Bayesian Dynamic Modeling for Space-time Data in R

Andrew O. Finley

April 17, 2017

We make use of several libraries in the following example session, including:

- `library(fields)`
- `library(geoR)`
- `library(MBA)`
- `library(spBayes)`

There are many different kinds of spatio-temporal data and extensive statistical literature that addresses most common settings. The approach adopted here applies to the setting where space is viewed as continuous, but time is taken to be discrete. Put another way, we view the data as a time series of spatial process realizations and work in the setting of dynamic models. Building upon previous work in the setting of dynamic models by West and Harrison (1997), several authors, including Tonellato (1997), Stroud et al. (2001) and Gelfand et al. (2005), proposed dynamic frameworks to model residual spatial and temporal dependence. These proposed frameworks are flexible and easily extended to accommodate nonstationary and multivariate outcomes.

1 Dynamic spatio-temporal models

Dynamic linear models, or state-space models, have gained tremendous popularity in recent years in fields as disparate as engineering, economics, genetics, and ecology. They offer a versatile framework for fitting several time-varying models (West and Harrison 1997). Gelfand et al. (2005) adapted the dynamic modeling framework to spatio-temporal models with spatially varying coefficients. Alternative adaptations of dynamic linear models to space-time data can be found in Stroud et al. (2001) and Tonellato (1997).

Here we consider a fairly basic, yet flexible, formulation. Suppose, $y_t(\boldsymbol{s})$ denotes the observation at location $\boldsymbol{s}$ and time t. We model $y_t(\boldsymbol{s})$ through a *measurement equation* that provides a regression specification with a space-time varying intercept and serially and spatially uncorrelated zero-centered Gaussian disturbances as measurement error $\epsilon_t(\boldsymbol{s})$. Next a *transition equation* introduces a $p \times 1$ coefficient vector, say $\boldsymbol{\beta}_t$, which is a purely temporal component (i.e., time-varying regression parameters), and a spatio-temporal component $u_t(\boldsymbol{s})$. Both these are generated through transition equations, capturing their Markovian dependence in time. While the transition equation of the purely temporal component is akin to usual state-space modeling, the spatio-temporal component is generated using Gaussian spatial processes. The overall model is written as

$$\begin{aligned} y_t(\boldsymbol{s}) &= \boldsymbol{x}_t(\boldsymbol{s})'\boldsymbol{\beta}_t + u_t(\boldsymbol{s}) + \epsilon_t(\boldsymbol{s}), \quad \epsilon_t(\boldsymbol{s}) \overset{ind.}{\sim} N(0, \tau_t^2)\ ; \\ \boldsymbol{\beta}_t &= \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \overset{i.i.d.}{\sim} N(0, \boldsymbol{\Sigma}_\eta)\ ; \\ u_t(\boldsymbol{s}) &= u_{t-1}(\boldsymbol{s}) + w_t(\boldsymbol{s}), \quad w_t(\boldsymbol{s}) \overset{ind.}{\sim} GP\left(\mathbf{0}, C_t(\cdot, \boldsymbol{\theta}_t)\right), \quad t = 1, 2, \ldots, N_t\ , \end{aligned} \tag{1}$$

where the abbreviations *ind.* and *i.i.d* are *independent* and *independent and identically distributed*, respectively. Here $\boldsymbol{x}_t(\boldsymbol{s})$ is a $p \times 1$ vector of predictors and $\boldsymbol{\beta}_t$ is a $p \times 1$ vector of coefficients. In addi-

