Evaluating the Performance of Bayesian Treed Gaussian Process Models for Predicting the Velocity of Ice in the Arctic Sea-Ice Computer Model

Nirupama Tamvada

April 21, 2022

Abstract

Stationary Gaussian Processes (GP's) have become a popular standard in a computer model framework due to their many desirable properties. However, for many real-life applications, the stationarity assumption can often be restrictive, particularly when it comes to modelling discontinuities, as well as accounting for differing regional variability in the parameter space. Thus, Gramacy and Lee (2008) proposed relaxing the stationarity assumption in a computationally scalable manner, by partitioning the input space through a tree-based approach, and subsequently fitting stationary Bayesian Gaussian Processes within each partition region. In this report, we apply four such Bayesian tree-based approaches: the classification and regression tree, random forests, treed Gaussian Processes and treed Gaussian Processes with a limiting linear model comparion to a stationary Gaussian Process. We do so to the end of predicting the ice velocity in the Arctic Sea-Ice Computer Model, which is speculated to be difficult to predict due to the numerical model displaying differential variability in some regions of the parameter space. Overall, we found that while the RMSE of the treed approaches (random forests and treed GP's) are marginally lower than that of the stationary GP, they are not significantly different enough to claim that the tree-based approach showed a clear, superior performance. The Random forests algorithm also had the lowest RMSE. Thus, while ice-velocity remains difficult to predict, it can be said that the more parsimonious random forests may be preferable for this application compared to the tree-based GP's as well as a stationary GP.

Introduction

In a computer model framework, using a Gaussian Process (GP) is a popular choice. This is because GPs are conceptually straightforward and satisfy several desirable properties: predictive uncertainty can be quantified, prior knowledge can be easily accounted for in the form of covariance functions and Bayesian approaches, a GP conditionally to given data (equality constraints) is also a GP, and the partial derivatives of a GP are also GPs. A standard approach in the literature for emulation is the use of a stationary smooth Gaussian Process. However, there are some particular disadvantages of using this standard form of a GP. In particular, the stationarity assumption, which leads to the usage of the same covariance structure being used throughout the space, often does not scale well to a wide range of applications. This assumption can be too restrictive when it comes to modelling discontinuities, and sharp changes of the function surface. Importantly, the estimated predictive error of a stationary GP only depends on a global measure of error that uses all of the discrepancies between observations and predictions. However, it is often the case that some regions of the space exhibit larger variability compared to others. In many applications, it is more desirable to model the uncertainty over the output space in a non-uniform manner, so that the predictive error also depends on the locally observed response values.

Thus many real-world applications require the use of more flexible, non-stationary Gaussian processes. However, fully non-stationary Bayesian Gaussian Processes can be difficult to fit and do not scale well computationally to a large number of data-points. Gramacy and Lee (2008) address this problem by proposing partitioning the input space into regions. A stationary Gaussian Process can then fit within each region. Such a partitioning is attractive, as if offers a straightforward mechanism for creating a less computationally-demanding non-stationary Gaussian Process. Under a Bayesian framework, the predictive uncertainty can also be estimated by applying a Bayesian model averaging approach. This approach is capable of flexibly modelling both continuous functions, as well as discontinuities. It additionally also resolves the variability issue, as the estimated predictive error estimated can now account for differing variability between regions of the space.

In particular, Gramacy and Lee (2008) apply a treed partitioning approach. A Classification and Regression Tree (CART), which which fits a constant surface in each leaf, is used as it is easy to implement and interpret.

Their approach involves applying a tree-generating prior to a CART. A Gaussian Process (or any suitable model) is then applied to each partition/leaf of this tree, with hierarchical priors specified for the hyper-parameters. Bayesian model-averaging is then finally applied to average over in the posterior in order to make predictions and estimate predictive uncertainty. This approach was applied to two applications: a computer model simulating the launch of NASA's Langley Glide-Back rocket booster, as well to a real-life application for predicting the acceleration of the head of a motorcycle rider as a function of time. In both cases, the treed Bayesian Gaussian Process showed a superior performance compared to a stationary Gaussian Process, as it was able to model discontinuities (particularly those those by numerical instability of the simulation as well) and account for for the differing levels of uncertainty in the space.

