Particle Swarm Optimization for Spatial Design

Matthew Simpson[1]

Abstract

KEY WORDS:

[1] (to whom correspondence should be addressed) Department of Statistics, University of Missouri, 146 Middlebush Hall, Columbia, MO 65211-6100, themattsimpson@gmail.com

1 Particle swarm optimization

We briefly describe PSO here; refer to Blum and Li (2008) for an excellent introduction and Clerc (2010) for more detail. Suppose that we wish to maximize some objective function $Q(\boldsymbol{\theta}) : \Re^D \to \Re$. Let $i = 1, 2, \ldots, n$ index a set of particles over time, $t = 1, 2, \ldots, T$, where in every period each particle consists of a location $\boldsymbol{\theta}_i(t) \in \Re^D$, a velocity $\boldsymbol{v}_i(t) \in \Re^D$, a personal best location $\boldsymbol{p}_i(t) \in \Re^D$, and a group best location $\boldsymbol{g}_i(t) \in \Re^D$. Here we mean "best" in the sense of maximizing Q, so $Q(\boldsymbol{p}_i(t)) \geq Q(\boldsymbol{\theta}_i(s))$ for any $s \leq t$. The group best location is defined with respect to some neighborhood $\mathcal{N}_i$ of particle i; that is, $\boldsymbol{g}_i(t) = \arg\max_{\{\boldsymbol{p}_j(t) | j \in \mathcal{N}_i\}} Q(\boldsymbol{p}_j(t))$. In the simplest case where the entire swarm is the neighborhood of each particle, $\boldsymbol{g}_i(t) \equiv \boldsymbol{g}(t) = \arg\max_{\{\boldsymbol{p}_j(t) | j \in 1:n\}} Q(\boldsymbol{p}_j(t))$. The generic PSO algorithm updates as follows for each particle i:

$$
\begin{aligned}
\boldsymbol{v}_i(t+1) &= \omega \boldsymbol{v}_i(t) + \phi_1 \boldsymbol{r}_{1i}(t) \circ \{\boldsymbol{p}_i(t) - \boldsymbol{\theta}_i(t)\} + \phi_2 \boldsymbol{r}_{2i}(t) \circ \{\boldsymbol{g}_i(t) - \boldsymbol{\theta}_i(t)\}, \\
\boldsymbol{\theta}_i(t+1) &= \boldsymbol{\theta}_i(t) + \boldsymbol{v}_i(t+1), \\
\boldsymbol{p}_i(t+1) &= \begin{cases} \boldsymbol{p}_i(t) & \text{if } \ Q(\boldsymbol{p}_i(t)) \geq Q(\boldsymbol{\theta}_i(t+1)) \\ \boldsymbol{\theta}_i(t+1) & \text{otherwise,} \end{cases} \\
\boldsymbol{g}_i(t+1) &= \arg \max_{\{\boldsymbol{p}_j(t+1) | j \in \mathcal{N}_i\}} Q(\boldsymbol{p}_j(t+1)), \qquad (1)
\end{aligned}
$$

where $\circ$ denotes the Hadamard product (element-wise product), $\boldsymbol{r}_{1i}(t)$ and $\boldsymbol{r}_{2i}(t)$ are each vectors of D random variates independently generated from the $U(0,1)$ distribution, and $\omega > 0$, $\phi_1 > 0$, and $\phi_2 > 0$ are user-defined parameters. The term $\omega \boldsymbol{v}_i(t)$ controls the particle's tendency to keep moving in the direction it is already going, so ω is called the inertia parameter. For $\omega < 1$ velocities tend to decrease over time, while for $\omega > 1$ they tend to increase over time. Similarly $\phi_1 \boldsymbol{r}_{1i}(t) \circ \{\boldsymbol{p}_i(t) - \boldsymbol{\theta}_i(t)\}$ controls the particle's tendency to move towards its personal best location while $\phi_2 \boldsymbol{r}_{2i}(t) \circ \{\boldsymbol{g}_i(t) - \boldsymbol{\theta}_i(t)\}$ controls its tendency to move toward the group's best location, so ϕ_1 and ϕ_2 are called the cognitive correction factor and social correction factor, respectively (Blum and Li, 2008). This version of PSO is equivalent to Clerc and Kennedy (2002)'s constriction type I particle swarm. There are many variants of the PSO algorithm, often obtained through choosing (ω, ϕ_1, ϕ_2) in special ways or sometimes even dynamically. A default version of the algorithm sets $\omega = 0.7298$ and $\phi_1 = \phi_2 = 1.496$; see Clerc and Kennedy (2002) and Blum and Li (2008) for justification of these choices. Even when $\omega < 1$, if ϕ_1 and ϕ_2 are set high enough the velocities of the particles can continually increase and cause the swarm to make jumps that are much too large. A heavy handed way to solve this problem is by setting an upper bound on the velocity of any particle in any given direction, called velocity clamping. However, the default parameter values suggested by Clerc and Kennedy (2002) are also designed to prevent exactly this sort of velocity explosion.