tion to an intercept, $\boldsymbol{x}_t(\boldsymbol{s})$ can include location specific variables useful for explaining the variability in $y_t(\boldsymbol{s})$. The $GP(\mathbf{0}, C_t(\cdot, \boldsymbol{\theta}_t))$ denotes a spatial Gaussian process (a Gaussian process defined over an Euclidean spatial domain; see, e.g., Cressie 1993) with covariance function $C_t(\cdot; \boldsymbol{\theta}_t)$. We customarily specify $C_t(\boldsymbol{s}_1, \boldsymbol{s}_2; \boldsymbol{\theta}_t) = \sigma_t^2 \rho(\boldsymbol{s}_1, \boldsymbol{s}_2; \phi_t)$, where $\boldsymbol{\theta}_t = \{\sigma_t^2, \phi_t\}$ and $\rho(\cdot; \phi)$ is a *correlation function* with ϕ controlling the correlation decay and σ_t^2 represents the spatial variance component. An exponential function is often used to define the spatial correlation structure, e.g., $C_t(\boldsymbol{s}_1, \boldsymbol{s}_2; \boldsymbol{\theta}_t) = \sigma_t^2 \exp(-\phi_t \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|)$, where $\|\boldsymbol{s}_1 - \boldsymbol{s}_2\|$ is the Euclidean distance between the sites $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. However, any *valid* spatial correlation function could be used, see, e.g., Cressie 1993, Chilés and Delfiner 1999, and Banerjee et al. 2004. We further assume $\boldsymbol{\beta}_0 \sim N(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$ and $u_0(\boldsymbol{s}) \equiv 0$, which completes the prior specifications leading to a well-identified Bayesian hierarchical model and also yield reasonable dependence structures. In practice, estimation of model parameters are usually very robust to these hyper-prior specifications. Also note that (1) reduces to a simple spatial regression model for $t = 1$.

We consider settings where the inferential interest lies in spatial prediction or interpolation over a region for a set of discrete time points. We also assume that the same locations are monitored for each time point resulting in a space-time matrix whose rows index the locations and columns index the time points, i.e. the (i, j)-th element is $y_j(\boldsymbol{s}_i)$. Our algorithm will accommodate the situation where some cells of the space-time data matrix may have missing observations, as is common in monitoring environmental variables.

2 Data and computing

For this illustrative analysis, we consider monthly temperature data observed on a network of weather stations between January 2000 and December 2002. For brevity we only consider stations over a portion of New England. We also have elevation in meters for each station, which will serve as a covariate in the subsequent analysis. In the code below, `N.t` is the number of months and `n` is the number of observations per month.

```
> set.seed(1)
> data("NETemp.dat")
> ne.temp <- NETemp.dat
> ne.temp <- ne.temp[ne.temp[, "UTMX"] > 5500000 & ne.temp[, "UTMY"] >
+     3e+06, ]
> y.t <- ne.temp[, 4:27]
> N.t <- ncol(y.t)
> n <- nrow(y.t)
```

Here we set some observations to `NA` to illustrate the predictive ability of this modeling framework. The true values of the holdout observations are retained for subsequent comparison.

```
> miss <- sample(1:N.t, 10)
> holdout.station.id <- 5
> y.t.holdout <- y.t[holdout.station.id, miss]
> y.t[holdout.station.id, miss] <- NA
> coords <- as.matrix(ne.temp[, c("UTMX", "UTMY")]/1000)
> max.d <- max(iDist(coords))
> plot(coords, xlab = "Easting (km)", ylab = "Northin (km)")
```

