Probabilistic weather forecast using the GOP method

The ProbForecastGOP package

Veronica J. Berrocal, Yulia Gel, Adrian E. Raftery, Tilmann Gneiting

January, 2005

Contents

1 Overview

The `ProbForecastGOP` package contains a main function, called `ProbForecastGOP`, and some other functions, to produce probabilistic weather forecasts using the Geostatistical Output Method (GOP) of Gel, Raftery, and Gneiting (2004). The GOP method consists of three steps:

- A first step in which the forecasts are regressed on the observations, and the empirical variogram of the forecast errors is computed;
- A second step in which a parametric geostatistical model is fitted to the empirical variogram and the parameters are estimated using Weighted Least Squares; and

- A third step in which ensemble members are generated by simulating realizations of the forecast errors random field using the estimated parametric geostatical model, and by adding the bias-corrected forecast to them.

Because the number of observations may be very large, the computing time needed to calculate the empirical variogram can be long. As a result, we have included in the package functions that can perform each step separately. So, for example, `Emp.variog` and `EmpDir.variog` compute, respectively, the empirical and the directional variogram, `Variog.fit` fits a parametric variogram model to the empirical variogram and estimates the parameters, and `Field.sim` simulates realizations of the weather random field. In addition, the main function `ProbForecastGOP` can be run so that either only the first or the first two steps of the GOP method are performed. Our experience supports using the separate functions approach, that is to compute the empirical variogram once, save the output in a file, fit a parametric variogram to it using different models and then, once the best fitting variogram model have been identified and its parameters have been estimated, simulate ensemble members. We have also included a wrapper function, `ProbForecastGOP`, that performs all the operations at once. The package also includes a plotting utility, `plotfields`, to plot weather random fields, and a function that computes the parametric variogram model at given distances, `linesmodel`, generally used to check visually the goodness of fit of a parametric variogram model.

Help files. As with any R package, detailed information on functions, their arguments and value can be obtained in the help files. For instance, to view the help file for the function `ProbForecastGOP` in a browser, type `?ProbForecastGOP`.

Note. In order to be able to run some of the functions included in the `ProbForecastGOP` package, the two `R` packages `fields` and `RandomFields` must be installed and loaded.

2 Using the package

In the following sections we will show how to use the `ProbForecastGOP` package on the data sets `phase1temp`, `forecast.grid`, `gridlat`, and `gridlong`.

All the data sets contain observed and/or 48-hr forecast temperature for the North-American Pacific Northwest as provided by the University of Washington MM5 Mesoscale Ensemble, initialized with the Aviation model of the National Weather Service. The data set `phase1temp` contains 48-hr forecast and observed temperatures during the period January-June 2000, `forecast.grid` contains 48-hr forecast temperatures on the 12-km domain initialized on January 10, 2002 at 00Z and verifying on January 12, 2002 at 00Z, and `gridlat` and `gridlong` contain, respectively, the latitudes and longitudes of the grid points in the 12-km domain.

2.1 Computing the empirical variogram

To load the 4 datasets, use `data(phase1temp)`, `data(forecast.grid)`, `data(gridlat)`, and `data(gridlong)`. For more information on the data, type `?phase1temp`, etc.

```
library(fields)
library(RandomFields)
library(ProbForecastGOP)
data(phase1temp)
day ← phase1temp$date.obs
obs ← phase1temp$obs
forecast ← phase1temp$forecast
id ← phase1temp$name.stat
coord1 ← phase1temp$lon.stat
coord2 ← phase1temp$lat.stat
data(forecast.grid)
forecast.grid ← forecast.grid$gridded.forecast
data(gridlat)
gridlat ← gridlat$gridded.lat
data(gridlong)
gridlong ← gridlong$gridded.long
```

To compute the (binned) empirical variogram, both the `Emp.variog` or the `ProbForecastGOP` function with the field `out` set equal to `"VARIOG"` can be used. The bins to be used in the empirical variogram computation can be either specified directly by entering a vector of cutpoints, or they can be determined so that there is approximately the same number of observations per bin.