Any PSO variant can also be combined with various neighborhood topologies that control how the particles communicate to each other. The default global topology allows each particle to see each other particle's previous best location for the social components of their respective velocity updates, but this can cause inadequate exploration and premature convergence. Alternative neighborhood topologies limit how many other particles each particle can communicate with. For example, particle 5 may only look at itself and particles 4 and 6 when determining what its group best location is. This allows information about high value locations in the domain of the objective function to eventually reach every particle in the swarm, but much more slowly so that each particle has an opportunity to explore the space more fully first. Appendix **??** contains a short description of the ring topologies, but there are many alternatives in the literature.[CITATION]

A variant of PSO, the bare bones PSO algorithm (BBPSO) is a PSO algorithm introduced by Kennedy (2003) that strips away the velocity term and removes the need for the user to choose parameters outside of the swarm size. Let $\theta_{ij}(t)$ denote the jth coordinate of the position for the ith particle in period t, and similarly for $p_{ij}(t)$ and $g_{ij}(t)$. Then the BBPSO algorithm obtains a new position coordinate θ_{ij} via

$$\theta_{ij}(t+1) \sim N\left(\frac{p_{ij}(t)+g_{ij}(t)}{2}, |p_{ij}(t)-g_{ij}(t)|^2\right) \tag{2}$$

where $N(\mu, \sigma^2)$ is the normal distribution with mean μ and standard deviation σ^2. The updates of $\boldsymbol{p}_i(t)$ and $\boldsymbol{g}_i(t)$ are the same as in (1). There are several variants of this algorithm proposed including using distributions from different location-scale families — e.g., using the t-distribution and modifying the location or the scale parameters, for example Krohling et al. (2009), Hsieh and Lee (2010), Richer and Blackwell (2006), and Campos et al. (2014). Appendix **??** contains more detail on some of thees BBPSO variants.

1.1 Adaptively tuned BBPSO

BBPSO adapts the size effective search space of the swarm over time through the variance term, $|p_{ij}(t)-g_{ij}(t)|^2$. As the personal best locations of the swarm move closer together, these variances decrease and the swarm tries locations which are closer to known high value areas in the space. This behavior is desirable, but the adaptation is forced to occur through one channel: the personal and group best locations. If the personal best locations of the swarm are arranged in a rough ring around the global optimum, smaller variances are desirable so that the new particle locations have a tendency to be in the center of the ring. On the other hand, if the personal best locations of the swarm are all to one side of the optimum and fairly far away, larger variances are desirable so that the particles can quickly approach the neighborhood of the optimum. BBPSO cannot distinguish between these two cases. [CAN ADD A GRAPHIC ILLUSTRATING THIS. WORTH IT?]

In order to allow it to adapt in a more flexible manner, we modify the BBPSO variance to $\sigma^2(t)|p_{ij}(t)-g_{ij}(t)|^2$ and tune $\sigma^2(t)$ in a manner similar to adaptive random walk Metropolis MCMC algorithms (Andrieu and Thoms, 2008). Define the improvement rate of the swarm in period t as $R(t) = \#\{i : Q(\boldsymbol{p}_i(t)) > Q(\boldsymbol{p}_i(t-1))\}/n$ where if A is a set then $\#A$ is the number of members of that set, and let R^* denote the target improvement rate. Then what we term adaptively tuned BBPSO (AT-BBPSO) updates personal best and group best locations as in (1), then updates particle locations as follows:

$$\begin{aligned} \theta_{ij}(t+1) &\sim t_{df}\left(\frac{p_{ij}(t)+g_{ij}(t)}{2}, \sigma^2(t)|p_{ij}(t)-g_{ij}(t)|^2\right), \\ \log\sigma^2(t+1) &= \log\sigma^2(t) + c\times\{R(t+1)-R^*\}, \end{aligned} \tag{3}$$

where *df* is a user chosen degrees of freedom parameter and c is another user chosen parameter that controls the speed of adaptation. We use a t kernel instead of a Gaussian in order to allow for more flexibility, and smaller values of *df* appear to combine well with adaptively tuning $\sigma^2(t)$ — in particular $df = 1$. Since the target rate in AT-BBPSO is similar to the target rate in an adaptive random walk Metropolis algorithm, a priori values around 0.25 and in particular below 0.5 seem reasonable (Gelman et al., 1996). In practice we find that values from $R^* = 0.3$ to $R^* = 0.5$ tend to yield good AT-BBPSO algorithms. The parameter c controls the speed of adaptation so that larger values of c mean the algorithm adapts $\sigma^2(t)$ faster. We find that $c = 0.1$ to be a good value, though anything within an order of magnitude often yields similar results. The initial value $\sigma^2(0)$ also needs to be chosen, though this does not have much impact on the algorithm as long as c is not too small. We use $\sigma^2(0) = 1$ to initialize the algorithm at the standard BBPSO algorithm.