On the usefulness of computer models, sea-ice models have become increasingly useful in the study of the high-latitude climate system. The Arctic Sea-Ice Computer Model in particular, offers valuable information on the effect of ice physics parameters such as drag coefficients, and snowfall rate on outputs such as ice-thickness that are difficult to quantify. While stationary Gaussian Processes have shown good performance in modelling specific outputs such as ice area, they showed a lesser predictive accuracy in modelling ice velocity in particular. Chapman, Welch, Bowman, Sacks and Walsh (1994) speculated that this may be due to the numerical model being erratic in some regions of the parameter space, which may have reduced predictive power. Given the flexibility of the treed Bayesian Gaussian Process in modelling discontinuities and in accounting for differing variability in the parameter space, it was of interest to see if tree-based approaches can significantly improve the prediction accuracy of ice velocity in the Arctic Sea-Ice Computer Model.

Specifically, the objective of this report is to compare the performance of a Bayesian stationary Gaussian Process to four tree-based, Bayesian methods: Classification and Regression Trees, Random Forests, Treed Gaussian Processes, and Treed Gaussian Processes with Limiting Linear Regression Models, to the end of potentially improving the predictive accuracy of ice velocity in the Arctic Sea-Ice Computer Model.

Methods

Computer Model

The Arctic Sea-Ice computer model is a dynamic formulation based on a momentum balance for a mass of ice within a grid cell. The specific details of the thermodynamic formulation can be found in [1]. The purpose of this computer model is to estimate sensitivities of ice physics input parameters on outputs that are key sea ice properties. The model was run with a daily time-step from the period of January 1, 1960-1988. It was run on a 110kn polar grid, which covers the Arctic Ocean and nearby bodies of water. Specifically, the data-set has 6 output parameters: Ice mass, Ice area, Ice velocity and Range of Area. For this project, we focus on the Ice velocity variable, which has been shown to be comparatively harder to predict accurately using a stationary GP. The model has 13 ice physics input parameters. These are as follows:

- **Drag coefficients:** These describe the drag forces exerted on ice by air (`AtmosDrag`) and ocean currents (`OceanicDrag`).
- **Ice Strength:** This measure is a function of the grid cell ice thickness and compactness (`LogIceStr`). It has been log-transformed in the data
- **Minimum lead fraction:** Permits the inclusion of the effect of small-scale motions within a grid cell. These motions tend to maintain a small fraction (several percent) of open water or very thin ice within the pack even during winter, which as most of the ocean-atmosphere exchanges of sensible and latent heat, as well as new ice growth and salt rejection ocur in these regions (`MinLead`)
- **Albedos:** Measure of the diffuse reflection of solar radiation for snow (`SnowAlbedo`), ice (`IceAlbedo`) and open-water (`OpenAlbedo`)
- **Exchange coefficient, surface sensible heat:** Bulk transfer coefficient for sensible heat (`SensHeat`)
- **Exchange coefficient, surface latent heat:** Bulk transfer coefficient for latent heat (`LatentHeat`)
- **Snowfall rate:** Snowfall at prescribed rates that vary seasonally (`Snowfall`)
- **Cloud depletion of solar flux:** . Downcoming fluxes of solar radiation which vary from the winter to the summer (`Shortwave`)
- **Cloud enhancement of longwave flux:** Downcoming fluxes of longwave radiation which vary from the winter to the summer (`Longwave`)
- **Oceanic heat flux:** prescribed vertical flux of oceanic heat into the mixed layer, which can determine the rate of ice growth and melt (`OceanicHeat`)

Stationary Gaussian Processes

A real-valued stochastic process $\{X_t, t \in T\}$ where X_t is a set of random variables indexed by T is a Gaussian Process having a jointly Gaussian distribution for a finite subset of indices if all the finite-dimensional distributions have a multivariate normal distribution. For any choice of distinct indices, the random vector $\mathbf{X} = (X_{t_1}, ..., X_{t_k})'$ has a multivariate normal distribution. A Gaussian Process is specified by a mean function $\mu(\mathbf{x}) = E(Z(\mathbf{x}))$, and a correlation function $R(\mathbf{x}, \mathbf{x}')$.

