Take Me Out to (Analyze) the Ballgame

Chris Comiskey

May 2, 2017

Contents

1 Introduction **2**

2 Variable-Resoluation Heat Maps **3**

2.1 Conventional Heat Maps 3

2.2 Empirical Heat Maps and Resolution Selection 4

2.3 Empirical Heat Maps and Spatially Varying Resolution 7

2.4 Appendix 12

2.4.1 VarResHM, An R Package 12

3 Shiny Heat Map Confidence Intervals **12**

3.1 Generalized Linear Models for Hitter Success Probabilities 12

3.2 Biomechanically Interpretable Covariates 13

3.3 Generalized Linear Model with Biomechanically Interpretable Covariates 14

3.4 Hosmer-Lemeshow Goodness of Fit Test 15

3.5 Appendix 16

3.5.1 ShinyHMCI, An R Package 16

4 Spatial Generalized Linear Mixed Models **16**

4.1 Introduction 16

4.1.1 Gaussian Random Field 16

4.1.2 Exponential Covariance 16

4.1.3 Spatial Generalized Linear Mixed Model 17

4.1.4 Markov Chains . . . 17
4.1.5 "Big N Problem" . . . 18
4.2 Numerical Optimization; Hamiltonian Monte Carlo in Stan . . . 18
4.2.1 Hamiltonian Dynamics and MCMC . . . 18
4.2.2 Hamilton Equations for MCMC . . . 18
4.2.3 Optimizing in Stan . . . 20
4.3 Dimension Reduction; Predictive Process Models . . . 23
4.3.1 PPM Procedure . . . 23
4.3.2 Improved Predictive Process Models . . . 24
4.4 Approximation; SPDE and INLA . . . 24
4.4.1 Gaussian Markov Random Fields . . . 25
4.4.2 Stochastic Partial Differential Equation (SPDE) . . . 25
4.4.3 Integrated Nested Laplace Approximations (INLA) . . . 28
4.4.4 Bayesian Inference in R-INLA . . . 30

A Stan Code **30**

B R Code, spBayes **31**

C R-INLA Code **31**

D Kriging **31**

1 Introduction

Graphical displays are an irreplaceable tool for the statistician, and the statistical community. They allow us to quickly communicate information about a data set to another statistician, or, perhaps even more importantly, to non-statisticians. As technology generates more data the importance of this area of statistics expands, statistical analysis becomes more widespread, and our graphic-making abilities improve. The R data visualization package `ggplot2` highlights (i) the importance of graphical diplays, because it is among the most downloaded R packages; and (ii) the need for innovation in this area, because its popularity explosion signals it met a need.

In this chapter we focus on one type of graphical display, the heat map. We show the way it fails to adequately communicate spatial data dispersion and distribution attributes, and we innovate a solution. This innovation improves our ability to communicate spatial data density and dispersion attributes to the viewer using a heat map.

2 Variable-Resoluation Heat Maps

2.1 Conventional Heat Maps

We start with a baseball heat map example, as baseball data motivates this research. Consider the empirical heat map in Figure 1, of the two dimensioned vertical face of the strike zone, where grid box colors represent empirical batting averages at pitch locations. In Figure 1 we show PITCHf/x® data on 1,932 right-handed hitters, taking 1,582,581 swings between 2008 and 2015. Let $b = 1, \ldots, 627$ index grid boxes. Let $i = 1, \ldots, 1,582,581$ index swings, and define $n_b = \sum_i \mathrm{I}_{\{i \in b\}}$ as the total number of swings in box b. Define a Bernoulli random variable, S_i, that equals one for swing success and zero for swing failure, and let $\hat{p}_b = \frac{1}{n_b} \sum_i S_i \cdot \mathrm{I}_{\{i \in b\}}$ be the empirical box b success probability. Figure 1 displays the resulting empirical heat map for 1,932 right-handed hitters, taking 1,582,581 swings between 2008 and 2015. The heat map graphic maps the empirical success probability, $\hat{p}_b$, of hitters swinging at pitches that passed through the space represented by that grid box, to a color on a spectrum.

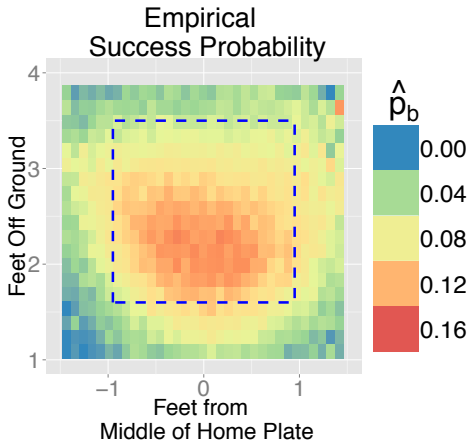

Figure 1: The gridded hitting zone with 3/4 inch by 3/4 inch boxes, from the catcher's point of view. The color of the box represents the empirical batting average ($\hat{p}_b$) for right handed hitters, swinging at pitches in that location. Calculations based on 1,932 right handed hitters, swinging at 1,582,581 pitches between 2008 and 2015. The dashed line marks the boundaries of the called strike zone for a 6'2" hitter.

While not sophisticated statistically, the graphic efficiently conveys empirical spatial success probabilities; it maps the statistic $\hat{p}_b$ to a color on a spectrum. Though it can easily go unnoticed, a heat map's creater *chooses* the grid box size that, hopefully, best communicates the data's information content; here, average spatial hitting ability estimates. The data's varying spatial density through the strike zone is not part of the communicated information, no matter the grid box size selected. This missing information is important, because in general more data means better estimates, and the viewer gets no such indication in Figure 1. Traditional heat maps do not communicate this information. We propose an innovation that addresses this shortcoming. To illustrate, consider a heat map for an individual hitter.

The heat map above divided the strike zone into relatively small boxes, because the data supported it. By "supported it" we mean the small, spatially specific boxes retain a sample size large enough to keep the variance of $\hat{p}_b$ acceptably [1] This is important because individual hitter datasets vary dramatically in size, with swing totals ranging from a single swing to over 10,000 swings. Similar to bin width selection for a histogram, the choice of heat map resolution can dramatically affect how the data is represented, and the usefulness of the parameter estimates of interest. The resolution decision depends on the size and nature of the data set in question, and its spatial dispersion through the domain. To explore this decision in detail we look at batter 425509, a veteran player named Jhonny Peralta, whose data set contains 9,177 swings.

2.2 Empirical Heat Maps and Resolution Selection

The heat map in Figure 2 divides the central region of the strike zone into 16 equally sized boxes. Each box maps $\hat{p}_b$ to a color, and the box sample size, n_b, is printed on the box center. For convenient referencing in this chapter, we adopt the convention of numbering boxes with a horizontal component (left to right) and a vertical component (bottom to top [**Alix: "WHY?"**]). For example, we call the top-left box (1,4), and the bottom-right box (4,1).

