SUMO - Supermodeling by combining imperfect models

Workpackage 2: Year 3

Wim Wiegerinck, Willem Burgers, Greg Duane, Frank Selten, Christiaan Schoenaker

May 16, 2014
# Contents

**Comments of the authors** ........................................ 3  
**Summary** .................................................................... 6  

1 **Comparing error reduction in interactive and non-interactive ensemble approaches** ................................................................. 7  
\hspace{1em} 1.1 Setting up the analysis of the error reduction ................................................................. 7  
\hspace{1em} 1.2 An abstract counter example .................................................................................. 9  
\hspace{1em} 1.3 Smooth model spaces ......................................................................................... 9  
\hspace{1em} 1.4 Imperfect models with optimized parametrization .................................................. 12  
\hspace{1em} 1.5 Conclusion ........................................................................................................ 15  

2 **Attractor learning** ........................................................ 16  
\hspace{1em} 2.1 Bayesian learning ............................................................................................ 16  
\hspace{1em} 2.2 Vector field learning in linear models ................................................................. 17  
\hspace{1em} 2.3 Attractor learning ............................................................................................ 18  
\hspace{1em} 2.4 Numerical experiments ..................................................................................... 19  
\hspace{1em} 2.5 Conclusion ........................................................................................................ 21
Comments of the authors

The objective of this WP is to research and develop efficient, robust and scalable learning strategies to optimize connection coefficients for dynamical systems of low and intermediate size complexity (up to 1000 variables). The resulting learning strategies are to be used to guide the development of methods in WP3 and WP4.

This WP is one of exploring approaches, developing methodologies and doing many simulations and analyses in a variety of ways on smaller-scale problems. The scalability to the higher dimensional climate models of WP4 and WP5 is an important issue.

The planned work was described in Task 2.2 of the description of work:

**Potential of learning of supermodels**

Task 2.3 is concerned with the second objective of this WP and addresses the question: “how good are the supermodels that we obtain by learning?”, and the related but very different question: “how can we predict the quality of the supermodels in advance?”.

In the scope of this WP, the supermodels are straightforward to validate, since in the simulated world of this WP, we have access to the ground truth model. The first question, concerns the application range of supermodeling learning, in particular where the approach might be beneficial (i.e. better than any single model and/or average of all models, and better than a manually set initialization of the connection coefficients). More specific questions are: Is there a minimum amount of data that is needed to learn the supermodel (beyond its manually set initialization)? How many observed variables and how many data points for each variable are needed? Are there limitations on the assumed measurement errors in the observations? To what extent will imperfect models that are very far from the truth obstruct the feasibility of supermodel learning? To what extent does the existence of slow and fast time scales hinder learning? In particular, the learnability of the supermodel will be studied in relation to the number of independent connection coefficients as it is expected that in the range in which the number of coefficients is large compared to the number of data, the supermodel might have the risk to over fit the data. The performances will be compared to the standard competing methods (average model, etc.). The answers to these questions will be primarily obtained by gathering statistics over many simulations and instances.

The second more practical question, is whether we can predict for a supermodel that is trained on a given set of data how good it will perform. Therefore we will research methods and measures that aim to predict the supermodels future performance based on current data. A starting point for these methods and measures will be related to cross validation techniques in machine learning [1]. These and related methods might be applied directly with the defined cost function, or using other metrics defined in WP1.

Relevant parts of the work has been addressed in the previous year. In the previous year, we noticed an important issue in learning and evaluation of in the supermodeling,
and that is the issue of vectorfield learning (short term prediction, weather prediction) versus attractor learning (long term prediction of the statistics, climate projection). In particular we noticed that super-models that are optimized for weather prediction, i.e. for short term prediction, could perform badly on the long term statistics. So, learning of supermodels that should perform on long term statistics should be based, or at least validated on a similar long term statistics measure \cite{2, 3}.

Furthermore, reviewers made the remarks that we should better understand why supermodeling works. In particular, among others, why the dynamic fusion of models is better than a posteriori ensemble methods. Another point is the issue of uncertainty in supermodels. Earlier results suggested so \cite{4, 5, 6}. But the ensembles in these studies were taken nicely distributed around an optimal value. No systematic study with more general perturbed models has been performed.

In the last year we focussed on these issues, while further exploring and advancing on the above-mentioned earlier results.

- In the first place, we studied the performance of supermodeling prior to learning. Prior to learning, one could consider a (weighted) supermodel as as a dynamical model-averaging method. In this method, the model outcomes are averaged at each step during integration and used as state to compute the next step. The question is then, if there is an advantage (in terms of estimating attractor statistics) in a dynamical averaging approach compared to the straightforward a posteriori averaging in a conventional non-interactive ensemble.

- In the second place, we further explored methods for simultaneously learning short and long term statistics in low dimensional dynamical systems, and in particular started an attempt of Bayesian learning, in other words, learning strategies that explicit address model uncertainty.
Summary

Overview of the work done in the third year in work-package 2 of the SUMO project.

