MOS, Perfect Prog, and Reanalysis

Caren Marzban[1,2]* Scott Sandgathe[3], Eugenia Kalnay[4]

[1] Department of Statistics, University of Washington

[2] Center for Analysis and Prediction of Storms

University of Oklahoma

[3] Applied Physics Laboratory, University of Washington

[4] Department of Meteorology, University of Maryland

*Corresponding Author: marzban@caps.ou.edu

Abstract

Statistical postprocessing methods have been successful in correcting many defects inherent in numerical weather prediction model forecasts. Among them, MOS and Perfect Prog have been most common, each with its own strengths and weaknesses. Here, an alternative method (called RAN) is examined that combines the two, while at the same time utilizes the information in reanalysis data. The three methods are examined from a purely formal/mathematical point of view. The results suggest that whereas MOS is expected to outperform Perfect Prog and RAN in terms of mean squared error, bias, and error variance, the RAN approach is expected to yield more certain and bias-free forecasts. It is, therefore, suggested that a real-time RAN-based postprocessor be developed for further testing.

1 Introduction

It is well-known that forecasts of numerical weather prediction (NWP) models have certain defects which can be removed by statistically postprocessing their output (Wilks 1995). Two of the more popular postprocessing approaches are Model Output Statistics (MOS) and Perfect Prog (Glahn and Lowry, 1972; Klein, et al. 1959), both of which are based on the idea of relating model forecasts to observations through linear regression. A great many variations on the original ideas have been proposed: Vislocky and Fritsch (1997), for example, include observations as both predictor and predictand, and Marzban (2003) additionally allows for nonlinear relationships among the various variables. In response to the necessity of deriving new regression equations every time a NWP model changes, Wilson and Vallée (2002) have developed an updateable version of MOS whose training set is a blend of "old" and "new" data. Vislocky and Young (1989) even consider Perfect Prog as a further postprocessor of MOS.

In spite of the development of increasingly more sophisticated versions of MOS and Perfect Prog, they have certain characteristics which have remained distinct, resulting in certain advantages or disadvantages of the respective approaches. For example, although MOS is known to remove the bias from NWP forecasts, its development generally requires not-readily-available large data sets involving both model variables and observations. The development of Perfect Prog, on the other hand, is simpler because it requires only observations as both predictor and predictand.

However, Perfect Prog forecasts are not bias-free. Meanwhile, Brunet, et al. (1988) have shown that Perfect Prog outperforms MOS in short-term forecasts. Wilson and Vallée (2003) found that their updateable MOS outperforms both Perfect Prog and MOS, at least in forecasting 2-m temperature, 10-m wind direction and speed, and probability of precipitation.

MOS is known to maintain reliability but loses sharpness and converges to climatology for longer forecast projections. In ensemble systems, this is disadvantageous because a MOS-based ensemble forecast system fails to adequately forecast rare events. For this reason, some operational centers now rely on Perfect Prog as their method of choice in postprocessing ensemble outputs (Bertrand and Verret 2004). On the other hand, a MOS-type approach to postprocessing of ensemble forecasts has been developed (Stephenson, et al. 2004), which shows improved forecasts in comparison with individual model forecasts as well as the ensemble mean.

In the development of all the variations on MOS and Perfect Prog, a limitation is the amount of data available for developing (or training) the regression model. A possible solution to this data shortage problem was proposed by Kalnay (2003). There, in Appendix C.2, it is suggested to utilize reanalysis data (Kalnay, et al, 1996; Kistler, et al. 2001) to develop a postprocessor with the advantages of both MOS and Perfect Prog, but without the weaknesses due to limited training data. In this article, this method is referred to as RAN (for reanalysis). As described in the next section, RAN also has the added virtue of separating the loss of information between predictor and predictand into its components - one due to the inadequacies of the

NWP model, and the other due to chaos in the atmosphere itself.

The object of this work is to compare MOS, Perfect Prog, and RAN within the confines of a simple statistical model that allows for exact/analytic conclusions. A data-based comparison will be performed elsewhere. The approach assumes a single underlying numerical model, and not an ensemble of models. Finally, the postprocessors are for processing model data at a given station, and not for all points on a grid. The next section addresses the three approaches in more detail, and the mathematical framework is presented in section 3. Two subsections compare the three postprocessors in terms of bias, error variance, and uncertainty of the forecasts. The paper ends with conclusions and discussions in section 4. This last section also discusses questions regarding the effect (on the three postprocessors) of changes in the NWP models, as well as the consequences of the low resolution of the re-analysis data.