[1] Defining "acceptable" variance ranges and thresholds will depend on context and analysis objectives. For example, a pitching coach may be satisfied with estimates accurate 95% of the time to within 20 batting average points. This margin of error, 0.02, requires a sample size of 32 when $p_b = 0.09$. Note that the variance depends on the mean for a Bernoulli random variable.

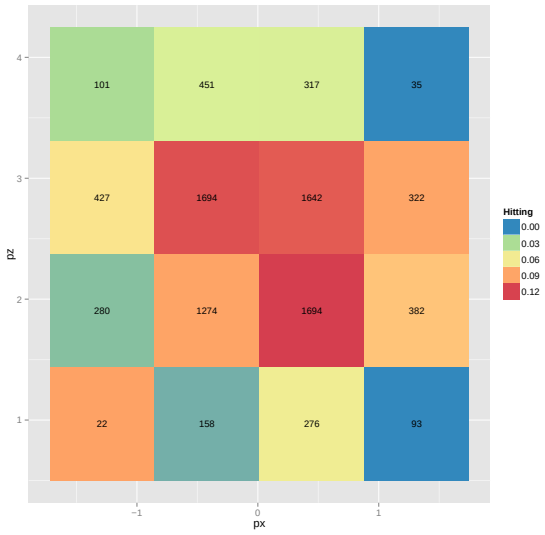

Figure 2: This four by four heat map conveys the empirical batting average of batter 425509, Johnny Peralta, in each of 16 square regions of the hitting zone. Each box maps $\hat{p}_b$ to a color. The number printed on each box represents the number of pitches the hitter swung at that passed through that box.

Peralta swang at 22 pitches in the region of space represented by box (1,1), and he swung at 1694 pitches in box (2,3). We interpret these two box sample sizes to contextualize. Three primary factors influence pitch location: pitcher game theoretic strategy, pitch-swing location margin of error (distance by which a pitch misses its intended target), and the game state. Game theoretic strategy concerns the pitcher's knowledge of the hitter's strenths and weaknesses, and the hitter's reciprocal knowledge. Margin of error concerns the pitcher's usual outcome of not exactly hitting his target.[2] Game state characteristics include the at-bat count, the number of outs, and if runners occupy bases.[3] Peralta probably swung at only 22 pitches in box (1,1) because he did not see many pitches there. We can speculate this is because pitches there are mostly out of the strike zone, both in the horizontal and vertical directions. Therefore, it is less likely to induce a swing at a bad pitch to hit, and unlikely to be called a strike despite being out of the strike zone. With $n_{(1,1)} = 22$, the four by four resolution is sufficiently fine to present $\hat{p}_{(1,1)}$. Box (2,3), with $n_{(2,3)} = 1694$ pitches, can support more location specific, but still reliable estimates of p. This motivates finer resolution in that region of space. Peralta has relatively high success in Box (2,3), and he undoubtedly swings at as many pitches in that box as possible. The pitcher knows this, so will seldom aim there. However, by virtue

[2]If you visualize a 12 inch diameter archery target where the pitcher aims, he will usually hit the target somewhere, but not the bull's-eye.

[3]Two example game state pressures include the increased penalty for throwing a pitch outside the strike zone on a three ball count (the runner gets on base at four balls); the increased penalty for a hit with a runner in scoring position (runner on second or third base).

of being closer to the center of the strike zone, this location collects more pitch location mistakes.

Because, as mentioned, a finer resolution is justified for box (2,3), we subdivide all boxes further. For simplicity, without implying this is the only or best way to increase resolution, we divide each box into four equally sized sub-boxes. Figure 3 shows the 16 by 16 result.

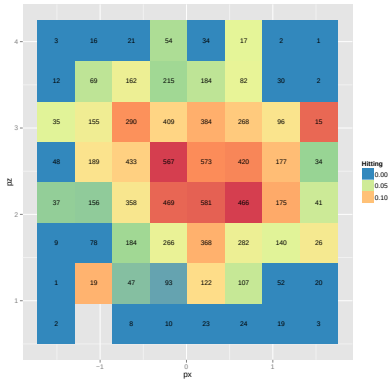

Figure 3: This 16 by 16 heat map conveys the empirical batting average of batter 425509, Johnny Peralta, in each of 64 square regions of the hitting zone. Each box maps $\hat{p}_b$ to a color. The number printed on each box gives the number of pitches the hitter swung at that passed through that box. A grey box indicates no pitches passed through that box.

Boxes (3,5), (3,6), (4,5), and (4,6)—the boxes created by dividing box (2,3) at the four by four resolution—still contain sample sizes sufficient to support low variance p estimates. More generally, 24 boxes still have a sample size greater than 150; and 15 boxes still have a sample size of greater than 250. These boxes could support further subdivision. On the other hand, numerous boxes—corner and edge boxes in particular—now contain sample sizes generally insufficient to support low variance estimates of p_b. Twenty-nine boxes have a sample size of less than 50, and 17 boxes have a sample size of less than 20. At this resolution one box recorded zero swings.

In this way, due to the particular dispersion of the data, a heat map at any resolution will contain boxes of exceedingly small sample sizes (high variance), and/or boxes of unnecessarily large sample size (unnecessarily low variance). Figure 3 shows six different heat map resolutions, constructed with the same data from Peralta. We started with one box, and subdivided each box into four at each iteration. We chose this simple resolution increasing algorithm to illustrate the resolution selection challenge, and to provide a foundation for our innovation in the next section.