If the `Emp.variog` is used, an empirical variogram with a specified number of bins is computed by typing the command:

```
empir.var ← Emp.variog(day,obs,forecast,id,coord1,coord2,
cut.points=NULL,max.dist=NULL,nbins=300)
```

A plot of the empirical variogram can be produced using the command:

```
plot(empir.var$bin.midpoints,empir.var$empir.variog,
xlab="Distance",ylab="Semi-variance")
```

Figure 1 shows such a plot.

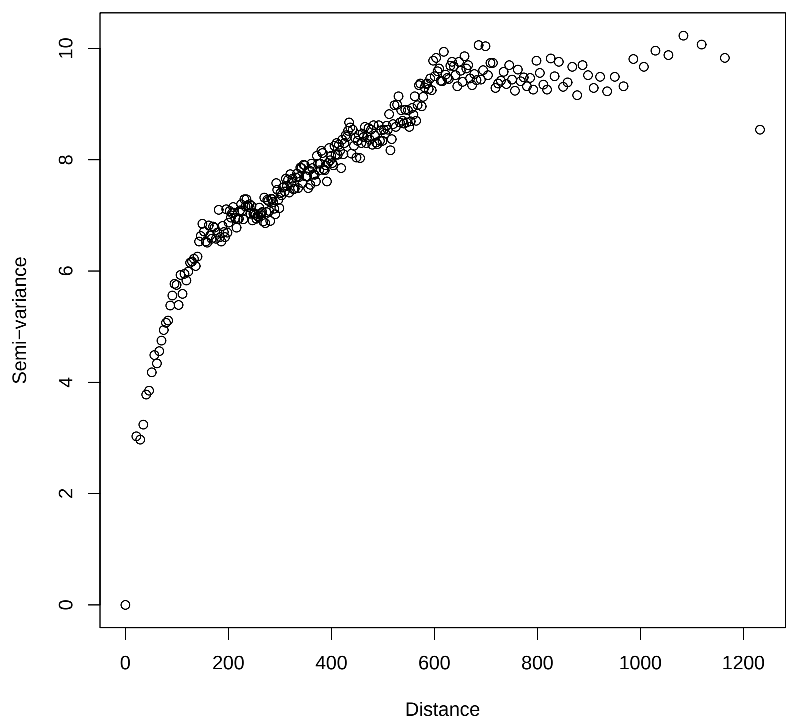

Figure 1: Empirical Spatial Variograms of Temperature Forecast Errors in the North American Pacific Northwest, January–June 2000.

If, on the other hand, the ProbForecastGOP wrapper function is used, an empirical variogram with a specified number of bins, is determined by typing:

```
empir.var ← ProbForecastGOP(day,obs,forecast,id,coord1,coord2,
cut.points=NULL,max.dist=NULL,nbins=300,variog.model=NULL,
max.dist.fit=NULL, init.val=NULL,fix.nugget=NULL,coord1.grid=gridlong,
coord2.grid=gridlat, forecast.grid=forecast.grid, n.sim=NULL,
out="VARIOG", n.displ=NULL, qt.displ=NULL)
```

The ProbForecastGOP function automatically gives a plot of the empirical variogram.
Empirical directional variograms can be computed using the EmpDir.variog function (Gel, Raftery, Gneiting, and Berrocal 2004). The function has the same usage as the Emp.variog function, with the only difference being that in this case, two tolerance angles and a direction must be entered. They determine a valid range for the angle between the locations. The EmpDir.variog function then computes the semi-variance only for those locations that yield an angle within the valid range. To produce an empirical directional variogram with a specified number of bins and to plot it, type the commands:

```
dir.var ← EmpDir.variog(day,obs,forecast,id,coord1,coord2,
tol.angle1=45,tol.angle2=135, cut.points=NULL,
max.dist=NULL,nbins=300,type="E")

plot(dir.var$bin.midpoints,dir.var$dir.variog,
xlab="Distance",ylab="Semi-variance")
```

Figure 2 shows both the East-West and North-South directional variograms.