2   MOS, Perfect Prog, and the Reanalysis Method

First, it is worthwhile to examine MOS more closely. Figure 1 displays two schematic versions of MOS; one where the predictor and predictand are contemporaneous (MOS1), and one that allows for different times for the two variables (MOS2) - t for the forecast time, and T for the valid time. The left portion of the figures refers to the development stage of the equations, where linear regression equations are developed to relate the relevant quantities. The right portion displays the manner in which the devel-

oped regression equations are utilized for making forecasts. The vertical lines mark the initialization/analysis time. The variable x denotes a predictor, and y represents the predictand. The subscript "o" indicates observation, "m" stands for model, and "f" represents forecast. These definitions are tabulated in Table 1. As an example, one may think of x as the 700mb height, and y as the surface temperature.

Which MOS in Figure 1 is better? To answer that question, it is reasonable to assume that the best predictor of y_o is x_o, and not x_m. This is so because x_m can, at best, be equal to x_o. In other words, if the NWP model is perfect, then $x_m = x_o$. Therefore, if the NWP model is perfect, then MOS1 will outperform MOS2, because at every point in time, the best predictor of y is the x_o at that same time.

However, no NWP model is perfect. The value of x_m agrees with x_o at the initialization/analysis time, but thereafter diverges from x_o.[1] This loss of information/skill is only partially due to the imperfect nature of the model. There is another loss due to the chaotic nature of the atmosphere itself. That loss reflects itself in the divergence between x_o and y as the time difference between them increases. In other words, quite independently of the model, the skill in forecasting y from x_o decreases as a function of the time between them. If the loss of information due to the imperfect nature of model is larger than that due to chaotic growth, then MOS2 will outperform MOS1.

In short, whether MOS1 is better than MOS2 depends on the rate at which the mutual information between x and y is lost in the atmosphere relative to the rate at

---

[1]To be exact, the value of x_m never agrees with x_0, but their difference is minimum at the initialization time.

which the model loses information as it propagates a state into the future. At one extreme, if the model is perfect, then MOS1 is better, because one would run the model forward all the way to the valid time, confident that x_m at that time has not diverged from x_o. At the other extreme, if the model is completely flawed, then y is better predicted from x_m if the model is not run at all, for running the model forward for any duration would only worsen x_m. Given that most models reside somewhere in the range between perfect and completely flawed, it follows that there exists an "optimal" time lag between the predictor and predictand.

The same argument applies to Perfect Prog (See Figure 2). The assumption of Perfect Prog is that the NWP model is perfect in the sense that $x_m = x_o$ for all times. Then one can run the model forward to the valid time (T), and produce a regression forecast (y_f) based on the value of x_m at that time. As such, it would be sufficient to produce the regression equations from contemporaneous values of x_o and y_o at the valid time. But if the model is completely flawed, then the model should not be run forward at all. It follows, again, that a generalization of Perfect Prog where the predictor and predictand are measured at different times may outperform the conventional Prefect Prog. Hereafter, "Perfect Prog" refers to this more general definition of the model.

The arguments presented above for allowing a time lag between predictor and predictand is predicated on the acknowledgment that information is lost in two ways - one due to model deficiency and another due to the atmosphere itself. This raises the possibility of modeling the two components, separately. In other words, as a first step,

one may develop a regression equation that maps x_m to x_o. This regression model would capture only model deficiencies, since it maps the model value of a quantity to the observed value of that same quantity.[2] The second step would then involve mapping x_o to y_o. Since this regression does not involve the model at all, it captures loss of information due to chaos.[3] In practice, one may employ this two-step approach to produce a forecast for y. In a way, this method can be considered as a hybrid of MOS and Perfect Prog, since both x_o and x_m are employed in the forecasting y.

One draw-back of this two-step approach is that it does not allow for x and y to be the same physical quantities, because the second step of the model would then involve mapping a variable to itself. This is a draw-back because one would expect that the model forecast for, say, temperature is a better predictor of surface temperature than any other single predictor. To overcome this problem, one may replace x_o with a "reanalysis" value of x. This will allow same-variable regression models, for at no point in the development stage will a variable be regressed onto itself. Meanwhile, the reanalysis value of x (x_r) encapsulates knowledge of the observed value, and so the information in the observations is not lost. In this article, this reanalysis-based approach is denoted RAN. Note that during the training phase, the NWP model is utilized to provide the best estimate of the reanalysis, followed by a regression model

---

[2]This step is effected by more than model deficiencies; for example, observational and assimiliation errors still come into play. However, for the present analysis, only two sources of "error" are considered - one due to an imperfect model, and another due to chaos in the atmosphere.