Both using a t kernel and adding a fixed scale parameter have been discussed in the BBPSO literature, but as Kennedy (2003) notes, something about setting $\sigma = 1$ is special that causes the algorithm to work well. Another similar BBPSO algorithm in the literature comes from Hsieh and Lee (2010). They propose a modified version of BBPSO with

$$\theta_{ij}(t+1) \sim N\left(\omega\frac{p_{ij}(t)+g_{ij}(t)}{2}, \sigma^2|p_{ij}(t)-g_{ij}(t)|^2\right),$$

where $\omega \leq 1$ and $\sigma^2 \leq 1$ are constriction parameters that are eventually both set to one after enough iterations of the algorithm. The authors suggest dynamically adjusting the constriction parameters in the early stage of the algorithm before they are set to one, but give no suggestion for how to do this. The default BBPSO algorithm essentially sets $\sigma^2(t) = 1$ for all t and Hsieh and Lee (2010)'s algorithm deterministically adjusts $\sigma^2(t)$, but our AT-BBPSO algorithm is able to adapt its value on the fly based on local knowledge about the objective function. If too much of the swarm is failing to find new personal best locations, AT-BBPSO proposes new locations closer to known high value areas. If too much of the swarm is improving, AT-BBPSO proposes bolder locations in an effort to make larger improvements. This ability to adapt to local information about the objective function allows AT-BBPSO to more quickly traverse the search space towards the global optimum, though by using local information AT-BBPSO does risk premature convergence to a local optimum. The adaptively tuned component of AT-BBPSO can also be combined with most BBPSO variants, some of which are outlined in Appendix **??**. In Appendix **??** we conduct a simulation study on several test functions that compares AT-BBPSO variants to other PSO and BBPSO variants in order to justify the parameter settings discussed above and demonstrate AT-BBPSO variants are attractive PSO algorithms.

1.2 Adaptively tuned PSO

In AT-BBPSO variants, the parameter $\sigma(t)$ partially controls the effective size of the swarm's search area, and we increase or decrease $\sigma(t)$ and consequently the search area depending on how much of the swarm is finding new personal best locations. In standard PSO there is no direct analogue to $\sigma(t)$, though the inertia parameter, denoted by ω in (1), is related. It controls the effective size of the swarm's search area by controlling how the magnitude of the velocities evolve over time — larger values of ω allow for larger magnitude velocities in future periods. In AT-BBPSO we use an analogy with tuning a random walk Metropolis-Hastings MCMC algorithm in order to build intuition about how to tune $\sigma(t)$. The analogy is much weaker in this case; nonetheless, we allow $\omega(t)$ to be time-varying and use the same mechanism in order to tune it as we did for $\sigma(t)$.

The idea of time-varying $\omega(t)$ has been in the PSO literature for some time. An early suggestion was to set $\omega(0) = 0.9$ and deterministically decrease it until it reaches $\omega(T) = 0.4$ after the maximum number of iterations allowed (Eberhart and Shi, 2000). In particular, Tuppadung and Kurutach (2011) suggest defining $\omega(t)$ via the parameterized inertia weight function

$$\omega(t) = \frac{1}{1 + \left(\frac{t}{\alpha}\right)^{\beta}} \tag{4}$$

where α and β are user-defined parameters. Roughly, α controls how low $\omega(t)$ can go and β controls how fast it gets there, so α and β can be thought of as intercept and slope parameters respectively. The suggestion in Tuppadung and Kurutach (2011) is to set α to a small fraction of the total amount of iterations in which the algorithm is allowed to run (e.g., 10% or 20%), and set β between one and four. We call this type of PSO algorithm deterministic inertia PSO (DI-PSO).

DI-PSO tends to improve on standard PSO if $\omega(t)$'s progression is set appropriately, but it invariably makes using PSO more difficult for the average user. Additionally, depending on the problem, it may be more useful to let the swarm explore the space for more or less iterations, necessitating different progressions of $\omega(t)$. A priori it may not be clear exactly which approach is best for any given problem, so an automatic method is desirable. adaptively tuned PSO (AT-PSO) is just that — it provides an automatic method to adjust the value of $\omega(t)$ depending on local information obtained by the particle swarm. Formally, AT-PSO updates personal and group best locations as in (1) and updates $\omega(t)$ and particle locations as follows:

$$\begin{aligned}
\boldsymbol{v}_i(t+1) &= \omega(t+1)\boldsymbol{v}_i(t) + \phi_1 \boldsymbol{r}_{1i}(t) \circ \{\boldsymbol{p}_i(t) - \boldsymbol{\theta}_i(t)\} + \phi_2 \boldsymbol{r}_{2i}(t) \circ \{\boldsymbol{g}_i(t) - \boldsymbol{\theta}_i(t)\}, \\
\boldsymbol{\theta}_i(t+1) &= \boldsymbol{\theta}_i(t) + \boldsymbol{v}_i(t+1), \\
\log\omega(t+1) &= \log\omega(t) + c \times \{R(t+1) - R^*\}, \qquad (5)
\end{aligned}$$

where $R(t)$ is the improvement rate of the swarm in iteration t, R^* is the target improvement rate, and c controls how much $R(t)$ changes on a per iteration basis. For AT-BBPSO we used an analogy with random walk Metropolis-Hastings algorithms to suggest that a good value for the target improvement rate is smaller than 0.5 but not too small and this turned out to be correct. The analogy does not apply as well here, though we still find in Appendix **??** that $R^* = 0.3$ or 0.5 still seems to work well for AT-PSO. The value of c controls the speed of adaptation, and in particular if c is small and $\omega(0)$ is large, AT-PSO can mimic DI-PSO to some extent. This turns out to produce poor AT-PSO algorithms, however. We use $c = 0.1$ as a default value and in simulations not reported here, we find that the gains from optimizing c appear to be small. However, very small values like those suggested by an attempt to mimic DI-PSO turn out to cause the algorithm to perform very poorly.

A major strength of AT-PSO relative to DI-PSO and standard PSO is that AT-PSO can increase $\omega(t)$ when information from the swarm suggests there is an unexplored high value region of the space — when too much of the swarm is improving on their personal best locations AT-PSO increases $\omega(t)$ until velocities start increasing, the swarm starts exploring a larger amount of the nearby space, and more of the particles fail to find improvements on their personal best. Just like in AT-BBPSO, this mechanism provides a way for the swarm to adapt its behavior on the fly based on local conditions and speed up convergence by allowing the particles that do improve to make larger improvements, but it can also cause premature convergence to a local optimum. While DI-PSO monotonically decreases $\omega(t)$ toward some minimum value, AT-PSO typically oscillates $\omega(t)$ so that the algorithm alternates between exploring and exploiting more, relative to standard PSO. Appendix **??** contains an extended simulation study comparing a variety of these PSO and BBPSO algorithms on a suite of test functions that demonstrates some of the behavior detailed above and shows that AT-PSO is an attractive PSO algorithm.

2 The Spatial Design Problem

Suppose we are interested in the latent spatial field of some response variable $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subseteq \Re^2$. Specifically, we are interested in predicting $Y(\boldsymbol{u})$ at a set of target locations $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_{N_t} \in \mathcal{D}$. We have the ability to sample N locations anywhere in $\mathcal{D}$, and we wish to place them in order to optimize the amount of information, in some sense, that we learn about the latent spatial field. This is a bit naieve since typically the sampled points represent fixed monitoring stations and the question is where to put new monitoring stations. Let $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{N_s} \in \mathcal{D}$ denote the N_s fixed sampling locations and let $\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{N_d} \in \mathcal{D}$ denote the N_d new sampling locations, A.K.A. the design points. Suppose that $Y(\boldsymbol{u})$ is a geostatistical process with mean function $\mu(\boldsymbol{u}) = \boldsymbol{x}(\boldsymbol{u})'\boldsymbol{\beta}$ for some covariate $\boldsymbol{x}(\boldsymbol{u})$ known at every point in $\mathcal{D}$ and some covariance function $C(\boldsymbol{u}, \boldsymbol{v})$ for $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{D}$. Not all covariates can be known a priori at every point in the spatial domain, but e.g. covariates that are known functions of the location satisfy this constraint. Once the design points are selected, we observe $Z(\boldsymbol{d}_i)$ for $i = 1, 2, \ldots, N_d$ and $Z(\boldsymbol{s}_i)$ for $i = 1, 2, \ldots, N_s$ where $Z(\boldsymbol{u}) = Y(\boldsymbol{u}) + \varepsilon(\boldsymbol{u})$ and $\varepsilon(\boldsymbol{u})$ is mean zero white noise with variance σ^2_ε, representing measurement error. Typically $\boldsymbol{\beta}$, σ^2_ε, and $C(.,.)$ are unknown and must be estimated, though for now we will treat them as known perhaps through estimation with previous measurements at the fixed sampling locations.

To completely specify the problem we need to define an informative design criterion. Intuitively, we can compute the mean square prediction error (MSPE), i.e. the kriging variance, at each of the target locations and optimize some function of these variances. Common choices are to minimize the mean kriging variance or the maximum kriging variance over all target locations. These criteria are naive since they they ignore parameter uncertainty, and taking that into account will often change the optimal design (Zimmerman, 2006). For now we will treat only the latent spatial field as unknown and discuss minimizing both mean and maximum kriging variance.