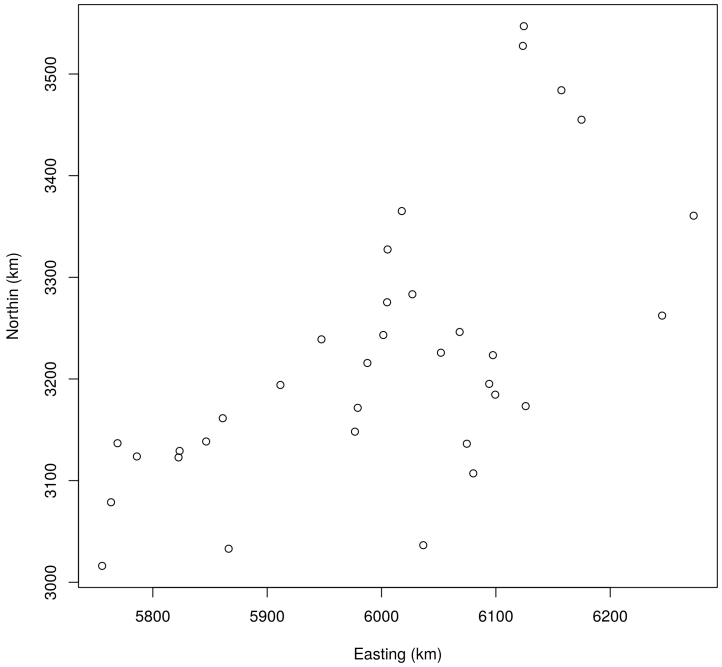

Figure 1: Weather station locations.

Next we need to specify priors for the numerous parameters. A variogram analysis for each time step is useful for guiding prior and hyperparameter specification. In the code below, `p` is the number of regression coefficients, i.e., corresponding to the intercept and elevation.

```
> p <- 2
> starting <- list(beta = rep(0, N.t * p), phi = rep(3/(0.5 * max.d),
+     N.t), sigma.sq = rep(2, N.t), tau.sq = rep(1, N.t), sigma.eta = diag(rep(0.01,
+     p)))
> tuning <- list(phi = rep(0.75, N.t))
> priors <- list(beta.0.Norm = list(rep(0, p), diag(1000, p)), phi.Unif = list(rep(3/(0.9 *
+     max.d), N.t), rep(3/(0.05 * max.d), N.t)), sigma.sq.IG = list(rep(2,
+     N.t), rep(10, N.t)), tau.sq.IG = list(rep(2, N.t), rep(5, N.t)),
+     sigma.eta.IW = list(2, diag(0.001, p)))
```