Work not described in this document

We have performed ensemble studies with three imperfect T21 models compared and one assumed ground truth T42 model. One of the imperfect models has the same parameters as the T42 model. The other two models are symmetrically perturbed around this “optimal” imperfect model. In addition, the imperfect models have forcings that are tuned to the observed climatology of the ground truth. In this study we compared a uniformly fused supermodel (weighted as well as connected) to the individual models and the posterior ensemble approach. We found that the uniform supermodel improved on the predicted climate variance [6].

Work described in more detail later in this document

The issue that we address in this document is two-fold:

- The question is considered whether there is an advantage in a dynamical averaging approach compared to the more straightforward a posteriori averaging in a conventional uncoupled ensemble of models.

  The answer to this question sheds light to the question how important learning is in supermodeling. If uniformly fusing dynamical models on its own were often already helpful, then the role of learning from data and efficient learning algorithms is less relevant than when this is not helpful.

  We performed several experiments, some in the perfect model class setting (where imperfect models and ground truth are in the same model class) and some in the imperfect model class setting (where ground truth model has a higher complexity than the imperfect models). We validated performance on long term statistics, either in terms of some metrics, or by visual inspection of the attractor.

  We did some analysis of attractor statistics. We considered settings were the model dynamics is linear in the model parameters. The attractor statistics is also assumed to be a function of these model parameters. With linear attractor statistics, interactive ensembles results in the same statistics as the mean of the model statistics in a non-interactive ensemble. Since non-interactive ensembles improve the statistics compared to individual models, this means that with supermodeling, prior to learning, some error reduction can also already be expected (compared to the individual models). When the attractor statistics are nonlinear, results are strongly dependent on the situation at hand and an interactive ensemble approach could lead to undesired effects due to nonlinearities in the integration, even worse than individual models.
The conclusion is that in general one should take care and that validation and fine tuning e.g. by learning weights, in supermodels according to appropriate metrics may be necessary to obtain appreciable results [7].

- The other item is learnability of dynamical systems with respect to both short term (vector field) and long term (attractor) behaviour. In particular we are interested in learning in the imperfect model class setting, in which the ground truth has a higher complexity than the models, e.g. due to unresolved scales. We start with a Bayesian point of view and we define a joint loglikelihood that consists of two terms, one is the vector field error and the other is the attractor error, for which we take the $L_1$ distance between the stationary distributions of the model and the assumed ground truth. In the context of linear models (like weighted supermodels), and assuming a Gaussian error model in the vector fields, vector field learning leads to a tractable Gaussian solution. These solution can then be used as a prior for attractor learning with the attractor likelihood. This is done by elliptical slice sampling [8], a sampling method for systems with a Gaussian prior and a non Gaussian likelihood. Simulations with the partially observed driven Lorenz 63 system shows that solutions can be found with less than 100 samples.
1 Comparing error reduction in interactive and non-interactive ensemble approaches

Ensemble methods in general have a two-fold advantage over single run methods. The first is that they give information about uncertainty about model outcomes in the presence of model error. The second is that by averaging model outcomes better predictions may be obtained due to cancellation of independent errors in the model outcomes.

In this project, methods for model fusion by dynamically combining model components in an interactive ensemble have been proposed. In these proposals, fusion parameters have to be learned from data. However, in our earlier studies with low and intermediate dimensional dynamical systems, promising results have been reported even when fusion parameters are taken to be uniform [4, 5, 6].

One could view interactive ensembles with uniform fusion parameters as a dynamical model averaging method. The question then arises whether there is an advantage (in terms of estimating attractor statistics) in such an dynamical approach compared to the more straightforward a posteriori averaging in a conventional uncoupled ensemble of models. In this report we address this question in the context of estimating attractor statistics in low dimensional chaotic systems using ensembles of imperfect models.

In the above mentioned earlier studies, we mostly have considered only one instance (of a manually constructed set) of imperfect models. This might lead to a overoptimistic bias in results. Therefore we now consider distributions of imperfections, and consider averages over these distributions.

1.1 Setting up the analysis of the error reduction

We consider a setting with a ground truth dynamics of the form

\[
\dot{x} = f_{gt}(x)
\]  

(in which “gt” stands for ground truth) and imperfect models with parameter vector \(\theta\), which are of the form

\[
\dot{x} = f(x, \theta)
\]

in which the states \(x \in \mathbb{R}^n\), and the parameters \(\theta \in \mathbb{R}^m\). In the following we assume that for both the ground truth and the imperfect models, the dynamics leads to a stationary distribution \(\rho(x)\) of the states. For the imperfect models, this distribution will depend on the parameters, so we have \(\rho(x) = \rho(x; \theta)\). Any statistics \(S\) can in principle be derived from this distribution in \(x\). For example, the mean value is \(\int x \rho(x; \theta) dx\). For the ground truth, we have \(S = S_{gt}\), which is treated as given value. For the imperfect models we have \(S = S(\theta)\), which is a function of the parameter vector \(\theta\).