2.1 Simple Kriging

In simple kriging, $C(.,.)$, $\boldsymbol{\beta}$, and σ^2_ε are all treated as known. Let $\boldsymbol{Z} = [Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_2), \ldots, Z(\boldsymbol{s}_{N_s}), Z(\boldsymbol{d}_1), Z(\boldsymbol{d}$ $\boldsymbol{X} = [\boldsymbol{x}(\boldsymbol{s}_1), \boldsymbol{x}(\boldsymbol{s}_2), \ldots, \boldsymbol{x}(\boldsymbol{s}_{N_s}), \boldsymbol{x}(\boldsymbol{d}_1), \boldsymbol{x}(\boldsymbol{d}_2), \ldots, \boldsymbol{x}(\boldsymbol{d}_{N_d})]'$, $\boldsymbol{C}_Z = \mathrm{cov}(\boldsymbol{Z})$ where $\mathrm{cov}[Z(\boldsymbol{u}), Z(\boldsymbol{v})] = C(\boldsymbol{u}, \boldsymbol{v}) + \sigma^2_\varepsilon 1(\boldsymbol{u} = \boldsymbol{v})$, and $\boldsymbol{c}_Y(\boldsymbol{t}_i) = \mathrm{cov}[Y(\boldsymbol{t}_i), \boldsymbol{Z}]$ where $\mathrm{cov}[Y(\boldsymbol{t}_i), Z(\boldsymbol{u})] = C(\boldsymbol{t}_0, \boldsymbol{u})$. Then the simple kriging predictor of $Y(\boldsymbol{t}_i)$ is (Cressie and Wikle, 2015, Section 4.1.2)

$$\widehat{Y}_{sk}(\boldsymbol{t}_i; \boldsymbol{d}) = \boldsymbol{x}(\boldsymbol{s}_i)'\boldsymbol{\beta} + \boldsymbol{c}_Y(\boldsymbol{t}_i; \boldsymbol{d})'\boldsymbol{C}_Z^{-1}(\boldsymbol{d})[\boldsymbol{Z}(\boldsymbol{d}) - \boldsymbol{X}(\boldsymbol{d})\boldsymbol{\beta}]$$

with MSPE

$$\sigma^2_{sk}(\boldsymbol{t}_i; \boldsymbol{d}) = C(\boldsymbol{t}_i, \boldsymbol{t}_i) - \boldsymbol{c}_Y(\boldsymbol{t}_i; \boldsymbol{d})'\boldsymbol{C}_Z^{-1}(\boldsymbol{d})\boldsymbol{c}_Y(\boldsymbol{t}_i; \boldsymbol{d}).$$

The simple kriging MSPE a function of the design points, $\boldsymbol{d} = (\boldsymbol{d}_1', \boldsymbol{d}_2', \ldots, \boldsymbol{d}_{N_d}')'$, through $\boldsymbol{c}_Y(\boldsymbol{t}_i)$ and $\boldsymbol{C}_Z^{-1}$, and we have made this explicit in the above equations. Then the mean kriging variance is given by

$$\overline{Q}_{sk}(\boldsymbol{d}) = \frac{1}{N_t}\sum_{i}^{N_t} \sigma^2_{sk}(\boldsymbol{t}_i; \boldsymbol{d})$$

while the maximum kriging variance is given by

$$Q^*_{sk}(\boldsymbol{d}) = \max_{i=1,2\ldots,N_t} \sigma^2_{sk}(\boldsymbol{t}_i; \boldsymbol{d}).$$

2.2 Universal Kriging

A major limitation of simple kriging is that typically $\boldsymbol{\beta}$, σ^2_ε, and $C(.,.)$ are unknown. Universal kriging attempts to remedy this to some extent by allowing $\boldsymbol{\beta}$ to be unknown. The universal kriging predictor of $Y(\boldsymbol{t}_i)$ is (Cressie and Wikle, 2015, Section 4.1.2)

$$\widehat{Y}_{uk}(\boldsymbol{t}_i;\boldsymbol{d}) = \boldsymbol{x}(\boldsymbol{t}_i)'\widehat{\boldsymbol{\beta}}_{gls} + \boldsymbol{c}_Y(\boldsymbol{t}_i)'\boldsymbol{C}_Z^{-1}(\boldsymbol{Z} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}_{gls})$$

where $\widehat{\boldsymbol{\beta}}_{gls} = [\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X}]^{-1}\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{Z}$ is the generalized least squares estimate of $\boldsymbol{\beta}$, and the MSPE of $\widehat{Y}_{uk}(\boldsymbol{t}_i)$ is

$$\begin{aligned}\sigma^2_{uk}(\boldsymbol{t}_i;\boldsymbol{d}) &= C(\boldsymbol{t}_i,\boldsymbol{t}_i) - \boldsymbol{c}_Y(\boldsymbol{t}_i)'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{t}_i)\\ &\qquad + [\boldsymbol{x}(\boldsymbol{t}_i) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{t}_i)]'[\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X}]^{-1}[\boldsymbol{x}(\boldsymbol{t}_i) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{t}_i)].\end{aligned}$$