Like `spMvLM`, `spDynLM` will take a list of symbolic model formula. However, unlike `spMvLM`, each time step must have the same covariates, although their values can change over time.

```
> mods <- lapply(paste(colnames(y.t), "elev", sep = "~"), as.formula)
> n.samples <- 4000
> m.1 <- spDynLM(mods, data = cbind(y.t, ne.temp[, "elev", drop = FALSE]),
+     coords = coords, starting = starting, tuning = tuning, priors = priors,
+     get.fitted = TRUE, cov.model = "exponential", n.samples = n.samples,
+     n.report = 1000)

----------------------------------------
        General model description
```

```
----------------------------------------
Model fit with 34 observations in 24 time steps.

Number of missing observations 10.

Number of covariates 2 (including intercept if specified).

Using the exponential spatial correlation model.

Number of MCMC samples 4000.

Priors and hyperpriors:
        beta normal:
        m_0:        0.000        0.000
        Sigma_0:
        1000.000        0.000
        0.000        1000.000

        sigma.sq_t=1 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=1 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=1 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=2 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=2 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=2 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=3 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=3 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=3 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=4 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=4 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=4 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=5 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=5 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=5 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=6 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=6 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=6 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=7 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=7 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=7 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=8 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=8 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=8 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=9 IG hyperpriors shape=2.00000 and scale=10.00000
```

```
tau.sq_t=9 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=9 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=10 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=10 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=10 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=11 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=11 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=11 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=12 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=12 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=12 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=13 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=13 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=13 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=14 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=14 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=14 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=15 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=15 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=15 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=16 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=16 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=16 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=17 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=17 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=17 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=18 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=18 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=18 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=19 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=19 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=19 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=20 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=20 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=20 Unif hyperpriors a=0.00516 and b=0.09281
---
sigma.sq_t=21 IG hyperpriors shape=2.00000 and scale=10.00000
tau.sq_t=21 IG hyperpriors shape=2.00000 and scale=5.00000
phi_t=21 Unif hyperpriors a=0.00516 and b=0.09281
---
```

```
        sigma.sq_t=22 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=22 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=22 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=23 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=23 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=23 Unif hyperpriors a=0.00516 and b=0.09281
        ---
        sigma.sq_t=24 IG hyperpriors shape=2.00000 and scale=10.00000
        tau.sq_t=24 IG hyperpriors shape=2.00000 and scale=5.00000
        phi_t=24 Unif hyperpriors a=0.00516 and b=0.09281
        ---
-------------------------------------------------
                Sampling
-------------------------------------------------
Sampled: 999 of 4000, 24.98%
Report interval Mean Metrop. Acceptance rate: 77.58%
Overall Metrop. Acceptance rate: 77.66%
-------------------------------------------------
Sampled: 1999 of 4000, 49.98%
Report interval Mean Metrop. Acceptance rate: 77.81%
Overall Metrop. Acceptance rate: 77.73%
-------------------------------------------------
Sampled: 2999 of 4000, 74.97%
Report interval Mean Metrop. Acceptance rate: 77.85%
Overall Metrop. Acceptance rate: 77.77%
-------------------------------------------------
Sampled: 3999 of 4000, 99.97%
Report interval Mean Metrop. Acceptance rate: 77.30%
Overall Metrop. Acceptance rate: 77.65%
-------------------------------------------------
```

```
> burn.in <- floor(0.75 * n.samples)
> quant <- function(x) {
+     quantile(x, prob = c(0.5, 0.025, 0.975))
+ }
> beta <- apply(m.1$p.beta.samples[burn.in:n.samples, ], 2, quant)
> beta.0 <- beta[, grep("Intercept", colnames(beta))]
> beta.1 <- beta[, grep("elev", colnames(beta))]
> par(mfrow = c(2, 1))
> plot(1:N.t, beta.0[1, ], pch = 19, cex = 0.5, xlab = "months", ylab = "beta.0",
+     ylim = range(beta.0))
> arrows(1:N.t, beta.0[1, ], 1:N.t, beta.0[3, ], length = 0.02, angle = 90)
> arrows(1:N.t, beta.0[1, ], 1:N.t, beta.0[2, ], length = 0.02, angle = 90)
> plot(1:N.t, beta.1[1, ], pch = 19, cex = 0.5, xlab = "months", ylab = "beta.1",
+     ylim = range(beta.1))
> arrows(1:N.t, beta.1[1, ], 1:N.t, beta.1[3, ], length = 0.02, angle = 90)
> arrows(1:N.t, beta.1[1, ], 1:N.t, beta.1[2, ], length = 0.02, angle = 90)


> theta <- apply(m.1$p.theta.samples[burn.in:n.samples, ], 2, quant)
> sigma.sq <- theta[, grep("sigma.sq", colnames(theta))]
```

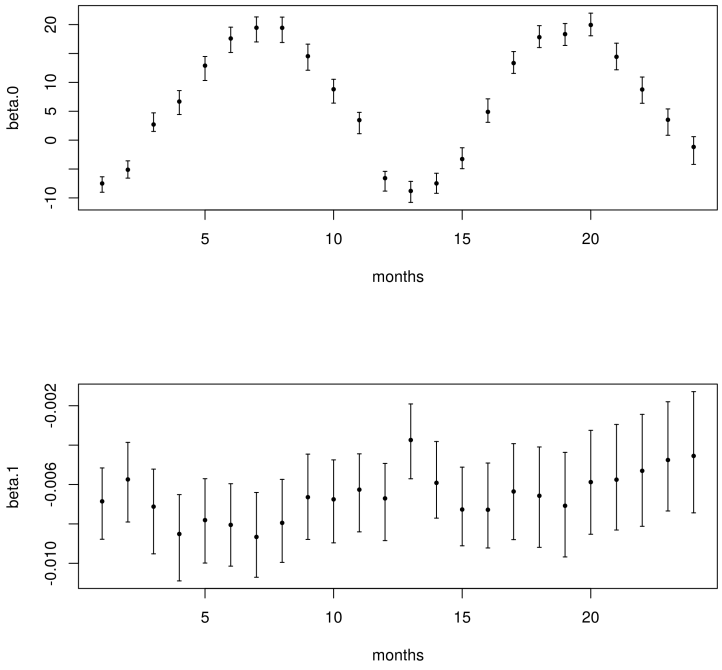

Figure 2: Evolution of the regression coefficients.

