R}} = 0$$

Note that this gradient is not taken over physical space but rather is taken across the model space defined by **x**. It is not computationally feasible to obtain an analytic solution for the analysis in the multidimensional problem in this way. Instead, an iterative procedure is typically used to minimize the cost function and obtain the updated analysis. One might start by assuming that the analysis equals the background to obtain an initial cost-function estimate. In the next iteration, one might assume that the analysis equals the observations to obtain a second cost-function estimate. The third and subsequent iterations would proceed to find the analysis between the background and observation that minimizes the cost function. Fig. 1 provides an example of cost-function minimization in a two-dimensional model space.

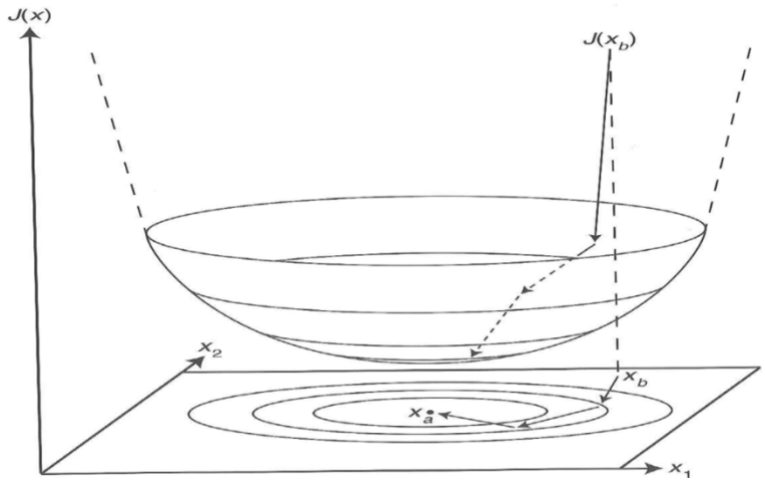
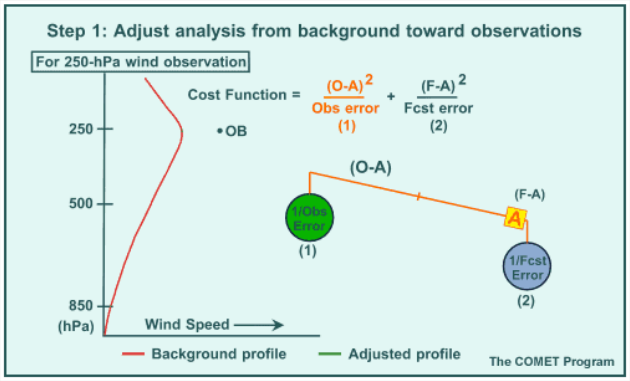


Figure 1. Idealized schematic showing the process of cost function minimization within a two variable (x_1, x_2) model space. In this example, the cost function takes the shape of a parabola. The background $\mathbf{x_b}$, where $J(\mathbf{x_b})$ is minimized, provides the initial guess for the analysis. Two iterations are used in this example to find the cost function minimum that defines the analysis $\mathbf{x_a}$. Figure reproduced from Warner (2011), their Fig. 6.12.

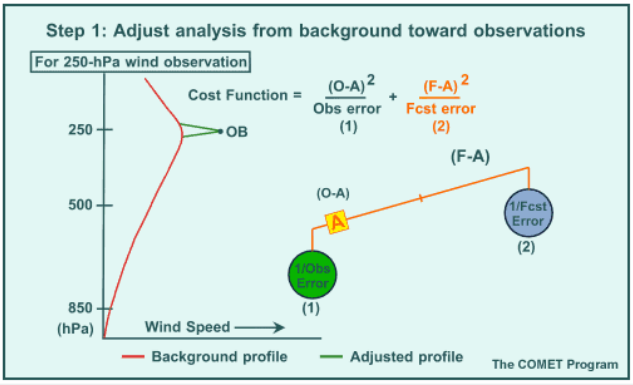
Practical implementations of iterative procedures for cost-function minimization are designed such that the minimum is approached over a relatively small number of iterations. In this regard, the iterative procedure is intend to obtain the greatest amount of minimization without passing a point where the added computational expense of further minimizing the cost function outweighs the benefit to the analysis of doing so. This can be informed by the cost function's Laplacian, which defines the slope of the cost function's gradient (e.g., how quickly you are approaching the cost function's minimum). In practice, on the order of 100 iterations may be required to minimize the cost function.