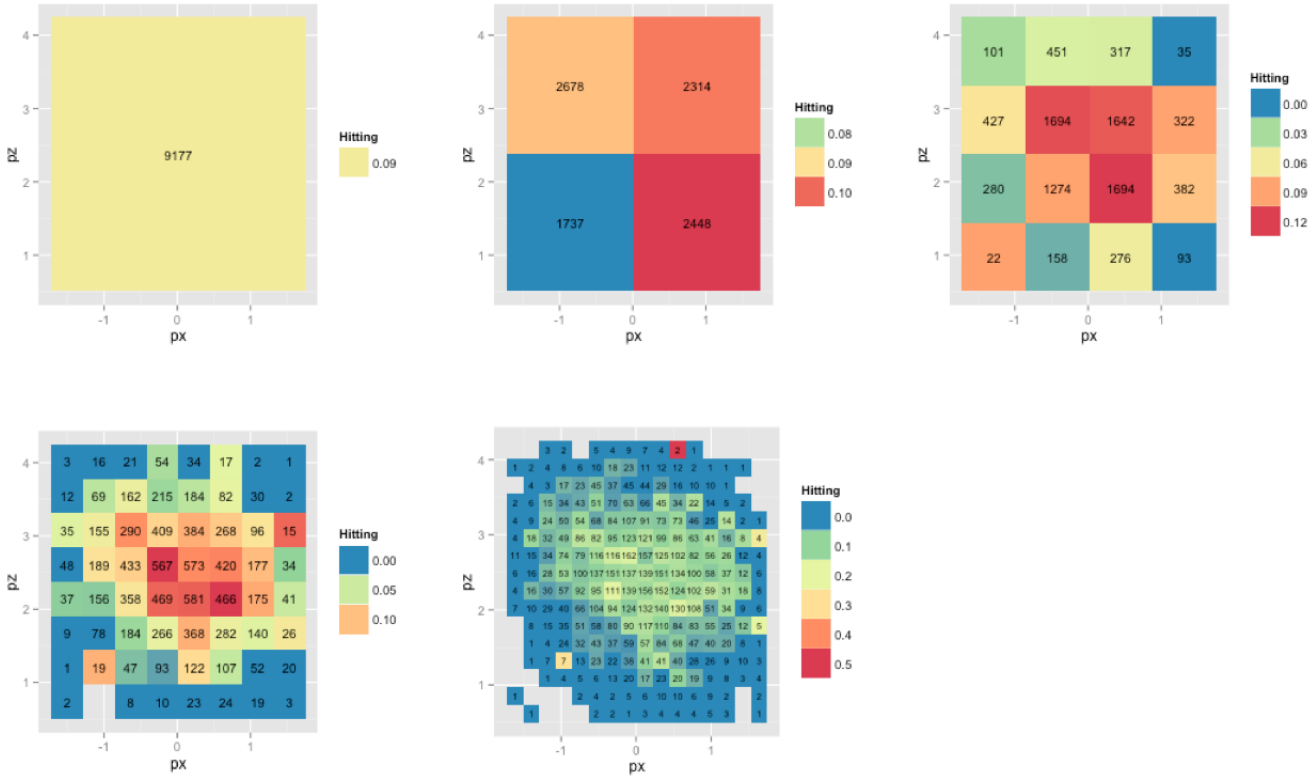

Figure 4: These **six (?)** heat maps show the same data, 9177 swings by batter 425509, Johnny Peralta, at increasing resolutions. Heat map one is unnecessarily coarse, while heat map six is excessively fine. Note how dramatically the visual imact and impression varies as the resolution increases. Which resolution best conveys the data?

It is unclear which of these five resolutions best combines spatially precise estimates of p where possible, and box sample sizes with $\text{Var}(\hat{p}_b)$ in a desirable range. The viewer interested in the center of the strike zone should prefer the last (**need labels** heat map, as the box sample sizes are sufficient to provide such spatially specific low variance estimates. The boxes closer to the edges of the strike zone contain higher variance, and thus less reliable estimates, due to prohibitively small sample sizes. We propose a new heat map approach that combines resolutions according to the data's varying spatial density.

2.3 Empirical Heat Maps and Spatially Varying Resolution

Consider again the heat map in Figure 2. Notice box (1,1) contains data on 22 swings, a sample size where subdividing would yield sample sizes uselessly small, and thus estimate variances prohibitively high. Box (2,3), in contrast, contains data on 1694 swings, which would support estimates that are more spatially accurate without $\text{Var}(\hat{p}_b)$ increasing past acceptable levels. We propose defining a stopping rule and a

subdividing method, and subdividing boxes further accordingly. For example, in Figure 4 we subdivide, into four equally sized boxes, all boxes where $n_b > 200$, . One iteration through all boxes at their current size, subdividing according to this rule, converts the heat map on the left to the heat map on the right.

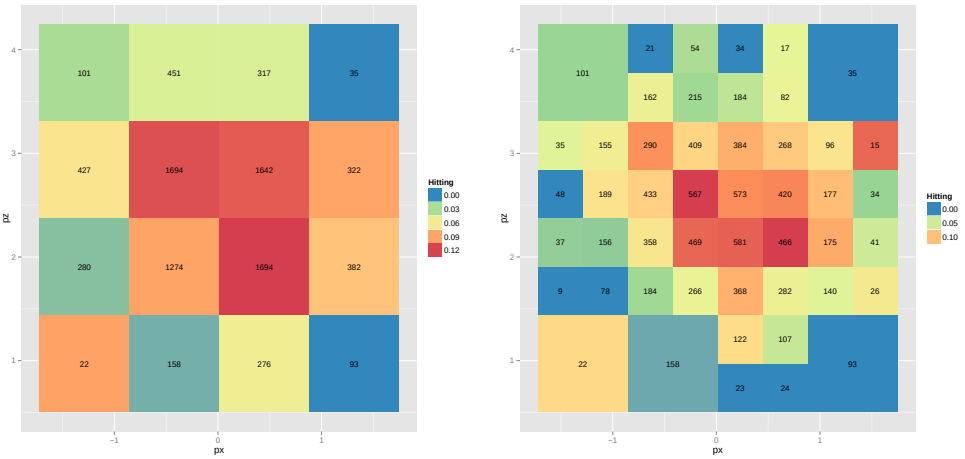

Figure 5: These heat maps convey the empirical batting average of Johnny Peralta in each square region of the hitting zone. Each box maps $\hat{p}_b$ to a color. The number printed on each box represents the number of pitches Peralta swung at that passed through that box. Notice that all boxes with a sample size greater than 200 in the heat map on the left, have been subdivided in the heat map on the right.

Notice all four corner boxes have not subdivided, indicating Peralta seldom sees and swings at pitches in these locations. The boxes toward the middle of the map tend to have larger sample sizes, and higher $\hat{p}_b$. Pitches pass through the middle of the hitting zone more frequently because many pitch target margin of error circles overlap there; and it is the region where pitch target margin of error circles are entirely inside the strike zone. Sixteen boxes still have a sample size greater than 200, and 11 still have a sample size greater than 300. We iterate again, and further subdivide 16 boxes where $n_b > 200$.

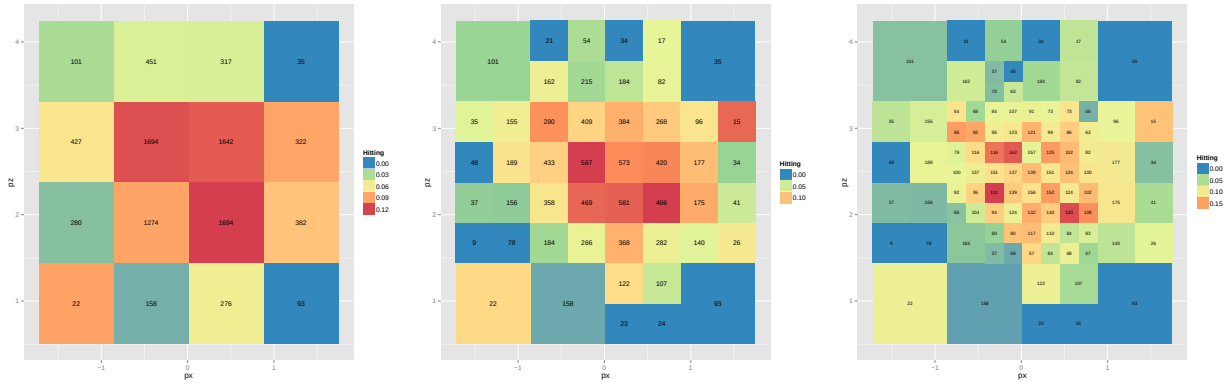

Figure 6: These heat maps convey the empirical batting average of batter 425509, Johnny Peralta, in each boxed region of the hitting zone. Each box maps $\hat{p}_b$ to a color. The number printed on each box represents the number of pitches the hitter swung at that passed through that box. All boxes with a sample size greater than 200 in the heat map on the left, have been subdivided in the heat map in the middle. All boxes with a sample size greater than 200 in the heat map in the middle, have been subdivided in the heat map on the right.