[3]Technically, this regression models the dynamic relation between x_o and y_o, and so it captures all contributions unrelated to the NWP model. For example, even instrument error may be captured.

to provide the best estimate of y_o (Kalnay, 2003).

Moreover, note that the optimal time lag for the three postprocessors (MOS, Perfect Prog, and RAN) may be mutually different. This is indicated in Figures 1 and 2 by marking the time at which the forecast is made (t) at different points along the time line. As such, in comparing the three approaches, one must first derive the optimal value of t for each approach, and then compare the models at their respective optimal values of t. This is an important point which will be further addressed in the next section.

In summary, better postprocessors may be obtained if 1) the predictor is measured at some "optimal" time preceding the time at which the predictand is measured, and 2) the two-step approach involving the reanalysis field is employed. Figure 2 illustrates the general situation underlying the three statistical forecasting approaches. The subscript "r" represents the reanalysis value of the predictor (see Table 1). Note that the above arguments for allowing a time lag between predictor and predictands applies to each of the two steps in RAN, introducing an additional time parameter (τ). The aim of this article is to assess the relative expected performance of the three postprocessors displayed in Figure 2.

3   Set Up

The statistical model is developed over some historical data on predictors at time t, and predictands at time T. The Perfect Prog and MOS regression equations, and

forecasts (y_f) can be written as (See Figure 2)

$$\text{Perfect Prog:} \quad y_o(T) \ = \ f[x_o(t)] + \text{error}, \quad y_f(t, T) = f[x_m(t)] \quad (1)$$

$$\text{MOS:} \quad y_o(T) \ = \ f[x_m(t)] + \text{error}, \quad y_f(t, T) = f[x_m(t)], \quad (2)$$

where the function f is usually a linear equation whose parameters are estimated from data. The same function with the estimated parameters is then utilized for making the forecasts y_f for time T. It is assumed that there exists only a single predictor; the analysis can be generalized to the multiple predictor case in a straightforward manner.

The RAN method involves two regression equations, and one forecast equation:

$$\text{RAN:} \quad x_r(\tau) \ = \ f[x_m(t)] + \text{error} \quad (3)$$

$$y_o(T) \ = \ g[x_r(\tau)] + \text{error}$$

$$y_f(t, \tau, T) \ = \ g\left[f[x_m(t)]\right],$$

where the functions f and g are linear functions whose parameters are again estimated from data. Note that for RAN, y_f depends also on the "reanalysis time", τ. In the following, this τ-dependence will not be expressed unless necessary.

Although the choice of the performance measure is arbitrary, here only the mean-squared error, and its decomposition into bias and error variance is considered. Therefore, the model parameters are estimated by minimizing the mean of the squared errors in equations (1) through (3), and the verification measure is computed as

$$E(t, T) = \overline{(y_f(t, T) - y_o(T))^2} \ , \quad (4)$$

10

where the over-bar denotes average. A useful decomposition of E is $E = V + B^2$, where V and B are called the error variance and bias, respectively, i.e.,

$$V(t,T) = \sigma^2[y_o(T) - y_f(t,T)] \ , \quad B(t,T) = \overline{y_f(t,T)} - \overline{y_o(T)} \ , \qquad (5)$$

where $\sigma^2[x(t)]$ denotes the variance of $x(t)$. Good forecasts yield small values of V and B.