Let's consider a practical example of three-dimensional variational data assimilation. All figures in this example are reproduced from the UCAR MetEd tutorial, "Understanding Assimilation Systems: How Models Create Their Initial Conditions" (account and login required).

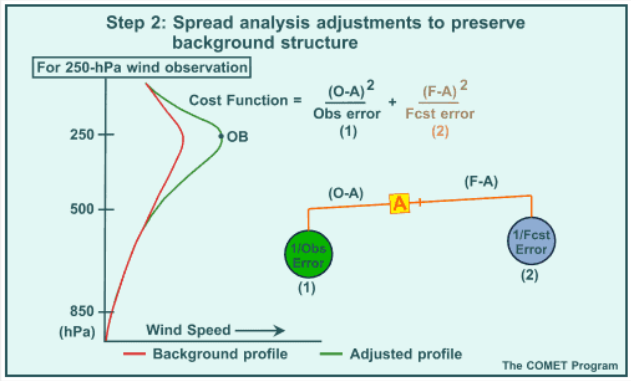
Consider a wind observation at 250 hPa (OB). The first guess for the analysis (A) is provided by the background (red line). This means that the background's cost is zero, such that an initial cost-function estimate equal to the observation's cost (assuming it is imperfect) is obtained.



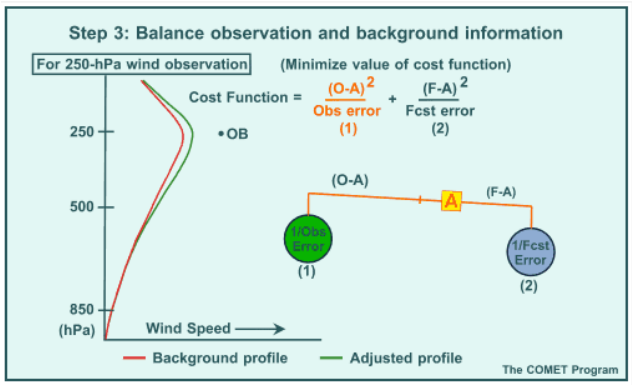
The first iteration occurs as the analysis is adjusted to match the observation. This means that the observation's cost is zero, such that a second cost-function estimate equal to the background's cost is obtained. Implicitly, both of the obtained cost-function estimates are too large, with further iteration needed to determine the minimum cost-function value.



Adjusting the analysis to match the observation has resulted in a profile shape that departs significantly from that of the background. The next iteration can involve updating the analysis of this variable at altitudes above and below that of the observation such that the profile shape more closely resembles that of the background. This enables another cost-function estimate to be obtained with non-zero contributions from both the background and observation. This estimate is smaller than before because of a better overall fit (considering the full profile structure, not just the level of the observation) to the background.



Subsequent iterations can involve updating the analysis to better match the background. The extent to which the analysis is updated to better match the background depends, as we discussed earlier, on the error characteristics of the background and the observation. This process is repeated until the cost function's minimum is found, defining the new wind speed profile.

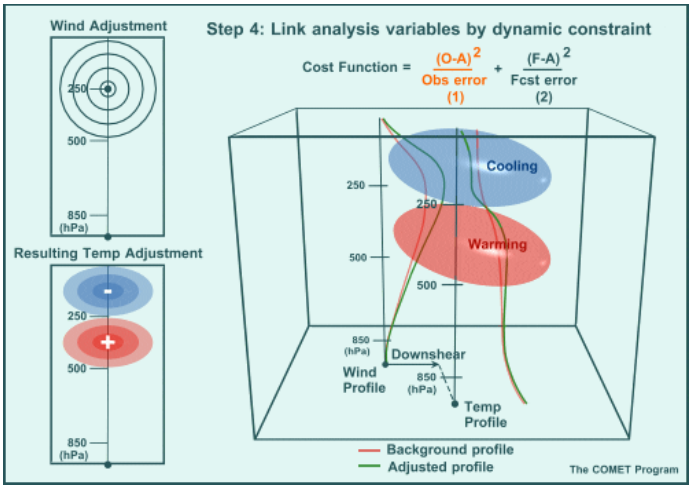


The assimilated wind observation has produced a minor update to the wind speed at the observation location roughly between 500-150 hPa. Intuitively, we know that other fields such as height and temperature are related to the wind (e.g., through geostrophic and thermal wind balance). Thus, physically, assimilating this wind observation should also update these fields.