In Figure 6, the middle heat map has 16 boxes with $n_b > 200$. In the heat map on the right these 16 boxes have been subdivided into four boxes each. After this iteration, the heat map on the far right consists of 97 boxes, with a mean box sample size of 94.57, and median of 94. The minimum box sample size is 9, and the maximum is 189. The first quartile box sample size is 63, and the third quartile is 125. Regions with a higher density of pitch-swings necessarily have smaller boxes, which acts to convey additional information to the reader, compared to a heat map on a uniform grid. Note that the stopping rule and subdivision algorithm can be defined by the map's creator, offering flexibility to create the heat map structure that suits the data.

Figure 8 gives the full sequence of heat maps that result from applying the stopping rule $n_b < 100$, starting with a single box.

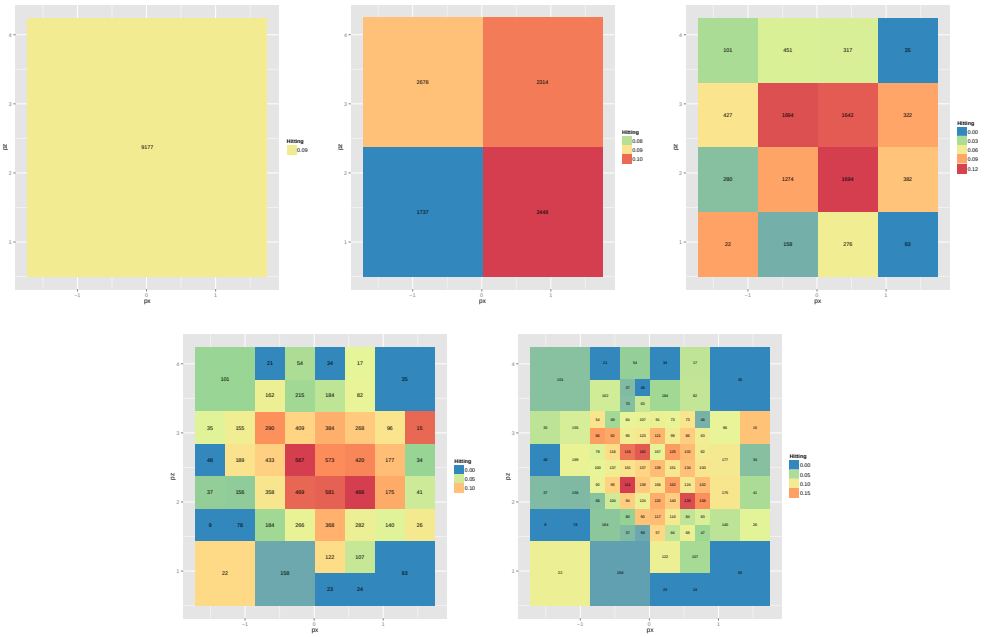

Figure 7: These heat maps convey the empirical batting average of batter 425509, Johnny Peralta, in each boxed region of the hitting zone. Each box maps $\hat{p}_b$ to a color. The number printed on each box represents the number of pitches the hitter swung at that passed through that box. All boxes with a sample size greater than 200 in each heat map have been subdivided in the subsequent heat map.

To demonstrate the flexibility, consider a different stopping rule, $n_b < 100$. Figure 8 gives the sequence of heat maps that result from applying this stopping rule, with the same subdividing algorithm (***need to delineate what this algorithm is).

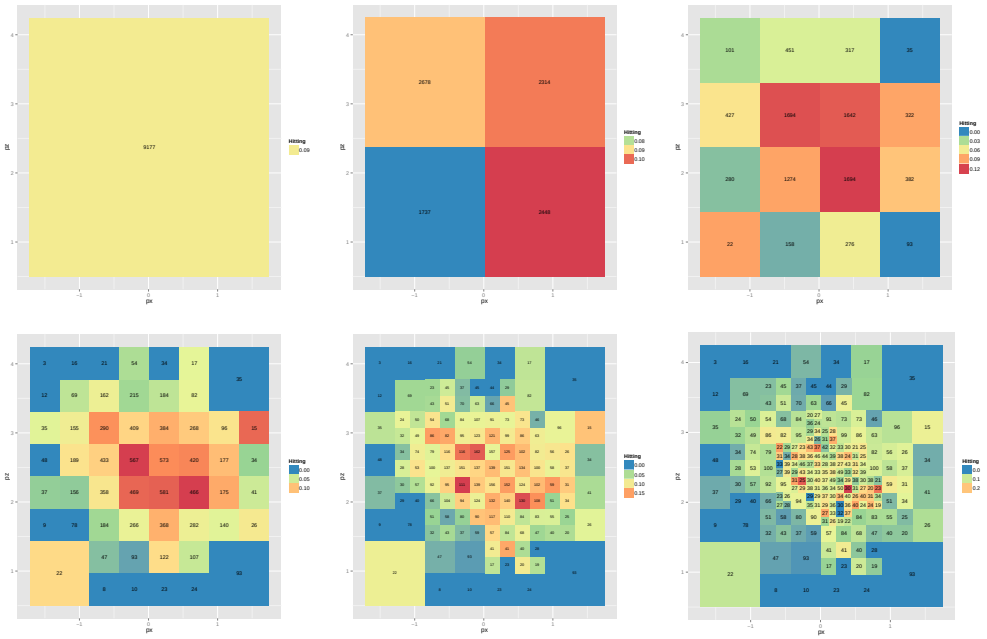

Figure 8: These heat maps convey the empirical batting average of batter 425509, Johnny Peralta, in each boxed region of the hitting zone. Each box maps $\hat{p}_b$ to a color. The number printed on each box represents the number of pitches the hitter swung at that passed through that box. All boxes with a sample size greater than 100 in each heat map have been subdivided in the subsequent heat map.

Compare this sequence to Figure 7, where the stopping rule was $n_b < 100$. The top row of heat maps in Figure 7 and Figure 8 are identical, but notice in the four by four heat map that $100 < n_{(2,1)} < 200$, and $100 < n_{(1,4)} < 200$. This implies one stopping rule applies, but the other does not.