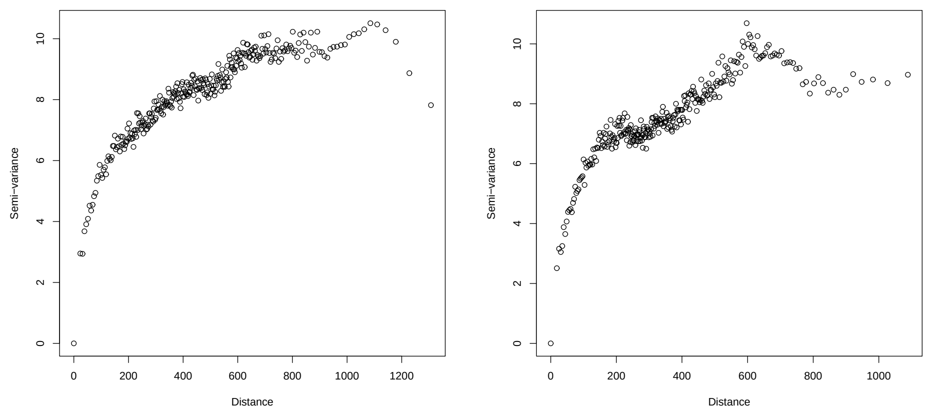

Figure 2: East-West and North-South Directional Empirical Spatial Variograms of Temperature Forecast Errors in the North American Pacific Northwest, January–June 2000.

Discrepancies among plots of the empirical variograms and the directional variograms are indications of anisotropy in the random field. The GOP method has been developed for the case of an isotropic weather random field, but it can be extended to anisotropic weather random fields; see for example Cressie (1993).

2.2 Fitting a variogram model

To fit a parametric variogram model to an empirical variogram model, either the function `Variog.fit`, or the function `ProbForecastGOP` with the setting `out="FIT"`, could be used. The parameters of the parametric variogram model are estimated by Weighted Least Squares using the `nlm` function in `R`. Initial values for the parameters can be provided, even though by default, initial values are determined automatically by the two functions, depending on the empirical variogram and on the parametric variogram model to be fitted. Choices for the parametric variogram model are: exponential, spherical, gaussian, generalized Cauchy and Whittle-Matern. The nugget effect can be kept fixed to some constant value or can be estimated iteratively along with the other parameters.
The following commands show how to use the `Variog.fit` function to fit an exponential variogram to the empirical variogram computed above with the `Emp.variog` functions:

```
param.var ← Variog.fit(emp.variog=empir.var,
variog.model="exponential",max.dist.fit=NULL,init.val=NULL,
fix.nugget="FALSE")
```

To plot the empirical variogram model with the estimated parametric variogram model superimposed, the function `linesmodel` can be used:

```
plot(empir.var$bin.midpoints,empir.var$empir.variog,
xlab="Distance",ylab="Semi-variance")

lines(empir.var$bin.midpoints,linesmodel(distance=
empir.var$bin.midpoints, variog.model="exponential",
param=c(param.var$nugget, param.var$variance,param.var$range)))
```

If the ProbForecastGOP function is used to fit a parametric variogram to an empirical variogram, the following command yields both estimates of the parameters and a plot of the empirical variogram with the parametric variogram superimposed:

```
param.var ← ProbForecastGOP(day,obs,forecast,id,coord1,coord2,
cut.points=NULL,max.dist=NULL,nbins=300,variog.model="exponential",
max.dist.fit=NULL,init.val=NULL,fix.nugget="FALSE",
coord1.grid=gridlong, coord2.grid=gridlat,
forecast.grid=forecast.grid, n.sim=NULL,out="FIT",
n.displ=NULL,qt.displ=NULL)
```