As an error measure for a method that result in statistics \(S\) we define the loss \(L(S)\). An example loss function is quadratic loss function,

\[
L(S) = |S - S_{gt}|^2.
\]
When the $N$ imperfect models have parameters $\theta_{\mu}$, a weighted supermodel with uniform weights (abbreviated as uniform supermodel) is of the form
\[
\dot{x} = \frac{1}{N} \sum_{\mu} f(x, \theta_{\mu})
\] (4)

If, in addition, the imperfect models are linear in the parameters, the uniform supermodel is just the model with averaged parameters, i.e.
\[
\dot{x} = f(x, \bar{\theta})
\] (5)

with $\bar{\theta} \equiv \frac{1}{N} \sum_{\mu} \theta_{\mu}$. An example of a model that is linear in the parameters is the Lorenz 63 model [9] with parameters $\theta = (\sigma, \rho, \beta)$,
\[
\begin{align*}
\dot{x} &= \sigma(y - x), \\
\dot{y} &= x(\rho - z) - y, \\
\dot{z} &= xy - \beta z
\end{align*}
\]

If the imperfect models are linear in the parameters, the statistics $S_{\text{us}}$ of the uniform supermodel reduces to $S(\bar{\theta})$,
\[
S_{\text{us}} = S(\bar{\theta})
\] (6)

Here, we also consider the conventional (non-interactive) ensemble. There are several ways to define the ensemble prediction of the statistics. For instance, one can just take the average of the statistics of each of the individual models,
\[
S_{\text{ens}} = \frac{1}{N} \sum_{\mu} S(\theta_{\mu}) \equiv \overline{S(\theta)}
\] (7)

An alternative is to consider the averaged density $\frac{1}{N} \sum_{\mu} \rho(x; \theta_{\mu})$ and compute the statistics of this density. When the statistics is a linear functional of the stationary distribution, $S(\theta) = \int g(x) \rho(x, \theta) dx$, then both definitions coincide. In general, e.g. when the variance is considered, they are different. For simplicity of analysis, we take in this section (7) as a definition.

To analyze the performance of ensemble methods, we assume that $\theta_{\mu}$ is “drawn” according to a probability density $P(\theta)$. Suppose that given a drawn ensemble of models, a certain method applied to this ensemble leads to a statistic $S_{\text{method}}$, then, given that the loss function of interest is $L(S)$, we can define the expected loss of this method as
\[
R_{\text{method}} = \mathbb{E}_{\theta}(L(S_{\text{method}}))
\] (8)

For the quadratic loss function (3), this is
\[
R_{\text{method}} = \mathbb{E}_{\theta}(|S_{\text{method}} - S_{gt}|^2)
\] (9)

We analyze the expected loss of the uniform supermodel and the conventional ensemble average method. For completeness, we also consider the random model strategy, i.e.
we consider the expected loss of a random model drawn according to $P(\theta)$. For linear imperfect models, the expected squared loss of the three methods is then

\[
R_{\text{rnd}} = \mathbb{E}_\theta(|S(\theta) - S_{gt}|^2)
\]

\[
R_{\text{pe}} = \mathbb{E}_\theta(|\frac{1}{N} \sum_\mu S(\theta_\mu) - S_{gt}|^2)
\]

\[
R_{\text{us}} = \mathbb{E}_\theta(|\frac{1}{N} \sum_\mu \theta_\mu - S_{gt}|^2)
\]

1.2 An abstract counter example

With the quadratic loss, it is well known that

\[R_{\text{pe}} \leq R_{\text{rnd}}\] \hspace{1cm} (10)

due to the reduction in variance, $\mathbb{E}_\theta(S - \mathbb{E}_\theta(S))^2 \geq \mathbb{E}_\theta(\bar{S} - \mathbb{E}_\theta(S))^2$.

For the uniform supermodel such a relation does not hold, and in fact an artificial counter-example can be constructed. Suppose there are two clusters of models, the first around $\theta = -1$ and the other around $\theta = 1$. Further suppose that the performance of these models is relatively good and that the performance of models between the two clusters is bad. Then obviously the performance of uniform supermodels will on average deteriorate the performance of the models in the ensemble. The posterior averaging of the statistics will still be helpful. In Figure 1 this example is illustrated.

![Figure 1: Left: assumed probability density $P(\theta)$ from which model parameters $\theta$ are drawn. Right: assumed loss $L(S(\theta))$. In this example, $R_{\text{rnd}} = 0.27$, $R_{\text{pe}} = 0.22$, $R_{\text{us}} = 0.91$](image)