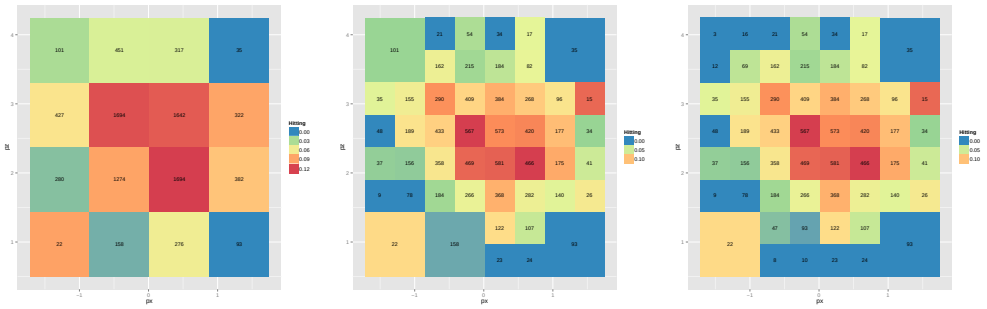

Figure 9: ...(these images, and others, need labels: (A) (B) (C) etc)

For this reason the bottom left heat maps in Figures 7 and 8, shown in Figure 9, differ in the number of boxes of each size, and the total number of boxes. This divergence continues at the next iteration, where the stopping rule $n_b < 100$ requires 28 box subdivisions in Figure 8, map three; and $n_b < 200$ gives 16 box

subdivisions in Figure 7, map three.

2.4 Appendix

2.4.1 VarResHM, An R Package

3 Shiny Heat Map Confidence Intervals

3.1 Generalized Linear Models for Hitter Success Probabilities

Our goal is to create a statistical model for the heat map of success probabilities. Nonparametric methods, while straightforward, sacrifice interpretability; while they achieve a modeled heat map, there may not be interpretable components of the model. Nonparametric models cannot relate spatially varying hitter success probabilities to hitter attributes. We propose a parametric approach using biomechanically interpretable covariates. Existing research analyzes the biomechanics of the baseball swing [Welch et al., 1995], but no research integrates those results with spatial swing outcomes in a statistical model.

Let success indicator variable, Y_{ijklm}, be a Bernoulli random variable with spatially varying mean [Sheldon et al., 2002]. Subscript $i = 1, \ldots, n_{jklm}$ indexes hitter j's swings in at bat k against pitcher l in year m. Subscript $k = 1, \ldots, n_{jlm}$ indexes hitter j's at bats against pitcher l in year m. Subscript $l = 1, \ldots, n_{jm}$ indexes pitchers hitter j faced, where n_{jm} is the total number of pitchers hitter j faced; and $m = 2007, \ldots, 2016$ indexes year. Let $\boldsymbol{s}_{ijkl} = (px_{ikl}, pz_{ijkl}) \in \boldsymbol{D} \subseteq \boldsymbol{R}^2$ be the horizontal and vertical locations, respectively, of pitch $ijkl$ as it passes through the two dimensioned vertical face of the hitting zone. The origin, $\boldsymbol{s}_{\cdot} = (0, 0)$, is the midpoint of the front edge of home plate, at ground level. From the pitcher's point of view, pitches to the left (right) of the center of home plate correspond to negative (positive) values of $px_{\cdot}$. Pitches that bounce before reaching home plate correspond to negative values of $pz_{\cdot}$.

In this study we make the simplifying assumption that location success probabilities depend on only location and hitter. This means we dispense with subscripts k, l, and m. We also assume that, given pitch location to hitter j, $\boldsymbol{s}_{ij} = (px_{ij}, pz_{ij})$, swings are independent Bernoulli trials. This gives $Y_{ij}|\boldsymbol{s}_{ij} \sim$ Bernoulli(p_{ij}), where $\mathrm{E}[Y_i|\boldsymbol{s}_{ij}] = p_{ij}$

Accordingly, let $i = 1, \ldots, n_j$ index hitter j's swings, out of n_j total swings on record. Let $\boldsymbol{X}_{ij}(\boldsymbol{s}_i)$ be covariates specific to hitter j and location $\boldsymbol{s}_{ij}$ on swing i. A Bernoulli random variable suggests a generalized linear model with logit link function for relating success probability to covariate information:

$$\text{logit}(p_{ij}|\boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})) = \boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\beta}_j, \tag{1}$$

where $\boldsymbol{\beta}_j$ is the vector of covariate coefficient parameters specific to hitter j [Myers et al., 2012]. Next, we discuss and develop covariates.

3.2 Biomechanically Interpretable Covariates

Why does Peralta, and why do hitters in general, hit pitches in some locations better than others? We submit biomechanics as potentially part of the answer. Biomechanics underpin why hitters prefer some pitch locations more than others. Given the choice, athletes select a specific place for the ball before swinging. Consider golf, a sport where the ball is stationary, and the acting athlete chooses where to stand in relation to the ball. In fact, golfers position themselves very precisely in relation to the ball to achieve impact at the optimal point in their swing [Cochran and Stobbs, 2005]. If the impact point deviates from the ideal location, performance suffers. Consider tennis, a step closer to baseball, in that the ball approaches, but the player has time to position himself relative to the incoming ball. Once again, tennis players strive to hit the ball at a specific point in their forehand, a precise distance from the ground and from their body [Elliott, 2006]. As with golf, if the point of impact deviates from this location, performance suffers. Note that in both sports the ideal player to ball positioning depends on, at the very least, anatomy, biomechanics, and equipment. We submit the same dynamics affect baseball hitting. However, in baseball the hitter cannot predtermine ball location, nor does he have time to reposition himself in response to the location and trajectory of the incoming pitch. For these reasons, meaningful measurements of hitter to ball distance and angle are reasonable covariates. Polar coordinate pitch locations would inherently provide this type of meaningful covariate for use and interpretation in our models.

To illustrate, in Figure 3 we shift the origin to a hitter's approximate center of gravity in his stance, where the extended bat line intersects his axis of rotation at the moment of contact [Welch et al., 1995].

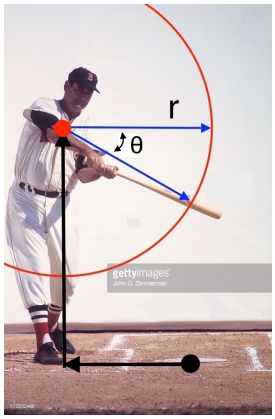

Figure 10: The ground level black dot represents the origin (0,0) in the rectangular coordinate system. The translated origin (red dot) coincides with the hitter's approximate center of gravity, and thus the polar origin. The bold arrows show the origin shift. The length of the arrows moving out from his center of gravity to specific grid locations represent r, and the angle between the same two arrows represent θ.

Referring to Figure 3, let r measure the distance from the hitter's center of gravity to the ball at impact, and let θ be the angle below horizontal of the line segment connecting the center of gravity and the ball at impact. As in golf and tennis, ball location—too far/close to the hitter, or above/below the ideal point of impact— affects hitting performance. Letting $\boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})$ in (1) be comprised of r_{ij} and θ_{ij} terms provides an exploratory starting point.