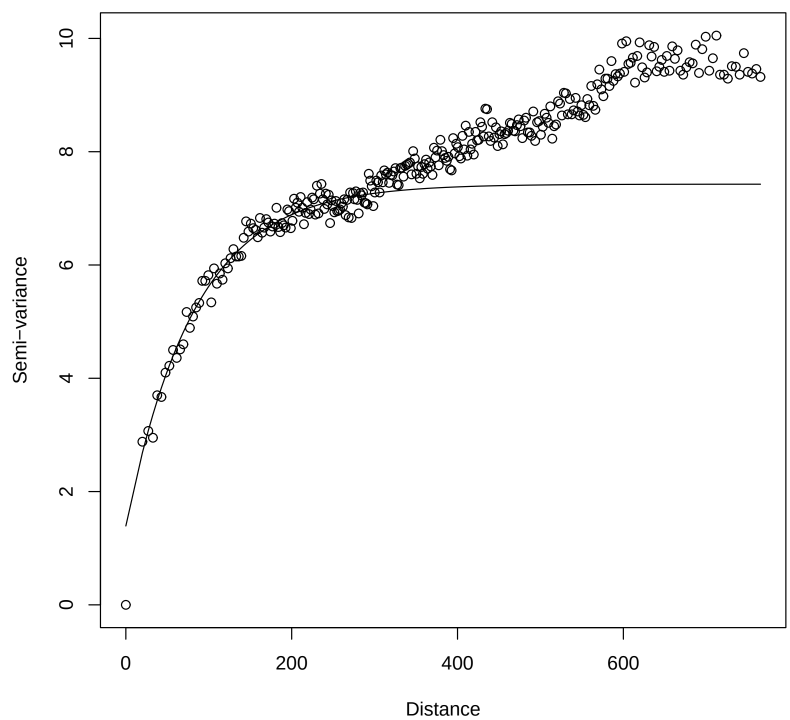

Figure 3: Empirical Spatial Variograms of Temperature Forecast Errors with the Fitted Exponential Variogram Model Superimposed.

2.3 Generating ensemble members

Realizations of the weather random field can be simulated using the two functions `Field.sim`, and `ProbForecastGOP` with the setting `out="SIM"`. To simulate ensemble members using the function `Field.sim`, the command is:

```
ens.forecast ← Field.sim(obs, forecast, coord1.grid=gridlong,
coord2.grid=gridlat, forecast.grid=forecast.grid,
variog.model="exponential", param.est=c(param.var$nugget,
param.var$variance,param.var$range), n.sim=99,
n.displ=4, qt.displ=c(10,50,90))
```

If instead the function `ProbForecastGOP` is used, realizations of the weather random field are obtained by typing the command:

```
ens.forecast ← ProbForecastGOP(day,obs,forecast,id,coord1,coord2,
cut.points=NULL,max.dist=NULL,nbins=300,variog.model="exponential",
max.dist.fit=NULL,init.val=NULL,fix.nugget="FALSE",
coord1.grid=gridlong, coord2.grid=gridlat,
forecast.grid=forecast.grid,n.sim=99,out="SIM",
n.displ=4,qt.displ=c(10,50,90))
```