Assume that the wind profile depicted in the above figures is of the zonal wind. Assimilating the wind observation increased westerly vertical wind shear between ~500-250 hPa and increased easterly vertical wind shear between ~250-150 hPa. From thermal wind balance, which describes the relationship of the horizontal layer-mean temperature gradient to the vertical wind shear, the magnitude of the horizontal layer-mean temperature gradient should increase in both layers. To accomplish this, warming to the south and cooling to the north below 250 hPa, and cooling to the south and warming to the north above 250 hPa, are needed.

This update is encapsulated in the cost-function minimization procedure outlined above. However, though we treated it separately here, it actually occurs concurrently with the wind update. It should be emphasized that this update does *not* necessarily occur from an explicit physical balance statement within the model but rather through the covariance between errors in

each field, representing a **statistical** relationship between the fields. It is hoped that the physical relationship is conveyed through the statistical relationship, but in practice this is approximate at best because of issues like sampling error (estimating correlations with a subset of the true sample).



Let us now consider a 500 hPa temperature observation at the same location as the 250 hPa wind observation. The procedure outlined above is followed to produce an updated analysis for both temperature and its related fields (e.g., the horizontal winds at multiple vertical levels, also via thermal wind balance). As above, this also is encapsulated in the cost-function minimization procedure; ultimately, the analysis is determined iteratively using all observations.

Most practical implementations of 3DVar use flow-independent specifications for **B** in their formulation. This, naturally, will have an impact to how the background influences the analysis relative to that if a flow-dependent specification for **B** were used. As noted above, however, it is possible to use a hybrid ensemble-variational method to obtain a flow-dependent **B** that may be used in the cost-function minimization process.

Four-Dimensional Variational Data Assimilation

Four-dimensional variational data assimilation, or 4DVar, is a generalization of 3DVar to allow for the continuous assimilation of all available observations over some assimilation interval.

Recall that the 3DVar formulation of the cost function is given by:

$$J(\vec{x_a}) = \frac{(\vec{x_a} - \vec{x_b})(\vec{x_a} - \vec{x_b})^T}{\vec{B}} + \frac{\left(\vec{H}(\vec{x_a}) - \vec{y}\right)\left(\vec{H}(\vec{x_a}) - \vec{y}\right)^T}{\vec{R}}$$

The 4DVar formulation is similarly expressed as:

$$J\left(\overrightarrow{x_a}(t_0)\right) = \frac{\left(\overrightarrow{x_a}(t_0) - \overrightarrow{x_b}(t_0)\right)\left(\overrightarrow{x_a}(t_0) - \overrightarrow{x_b}(t_0)\right)^T}{\overrightarrow{B}_{t_0}} + \sum_{i=0}^{n} \frac{\left(\overrightarrow{H}\left(\overrightarrow{x_a}\right)_i - \overrightarrow{y}_i\right)\left(\overrightarrow{H}\left(\overrightarrow{x_a}\right)_i - \overrightarrow{y}_i\right)^T}{\overrightarrow{R}_i}$$

Here, t_0 is the initial/analysis time, t_i is an intermediate time at which one or more observations are assimilated, and t_n is the time at the end of the assimilation window. As compared to 3DVar, there is no change to the background portion of the cost function's formulation. The only change is found with the observation portion of the cost function's formulation, in which observations over a series of times between t_0 ($i = 0$) and t_n ($i = n$) instead of at just t_0 are assimilated. Each set of observations are assimilated at the time at which they are taken rather than at a single analysis time. Consequently, the analysis state vector in the observation portion of the cost function is that valid at the time of the observation. In the case of observations all being valid at t_0, this formulation is identical to 3DVar.