1.3 Smooth model spaces

So counter examples can be constructed where uniform supermodels perform quite badly. However, one may ask what if the model performance is a more smooth function in model
space, and imperfect models are drawn from a distribution that is centered around an optimal model. This we try to illustrate in the following. We generate models by taking $P(\theta)$ the standard normal distribution. We consider a (hypothetical) “linear statistic”, $S(\theta) = \theta$, a “sub-linear statistic” $S(\theta) = \frac{1}{2}(\tanh(\theta) + 1)$ and a “super-linear statistic” $S(\theta) = \theta^3$. The ground truth statistic $S_{gt}$ is defined by the optimal model $\theta_{gt} = 0$, so $S_{gt} = S(0)$. The resulting quadratic loss functions $L(S(\theta)) = (S(\theta) - S_{gt})^2$ as function of $\theta$ are plotted in Figure 2. The intuition is that with linear statistics, first averaging over parameters and then computing the statistic is equivalent to first computing the statistic for each model and then averaging. In the sub-linear case, the statistics function $S$ has a damping effect on larger errors in the parameters, and therefore it is favorable to first first compute the statistic for each model and then average the statistics. In the super-linear case, statistics has an amplifying effect on larger errors in the parameters. Therefore first averaging over parameters and then computing the statistic is favorable. Numerical results for expected squared loss in ensembles of size $N = 20$ confirming these observations are tabulated in Table 1.

<table>
<thead>
<tr>
<th>$S(\theta)$</th>
<th>$R_{std}$</th>
<th>$R_{pe}$</th>
<th>$R_{us}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>1.1</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>$\frac{1}{2}(\tanh(\theta) + 1)$</td>
<td>0.14</td>
<td>0.007</td>
<td>0.012</td>
</tr>
<tr>
<td>$\theta^3$</td>
<td>2.9</td>
<td>0.15</td>
<td>0.0023</td>
</tr>
</tbody>
</table>

Table 1: Expected costs for an ensemble of $N = 20$ models, with a standard normal parameter distribution.

These effects of damping and amplifying are also visible in an experiment with the Lorenz 63 system. As the ground truth, we took the Lorenz 63 system with standard parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. We considered the statistic $S = \text{std}(z)$. As imperfect models we took L63 models with perturbed $\rho$-parameter. The statistic as a function of $\rho$ is computed numerically and plotted in Figure 3. Clearly there are two regimes: for small $\rho$, the system converges to a point attractor (or at least close to it in our simulations). For large $\rho$, there is the butterfly shaped attractor, which has a size that grows approximately linearly with $\rho$.

We generated ensembles of $N = 3$ imperfect models by adding zero mean Gaussian noise to $\rho$, with a fixed standard deviation $\text{std}(\rho)$. Then for each method the expected loss is computed by taking repeated draws of ensembles and assess results using a linear interpolation of the statistics $\text{std}(z)$ as a function of $\rho$. In Figure 4 we plotted for different values of $\text{std}(\rho)$ the results of individual realisations of the estimate of $\text{std}(z)$ according to the different methods. We see that for small imperfections ($\text{std}(\rho)/\rho = 0.05$), the two ensemble methods have about the same performance. This can be understood since for small perturbations, the function $S(\rho)$ is almost linear. For larger imperfections ($\text{std}(\rho)/\rho = 0.1$), the uniform supermodel predicts best. This is because some of the models fall left of the transition and therefore predict a very small $\text{std}(z)$. In the posterior ensemble, the predicted $\text{std}(z)$ is therefore often decreased by a factor $1/3$,
Figure 2: Top left: assumed standard normal probability density $P(\theta)$ from which model parameters $\theta$ are drawn. Other graphs are the quadratic loss functions $L(S(\theta)) = (S(\theta) - S_{gt})^2$ for different hypothetical statistics $S(\theta)$. Top right: $S(\theta) = \theta$. Bottom left: $S(\theta) = \frac{1}{2}(\tanh(\theta) + 1)$. Bottom right: $S(\theta) = \theta^3$. $S_{gt} = S(0)$ in all cases.

while the uniform supermodel mostly corresponds with a parameter that is still right of the transition. In a way, we are here in the superlinear regime (due to the transition in $\text{std}(z)$). If we further increase the perturbations, $\text{std}(\rho)/\rho = 0.3$ and $\text{std}(\rho)/\rho = 0.3$, we see that the uniform supermodel corresponds more and more often with a parameter that is left of the transition, while often there is still one model or even two models right of the transition. In these cases, the posterior ensemble performs better. In a way, we are here in the sublinear regime (due to the asymptotic values of $\text{std}(z)$, left and right of the transition). In figure 5, we plotted the simulated expected squared loss $R$ as function of $\text{std}(\rho)/\rho$, based on 10000 randomly drawn ensembles. Here we clearly see that both ensemble methods perform better than a randomly chosen model. For small parameter perturbations, the uniform supermodel performs better, but with larger parameter perturbations, the posterior ensemble performs better.
Figure 3: Lorenz 63: the statistic \( \text{std}(z) \) as a function of the parameter \( \rho \) (other parameters have the standard value \( \sigma = 10, \beta = 8/3 \)). The graph is determined numerically by for each \( \rho \) simulating the Lorenz 63 system over 10000 time units, after a transient of 1000 time units. The \( \rho \) is incremented with steps of 0.28.

