Doubly stochastic models for replicated spatio-temporal point processes

The model

Gervini (2019) proposes an additive model for the logarithm of the (unobservable) spatio-temporal intensity functions $\lambda_1(s,t), \dots, \lambda_n(s,t)$ underlying a replicated point process:

$$\ln \lambda_i(s,t) = \tau + Z + \ln \lambda_{Ti}(t) + \ln \lambda_{Si}(s),$$

$$\ln \lambda_{Ti}(t) = \mu(t) + \sum_{k=1}^{p} u_{ik}\varphi_k(t),$$

$$\ln \lambda_{Si}(s) = \nu(s) + \sum_{k=1}^{q} v_{ik}\psi_k(s),$$

where the φ_ks and the ψ_ks are orthonormal and the random effects z_is, u_{ik}s, and v_{ik}s are assumed jointly Normal. We explain in this note how to estimate the model parameters and interpret the results in the context of a bike-sharing data analysis.

The data

The data consist of starting times and destinations of bike trips that took place between April 1st and November 30 of 2016 at station #166 (located on Ashland & Wrightwood avenues) in the Divvy system of the city of Chicago. The data is available in the Matlab file `data_166day.mat` in this package.

The data is organized as two 244 x 1 cell arrays `x` and `y`, where, for each day `i`, `x{i}` is a vector of starting times (on a 24-hour time scale) and `y{i}` is a two-column matrix with the respective destinations (first column is longitude, second column is latitude). The number of rows of `x{i}` and `y{i}` are then equal, since they correspond to the same trips. The `x{i}`s and `y{i}`s are, in fact, paired observations, but we keep them in separate cell arrays for easier manipulation.

Fitting the model

Estimation is done by the program `STPP`. The temporal mean and components, $\mu(t)$ and $\varphi_k(t)$s, are modeled as spline functions using B-spline bases with equally spaced knots, so we first specify the basis parameters: the time domain, the number of knots, and the spline order. For these data the domain is [0,24]. We use cubic splines (order 4) and 10 equally space knots, so we set

```
>> basis_x = struct('rng',[0 24],'or',4,'nk',10);
```

The spatial mean and components, $\nu(s)$ and $\psi_k(s)$s, are modeled using a renormalized Gaussian kernel basis (`NGbasis.m`) with equally spaced centroids on an initial rectangle $I_1 \times I_2$ that contains the spatial domain of interest (we use 10 knots for each dimension, giving a total of 100 centroids). The actual domain of interest (a sector of the city of Chicago, in this case) is usually very irregular, so it is defined

through a logical function (`Rdom_chi.m`) that, given input `s` with longitude-latitude coordinates, returns 1 if `s` is inside the desired domain or 0 otherwise. Then the spatial basis is specified as follows:

```
>> basis_y = struct('I1',[-87.840,-87.530],'I2',[41.800,42.030],...
                    'Rdom',@Rdom_chi,'nk',10);
```

We are going to estimate a model with two temporal components and two spatial components, so we set `p1=2` and `p2=2`. We also need to specify smoothing parameters for the means and the components; in principle four different parameters could be used, since the scales are different, but for these data we found that 10^{-5} works well for all functional parameters, producing estimates that are neither too irregular nor too smooth. So we set

```
>> sm = struct('muX',1e-5,'pcX',1e-5,'muY',1e-5,'pcY',1e-5);
```

There is an option to specify periodic temporal mean and components; we use this option here, since it is natural to expect that the intensity functions "wrap around" the interval [0,24], so we set `per=1`. Finally, we set the maximum number of iterations at `itmax = 100` (note that estimation is done in a cumulative way, starting from a mean-only model and adding one component at a time, so several passes of at most `itmax` iterations each are carried out.)

The parameter estimators are then computed by calling

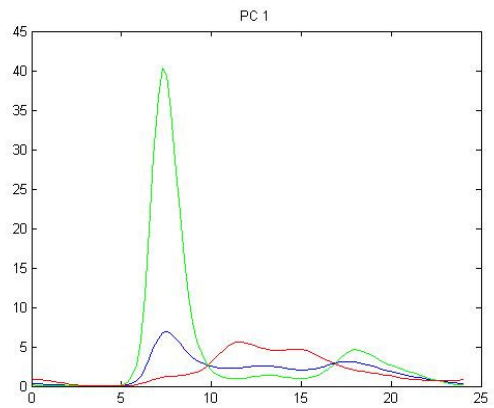
```
>> [param,effects,logf] = stpp(x,y,basis_x,basis_y,p1,p2,sm,per,itmax);
```

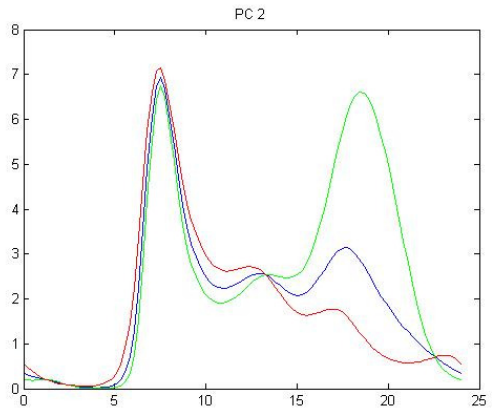
The output struct `param` contains the parameter estimators, `effects` the random-effect estimators, and `logf` the estimated log-density for each data point.