We assume the correlation function can be written as follows:

$$R(\mathbf{x}_j, \mathbf{x}_k|g) = R^*(\mathbf{x}_j, \mathbf{x\_k}) + g\delta_{j,k}(1)$$

Treed Partitioning

Heirarchial Priors

Estimation

Bayesian Model Averaging

Cross-Validation

Sensitivity Analysis

Results

The Random Forests Model shows the highest predictive accuracy

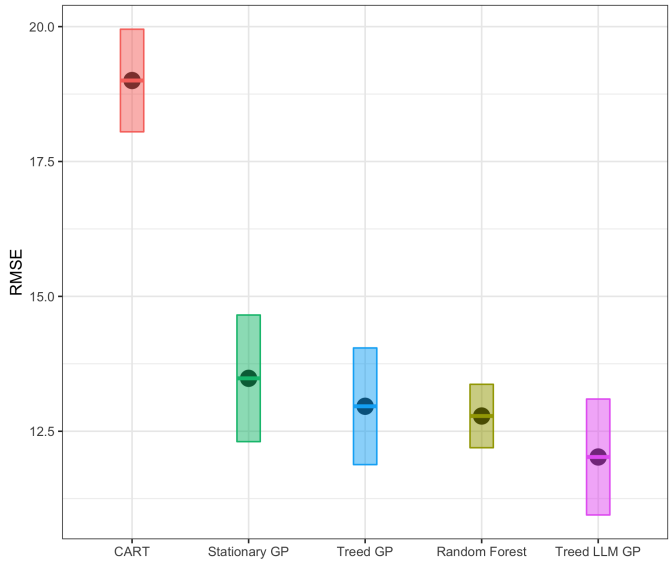

Figure 1: Summarizing root mean squared errors (RMSE) obtained with various models with the Arctic-Sea Ice computer model over 10-fold cross-validations (Mean ± SE)

Table 1: RMSE Summary

	CART	GP	Treed GP	Treed LLM GP	Random Forests
Mean	18.43	13.12	13.08	12.72	12.78
Median	19.01	12.78	12.65	11.31	13.15
5%	14.93	8.83	8.63	7.26	9.94
95%	22.75	18.07	17.40	15.94	15.14