1.4 Imperfect models with optimized parametrization

In [10] a six-dimensional model is derived by projecting the barotropic vorticity equation onto two zonal modes that only depend on latitude and two wave modes of one and two wavelengths along the zonal direction. This system is given by

\[
\begin{align*}
\dot{x}_1 &= \tilde{\gamma}_1 x_3 - C(x_1 - x_1^*) \\
\dot{x}_2 &= -(\alpha_1 x_1 - \beta_1) x_3 - C x_2 - \delta_1 x_4 x_6 \\
\dot{x}_3 &= (\alpha_1 x_1 - \beta_1) x_2 - \gamma_1 x_1 - C x_3 + \delta_1 x_4 x_5 \\
\dot{x}_4 &= \tilde{\gamma}_2 x_6 - C(x_4 - x_4^*) + \epsilon(x_2 x_6 - x_3 x_5) \\
\dot{x}_5 &= -(\alpha_2 x_1 - \beta_2) x_6 - C x_5 - \delta_2 x_4 x_3 \\
\dot{x}_6 &= (\alpha_2 x_1 - \beta_2) x_5 - \gamma_2 x_4 - C x_6 + \delta_2 x_4 x_2. 
\end{align*}
\] (11)

The various coefficients \( \alpha_m, \beta_m, \delta_m, \tilde{\gamma}_m, \epsilon \) and \( \gamma_m \) are given in [10]. The terms multiplied by \( \alpha_i \) model the advection of the waves by the zonal flow, the \( \beta_i \) terms are due to the Coriolis force; the \( \gamma \) terms are generated by the topography. The \( C \) terms take care of the Newtonian damping to the prescribed zonal profile. The \( \delta \)- and \( \epsilon \)-terms describe the nonlinear triad interaction between a zonal mode and the two waves. For suitable values of the six free parameters \( C, x_1^*, x_4^*, b, \beta \) and \( \gamma \) the model displays chaotic behavior with a-periodical transitions between two preferred regimes. These regimes correspond to a strong zonal flow and a flow with a strong wavenumber 1 amplitude (See [11] for more details). This behavior is also observed in the real atmosphere and is a challenge to reproduce in numerical models.

As we did earlier in the supermodeling paradigm, we assume that in the imperfect models not all scales are resolved. We construct two incomplete models, model A and
Figure 4: Lorenz 63: the statistic \( \text{std}(z) \) for different values of parameter noise \( \text{std}(\rho) \) (expressed as a ratio to \( \rho \)) (other parameters have the standard value \( \sigma = 10 \), \( \beta = 8/3 \)). Each graph shows 100 random realizations of estimates according to the different methods (black dashed: random model, red dashed: posterior ensemble, blue: uniform supermodel, black horizontal line: assumed ground truth).

model B. Both have a missing variable. In model A, variable \( x_6 \) is not dynamically modeled. In model B, variable \( x_5 \) is not modeled. In both models, the effect of the missing variable is modeled by a parametrization, a function that depends on the other dynamical variables in the imperfect model, and which is optimized on the basis of data. Then we combine these incomplete models into a single uniform supermodel and see what happens.

To be more explicit, the ground truth model is given by equations for \((x_1, \ldots x_6)\) which are given in (11). We write these formally as

\[
\begin{align*}
\dot{x}_1:4 &= f(x_1:4, x_5, x_6) \\
\dot{x}_5 &= g(x_1:4, x_5, x_6) \\
\dot{x}_6 &= h(x_1:4, x_5, x_6)
\end{align*}
\]

in which \( f \) stands for the first 4 components of the vector field defined in (11), \( g \) for the fifth and \( h \) for the sixth component.

So, imperfect model A is assumed to be of the form

\[
\begin{align*}
\dot{x}_1:4 &= f(x_1:4, x_5, \phi_A(x_1:4, x_5)) \\
\dot{x}_5 &= g(x_1:4, x_5, \phi_A(x_1:4, x_5))
\end{align*}
\]

in which \( \phi_A \) is a function of 5 variables, \( x_1, \ldots, x_5 \) and is a parametrization for the variable \( x_6 \) in the ground truth. It is optimized on a dataset of observations from the ground truth \( \{x_1(t), \ldots x_5(t)\} \), in such a way that for these data the difference between \( x_6(t) \) and \( \phi_A(x_1(t), \ldots, x_5(t)) \) is minimized.
Likewise, imperfect model B is assumed to be of a similar form,

\[
\begin{align*}
\dot{x}_{1:4} &= f(x_{1:4}, \phi_B(x_{1:4}, x_6), x_6) \\
\dot{x}_6 &= h(x_{1:4}, \phi_B(x_{1:4}, x_6), x_6)
\end{align*}
\]

in which now \(\phi_B\) is a function of 5 variables, \(x_1, \ldots, x_4, x_6\) representing a parametrization for the variable \(x_5\) in the ground truth. It is optimized in a similar manner as \(\phi_A\) in model A.

The uniform supermodel is now constructed by straightforward averaging of the corresponding variables in model A and model B.

\[
\begin{align*}
\dot{x}_{1:4} &= \frac{1}{2}(f(x_{1:4}, x_5, \phi_A(x_{1:4}, x_5)) \quad + f(x_{1:4}, \phi_B(x_{1:4}, x_6), x_6)) \\
\dot{x}_5 &= g(x_{1:4}, x_5, \phi_A(x_{1:4}, x_5)) \\
\dot{x}_6 &= h(x_{1:4}, \phi_B(x_{1:4}, x_6), x_6)
\end{align*}
\]

Variables that do not correspond to a variables in the other model are not averaged.