To avoid clutter we drop the explicit dependence on $\boldsymbol{d}$ in these equations, but $\boldsymbol{c}_Y(\boldsymbol{t}_i)$, $\boldsymbol{C}_Z$, $\boldsymbol{Z}$, $\boldsymbol{X}$, and $\widehat{\boldsymbol{\beta}}_{gls}$ all depend on $\boldsymbol{d}$. The mean universal kriging variance is given by

$$\overline{Q}_{uk}(\boldsymbol{d}) = \frac{1}{N_t}\sum_{i}^{N_t}\sigma^2_{uk}(\boldsymbol{t}_i;\boldsymbol{d})$$

while the maximum universal kriging variance is given by

$$Q^*_{uk}(\boldsymbol{d}) = \max_{i=1,2\ldots,N_t}\sigma^2_{uk}(\boldsymbol{t}_i;\boldsymbol{d}).$$

In this context, Zimmerman (2006) finds that the optimal design is highly dependent on the class of mean function of the geostatistical process. In practice we are often interested in predicting at the entire spatial domain rather than a finite set of target locations. This changes the mean and maximum kriging variances to

$$\overline{Q}_{uk}(\boldsymbol{d}) = \frac{1}{|\mathcal{D}|}\int_{\mathcal{D}}\sigma^2_{sk}(\boldsymbol{d};\boldsymbol{d})d\boldsymbol{u}$$

and

$$Q^*_{uk}(\boldsymbol{d}) = \max_{\boldsymbol{u}\in\mathcal{D}}\sigma^2_{sk}(\boldsymbol{u};\boldsymbol{d}).$$

respectively. We can approximate both of these with a large but finite sample of target locations from $\mathcal{D}$, though if the sample is too small some design critera may favor points directly on top of the sampled locations. In our experience this is more problematic for simple kriging.

2.3 Empirical Kriging

In both simple and universal kriging we assume that σ^2_ε and $C(.,.)$ are known, though this is not always reasonable. Suppose the covariance function is parameterized by $\boldsymbol{\theta}$. A common classical estimation strategy is to estimate $(\boldsymbol{\theta}, \sigma^2)$ via maximum likelihood or residual maximum likelihood, then plug these estimates into the universal kriging formulas for prediction. The MSPE of this estimator for $Y(\boldsymbol{t}_i)$ is unknown, but Zimmerman (2006) discusses some approximations using the Fisher information matrix. Then for the design problem, Zimmerman (2006) suggests minimizing either the maximum approximate MSPE of the target locations, or minimizing the mean approximate MSPE.

In the Bayesian context, the standard design criterion is the expected entropy gain from conditioning on the observations (Ebrahimi et al., 2010), defined by $\mathrm{E}\left(\log \frac{[\boldsymbol{Y}|\boldsymbol{Z}]}{[\boldsymbol{Y}]}\right)$ where $\boldsymbol{Y} = [Y(\boldsymbol{t}_1), Y(\boldsymbol{t}_2), \ldots, Y(\boldsymbol{t}_{N_t})]'$, the expectation is taken over $\boldsymbol{Y}$, $\boldsymbol{Z}$, and any model parameters, and $[.]$ denotes the (conditional) probability density of the enclosed random variable. For example, Fuentes et al. (2007) uses this criterion. It turns out that in the simple and universal kriging cases, i.e. all parameters known and all parameters but $\boldsymbol{\beta}$ known respectively, the entropy approach with a specific prior is equivalent to minimizing the log determinant of the MSPE matrix. The key difference between this and maximizing $\overline{Q}$ is that the entropy criterion takes into account the covariance between the prediction errors at the various target locations. We mention this for completeness, but will restrict ourselves to universal kriging in the example in the next section.

2.4 PSO for Minimizing Kriging Variance

In the many spatial design problems, e.g. the examples in Zimmerman (2006), the candidate sampling locations and the target locations are the same discrete subset of $\mathcal{D}$. We allow the candidate locations to be the entirety of $\mathcal{D}$, where $\mathcal{D}$ is an open or closed subset of $\Re^2$. This is partially a requirement of PSO — implicitly PSO assumes that the search space is $\Re^n$, but this can be easily replaced with a closed or open subset of $\Re^n$ by defining the objective function to be $-\infty$ outside of the candaidate set (assuming maximization). In principle, it also allows for better potential designs. It is not always the case that the optimal design samples only from the target locations. When there is uncertainty about the regression or covariance function parameters, the optimal design is much less likely to only use the target locations in our experience, though this depends on the design criterion as well.