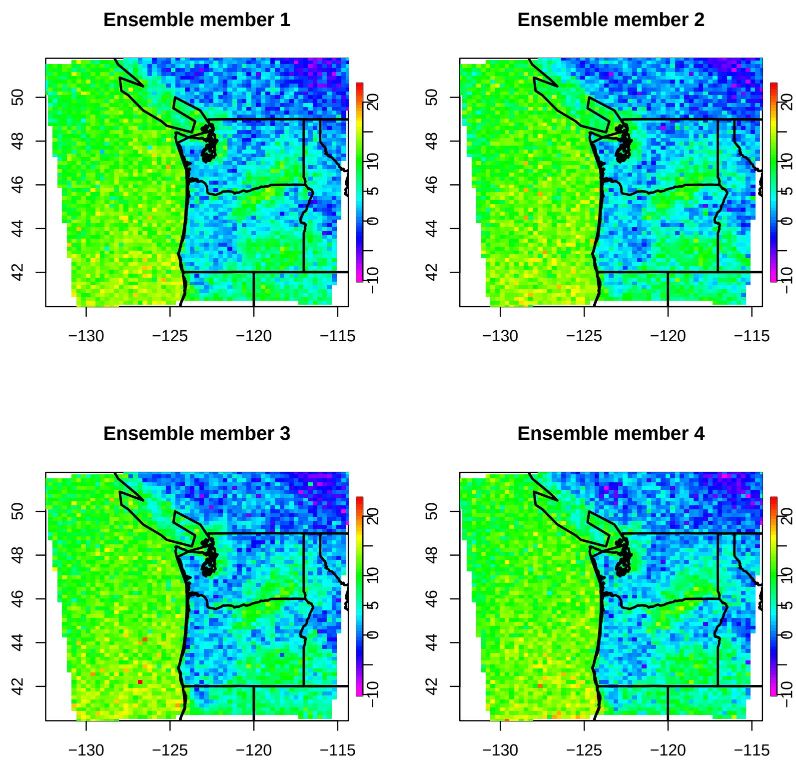

Figure 4: Ensemble of Simulated Forecasts for the Temperature on January 12, 2002.

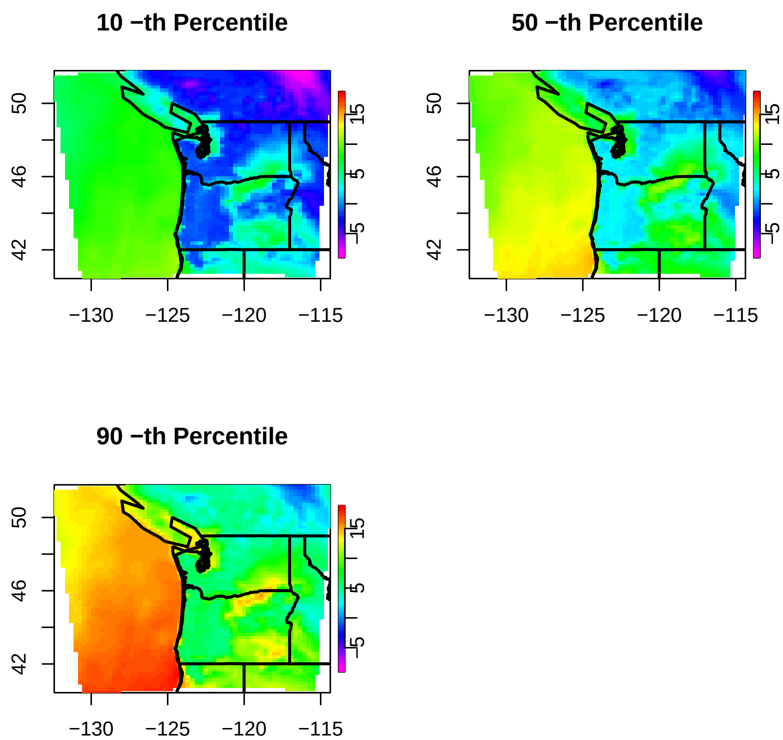

Figure 5: Tenth, Fiftieth and Ninetieth Percentiles of the Temperature Random Field on January 12, 2002.

Both functions simulate `n.sim` realizations of the weather random field and display `n.displ` of them together with the quantiles specified in the `qt.displ` field. However, the output obtained with the `ProbForecastGOP` function is more complete, since it also includes a plot of the empirical variogram with the estimated parametric variogram superimposed. On the other hand, computing time is much shorter with the `Field.sim` function, since the `ProbForecast` function with `out="SIM"` performs all the three steps of the GOP method, rather than only the last one. Notice that in order to be able to run both functions, the packages `fields` (Nychka 2004) and `RandomFields` citeSchlather2004 need to be installed. The weather random fields, in fact, are generated using the `GaussRF` function contained in the `RandomFields` package.