To illustrate possible different behaviour, we considered to types of parametrization functions \(\phi\). A linear-quadratic parametrization with functions with only linear and quadratic terms, i.e. \(\phi = w_0 + \sum_i w_i x_i + \sum_{ij} w_{ij} x_i x_j\) and so called random forests, a well known modeling class from machine learning. Trajectories of the resulting imperfect models and supermodels, both in comparison with a ground truth trajectory are plotted in Figures 7 and 8. As a reference, the ground truth trajectory is plotted in Figure 6.

In Figure 7 we see that the uniform supermodel can improve the performance of individual imperfect models. However in Figure 8 we see a counter example. This seems similar to the situation described schematically in Figure 1: the individual imperfect models are optimized and seem to do quite well, whereas dynamic averaging of these models seem to deteriorate results.
1.5 Conclusion

There are situations that uniform supermodeling, i.e. the uniform dynamic fusion of imperfect models improves upon the long term behavior of individual models. However, there is no guarantee that this is the case, unlike with posterior averaging. Counter examples in which uniform supermodels deteriorate results are easily constructed, and this may be more often the case than one might think at first. This means that supermodeling is better thought of as models that are constructed using the imperfect models as basis functions. As in e.g. regression, the weights in these combinations of basis functions have to be fitted to data, and like any fitting procedure care against overfitting must be taken.
2 Attractor learning

We have seen in the previous section that supermodels should be more seen as a parametrized model class with the imperfect models as basis functions, and the connecting parameters to be learned from data. Earlier, we have seen that learning of supermodels can be tedious. In particular learning on the basis of “weather prediction” (vector field learning) might be insufficient if one is interested in climate (attractor statistics). For example in systems with hidden variables, such as the driven Lorenz 63 system that we studied last year in the context of supermodeling, vector field learning can lead to supermodels which perform very bad attractor statistics. As a remedy, we introduced attractor learning, which in the constructed model ensemble at hand, indeed lead to a model which improved upon attractor statistics [2, 3].

In this constructed model ensemble, the vector field of each member of the model was close to the assumed ground truth. In general this might not be the case. If one is interested in learning dynamical systems, then the vector field remains an important feature. Basically we want to learn models that have both reasonable short term and long term performance. In this section we describe a Bayesian inspired formalism to do learning in dynamical systems, in which both vector field learning and attractor learning is combined.

2.1 Bayesian learning

Before we describe our proposed formalism, we give a very short review on Bayesian learning [1]. We consider learning of parameterized models with parameters $w$. In Bayesian learning one starts with a prior over the parameters $p(w)$. Furthermore, one needs a likelihood model $P(D|w)$ that describes how likely the observation of a set of observations $D$ is, given a parameter setting $w$. With these ingredients, one can compute the posterior distribution $P(w|D)$ of the parameters given a data set of observations $D$.
by applying Bayes’ rule,

\[ P(w|D) = \frac{P(D|w)P(w)}{P(D)} \]  \hspace{1cm} (12)

with \( P(D) = \int P(D|w)P(w)dw \). Since \( P(D) \) is only a normalization factor for the posterior distribution, we may write

\[ P(w|D) \propto P(D|w)P(w) \]  \hspace{1cm} (13)

In sequential Bayesian learning, the posterior distribution is continuously updated upon arrival of new data \( D_t \),

\[ P(w|D_1, \ldots, D_t) \propto P(D_t|w)P(w|D_1, \ldots, D_{t-1}) \]  \hspace{1cm} (14)

### 2.2 Vector field learning in linear models

We consider linearly parametrized models

\[ \dot{x}_i = \sum_{\mu=1}^{M} w_{i\mu} f_{i\mu}(x_1, \ldots, x_N) \]  \hspace{1cm} (15)

with parameters \( w_{i\mu} \) and basis functions \( f_{i\mu} \). We further consider the case where we have a dataset of observations \( D \) at a number of times \( t = t_1, \ldots t_T \) where we not only have the state but also the vector field, so

\[ D = \{(x(t_1), v(t_1)), \ldots, (x(t_T), v(t_T))\} \]  \hspace{1cm} (16)

If we would only be interested in a model that predicts the vector field value of the system given the current state, then we could apply standard Bayesian linear regression. Bayesian linear regression starts with assuming a Gaussian prior

\[ p(w) = \mathcal{N}(0, \alpha^{-1}I) \]  \hspace{1cm} (17)

and assuming Gaussian noise on the vector fields with inverse variance \( \sigma_{vv}^{-2} = \beta_{vv} \) Due to the Gaussianity and linearity of the model, the posterior is also Gaussian,

\[ p(w|D_{vv}) = \mathcal{N}(m, S) \]  \hspace{1cm} (18)

which has mean and covariance that reads for the \( i \)-th component (basically we have for each \( i \) an independent regression problem),

\[ m_i = \beta S_i^{-1} F_i^T V_i \]
\[ S_i^{-1} = \beta F_i^T F_i + \alpha I \]

with \( V_i \) is the vector with components \( V_i(t) = v_i(t), t = 1, \ldots, T \) and \( F_i \) the matrix with components \( F_i(t, \mu) = f_i(i, \mu)(x_1(t), \ldots, x_N(t)), \mu = 1, \ldots, M \) and \( t = 1, \ldots T \).