3 Houston Ozone Monitoring

The Texas Commission on Environmental Quality (TCEQ) publishes a variety of environmental data for Texas, including many environmental indicators that directly relate to public health. We focus on ozone in the Houston-Galveston-Brazoria area. The TCEQ measures ozone at a variety of monitoring locations in this area and publishes daily maximum eight-hour ozone concentrations (DM8s) in parts per billion (ppb) for each monitoring location. DM8s are computed as follows. First, the TCEQ creates publishes an hourly average (in ppb) for each monitoring location. Then an eight hour average is constructed at that location for each contiguous eight-hour period where all eight measurements were present for a given day. The maximum of these eight-hour averages for a given day is the published DM8. Days with less than 18 valid eight-hour averages have no published DM8.

In August 2016 there were 44 active monitoring locations in the Houston-Galveston-Brazoria area. For each location $\boldsymbol{u}$ we compute the monthly average DM8, which we denote by $Z(\boldsymbol{u})$. At one location, MRM-3 Haden Road, there are two DM8 observations of 0 ppb in the month of August. We assume that these were data errors and throw them out for the purposes of computing $Z(\boldsymbol{u})$ at that location. Of the 44 locations, one has 15 valid DM8 measurements of 31 possible valid measurements, another has 24 valid measurements, and the rest of the locations have at least 27 valid measurements.

The hypothetical design problem we consider is the addition of five new ozone monitoring locations to the Houston-Galveston-Brazoria monitoring network in Harris County, where Houston is located, with the goal of predicting ozone concentrations within Harris County. Of the 44 existing locations, 33 are already in Harris County, though the locations outside of the county are still useful for spatial prediction within Harris County.

The class of models we consider for the average DM8s is as follows. Let $Z(\boldsymbol{u})$ denote the measured average DM8 as location $\boldsymbol{u}$ and let $Y(\boldsymbol{u})$ denote the true DM8. We assume that $Z(\boldsymbol{u})$ is a noisy Gaussian measurement of $Y(\boldsymbol{u})$, i.e. the data model is $Z(\boldsymbol{u}) \sim N[Y(\boldsymbol{u}), s^2_Z(\boldsymbol{u}) + \tau^2]$. The measurement error is broken into two pieces which we assume are independent: pure measurement error with variance τ^2, and sampling error with variance $s^2_Z(\boldsymbol{u})$. Sampling error occurs from measuring DM8 on less than the full 31 days in August. We assume that the measured days are randomly sampled from the 31 possible days so that the sampling error for $Z(\boldsymbol{u})$ is $s^2_Z(\boldsymbol{u}) = s^2_{DM8}(\boldsymbol{u})[31 - n(\boldsymbol{u})]/[31n(\boldsymbol{u})]$ where $s^2_{DM8}(\boldsymbol{u})$ is the sample variance of the DM8 measurements at location $\boldsymbol{u}$ and $n(\boldsymbol{u})$ is the number of measurements at that location. Note that if $n(\boldsymbol{u}) = 31$, the maximum number of DM8 measurements, then $s^2_Z(\boldsymbol{u}) = 0$. This sampling model is almost certainly incorrect since missing DM8 measurements at a given location tend to be contiguous in time, but it is a useful approximation as long as the probability of a given day being measured is largely unrelated to the true DM8 at that location.

At the process level, we assume that $Y(\boldsymbol{u})$ is a geostatistical process with mean function $\mu(\boldsymbol{u}) = \boldsymbol{x}(\boldsymbol{u})'\boldsymbol{\beta}$ and exponential covariance function $C(\boldsymbol{u}, \boldsymbol{v}) = \sigma^2 \exp(-||\boldsymbol{u} - \boldsymbol{v}||/\phi)$. We assume that any fine scale variability is measurement error and captured by the data model variance $s_Z^2(\boldsymbol{u}) + \tau^2$. We considered several possible mean functions: constant in $\boldsymbol{u}$, linear in $\boldsymbol{u}$, and quadratic in $\boldsymbol{u}$. We fit each model using maximum likelihood and found that quadratic terms were unnecessary, but linear terms did significantly help explain variation in $Y(\boldsymbol{u})$, i.e. $\mu(\boldsymbol{u}) = \beta_0 + u_1\beta_1 + u_2\beta_2$. Further, both AIC and BIC were essentially indifferent between the linear mean and constant mean models. We use the linear mean model. Note that this model is a slight departure from the class considered in Section 2. We use the same notation in that section as in this section so that, e.g., $\boldsymbol{u}$ is a generic location inside $\mathcal{D}$, which can be considered to be the entire Houston-Galveston-Brazoria area, $\boldsymbol{s}_i$ is an existing monitoring location, $\boldsymbol{t}_i$ is a target location, and $\boldsymbol{d}_i$ is a proposed location for a new monitoring station. The only difference is that in Section 2 $\boldsymbol{C}_Z = \tau^2\boldsymbol{I} + \boldsymbol{C}_Y$, whereas in this section $\boldsymbol{C}_Z = \boldsymbol{S}_Z^2 + \tau^2\boldsymbol{I} + \boldsymbol{C}_Y$ where $\boldsymbol{S}_Z^2$ is a diagonal matrix with $[s_Z^2(\boldsymbol{s}_1), s_Z^2(\boldsymbol{s}_2), \ldots, s_Z^2(\boldsymbol{s}_{N_s}), 0, \ldots, 0]$ along the diagonal. This assumes that $s_Z^2(\boldsymbol{d}_i) = 0$ for $i = 1, 2, \ldots, N_d$, in other words that each new monitoring station will record a valid DM8 for every day of a given month so that there is no sampling error. [NOTE: WE SHOULD OBVIOUSLY FIT THE MODEL USING THE $s_Z^2(\boldsymbol{s}_i)$s, BUT FOR KRIGING SHOULD WE ASSUME $s_Z^2(\boldsymbol{s}_i) = 0$? OR MAYBE SOME OTHER VALUE? BASICALLY, SHOULD WE EXPECT SAMPLING ERROR AND TO WHAT DEGREE?]