3.3 Generalized Linear Model with Biomechanically Interpretable Covariates

Let covariate vector $\boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})$ in (1) be defined as $\boldsymbol{X}_{ij}(\boldsymbol{s}_{ij}) = \{r_{ij}, \theta_{ij}, r_{ij}\theta_{ij}, r_{ij}^2, \theta_{ij}^2, r_{ij}^2\theta_{ij}^2\}$. Substituting into (1) yields:

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}, r_{ij}, \theta_{ij}) = \beta_{j0} + \beta_{j1}r_{ij} + \beta_{j2}\theta_{ij} + \beta_{j3}r_{ij}\theta_{ij} + \beta_{j4}r_{ij}^2 + \beta_{j5}\theta_{ij}^2 + \beta_{j6}r_{ij}^2\theta_{ij}^2 \tag{2}$$

Note that given a hitter j, and pitch location $\boldsymbol{s}_{ij}$, the elements of $\boldsymbol{X}_{ij}$ are simply a trigonemtric function of $\boldsymbol{s}_{ij}$ and the translated origin. Thus, for convenience, we replace $\text{logit}(p_{ij}|\boldsymbol{s}_{ij}, r_{ij}, \theta_{ij})$ with $\text{logit}(p_{ij}|\boldsymbol{s}_{ij})$ for the remainder of this study.

We choose Johnny Peralta from Chapter 1 to illustrate, and let $j = P$ for convenience. We fit model (2) using Peralta's $n_P = 9177$ observed swings, find maximum likelihood estimates of $\boldsymbol{\beta}_P$ using an iteratively reweighted least squares algorithm [Myers et al., 2012].

(IT WOULD BE NICE: EMPIRICAL, ESTIMATES, FITTTED HEAT MAP)

Covariate	Parameter	MLE	SE	p
N/A	β_0	-4.08	0.70	< 0.001
r	β_1	1.19	0.51	0.018
θ	β_2	-1.93	1.90	0.311
$r * \theta$	β_3	-1.64	0.70	0.064
r^2	β_4	-0.32	0.09	< 0.001
θ^2	β_5	-3.92	1.10	< 0.001
$r^2 * \theta^2$	β_6	-0.46	0.21	0.025

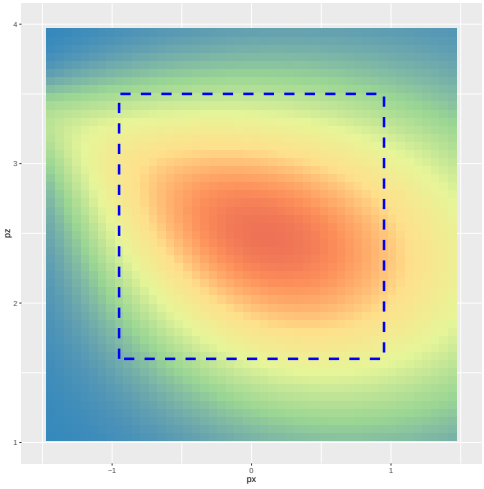

Figure 11:

3.4 Hosmer-Lemeshow Goodness of Fit Test

Generalized Linear Models [Myers et al., 2012]

- (pg 147) Hosmer-Lemeshow test = Logistic regression Goodness of Fit test.
- Like Pearson Chi-Sq, but for continuous covariates.
- Order all responses according to fitted $\hat{p}$, then group into deciles.
- Then basically $\chi^2 = \Sigma\Sigma\frac{(O-E)^2}{E}$
- p-value = 0.1513

3.5 Appendix

3.5.1 ShinyHMCI, An R Package

4 Spatial Generalized Linear Mixed Models

4.1 Introduction

No matter the performace of the previously fit model, we expect there to be unexplained spatial variation in the mean. The covariates are limited in scope and depth, and Tobler's First Law of Geography tells us that things close together in space tend to behave more similarly than things further apart [Tobler, 1970]. Accordingly, we enhance the model to capture the unexplained spatial variation in the mean, and compensate for unobserved covariates, by adding a spatially correlated random effect [Banerjee et al., 2008].

4.1.1 Gaussian Random Field

A Gaussian random field is a popular and practical distribution for spatial random effects [Gelfand et al., 2010]. Let random variable vector $\boldsymbol{w}(\boldsymbol{s})$, for vector of locations $\boldsymbol{s} \in \boldsymbol{D} \subseteq \boldsymbol{R}^2$, be distributed multivariate Normal with mean $\mathbf{0}$; with symmetric, positive definite covariance matrix $\Sigma(\boldsymbol{\theta})$, and covariance parameter vector $\boldsymbol{\theta}$ [Haran, 2011].

$$\boldsymbol{w}|\boldsymbol{\theta} \sim MVN(\mathbf{0}, \Sigma(\boldsymbol{\theta})) \tag{3}$$

Including a random effect defined in this way, with a valid covariance matrix, would retool (2) as a *spatial* generlatized linear *mixed* model (SGLMM). Next we define the covariance structure we use in this study.

4.1.2 Exponential Covariance

To add a spatial random effect, distributed as a Gaussian random field, to the linear predictor in (2), it remains to define a spatial correlation structure to $\boldsymbol{w}$. Let w_{ij} be defined as in **5.1**, with an exponential covariance structure. That is, the i,jth element of $\Sigma(\phi, \sigma^2)$ is:

$$\Sigma(\phi, \sigma^2)_{i,j} = \sigma^2 exp(-||\boldsymbol{s}_i - \boldsymbol{s}_j||/\phi), \tag{4}$$

where $||s_i - s_j||$ is the Euclidean distance between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, σ^2 is the scale parameter, and ϕ is the range parameter.

4.1.3 Spatial Generalized Linear Mixed Model

Inserting $\boldsymbol{w}(\boldsymbol{s})$ to the linear predictor in (1) gives the following spatial generalized linear mixed model (SGLMM):

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\beta}_j + w(\boldsymbol{s}_{ij}). \tag{5}$$

This spatial hierarchical model, with its latent Gaussian random field, gives p_{ij} a complicated correlation structure. Bayesian statistical methodologies, primarily Markov chain Monte Carlo (MCMC) methods, are very popular for fitting spatial models of this kind [Banerjee et al., 2014]. In fact, MCMC is one of the few practical approaches available to fit a 'big n' model with complex spatial correlation. This is because of the 'big n problem' [Lindgren et al., 2011]. Namely, the computational costs for SGLMMs increase at a rate of $\mathcal{O}(n^3)$ (REFERENCE). This rate of increase leads to prohibitively slow model fitting. To attempt to fit SGLMMs in practically useful time spans, we try:

To estimate model parameters for (4) we tried:

1. Computational optimization, C++, an efficient algorithm, with Hamiltonian Monte Carlo in Stan
2. Dimension reduction with Predictive process models in `spBayes`
3. INLA SPDEs and `INLA-R`

Note that the ultimate goal of this research is practical, real-time applications for baseball fans, broadcasts, players, scouts, and teams. Therefore, model fitting speed matters on a finer time scale than academic research demands.