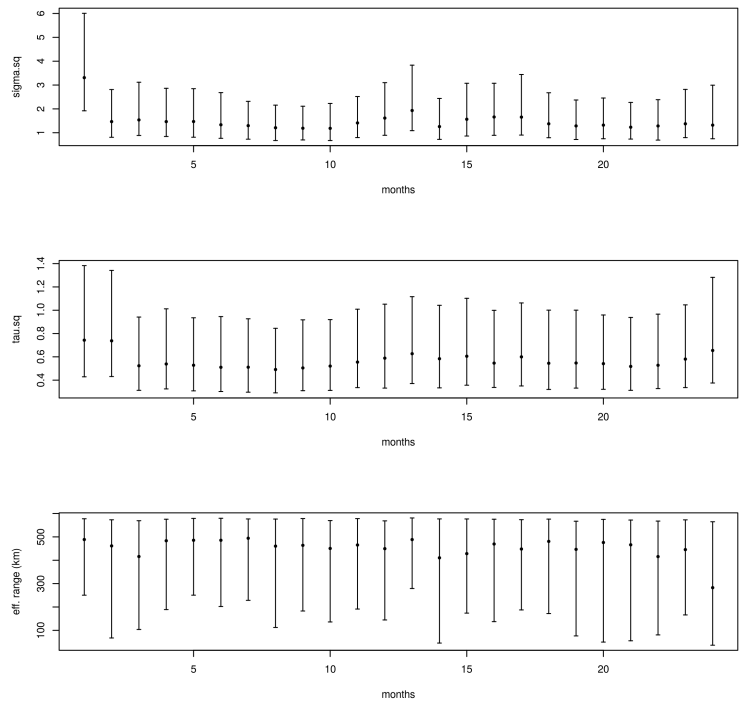

Figure 3: Evolution of the spatial covariance parameters.

```
> tau.sq <- theta[, grep("tau.sq", colnames(theta))]
> phi <- theta[, grep("phi", colnames(theta))]
> par(mfrow = c(3, 1))
> plot(1:N.t, sigma.sq[1, ], pch = 19, cex = 0.5, xlab = "months",
+     ylab = "sigma.sq", ylim = range(sigma.sq))
> arrows(1:N.t, sigma.sq[1, ], 1:N.t, sigma.sq[3, ], length = 0.02,
+     angle = 90)
> arrows(1:N.t, sigma.sq[1, ], 1:N.t, sigma.sq[2, ], length = 0.02,
+     angle = 90)
> plot(1:N.t, tau.sq[1, ], pch = 19, cex = 0.5, xlab = "months", ylab = "tau.sq",
+     ylim = range(tau.sq))
> arrows(1:N.t, tau.sq[1, ], 1:N.t, tau.sq[3, ], length = 0.02, angle = 90)
> arrows(1:N.t, tau.sq[1, ], 1:N.t, tau.sq[2, ], length = 0.02, angle = 90)
> plot(1:N.t, 3/phi[1, ], pch = 19, cex = 0.5, xlab = "months", ylab = "eff. range (km)",
+     ylim = range(3/phi))
> arrows(1:N.t, 3/phi[1, ], 1:N.t, 3/phi[3, ], length = 0.02, angle = 90)
> arrows(1:N.t, 3/phi[1, ], 1:N.t, 3/phi[2, ], length = 0.02, angle = 90)


> y.hat <- apply(m.1$p.y.samples[, burn.in:n.samples], 1, quant)
> y.hat.med <- matrix(y.hat[1, ], ncol = N.t)
> y.hat.up <- matrix(y.hat[3, ], ncol = N.t)
> y.hat.low <- matrix(y.hat[2, ], ncol = N.t)
> y.obs <- as.vector(as.matrix(y.t[-holdout.station.id, -miss]))
```