2.4 The plotting utility

The last function included in the package is a plotting utility to plot weather random fields. The function depends on the `fields` package (Nychka 2004), in particular on the functions `image.plot`, `US` and `world`. Here is an illustration of its usage:

```
field ← ens.forecast$sim.fields[,,1]

plotfields(field,x.lim=c(min(gridlong),max(gridlong)),
y.lim=c(min(gridlat),max(gridlat)),
title="Simulated weather field")
```

Figure 6 shows such the plot of a weather random field.

The contours in the plots in Figures 4, 5 and 6 were obtained by superimposing two maps available in `R`: a map of the world, that in this example provides the North American coastline, and a map of the US that gives the state boundaries. This is the approach used for plots of Weather Random Fields on the North American territory. For regions that do not overlap with the United States, plots of weather random fields are obtained by using the `R` built-in map of the world.

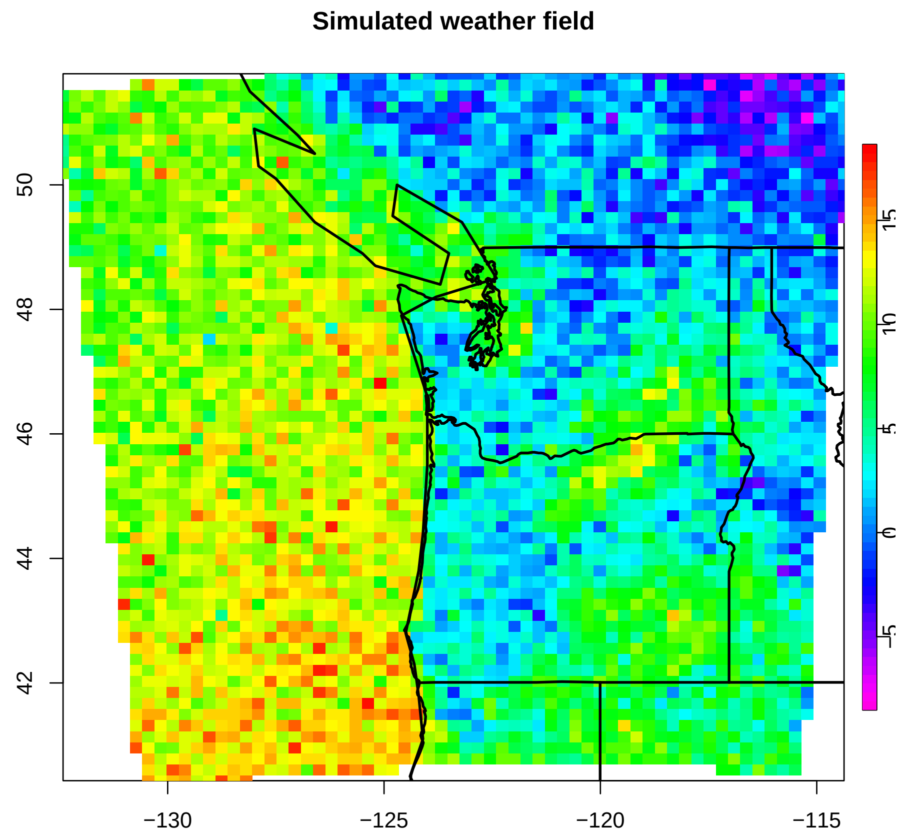

Figure 6: Plot of one member of the ensemble.

References

Cressie, N. A. C. (1993). *Statistics for Spatial Data* (revised ed.). New York: Wiley.

Gel, Y., A. E. Raftery, and T. Gneiting (2004). Calibrated mesoscale weather field forecasting: The Geostatistical Output Perturbation (GOP) method (with discussion). *Journal of the American Statistical Association 99*, 575-583.

Gel, Y., A. E. Raftery, T. Gneiting, and V. J. Berrocal (2004). Rejoinder. *Journal of the American Statistical Association 99*, 588-590.

Nychka, D. (2004). The `fields` package. http:\\cran.r-project.org\doc\packages\fields.pdf.

Schlather, M. (2004). The `RandomFields` package. http:\\cran.r-project.org\doc\packages\RandomFields.pdf.