To visualize the mean and components we can use the functions `plot_pc_t.m` for the temporal parameters and `plot_pc_sp.m` for the spatial parameters:

```
>> plot_pc_t(param.c0,param.C,param.s2u,basis_x)
```

produces the plots



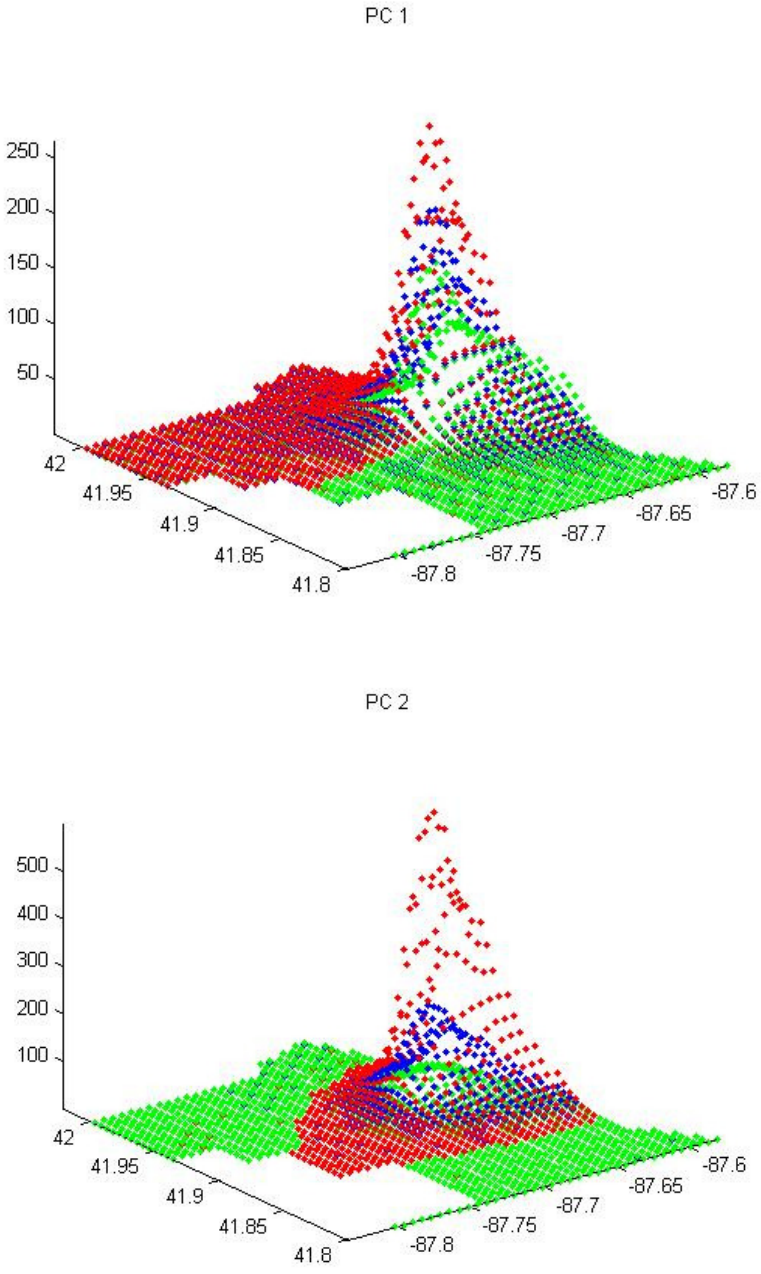


where the blue line is the estimated baseline intensity $\exp\{\mu(t)\}$, the green line is $\exp\{\mu(t) - c\varphi_k(t)\}$ and the red line is $\exp\{\mu(t) + c\varphi_k(t)\}$, for $c = 2\sigma_{uk}$, with $k = 1$ on the left and $k = 2$ on the right. We interpret, then, that the first component explains variation in the morning peak vs higher demand in the early afternoon, and the second component mostly explains variation in the evening peak.

Similarly, we plot the spatial components by calling

```
>> plot_pc_sp(param.d0,param.D,param.s2v,basis_y)
```

which produces the plots



Here the blue dots show the baseline spatial intensity $\exp\{\nu(s)\}$, the green dots $\exp\{\nu(s) - c\psi_k(s)\}$ and the red dots $\exp\{\nu(s) + c\psi_k(s)\}$, for $c = 2\sigma_{\nu k}$. We see that the type of variation explained by the first component is: trips mostly concentrated around the bike station vs more trips going downtown. The second component accounts for: more trips going far North or South vs more trips staying in the central part of the city.

The cross-correlations between the temporal component scores u_{ik}s and the spatial component scores v_{ik}s can be obtained as

```
>> diag(1./sqrt(param.s2u))*param.Suv*diag(1./sqrt(param.s2v))

ans =
      0.90234      0.31654
      -0.1182      0.10373
```

We see that the correlation between the first temporal component and the first spatial component is very high. The correlation between the first temporal component and the second spatial component is not as high, but probably statistically significant too.

To determine which correlations are statistically significant, we can use the asymptotic standard deviations of the estimators. The 'reduced' or 'simplified' asymptotic estimators, explained in the paper, are obtained by calling

```
>> V = AV_stpp_simp(x,y,basis_x,basis_y,param,effects);
>> n = length(x);
>> sd = sqrt(diag(V/n));
```

The vector `sd` contains the standard deviations for all model parameters except the functional basis coefficients. See the help of function `AV_stpp_simp.m` to find out the exact order in which the elements are given. We are only interested in the asymptotic standard deviations of the cross-covariances `param.Suv` here, which correspond to the indices

```
      >> i3 = p1+p2+1:p1+p2+p1*p2;
```

Then we obtain the asymptotic Z-scores as

```
>> z_Suv = param.Suv./reshape(sd(i3),[2 2])

z_Suv =
       4.6043       2.6893
     -0.62767      0.54915
```

We see that only the first row, the covariances of the first temporal component with both spatial components, is statistically significant.

References

- Gervini, D. (2019). Doubly stochastic models for replicated spatio-temporal point processes. *ArXiv 1903.09253.*