The parametric functions in (1)-(3) are written as follows

$$\text{Perf Prog:} \quad y_o(T) \ = \ \alpha_P(t,T) \ x_o(t) + \beta_P(t,T), \qquad (6)$$

$$\text{MOS:} \quad y_o(T) \ = \ \alpha_M(t,T) \ x_m(t) + \beta_M(t,T),$$

$$\text{RAN:} \quad x_r(\tau) \ = \ \alpha_1(t,\tau) \ x_m(t) + \beta_1(t,\tau),$$

$$y_o(T) \ = \ \alpha_2(\tau,T) \ x_r(\tau) + \beta_2(\tau,T).$$

The minimization of E in (4) yields the well-known Ordinary Least Squares (OLS) estimates:

$$\alpha_P(t,T) = \sigma[x_o(t), y_o(T)]/\sigma^2[x_o(t)] \quad , \quad \beta_P(t,T) = \overline{y_o(T)} - \alpha_P(t,T) \ \overline{x_o(t)}, \quad (7)$$

$$\alpha_M(t,T) = \sigma[x_m(t), y_o(T)]/\sigma^2[x_m(t)] \quad , \quad \beta_M(t,T) = \overline{y_o(T)} - \alpha_M(t,T) \ \overline{x_m(t)},$$

$$\alpha_1(t,\tau) = \sigma[x_m(t), x_r(\tau)]/\sigma^2[x_m(t)] \quad , \quad \beta_1(t,\tau) = \overline{x_r(\tau)} - \alpha_1(t,\tau) \ \overline{x_m(t)},$$

$$\alpha_2(\tau,T) = \sigma[x_r(\tau), y_o(T)]/\sigma^2[x_r(\tau)] \quad , \quad \beta_2(\tau,T) = \overline{y_o(T)} - \alpha_2(\tau,T) \ \overline{x_r(\tau)},$$

where $\sigma[x(t), y(T)]$ denotes the covariance between x at time t and y at time T. Note that these OLS estimates are denoted by the same symbol as the arbitrary parameters appearing in the regression equations (6). Henceforth, any reference to the α, β parameters refers to their respective OLS values given in equation (7).

11

Substituting the OLS estimates into the forecasts equations in equations (1) through (3) yields the forecast quantities in each method:

$$\text{Perf Prog:} \quad y_f(t,T) \;=\; \alpha_P(t,T)\,x_m(t) + \beta_P(t,T) \tag{8}$$

$$\text{MOS:} \quad y_f(t,T) \;=\; \alpha_M(t,T)\,x_m(t) + \beta_M(t,T)$$

$$\text{RAN:} \quad y_f(t,\tau,T) \;=\; \alpha_R(t,\tau,T)\,x_m(t) + \beta_R(t,\tau,T) \;,$$

where

$$\alpha_R(t,\tau,T) = \alpha_1(t,\tau)\,\alpha_2(\tau,T)\,, \quad \beta_R(t,\tau,T) = \overline{y_o(T)} - \alpha_R(t,T,\tau)\,\overline{x_m(t)} \;. \tag{9}$$

To bring transparency to the equations, it is convenient to introduce the quantities

$$\rho_P(t,T) \;=\; r[x_o(t),y_o(T)]\,\frac{\sigma[x_m(t)]}{\sigma[x_o(t)]}\,\sigma[y_o(T)], \tag{10}$$

$$\rho_M(t,T) \;=\; r[x_m(t),y_o(T)]\,\sigma[y_o(T)],$$

$$\rho_R(t,\tau,T) \;=\; r[x_m(t),x_r(\tau)]\,r[x_r(\tau),y_o(T)]\,\sigma[y_o(T)],$$

where $r[x,y]$ is the linear correlation coefficient between x and y, defined as

$$r[x,y] = \frac{\sigma[x,y]}{\sigma[x]\,\sigma[y]} \;. \tag{11}$$

Given that $r[x,y]$ assesses the linear correlation between a pair of variables (usually a predictor and a predictand), the quantities ρ_P, ρ_M, and ρ_R also assess the "correlation", but in a generalized sense; if the variables are scaled to have unit variance, then the ρ's are determined entirely from the r's.

Then V and B in eq. (5) for the three methods become

$$V_P(t_P,T) \;=\; \sigma^2[y_o(T)] + \rho_P^2(t_P,T) - 2\,\rho_P(t_P,T)\,\rho_M(t_P,T) \tag{12}$$

12

$$V_M(t_M, T) = \sigma^2[y_o(T)] - \rho_M^2(t_M, T)$$

$$V_R(\tau_R, t_R, T) = \sigma^2[y_o(T)] + \rho_R^2(t_R, \tau_R, T) - 2\,\rho_R(t_R, \tau_R, T)\,\rho_M(t_R, T)$$

$$B_P(t_P, T) = \alpha_P(t_P, T)\left[\overline{x_o(t_P)} - \overline{x_m(t_P)}\right] \tag{13}$$

$$B_M(t_M, T) = 0\ ,$$

$$B_R(\tau_R, t_R, T) = 0\ ,$$

where the various times $(t,\ T,\ \tau)$ have been supplemented with an index - P, M, R - denoting the respective approaches (Perfect Prog, MOS, and RAN). These time parameters represent the respective optimal times for each method, as discussed in Section 2.