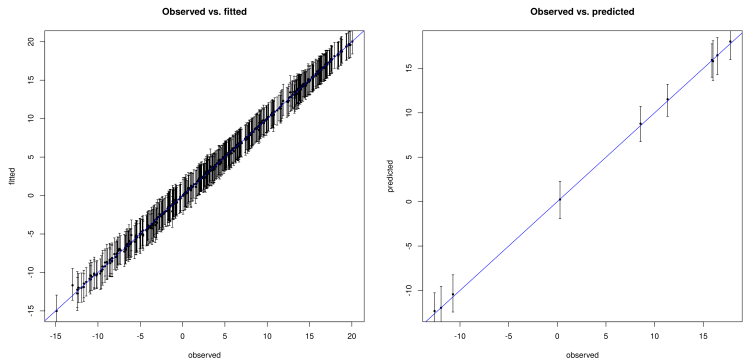

Figure 4: Fitted values (left) and holdout predicted values (right).

```
> y.obs.hat.med <- as.vector(y.hat.med[-holdout.station.id, -miss])
> y.obs.hat.up <- as.vector(y.hat.up[-holdout.station.id, -miss])
> y.obs.hat.low <- as.vector(y.hat.low[-holdout.station.id, -miss])
> y.ho <- as.matrix(y.t.holdout)
> y.ho.hat.med <- as.vector(y.hat.med[holdout.station.id, miss])
> y.ho.hat.up <- as.vector(y.hat.up[holdout.station.id, miss])
> y.ho.hat.low <- as.vector(y.hat.low[holdout.station.id, miss])
> par(mfrow = c(1, 2))
> plot(y.obs, y.obs.hat.med, pch = 19, cex = 0.5, xlab = "observed",
+     ylab = "fitted", main = "Observed vs. fitted")
> arrows(y.obs, y.obs.hat.med, y.obs, y.obs.hat.up, length = 0.02,
+     angle = 90)
> arrows(y.obs, y.obs.hat.med, y.obs, y.obs.hat.low, length = 0.02,
+     angle = 90)
> lines(-50:50, -50:50, col = "blue")
> plot(y.ho, y.ho.hat.med, pch = 19, cex = 0.5, xlab = "observed",
+     ylab = "predicted", , main = "Observed vs. predicted")
> arrows(y.ho, y.ho.hat.med, y.ho, y.ho.hat.up, length = 0.02, angle = 90)
> arrows(y.ho, y.ho.hat.med, y.ho, y.ho.hat.low, length = 0.02, angle = 90)
> lines(-50:50, -50:50, col = "blue")
```

3 References

Banerjee S, Carlin, BP, Gelfand AE (2004) *Hierarchical modeling and analysis for spatial data.* Chapman and Hall/CRC Press, Boca Raton, FL.

Chilés JP, Delfiner P (1999) *Geostatistics: modelling spatial uncertainty.* Wiley, New York.

Cressie NAC (1993) *Statistics for spatial data, 2nd edition.* Wiley, New York.

Gelfand AE, Banerjee S, Gamerman D (2005) Univariate and multivariate dynamic spatial modelling. *Environmetrics*, **16**:465–479

Stroud JR, Muller P, Sansó B (2001) Dynamic models for spatiotemporal data. *Journal of the Royal Statistical Society Series B*, **63**: 673–689

Tonellato S (1997) Bayesian dynamic linear models for spatial time series. *Tech report, Rapporto di riceria 5/1997*, Dipartimento di Statistica - Universita CaFoscari di Venezia, Venice Italy

West M, Harrison P (1997) *Bayesian forecasting and dynamic models.* 2nd edition. Springer, New York.