In contrast to 3DVar, which requires no model integration, the 4DVar algorithm requires integrating the model forward from t_0 to t_n as observations are assimilated. The model must be integrated backward from t_n to t_0 to finalize the analysis at t_0, which requires using an adjoint (or backward linear) version of the model. Since an adjoint operator is specific to a given forecast model, it must be updated each time that the forecast model is updated. This can be resource intensive. Further, integrating the model both forward and backward during assimilation results in added computational expense for 4DVar as compared to 3DVar. Historically, this has limited its operational use to the ECMWF model, which is not limited by the same operational timing constraints as NCEP.

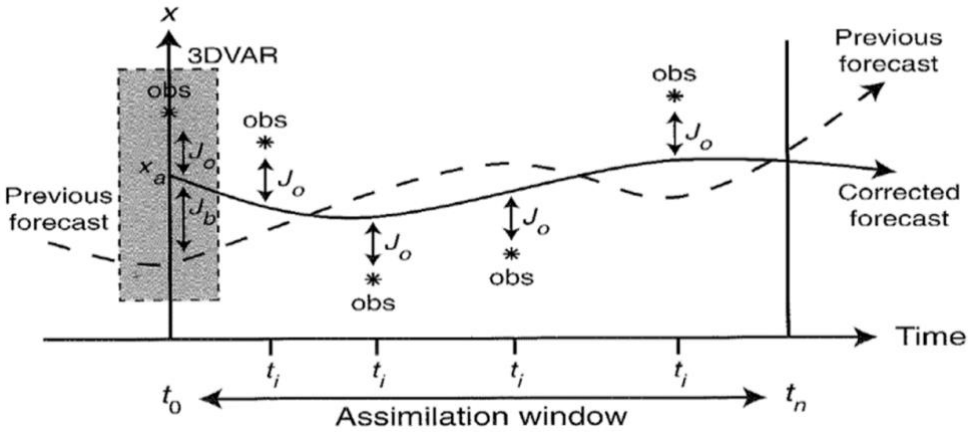


Figure 2. Schematic illustrating the 4DVar process. In this example, a previous model forecast is used to provide the background at t_0, as in 3DVar, but also at all subsequent intermediate times t_i. Observations are assimilated at t_0, all t_i, and t_n, seeking to produce the optimal combination of the background and observations throughout the assimilation window that is a valid solution to the model equations. Here, the model state vectors from the corrected and previous forecasts are identical at t_n, although it is more common for the corrected model state vector to be a better match to the available observations. Figure reproduced from Warner (2011), their Fig. 6.18.

In practice, 4DVar seeks to minimize the cost function throughout the assimilation window. It determines the analysis state vector $\mathbf{x_a}$ at t_0 that produces a model solution, given by the forecast state vector $\mathbf{x_f}$, that minimizes the cost function <u>*at all times*</u> between t_0 and t_n (Fig. 2). In the case of only assimilating observations at t_0, this is identical to 3DVar and reflects cost-function minimization at the analysis time t_0.

Extended Kalman Filter

The extended Kalman filter is a sequential (i.e., individual observations are assimilated one at a time rather than all at once) implementation of least-squares minimization that allows for a flow-dependent, time-varying background error covariance matrix to the calculated to better determine the weighting matrix **K**.

First, the background estimate of the model state $\mathbf{x_b}$ at time t+1 is obtained by integrating the model **M** forward from a previous analysis $\mathbf{x_a}$ at time t, i.e.,

$$\overrightarrow{x_b}(t+1) = \overrightarrow{M}_{t \to t+1}\left(\overrightarrow{x_a}(t)\right)$$

Next, the background error covariance matrix **B** valid at time t+1 is obtained. This is a function of (a) the linear forward propagation of analysis errors from time t to time t+1 and (b) the accumulated model error over the interval between times t and t+1:

$$\overrightarrow{B}(t+1) = \overrightarrow{M}_{t \to t+1}\,\overrightarrow{A}(t)\,\overrightarrow{M}^{T}_{t \to t+1} + \overrightarrow{Q}(t)$$

Here, **A** is the analysis error covariance matrix at time t (or the one from which we just integrated the model), **Q** is the forecast error covariance matrix over the interval between times t and t+1, **M** represents the linear forward model operator, and $\mathbf{M^T}$ represents the adjoint of **M**. Of these, **Q** can be particularly challenging to estimate or obtain, and this is left as a topic for other resources.