It is worth mentioning that $B_R(\tau_R, t_R, T) = 0$ only if one assumes that the average of x_m over the training data is equal to that over the data of the forecasting phase. In other words, it is assumed that $\overline{x_m}$ for the left portion of Figure 2 is the same as that of the right portion. The same is assumed for $\overline{x_r}$. This is a reasonable assumption if the sample employed for training is representative of the population.

3.1   Comparison of Bias and Variance

With a bit of algebra, one can show

$$V_P(t_P) - V_M(t_M) = [\rho_P(t_P) - \rho_M(t_P)]^2 + [\rho_M^2(t_M) - \rho_M^2(t_P)]\ , \tag{14}$$

$$V_R(\tau_R, t_R) - V_M(t_M) = [\rho_R(\tau_R, t_R) - \rho_M(t_R)]^2 + [\rho_M^2(t_M) - \rho_M^2(t_R)]\ ,$$

$$V_R(\tau_R, t_R) - V_P(t_P) = [\rho_R(\tau_R, t_R) - \rho_M(t_R)]^2 - [\rho_P(t_P) - \rho_M(t_P)]^2 + [\rho_M^2(t_P) - \rho_M^2(t_R)]\ ,$$

13

where, for brevity, the dependence of the ρ quantities on the valid time T has been suppressed. This form of the equations is particularly useful, because the last term on the right hand side of all three equations involves only ρ_M evaluated at different times. The remaining terms on the right hand side are all perfect squares, and hence nonnegative.

For example, consider the first equation in (14). The first term on the right hand side is non-negative. The second term is the difference between ρ_M at time t_M and time t_P. However, by the definition of t_M, it is the time at which the mean squared error for MOS is a minimum. But since the bias is zero at that time (eq. 13), then V_M, too, is minimized at time t_M. Finally, based on eq. (12), ρ_M is maximized at time t_M. In short, $\rho_M(t_M) \geq \rho_M(t_P)$, which implies that the right hand side of the first equation in eq. (14) is non-negative. It follows that $V_P(t_P) \geq V_M(t_M)$, i.e., MOS outperforms Perfect Prog in terms of the variance of the forecast errors.

The same argument applies to the second equation in (12), which in turn, implies that $V_R(\tau_R, t_R) \geq V_M(t_M)$, i.e., MOS outperforms RAN in terms of the variance of the forecast errors.

Given that the bias of MOS forecasts is zero, it then follows that MOS outperforms both Perfect Prog and RAN in terms of bias and error variance (and mean squared error).[4]

---

[4]That MOS is better than the alternatives is related to the Gauss-Markov theorem, which states that the best linear unbiased estimator is the OLS estimator. David Stephenson is acknowledged for pointing out this connection.

3.2 Comparison of Uncertainty

It is now possible to compute the uncertainty associated with the forecasts, namely the variance of the forecasts.[5] For brevity, the time dependence of the various quantities is suppressed in this section, because the central quantity in the analysis of these uncertainties is the sample size of the data. Note that the three approaches may be based on samples of different size. Let the sample size of the data employed in MOS be N_M, and that in Perfect Prog N_P. As for RAN, the samples in the two steps of that approach may again have different sizes. In practice, the first step of the approach involves model predictor, x_m, and as such will have a sample size of N_M. The second step of RAN involves reanalysis data, and for that reason, its sample size is denoted N_R.

The uncertainties can be computed as

$$
\begin{aligned}
\text{Perf Prog:} \quad \sigma^2[y_f] &= \frac{\sigma^2[y_o]}{N_P} \, [1 + (\frac{x_m - \overline{x_o}}{\sigma[x_o]})^2]. \\
\text{MOS:} \quad \sigma^2[y_f] &= \frac{\sigma^2[y_o]}{N_M} \, [1 + (\frac{x_m - \overline{x_m}}{\sigma[x_m]})^2], \\
\text{RAN:} \quad \sigma^2[y_f] &= \sigma^2[y_o] \, [\frac{1}{N_R} + (\frac{x_m - \overline{x_m}}{\sigma[x_m]})^2 \, (\frac{r^2[x_m, x_r]}{N_M} + \frac{r^2[x_r, y_o]}{N_R})].
\end{aligned}
\tag{15}
$$

A comparison of MOS and Perfect Prog uncertainties in equation (15) suggests that the latter may in fact have smaller uncertainties when $N_P > N_M$, at least for some values of x_m. However, this slight advantage is offset by the fact that Perfect

---

[5]The variance of the forecasts is to be contrasted with the error variance computed previously. The former is the variance of the *forecasts*, while the latter is the variance of the forecast *errors* (technically, residuals).