4.1.4 Markov Chains

Hierarchical Modeling and Analysis for Spatial Data [Banerjee et al., 2014]

- "Without doubt, the most popular computing tools in Bayesian practice today are Markov chain Monte Carlo (MCMC) methods."
- inference from posteriors of "...arbitrarily large dimension, essentially by reducing the problem to one of recursively solving a series of lower-dimensional (often unidimensional) problems."
- "...work by producing not a closed form for posterior, but a sample of values $\{\theta^{(g)}, g = 1, \ldots, G\}$ from this distribution." (G = number of draws from posterior)

- Two issues: MCMC algorithms produce *correlated* draws from poster (hence thinning, acf(), pacf(), and *convergence* diagnosis
- Two most popular MCMC algorithms: (1) Gibbs sampler (2) Metropolis Hastings algorithm

4.1.5 "Big N Problem"

4.2 Numerical Optimization; Hamiltonian Monte Carlo in Stan

4.2.1 Hamiltonian Dynamics and MCMC

[4] Trying to understand molecular states, Metropolis et al. [1953] created MCMC for "fast machines." Later, modeling molecular motion as a deterministic process, Alder and Wainwright [1959] introduced *Hamiltonian dynamics* as an alternate representation of Newtonian mechanics. Almost 30 years later, Duane et al. [1987] combined the two to create "hybrid Monte Carlo" to simulate certain quantum mechanical processes. Over time, this named morphed into *Hamilton* Monte Carlo (HMC), as it is known today. Eventually, Neal [1996] used HMC methods for explicitly statistical applications, studying neural networks.

HMC works by reframing the variables and distribution of interest as part of a physical system. From a physics standpoint, an object in a well defined three dimensional physical space can be completely characterized by its position and momemtum. For HMC, the variables of interest function as position variables, and auxiliary Gaussian variables are introduced serve as momentum variables. Simple updates for the auxiliary momentum variables generate, via a system of differential equations, proposals for Metropolis updates to the more important position variables (which represent the variabls of interest). The differential equation solutions estimate trajectories of the hypothetical physical object, which will then occupy a new position after some chosen time step. This crafty formulation enables distant, yet high probability, proposals for the variables of interest. Note that this contrasts favorably, in terms of mixing, to the random walk proposal generation process commonly used for Metropolis updates.

4.2.2 Hamilton Equations for MCMC

Let $q(t)$ be a d-dimensional (d parameters of interest) position vector that is a function of time t; and $U(q(t))$ represent the potential energy at time t. Let $p(t)$ give the d-dimensional momentum at time t, and $K(p(t))$

[4] The history and physics presented in this section owe heavily to, and are primarily informed by, [Neal et al., 2011]

represent kinetic energy at time t. Then the Hamilton equation,

$$H(q(t), p(t)) = U(q(t)) + K(p(t)), \tag{6}$$

measures the total energy of a system.

For HMC applications, we let the potential energy, $U(q)$, be minus the log of the probability density function of interest, plus any convenient constant[5]. Typically, HMC procedures define p as a d-dimensional zero mean Gaussian with covariance matrix M, and $K(p)$ as minus the log of the multivariate Gaussian probability density function. This gives:

$$H(q, p) = -\log f_q(q) + p^T \boldsymbol{M}^{-1} p/2. \tag{7}$$

This clever formulation provides useful partial derivatives for calculating the change in position and momentum over time. For $i = 1, \ldots, d$:

$$\frac{dq_i(t)}{dt} = \frac{\partial H}{\partial p_i}, \tag{8}$$

$$\frac{dp_i(t)}{dt} = -\frac{\partial H}{\partial q_i}. \tag{9}$$

Substituting in Hamilton's equation (5) and simplifying gives

$$\frac{dq_i(t)}{dt} = [\boldsymbol{M}^{-1} p]_i \tag{10}$$

$$\frac{dp_i(t)}{dt} = \frac{\partial \left[\log f_q(q)\right]}{\partial q_i} \tag{11}$$

The solutions to these two differential equations, that is $q(t)$ and p(t) such that (8) and (9) hold, give the instantaneous rate of change of position and momentum at time t.

subsubsectionMCMC Using Hamiltonian Dynamics [Neal et al., 2011] Steps.

- **Leapfrog method** = for calculating new position (q) and momentum (p) through tiny time steps
  - for **discretizing Hamilton equations**
  - akin to Taylor Series appoximations

[5]We omit t for clarity of presentation, here and elsewhere, but position and momentum remain functions of time t.

- Postion (q) (or momentum (p)) at t_0 plus time step times rate of change of position (q) (momentum (p)) variable at t_0
- Leapfrom Method does half step for momentum (p), full step for postion (q), other half step for momentum (p). Damn good.

- Short version: randomly sample from K(p) (kinetic, momentum), calculate U(k) (potential, position*) — that's your Metropolis proposal.

4.2.3 Optimizing in Stan

'Big n' computational burdens can also be mitigated somewhat, by certain program specific coding techniques. These techniques, as well as other techniques aimed to encourage model convergence, bolstered our modelling efforts. We highlight some such techniques for Stan here, and include the complete .stan script in Appendix A for reference.

Bayesian Motivated Techniques

Stan allows a user to omit prior distributions for parameters, but interprets non-inclusion as a non-informative, uniform prior. However, Gelman [2016], the first Stan developer, pointed out in correspondence that the exponential covariance length-scale parameter, ϕ in equation (3), requires an informative prior for model identifiability (CITATION?). Trangucci [2016] recommended, in particular, a sharp tailed prior distribution for the length-scale parameter, such as the normal or log-normal, to act as soft upper and lower bound constraints[6]. Even further, for practical computing time and convergence considerations, Trangucci [2016] said complex models such as spatial hierarchical models require proper priors for all β coefficients. Using the intial GLM estimates to inform coefficient prior distributions had exactly the intended effects.

Computational and Linear Algebra Motivated Techniques

For speed and efficiency, the Stan Users Manual recommends pure matrix algebra and vectors, over 'for loops' and scalars Stan Development Team [2016]. For example,

```
hit ~ bernoulli_logit(X*beta + Z)
```

is faster than

```
for (n in 1:N)
        hit[n] = bernoulli_logit(X[n]*beta[n] + Z[n]);
```

[6] "Without stronger priors on l, GP can act as a second constant term in your regression for large draws of length-scale and large draws of alpha."

Notice that $N \times 1$ column vectors `hit`, `beta`, and `Z` replace scalars `hit[n]`, `beta[n]`, and `Z[n]`; and $N \times p$ matix `X` replaces $1 \times p$ row vector `X[n]`.