Calculating **B** requires that we know **A**. In one dimension, the analysis error variance σ_a^2 is related to the background error variance σ_b^2 through the weighting factor k:

$$\sigma_a^2 = (1-k)\sigma_b^2$$

The corresponding multidimensional formulation is given by:

$$\vec{A}(t) = \left(\vec{I} - \vec{K}(t)\vec{H}(t)\right)\vec{B}(t)$$

Here, **I** is the identity matrix, defined as 1 along the diagonals and 0 elsewhere. For the case where t = 0, however, **B** and **K** are undefined (since there is only an analysis at t = 0) and thus some other method of defining **A** is needed to be able to calculate **B** at the first assimilation time.

The weighting matrix **K** at the future time (t + 1) can be determined once we know **B** at the future time. In the extended Kalman filter, this matrix is referred to as the *Kalman gain matrix*. It is identical to the generic multidimensional form of **K** in the last lecture except for being time-dependent:

$$\vec{K}\,(t+1) = \frac{\vec{B}\,(t+1)H^T(t+1)}{H(t+1)\,\vec{B}\,(t+1)H^T(t+1) + \vec{R}\,(t+1)}$$

Here, the Kalman gain matrix is equal to the background error covariance divided by the total error covariance (background plus observations). H is the linear forward observation operator and H^T is its adjoint.

The updated analysis may be obtained once **K** has been obtained. As before, this is equal to the background plus an optimally weighted innovation, i.e.,

$$\vec{x_a}\,(t+1) = \vec{x_b}\,(t+1) + \vec{K}\,(t+1)\left[\vec{y}\,(t+1) - \vec{H}\,(t+1)\,\vec{x_b}\,(t+1)\right]$$

Here, **y** is the vector of the observations to assimilate, **H** is the forward operator to convert from model to observation space, and the innovation is equal to the terms in brackets. It is implicitly assumed that the innovation is transformed back to model space during the assimilation process. We can then repeat this process for the next analysis time using this updated analysis (for which we can compute **A**).

Ensemble Kalman Filter

The ensemble Kalman filter extends the extended Kalman filter to an ensemble framework. We start with an ensemble of atmospheric state analyses. Prior to the first assimilation time, this is typically obtained by randomly perturbing a single analysis that is often drawn from another model's analysis. The random perturbations are often obtained from randomly sampling a static, climatological form of the background error covariance matrix **B**. The result is an ensemble of initial conditions $\mathbf{x_a}$ with members that vary randomly as a function of climatological model background errors. If the ensemble is to be used with a limited-area model, this perturbation procedure is also applied to a single set of temporally varying lateral boundary conditions to obtain an ensemble of perturbed lateral boundary conditions.

The model is then integrated to the next analysis time using the initial ensemble analyses as initial conditions. The ensemble forecasts valid at this subsequent analysis time provide the first guess $\mathbf{x_b}$. Observations are then assimilated to create an updated analysis. No further perturbations to the analysis are applied, and the cycle then repeats from here.

The ensemble Kalman filter differs from the extended Kalman filter in how the background error covariance matrix is specified. Recall the generic expressions for the background error variance (one dimension) and background error covariance matrix (multidimensional):

$$\sigma_b^2 = E(\varepsilon_b^2) = \overline{(x_b - x_t)^2}$$

$$\vec{B} = \overline{\left(\vec{x_b} - \vec{x_t}\right)\left(\vec{x_b} - \vec{x_t}\right)^T}$$

These represent the mean squared errors in the background estimates relative to the true state. With an ensemble of background estimates, if we assume that these estimates are unbiased, the true state is approximated relative by the ensemble mean of the background estimates:

$$\vec{x_t} = \overline{\vec{x_b}}$$

This enables the background error covariance matrix to be expressed as:

$$\vec{B} = \overline{\left(\vec{x_b} - \overline{\vec{x_b}}\right)\left(\vec{x_b} - \overline{\vec{x_b}}\right)^T}$$

Here, the exponent of *T* represents the matrix transpose operator. No adjoint is needed to obtain **B** in this method. Instead, the ensemble forecasts themselves provide a direct estimate of how analysis errors propagate and how model errors accumulate in time. This is a clear advantage of the ensemble Kalman filter relative to the extended Kalman filter, although the ensemble approach does come with the disadvantage of being more computationally expensive. It should be noted that this **B** is also flow-dependent, a clear advantage relative to traditional 3DVar and 4DVar implementations.