Prog forecasts are not bias-free (eq. 13).

As for MOS and RAN, it follows from (15) that, if $N_R > N_M$, then the uncertainty in RAN forecasts is smaller than that of MOS forecasts, for all values of x_m. Therefore, if the second step of RAN (i.e., the step with reanalysis predictors) is based on a larger data set than the first step (i.e., the step based on model predictors), then RAN forecasts have lower uncertainty than MOS forecasts. In short, if the reanalysis data is larger than NWP model data, then RAN forecasts have lower uncertainty than MOS and Perfect Prog forecasts. Therefore, as far as uncertainty is concerned, RAN outperforms the alternatives.

4 Conclusion and Discussion

MOS is expected to outperform Perfect Prog and RAN in terms of bias, error variance, and mean squared error. However, the uncertainty of MOS forecasts may be hindered by the limited size of available model data. Perfect Prog is less restrictive because its development is not limited by the availability of model data; however, its forecasts are biased and have higher error variance than MOS forecasts. On the other hand, RAN forecasts are bias-free, and have lower uncertainty than MOS and Perfect Prog. In short, even though MOS has higher performance (in terms of bias and error variance) than Perfect Prog and RAN, the latter has lower uncertainty associated with its forecasts if its sample size is larger than MOS's sample size.

These findings are based on several assumptions, including the following: First,

16

the underlying relations are assumed to be linear. It is this linearity that allows for analytic conclusions like those presented here. Nonlinear models do not lend themselves to such analysis, and are better assessed with real data. That task will be addressed in the future. Second, ordinarily, in regression modeling, the predictors are assumed to be error-free, and only the target is assumed to have errors. In RAN, however, there are two regression models - one from a model variable to a reanalysis variable, and a second from the reanalysis variable to an observation. Here, the reanalysis variable is assumed to be error-free during the first step, but with error during the second step. The more realistic analysis wherein both the predictors and predictands are assumed to have error is more complicated (see section 3.4 of Draper and Smith, 1998), and will be considered elsewhere.

Given all of these assumptions, it is important to emphasize that the results reported here regarding the relative performance of the three methods are only "expected" and not gauranteed. However, as shown here, there exists sufficient theoretical justification to develop and test a real-time, RAN-based postprocessor.

One question is whether changes in the numerical model call for retraining RAN. In other words, in an operational setting, is one expected to retrain RAN each time a numerical model undergoes some revision? The important point to note is that the difficulty in training MOS is in the requirement for large data sets involving observations. In RAN, however, observations play a role only in the second step, where they are mapped to the reanalysis data, not model data. When a new version of a model is released, only the first step of RAN must be retrained, but that is a relatively simple

task (compared to MOS), given that it involves mapping the "new" model data only to the reanalysis data, i.e., the reanalysis based on the "old" model compared to the "new" model, where a large amount of data exists on identical grids. Consequently, only a relatively short model rerun is required to retrain. This is especially important when dealing with rare phenomena, where MOS would require exceedingly large training sets mapping the "new" model to the observations. Moreover, it is possible to implement an updateable approach (Wilson and Valle 2002) for RAN. As such, model changes are expected to have minimal effect on RAN in an operational setting.

Another question is whether or not there is an advantage in using the NWP model used for the reanalysis also for producing the forecasts. One might expect that the correlation between the reanalysis (i.e., modeled "truth") and an NWP prediction based on the same underlying dynamics, physics and numerical approximations should be higher than the correlation based on an entirely different NWP system. If true, this would constitute a severe restriction to RAN. However, it is also possible that the benefits of a sufficiently long training period (as is possible in RAN) may outweigh any drawbacks of using different models for reanalysis and forecasts. This is especially true if differences between models are "systematic." This question will be further addressed in the future.