We use two design criteria in order to choose the five new locations in Harris County: $\overline{Q}_{ik}(\boldsymbol{d})$ and $Q_{uk}^*(\boldsymbol{d})$, both defined in Section 2. For both criteria we plug in the MLEs for τ^2, σ^2, and ϕ. We assume that the goal is the predict average DM8 in all of Harris County, so we approximate the continuous versions of $\overline{Q}_{ik}(\boldsymbol{d})$ and $Q_{uk}^*(\boldsymbol{d})$ with the finite sample versions using a sample of 1,000 locations drawn uniformly from Harris County. We try a variety of PSO algorithms in order to select the new locations. Since the design space only allows new monitoring locations within Harris County, we define $\overline{Q}_{uk}(\boldsymbol{d}) = Q_{uk}^*(\boldsymbol{d}) = \infty$ when any of the proposed locations is outside of Harris County.

[INSERT PSO RESULT HERE... CURRENTLY RUNNING, THOUGH WE MAY

WANT TO TWEAK IT - TAKES 2-3 DAYS TO RUN]

References

Andrieu, C. and Thoms, J. (2008). "A tutorial on adaptive MCMC." *Statistics and Computing*, 18, 4, 343–373.

Blum, C. and Li, X. (2008). "Swarm Intelligence in Optimization." In *Swarm Intelligence: Introduction and Applications*, eds. C. Blum and D. Merkle. Springer.

Campos, M., Krohling, R. A., and Enriquez, I. (2014). "Bare bones particle swarm optimization with scale matrix adaptation." *Cybernetics, IEEE Transactions on*, 44, 9, 1567–1578.

Clerc, M. (2010). *Particle swarm optimization*. John Wiley & Sons.

Clerc, M. and Kennedy, J. (2002). "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." *Evolutionary Computation, IEEE Transactions on*, 6, 1, 58–73.

Cressie, N. and Wikle, C. K. (2015). *Statistics for Spatio-Temporal Data*. John Wiley & Sons.

Eberhart, R. C. and Shi, Y. (2000). "Comparing inertia weights and constriction factors in particle swarm optimization." In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, 84–88. IEEE.

Ebrahimi, N., Soofi, E. S., and Soyer, R. (2010). "Information Measures in Perspective." *International Statistical Review*, 78, 3, 383–412.

Fuentes, M., Chaudhuri, A., and Holland, D. M. (2007). "Bayesian Entropy for Spatial Sampling Design of Environmental Data." *Environmental and Ecological Statistics*, 14, 3, 323–340.

Gelman, A., Roberts, G., and Gilks, W. (1996). "Efficient Metropolis jumping rules." *Bayesian statistics*, 5, 599-608, 42.

Hsieh, H.-I. and Lee, T.-S. (2010). "A modified algorithm of bare bones particle swarm optimization." *International Journal of Computer Science Issues*, 7, 11.

Kennedy, J. (2003). "Bare bones particle swarms." In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, 80–87. IEEE.

Krohling, R., Mendel, E., et al. (2009). "Bare bones particle swarm optimization with Gaussian or Cauchy jumps." In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 3285–3291. IEEE.

Richer, T. J. and Blackwell, T. M. (2006). "The Lévy particle swarm." In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 808–815. IEEE.

Tuppadung, Y. and Kurutach, W. (2011). "Comparing nonlinear inertia weights and constriction factors in particle swarm optimization." *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15, 2, 65–70.

Zimmerman, D. L. (2006). "Optimal Network Design for Spatial Prediction, Covariance Parameter Estimation, and Empirical Prediction." *Environmetrics*, 17, 6, 635–652.