The Gaussian posterior can be written as

\[ p_{vv}(w) \equiv p(w|D_{vv}) \propto \exp(-TE_{vv}(w)) \]  \hspace{1cm} (19)
2.3 Attractor learning

Regarding attractor learning, we make a number of assumptions. The first assumption is that the system is stationary, and that the states \( x(t) \) that are observed in the data can be considered to be a sample from the stationary distribution. Furthermore we assume a likelihood function based on an attractor error measure \( E_A(w) \) proportional to a distance function between stationary distributions of the model \( p_{\text{model}} \) and the empirical distribution \( p_{\text{emp}} \) defined by the data set. Here we take this distance measure to be proportional to the \( L_1 \) distance between the empirical distribution according to a trajectory of the model and the empirical distribution according to the data. To make the distance non-singular, we bin the state space (in the examples, the state space is only three dimensional, so that binning is still feasible). The binned \( L_1 \) distance between distributions \( p \) and \( q \) is defined as

\[
L_1(p, q) = \frac{1}{2} \sum_i |p(i) - q(i)|
\]  

in which \( p(i) \) and \( q(i) \) are the mass proportions falling into bin \( i \) according to the distributions \( p \) and \( q \). The prefactor \( \frac{1}{2} \) is to make sure that \( L_1 \) is between zero and one. By taking \( E_A \) equal to the \( L_1 \) distance, we have

\[
E_A(w) = L_1(p_{\text{model}}, p_{\text{emp}})
\]

By introducing a multiplier \( \gamma \), the likelihood is of the form \( \exp(-\gamma E_A(w)) \).

Now there is an issue with the multipliers. On the one hand, it is not clear what to take for \( \gamma \). On the other hand, with a larger number of data points \( T \), the Gaussian posterior (19) almost becomes singular, which may not be desirable if the model should be adapted to the attractor measure. The reason for the large multipliers \( T \) is the assumption of independent Gaussian noise on each data instance of the vector field. In our models, error is mostly not due to independent Gaussian noise, but rather due to model mismatch. This mismatch causes correlations between data, making the effective sample size smaller. So, we assume in (19) a multiplier \( \tau \ll T \). Then, using this adapted Gaussian vector field posterior as a prior for the second stage with the attractor measure, we get

\[
p_A(w) \equiv p(w|D) \propto \exp(-\gamma E_A(w) - \tau E_{vv}(w))
\]

Which values are to be taken for \( \gamma \) and \( \tau \) is not clear to us, so we determined these by rather informal empirical methods to obtain an acceptable and practical compromise in vector field error and attractor error.

Direct inference of \( p_A \) is infeasible. However, due to the Gaussian structure of \( \exp(\tau E_{vv}(w)) \) – which can be inferred exactly – we may resort to Elliptical slice sampling [8]. Elliptical slice sampling is a Metropolis-Hastings methods with moves

\[
w' = w \sqrt{1 - \epsilon^2} + \epsilon u
\]

with \( u \) generated from the Gaussian \( \exp(\tau E_{vv}(w)) \), and \( \epsilon \in [-1, 1] \). The move is accepted with a probability that is only based on the values of \( E_A \) before and after the proposed
move,

\[ p(\text{accept}) = \min\{1, \exp(-\gamma E_A(w') - E_A(w))\} \]  

(24)

In elliptical slice sampling, an automatic procedure to adapt \( \epsilon \) is provided in addition. The quantities \( E_A(w') \) is computed by doing a simulation of the model with parameters \( w' \).

2.4 Numerical experiments

We considered two experimental conditions. In the first condition, the ground truth model is a fully observable conventional Lorenz 63 system with standard parameter settings. In the second condition, we took the partially observable driven Lorenz 63 system as ground truth. In the previous year, the driven Lorenz 63 system was already introduced. It features hidden states, resembling unresolved scales in a real system. The driven Lorenz 63 system is defined by the equations

\[
\begin{align*}
\dot{x} &= \sigma(y - x) + \epsilon z_h, \\
\dot{y} &= x(r - z) - y, \\
\dot{z} &= xy - \beta z + \delta(x_h + \eta), \\
\dot{x}_h &= \sigma(y_h - x_h), \\
\dot{y}_h &= x_h(r - z_h) - y_h, \\
\dot{z}_h &= x_h y_h - \beta z_h.
\end{align*}
\]  

(25)

The subsystem \( h \) is the ordinary Lorenz 63 system which drives the non-indexed subsystem. The parameters \( \epsilon \) and \( \delta \) determine the magnitude of the driving, while \( \eta \) is some constant drift; \( \sigma, r \) and \( \beta \) are the parameters that appear in the Lorenz equations. For simplicity we have again taken the standard values in both subsystems \( \sigma = 10, r = 28 \) and \( \beta = 8/3 \). Furthermore, we assume that only the subsystem \( v \) is accessible for observations and our aim is to model its time series \( x(t), y(t) \) and \( z(t) \). In Figure 9 is shown the projection of the full attractor on the three dimensional space \((x, y, z)\) where it is clear that it looks like the familiar Lorenz attractor.