Finally, how does the relatively low resolution of the reanalysis data affect the practical utility of RAN? A RAN postprocessor will reproduce a response to larger-scale features but not very high-resolution features (small eddies, thunderstorms, etc.). But that is precisely what one would desire in a postprocessor - the ability

to calibrate to large-scale features that are highly predictable, rather than to "contaminate" the calibration with less predictable, small-scale features. (It is useful to note here that the NWS operational MOS was developed using a very low-resolution version NWP model, following this reasoning.) The first step of RAN calibrates the model to the reanalysis, correcting the model representation of the large scale. The second step of RAN calibrates the reanalysis to the observations, ensuring that site-specific properties of the observations are appropriately represented.

Acknowledgement

References

Bertrand, D., and R. Verret, 2004: Toward a new Canadian medium-range Perfect-Prog temperature forecast system. 17th Conference on Probability and Statistics in Atmospheric Sciences. The 84th American Meteorological Society Annual Meeting, Seattle, WA. Available at

http://ams.confex.com/ams/84Annual/techprogram/paper_67887.htm .

Brunet, N., Verret, R. and Yacowar, N., 1988: An objective comparison of model output statistics and perfect prog systems in producing numerical weather element forecasts. *Wea. Forecasting*, 3, 273-283.

Draper, N.R., and H. Smith, 1998: *Applied Regression Analysis*, John Wiley & Sons, NY, 706 pp.

Glahn, H. R., and D. A. Lowry, 1972: The use of model output statistics (MOS) in objective weather forecasting. *J. Appl. Meteor.*, **11**, 1203-1211.

Kalnay, E., M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph, 1996. The NCEP/NCAR 40-year reanalysis project. *Bull. Amer. Meteor. Soc.*, **77**, 437471.

Kalnay, E., 2003: *Atmospheric Modeling, Data Assimilation and Predictability*, Cambridge University Press, 341 pp.

Kistler, R., E. Kalnay, W. Collins, S. Saha, G. White, J. Woollen, M. Chelliah, W. Ebisuzaki, M. Kanamitsu, V. Kousky, H. van den Dool, R. Jenne, and M. Fiorino, , 2001. The NCEP-NCAR 50 year reanalysis: monthly mean CD-ROM and documentation. *Bull. Amer. Meteor. Soc.*, **82**, 247-267.

Klein, W.H., B.M. Lewis and I. Enger, 1959: Objective prediction of five-day mean temperature during winter. *J. Meteor.*, **16**, 672-682.

Marzban, C., 2003: A neural network for post-processing model output: ARPS. *Mon. Wea. Rev.*, **131** (6), 1103-1111.

Stephenson, D. B., C.A.S. Coelho, F. J. Doblas-Reyes, and M. Balmaseda, 2004: Forecast Assimilation: A unified framework for the combination of multi-model weather and climate predictions. *Tellus*, Special issue, in press. Available at http://www.met.rdg.ac.uk/cag/publications/2005/tellus.pdf .

Vislocky R.L. and J.M. Fritsch, 1997: An automated, observations-based system for short-term prediction of ceiling and visibility, *Wea. Forecasting*, **12**, 31-43.

Vislocky, R. L., G. S. Young, 1989: The use of Perfect Prog Forecasts to Improve Model Output Statistics Forecasts of Precipitation Probability, *Weather and Forecasting*, 4, 202209.

Wilks, D. S., 1995: *Statistical Methods in the Atmospheric Sciences.* Academic Press, NY. 467 pp.

Wilson, L.J. and M. Vallée, 2002; The Canadian updateable model output statistics (UMOS) system: Design and development tests. *Wea. Forecasting*, **17**, 206-222.