The computation of **K** and $\mathbf{x_a}$ follow from that for the extended Kalman filter.

The analysis error covariance matrix **A** is computed relative to the ensemble mean analysis:

$$\vec{A} = \overline{\left(\vec{x_a} - \overline{\vec{x_a}}\right)\left(\vec{x_a} - \overline{\vec{x_a}}\right)^T}$$

More information about ensemble Kalman filters for atmospheric and geophysical applications is given by Whitaker and Hamill (2002, *Mon. Wea. Rev.*), Hakim and Torn (2008, *Meteor. Monogr.*), and Houtekamer and Zhang (2016, *Mon. Wea. Rev.*).

Practical Considerations and Implementations of Ensemble Kalman Filters

Practically speaking, most ensemble filters used for atmospheric data assimilation, including the ensemble Kalman filter and the ensemble adjustment Kalman filter variant, apply Bayes' theorem to assimilate an observation and thus update an ensemble of background estimates.

To do so, these algorithms assume that the ensemble background estimate is *normally distributed*. The background error variance (one dimension) or background error covariance matrix (multiple dimensions) is calculated from the ensemble estimates themselves, as described above in multiple dimensions for the ensemble Kalman filter. These algorithms also assume that the observation that is to be assimilated can be expressed as a *normal distribution* with mean equal to the observation value and variance equal to the assumed observation error variance.

A normal distribution for any variable x can be expressed as:

$$exp\left(\frac{-(x-\mu_x)^2}{2\sigma_x^2}\right)$$

where μ_x is the mean of x and σ_x^2 is the variance of x.

This distribution can be normalized such that it represents a probability distribution function:

$$\frac{1}{\sigma_x\sqrt{2\pi}}exp\left(\frac{-(x-\mu_x)^2}{2\sigma_x^2}\right)$$

The application of Bayes' theorem to ensemble atmospheric data assimilation has the general form:

$$\textit{Posterior Probability} = \frac{\textit{Prior Probability * Observation Probability}}{\textit{normalization}}$$

Here, the normalization factor is simply the area underneath the curve cut out by the product in the numerator. Thus, the posterior (i.e., analysis) probability is the normalized product of the prior (or background) and observation probability distributions, which are both normal distributions. Note that the product of any two normal distributions is also a normal distribution!

The normal distribution from the product of two normal distributions has mean and variance of:

$$\mu_a = \frac{\mu_b\sigma_o^2 + \mu_o\sigma_b^2}{\sigma_o^2 + \sigma_b^2} = \mu_b\frac{\sigma_o^2}{\sigma_o^2 + \sigma_b^2} + \mu_o\frac{\sigma_b^2}{\sigma_o^2 + \sigma_b^2} = \mu_b(1-k) + \mu_0 k = \mu_b + k(\mu_0 - u_b)$$

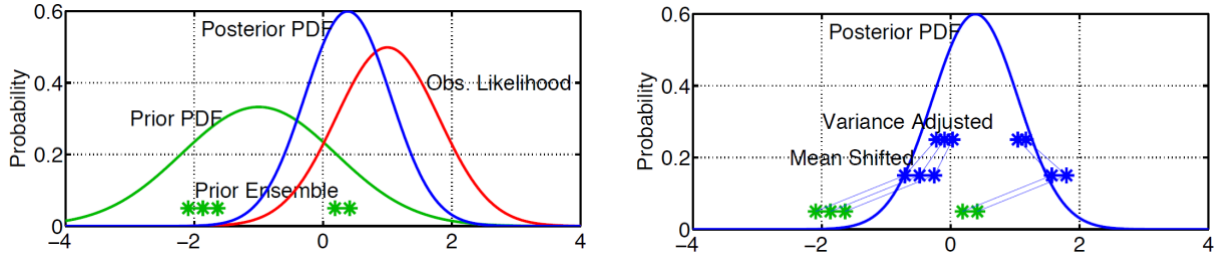
$$\sigma_a^2 = \frac{\sigma_o^2\sigma_b^2}{\sigma_o^2 + \sigma_b^2}$$

Here, subscripts of *b* indicate background, subscripts of *o* indicate observation, and subscripts of *a* indicate analysis or posterior to make each specific to the ensemble data assimilation application.