As basis functions of the model system we take polynomials in \( x, y, z \) up to order 3, so terms of type \( xyz \), and \( x^2z \) are included. These basis functions are used for all three components. We deliberately took polynomials up to order three to leave open the possibility of a sort of overfitting, rather than to make use that the Lorenz 63 system is described by polynomials up to order 2 and constrain the basis functions accordingly. This would also favour performance on the driven Lorenz 63 system, with its attractor very similar to the conventional Lorenz 63 system.

In both experimental conditions, we generated a data set of 1000 data points as follows. We took a random initialization, then run the assumed ground truth system for a period 10000 time units to get rid of the transient. Then we further run the system for 4000 time units to collect data, with four time units difference between each data point. We recorded for each data point both the state and the current vector direction.
Figure 9: The trajectory of the driven Lorenz63 system.

(the tendency). In the driven Lorenz system, we collected these only for the three visible variables.

With these data, we first trained models using the linear regression. In the case where the ground truth is the fully observed Lorenz 63 system, linear regression leads – as expected – to an almost perfect fit. The sum squared error averaged per data point is of order $10^{-11}$. The resulting model has a good long term behavior, close to the assumed ground truth. See Figure 10. Long term statistics (based on runs of $10^5$ time units) in terms of probability distributions is also illustrated by the three marginal distributions $p(x), p(y), p(z)$ according to both the model and the ground truth, plotted in Figure 11. The $L_1$ distance is found to be 0.023. Since the attractor fit is already so good, we did not apply attractor learning.

In the case where the ground truth is the partially observed driven Lorenz 63 system, linear regression leads to a less perfect fit. The mean sum squared error is about 1500. The resulting (most probable) model has a bad long term behavior, leading to a limit cycle of very small radius, see Figure 12. Long term statistics (based on runs of $10^5$ time units) in terms of probability distributions is also illustrated by the three marginal distributions $p(x), p(y), p(z)$ according to both the model (almost spikes) and the ground truth, plotted in Figure 13. The $L_1$ distance is found to be 0.977.

We then performed the second step by applying slice sampling. We took $\gamma = 1$ and $\tau = 10^{-5}$ so that the contributions from the “trainingserror” $E_{vv}$ and the “attractor
error" $E_{cv}$ are in balance. We terminated the sampling after 100 iterations. See Figure 14

After slice sampling, the sum squared error averaged per data point has increased to about 21000. The resulting model (the last sample), however, has a much better long term behavior, much closer to the assumed ground truth. See Figure 15 for plots of trajectories. Long term statistics (based on runs of $10^5$ time units) in terms of the three marginal distributions $p(x)$, $p(y)$, $p(z)$ according to both the model and the ground truth, are plotted in Figure 16 also reveal better performance. The $L1$ distance is found to be 0.32, which indicates a reasonable fit.

### 2.5 Conclusion

Learning dynamical systems based on short term predictions only can lead to models that have a bad long term behaviour. We proposed a Bayesian - inspired attractor learning method. With these models can be trained that not only perform on short term prediction, but also has a more reasonable long term behaviour. The number of iterations required might be feasible for larger models of practical interest. In this study we did not further study model uncertainty, which is still subject of future research. An other issue still to be resolved is a better and more fundamental understanding of the multipliers.

### References

Figure 11: The marginal distributions of the x, y and z components of both the learned model as the ground truth Lorenz63 model. Red: the learned model. Black: the ground truth model.


Figure 12: Projections of a long trajectory of the learned driven Lorenz model. Red: learned model after regression. Note the learned system ends up in a limit cycle of very small radius. Black: Driven Lorenz (ground truth) model. White dots: position of the data points.


Figure 13: The marginal distributions of the $x, y$ and $z$ components of both the learned model after regression of the vector fields and the ground truth driven Lorenz model. Red: the learned model. Black: the ground truth model.
Figure 14: The evolution of different quantities during the sampling process of 100 iterations. Top: the L1-distance. Middle: the “training error” $E_{\text{vv}}$. Bottom: the “log likelihood” $-\gamma E_A - \tau E_{\text{vv}}$, with $\tau = 10^{-5}$. 
Figure 15: Projections of a long trajectory of the learned driven Lorenz model. Red: learned model after full learning. Black: Driven Lorenz (ground truth) model. White dots: position of the data points.

Figure 16: The marginal distributions of the $x,y$ and $z$ components of both the learned model after full learning and the ground truth driven Lorenz model. Red: the learned model. Black: the ground truth model.