Discussion

Limitations and Future Directions

Appendix

0.1 Code

```
# Libraries
library(tgp)
library(randomForest)
library(MASS)
library(e1071)
library(tidymodels)
library(ggplot2)
library(forcats)

# Reading in Wonderland data
input <- read.csv("ice_x.csv")
output <- read.csv("ice_y.csv")

# Keeping only ice velocity
x <- input
y <- output$IceVelocity

# Combining into one dataset
dat <- data.frame(cbind(x, y))

# Create cross-validation folds
set.seed(112)
folds  <- vfold_cv(dat, v = 10)

# For Bayesian CART model
cv_fit1 <- function(splits,...) {
  ZZ <- assessment(splits)$y
 fit1 <- bcart(X = analysis(splits)[,-14],
               Z = analysis(splits)$y, XX = assessment(splits)[,-14])
 rmse <- sqrt(mean((fit1$ZZ.mean - ZZ)^2))
  return(rmse)
}

# Fitting entire training set
fit_cart <- bcart(X = x, Z = y)

plot(fit_cart)


# For Bayesian stationary model
cv_fit2 <- function(splits,...) {
  ZZ <- assessment(splits)$y
 fit2 <- bgp(X = analysis(splits)[,-14],
               Z = analysis(splits)$y, XX = assessment(splits)[,-14],
               bprior = "b0", R = 2, corr = "exp")
 rmse <- sqrt(mean((fit2$ZZ.mean - ZZ)^2))
  return(rmse)
}

# Fitting stationary GP
fit_bgp <-  bgp(X = x, Z = y,  bprior = "b0", corr = "exp", R = 2, trace = TRUE)
```

```r
# For Bayesian treed GP model
cv_fit3 <- function(splits,...) {
  ZZ <- assessment(splits)$y
 fit3 <- btgp(X = analysis(splits)[,-14], Z = analysis(splits)$y, XX = assessment(splits)[,-14],
              bprior = "b0", R = 2)
  rmse <- sqrt(mean((fit3$ZZ.mean - ZZ)^2))
  return(rmse)
}

# For Bayesian treed GP with limiting linear model
cv_fit4 <- function(splits,...) {
  ZZ <- assessment(splits)$y
 fit4 <- btgpllm(X = analysis(splits)[,-14], Z = analysis(splits)$y,
                 XX = assessment(splits)[,-14], bprior = "b0", R = 2)
 rmse <- sqrt(mean((fit4$ZZ.mean - ZZ)^2))
  return(rmse)
}

# Running GP LLM model
# d[0/] means LLM is active
fit_llm <-  btgpllm(X = x, Z = y,  bprior = "b0", R = 2, trace = TRUE)

# MAP tree
png(filename = "map-tree.png", width = 16, height = 14, units = "cm", res = 300)
tgp.trees(fit_llm)
dev.off()

# A few plots
png(filename = "statgp.png", width = 20, height = 14, units = "cm", res = 300)
plot(fit_bgp, proj = c(3, 12))

dev.off()
png(filename = "treedgp.png", width = 20, height = 14, units = "cm", res = 300)
plot(fit_llm, proj = c(3, 12))
dev.off()

# Extracting RMSE's
res_cv_train <-
  folds %>%
  mutate(res_rmse1 = map(splits, .f = cv_fit1) ,
    res_rmse2 = map(splits, .f = cv_fit2),
         res_rmse3 = map(splits, .f = cv_fit3),
    res_rmse4 = map(splits, .f = cv_fit4))

# Mean of RMSE's
rmse_cart <- sqrt(mean((unlist(res_cv_train$res_rmse1))^2))
rmse_gp <- sqrt(mean((unlist(res_cv_train$res_rmse2))^2))
rmse_tgp <- sqrt(mean((unlist(res_cv_train$res_rmse3))^2))
rmse_tgpllm <- sqrt(mean((unlist(res_cv_train$res_rmse4))^2))
var_cart <- sqrt(var((unlist(res_cv_train$res_rmse1))))/sqrt(10)
var_gp <- sqrt(var((unlist(res_cv_train$res_rmse2))))/sqrt(10)
var_tgp <- sqrt(var((unlist(res_cv_train$res_rmse3))))/sqrt(10)
var_tgpllm <- sqrt(var((unlist(res_cv_train$res_rmse4))))/sqrt(10)


# CV varying mtry and ntree
ice.tune <-  tune.randomForest(y ~., data = dat, mtry = 1:13, ntree=100*1:8,
                                  tunecontrol = tune.control(sampling = "cross",cross=5))
```

```r
# Random forests
# Setting up model with mtry = 13 and ntree = 300
rf_mod <-
  rand_forest(mtry = 13, ntree = 300) %>%
  set_engine("ranger") %>%
  set_mode("regression")

# Running random forests
rf_fit <-
  rf_mod %>%
  fit(y ~ ., data = dat)

# Workflow for cross-validation
rf_wf <-
  workflow() %>%
  add_model(rf_mod) %>%
  add_formula(y ~ .)

# Fit on 10-folds
rf_fit_rs <-
  rf_wf %>%
  fit_resamples(folds)

# Use tune package to collect metrics
collect_metrics(rf_fit_rs)

# Plotting RMSE's
rmses <- c(rmse_cart, rmse_gp, rmse_tgp, collect_metrics(rf_fit_rs)$mean[1], rmse_tgpllm)
sds <- c(var_cart, var_gp, var_tgp, collect_metrics(rf_fit_rs)$std_err[1], var_tgpllm)
mod <- c("CART", "Stationary GP", "Treed GP", "Random Forest", "Treed LLM GP")
# Data framing
dat_plot <- data.frame(cbind(rmses, sds, mod))
# Convert class
dat_plot[, 1:2] <- lapply(dat_plot[, 1:2], as.numeric)
png(filename = "RMSE.png", width = 16, height = 14, units = "cm", res = 300)
ggplot(dat_plot, aes(x= mod, y= rmses)) +
  geom_dotplot(binaxis='y', stackdir='center') +
  geom_crossbar(aes(ymin=rmses-sds, ymax=rmses+sds, col = mod, fill = mod, alpha = 0.01), width=.2,
                position=position_dodge(.9), alpha = 0.5) + theme_bw() +
  theme(legend.position="none") + aes(x = fct_inorder(mod)) + labs(col = "", x = "", y = "RMSE")
dev.off()
#
```