Note that the analysis mean and variance are both ***identical*** to the least-squares formulation! This is a nice outcome of assuming normal distributions for both the background and observation, even as the validity of this assumption is often questionable for the highly non-linear atmosphere.

How are the individual ensemble member estimates adjusted once the analysis mean has been computed, however? As we will discuss later this semester, the specifics vary between flavors of the ensemble Kalman filter, of which there are several.

The ensemble adjustment Kalman filter is a *deterministic* filter, as there is a clear correspondence between the ensemble member background and analysis distributions. Specifically, the ensemble background estimates are first uniformly (i.e., maintaining their distribution and spread) shifted so that they have a mean equal to the new analysis mean, then uniformly scaled so that they have a variance equal to the new analysis variance. This is depicted visually in the figures below from the excellent UCAR DART Tutorial reference:



Mathematically, this takes the form:

$$\vec{x}_a = \mu_a + (\vec{x}_b - \mu_b)\sqrt{\left(\frac{\sigma_o^2}{\sigma_o^2 + \sigma_b^2}\right)}$$

wherein the ensemble analysis estimates are equal to the analysis mean (calculated analytically as described above) plus the variance-scaled departure of each ensemble background estimate from the background mean. The variance scaling is equivalent to $\sqrt{1-k}$. This variance-scaled departure defines the departure of each analysis estimate from the analysis mean.

In the 1-D ensemble data assimilation case, an observation for a given variable and location is used to update ensemble estimates for that same variable and location. In the more common multivariate case, an observation for a given variable and location is used to update ensemble estimates of many variables and locations! How does this look in a practical sense, however?

Consider a simple multivariate scenario, wherein an observation of one variable at one location is used to update an ensemble of background estimates for a different variable at a different location. First, the observation and its variance are used to update the ensemble background estimates of the observed variable at the observation location. This allows us to obtain the *analysis increments*, or the adjustment needed to each background estimate to obtain its corresponding analysis estimate. We next obtain the slope of the linear regression line between the ensemble background estimates for the observed and unobserved variables. This provides a measure of the correlation between the observed and unobserved variables' backgrounds, from which we can obtain the analysis increments for the unobserved variable. These analysis increments are simply the analysis increments for the observed variable scaled by the slope of the linear regression line between the observed and unobserved variables' background estimates. This is depicted visually in Fig. 3 below from the UCAR DART Tutorial reference.