Wilson, L.J. and M. Vallée, 2003: The Canadian updateable model output statistics (UMOS) system: Validation against perfect prog. *Wea. Forecasting*, **18**, 288-302.
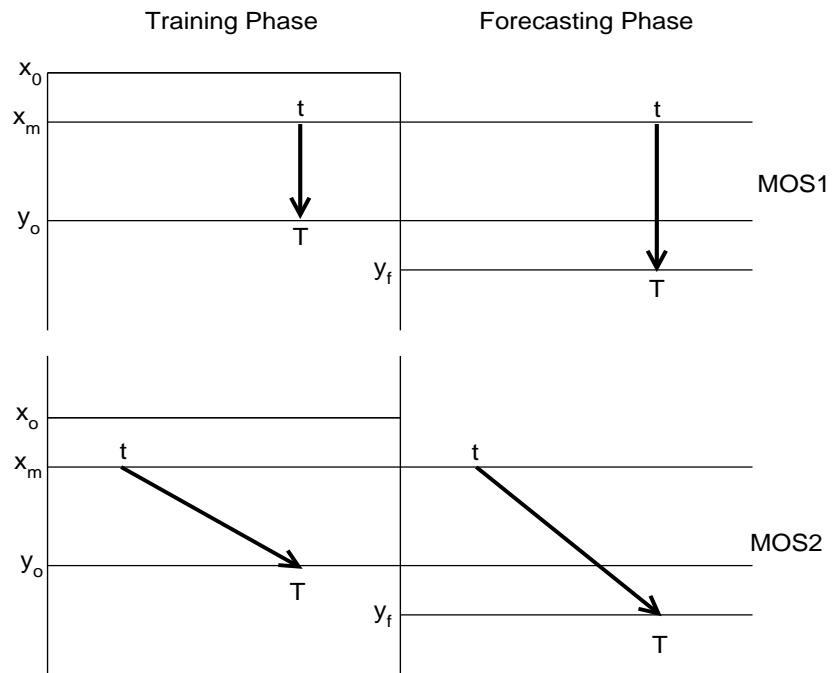
Figure 1. A graphical display of two versions of MOS. MOS1 represents the "conventional" MOS where the predictor and predictand are contemporaneous, while MOS2 allows for different times for the two variables. The various components of these figures are defined in Section 2.
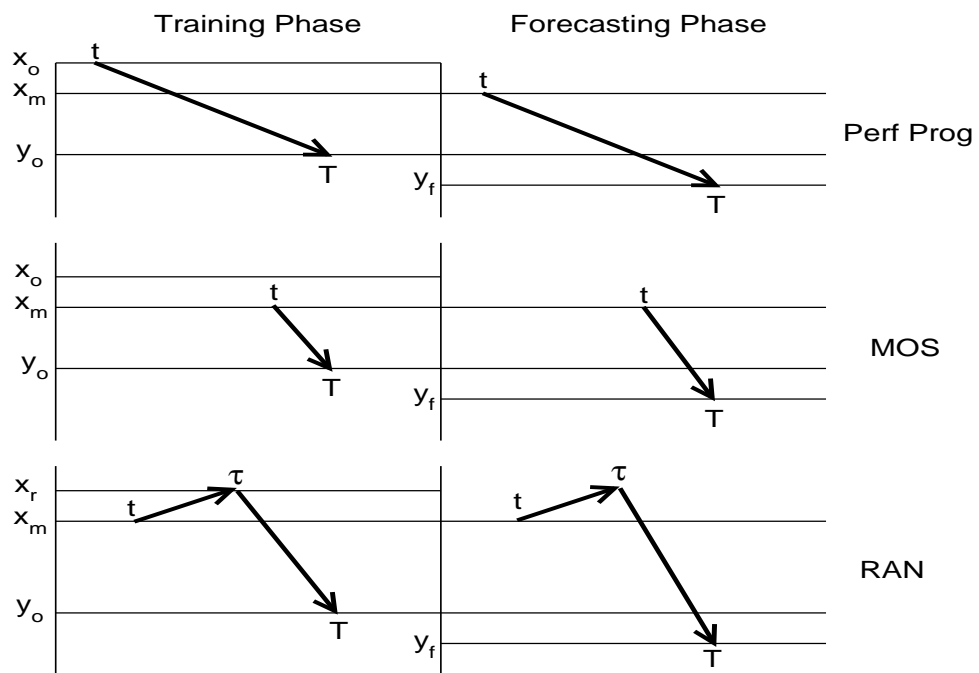
Figure 2. A graphical display of Perfect Prog, MOS, and the RAN methods.

Table 1. The definition and examples of the variables. Note the time-dependence of the various quantities.

Variable	Definition	Example
$x_o(t)$	Observed predictor	Observed 700mb height
$x_m(t)$	Model predictor	Model forecast of 700 mb height
$x_r(\tau)$	Reanalysis predictor	Reanalysis value of 700 mb height
$y_o(T)$	Observed predictand	Observed 2m temperature
$y_f(t,T)$	Forecast predictand	Forecast 2m temperature