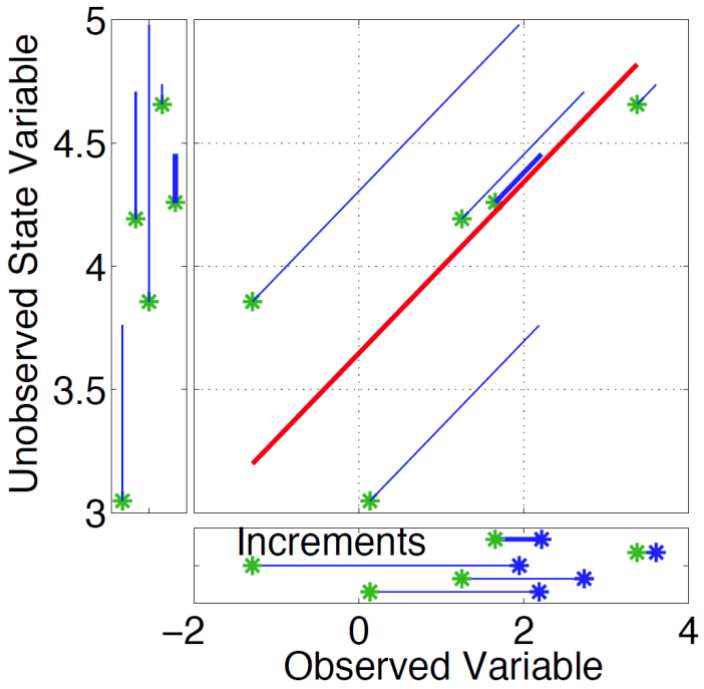


Figure 3. Graphical depiction of an observation of some variable at some location is used to update the background estimates of an unobserved variable at some (other) location. In this example, five ensemble member background estimates an observed variable (*x*-axis) and an unobserved variable (*y*-axis) are depicted by the green stars. The observation is first assimilated (here with the ensemble Kalman filter) to update the observed variable's background estimates, resulting in the blue lines (depicting its analysis increments) and blue stars on the lower inset axis. Next, the linear regression line (depicted in red) between the two backgrounds is obtained. Finally, the observed variable's analysis increments are scaled by the linear regression's slope to obtain the unobserved variable's analysis increments and analysis estimates. This results in the blue lines on the leftmost inset axis. Figure reproduced from the UCAR DART Lab Tutorial, Section 2.

Mathematically, the slope of the linear regression line between two sets of background estimates is given by:

$$\beta = \frac{cov(\vec{x}_{b,o}, \vec{x}_{b,u})}{var(\vec{x}_{b,o})} = \frac{\frac{\sum_{i=1}^{n}\left[\left(x_{b,o}^{i} - \mu_{b,o}\right)\left(x_{b,u}^{i} - \mu_{b,u}\right)\right]}{n-1}}{\sigma_{b,o}^{2}}$$

where…

- $\vec{x}_{b,o}$ is the ensemble of background estimates for the observed variable
- $x_{b,o}^{i}$ is the *i*th ensemble member's background estimate for the observed variable
- $\mu_{b,o}$ is the mean of $\vec{x}_{b,o}$
- $\sigma_{b,o}^{2}$ is the variance of $\vec{x}_{b,o}$
- $\vec{x}_{b,u}$ is the ensemble of background estimates for the unobserved variable
- $x_{b,u}^{i}$ is the *i*th ensemble member's background estimate for the unobserved variable
- $\mu_{b,u}$ is the mean of $\vec{x}_{b,u}$
- *n* is the total number of ensemble members

If the background estimates are uncorrelated, their covariance and thus the slope will be zero.

Once β has been determined, the ensemble analysis estimates for the unobserved variable ($\vec{x}_{a,u}$) are given by:

$$\vec{x}_{a,u} = \vec{x}_{b,u} + \beta\left[\left(\vec{x}_{a,o} - \vec{x}_{b,o}\right)\right]$$

The slope of the linear regression line between the observed and unobserved variable's background estimates is equal to the rise (the unobserved variable's analysis increments) over run (the observed variable's analysis increments). The run equals $\vec{x}_{a,o} - \vec{x}_{b,o}$, the former of which we compute using the equation on pg. 12 and the latter of which we know from the background estimates. We compute the slope using the equation on the top of this page. Given the run and the slope, we can find the rise as the product of the slope and the run. This is added to the unobserved variable's background estimates to obtain its analysis estimates (the ends of the vertical blue lines in Fig. 3's left inset).

A single equation for the unobserved variable's ensemble analysis estimates takes the form:

$$\vec{x}_{a,u} = \vec{x}_{b,u} + \beta\left[\mu_{a,o} + \left(\vec{x}_{b,o} - \mu_{b,o}\right)\sqrt{\left(\frac{\sigma_{o}^{2}}{\sigma_{o}^{2} + \sigma_{b,o}^{2}}\right)} - \vec{x}_{b,o}\right]$$

In this form of the equation, the first two terms inside the brackets are equivalent to $\vec{x}_{a,o}$ (as defined on pg. 12).

This approach assumes that the linear regression line between the two sets of background estimates realistically represents the true relationship between the observed and unobserved variables. This assumption is good when the linear correlation coefficient between the observed and unobserved variables is close to -1 or +1. It is not a good assumption when the linear correlation coefficient is close to 0 *and* the number of ensemble members *n* is small (less than 100, as is often the case for atmospheric and geophysical applications), however. In this case, the difference of the sample and true linear correlation coefficients – and thus also the sample and true linear regression lines – can be quite large, potentially negatively impacting the final analysis' quality. How to best handle such cases is an active area of research within the data assimilation community, and we will discuss a few methods for doing so in a later lecture.