Trangucci [2016] also suggested a QR factorization on covariate matrix $\boldsymbol{X}$, in the linear predictor, to increase computational efficiency. A QR factorization consists of factoring an $n \times p$ matrix into the product of an $n \times p$ orthogonal matrix $\boldsymbol{Q}$ and a $p \times p$ upper triangular matrix $\boldsymbol{R}$, such that $\boldsymbol{X} = \boldsymbol{QR}$.

$$\boldsymbol{X} = \boldsymbol{QR} \tag{12}$$

$$\boldsymbol{X\beta} = \boldsymbol{QR\beta} \tag{13}$$

To reparameterize for model fitting, let $\boldsymbol{\theta} = \boldsymbol{R\beta}$, so that $\boldsymbol{\beta} = \boldsymbol{R^{-1}\theta}$, which gives

$$\boldsymbol{X\beta} = \boldsymbol{Q\theta}, \text{ and} \tag{14}$$

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{Q}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\theta}_j + w_{ij}. \tag{15}$$

Now prior information about $\boldsymbol{\beta}$ should be incorporated into and given in the prior distributions of $\boldsymbol{\theta}$. Consider non-informative prior distributions on p dimensional parameter vector $\boldsymbol{\beta}$,

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I}_p),$$

where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix, and $\mathbf{0}$ is a $p \times 1$ zero vector. Notice the intended variance of the non-informative prior must be modified for $\boldsymbol{\theta}$.

$$\begin{aligned} \text{Var}(\boldsymbol{\theta}) &= \text{Var}(\boldsymbol{R\beta}) && (16)\\ &= \boldsymbol{R}\text{Var}(\boldsymbol{\beta})\boldsymbol{R}' && (17)\\ &= \boldsymbol{R}\sigma^2\boldsymbol{I}_p\boldsymbol{R}' && (18)\\ &= \sigma^2\boldsymbol{RR}' && (19) \end{aligned}$$

We add noise to the covariance matrix diagonal with the following snippet of code.

```
for (n in 1:N)
  Sigma[n, n] = Sigma[n, n] + 1e-6;
```

This added diagonal noise guarantees that the covariance matrix assembled by `cov_exp_quad(...)` remains numerically positive-definite Trangucci [2017]. This `cov_exp_quad(...)` function can generate numerically

non-positive-definite matrices when operating at high dimensions.

Finally, Carpenter [2016] recommended a Cholesky decomposition and tactical reparameterization, noting the efficiency of a vectorized scalar approach.

```
L = cholesky_decompose(Sigma);
Z ~ normal(0, 1);
Z_mod = L * Z;
hit ~ bernoulli_logit(Q*theta + Z_mod);
```

The first line performs a Cholesky decomposition on the covariance matrix `Sigma`. A Cholesky decomposition factors symmetric matrix `Sigma` such that $\Sigma = \mathbf{L}\mathbf{L}'$. The second, "vectorized scalar" line generates n standard normal random variables, by reusing `normal(0, 1)` for every element of `Z`. These two lines remove the dependence of random vector $\mathbf{Z}$, which must be generated, on unknown parameters to be estimated Trangucci [2017]. The third line transforms `Z` to have the desired distribution. Note that $\text{Var}(\mathbf{LZ}) = \mathbf{L}\text{I}_n\mathbf{L}' = \Sigma$, so that $\boldsymbol{LZ} \sim N(\mathbf{0}, \Sigma)$ as desired.

Evaluate Inverse you say?? Yes.

- $\text{logit}\{\text{EY(s)}\} = \boldsymbol{X}(s)\boldsymbol{\beta} + Z(s)$, with $Z(s) \sim MVN\{\mathbf{0}, \Sigma_s\}$
- $f(\boldsymbol{\beta}, \phi, \sigma^2, \boldsymbol{Z}|\boldsymbol{Y}) \propto f(\boldsymbol{Y}|\boldsymbol{\beta}, \phi, \sigma^2, \boldsymbol{Z})f(\boldsymbol{\beta})f(\boldsymbol{Z}|\phi, \sigma^2)f(\phi)f(\sigma^2)$
- M-H proposal, iteration i: $Z_{10,i}$

$$r = \frac{f(Z_{10,i}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \boldsymbol{\beta}_{i-1}, \phi_i, \sigma_i^2)}{f(Z_{10,i-1}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \boldsymbol{\beta}_{i-1}, \phi_i, \sigma_i^2)}$$

- Note: $f(z_1, z_2, z_3|\boldsymbol{Y}) = f(z_1|z_2, z_3, \boldsymbol{Y})f(z_2, z_3|\boldsymbol{Y})$. So...

$$r \propto \frac{f(\boldsymbol{Y}|\boldsymbol{\theta}_i)f(\boldsymbol{\beta})f(Z_{10,i}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)f(\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}|\phi_i, \sigma_i^2)f(\phi)f(\sigma^2)}{f(\boldsymbol{Y}|\boldsymbol{\theta}_{i-1})f(\boldsymbol{\beta})f(Z_{10,i-1}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)f(\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}|\phi_i, \sigma_i^2)f(\phi)f(\sigma^2)}$$

$$r \propto \frac{f(\boldsymbol{Y}|\boldsymbol{\theta}_i)f(Z_{10,i}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)}{f(\boldsymbol{Y}|\boldsymbol{\theta}_{i-1})f(Z_{10,i-1}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)}$$

- And $f(\boldsymbol{Z})$, $f(Z_i|\boldsymbol{Z}_{-i})$, etc. are $MVN\{\cdot, \Sigma^*\}$, where Σ^* either is, or is some function of, $\Sigma_{\boldsymbol{s}}$; with PDF kernal containing $\Sigma_{\boldsymbol{s}}^{*-1}$, (containing ϕ_i, σ_i^2).

4.3 Dimension Reduction; Predictive Process Models

Predictive process models (PPMs) provide a method for attempting to circumvent the "big N problem" in the case of Bayesian hierarchical models with latent Gaussian random effects. Numerous methods exist, for example [Cressie and Johannesson, 2008], (CITE OTHERS), but predictive process models provide a competitive modeling approach with computational advantages for hierarchical models with a Gaussian random field (GRF) at the second level of specification [Banerjee et al., 2008]. Latent GRFs prove challenging because they are only implicitly observed, through a binomial response in our case. This means the GRF parameters (hyperparameters) and their prior distributions comprise the third level of the hierarchical model. Gaussian PPMs achieve dimension reduction by projecting the original process onto a lower dimensioned subspace, at a set of locations called knots [Banerjee et al., 2008].

4.3.1 PPM Procedure

Consider again the SGLMM (5), and Gaussian random field $\boldsymbol{w}(\boldsymbol{s})$.

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{X}_{ij}(s_{ij})\boldsymbol{\beta}_j + w(\boldsymbol{s}_{ij}) \tag{20}$$

$$\boldsymbol{w}(\boldsymbol{s})|\boldsymbol{\theta} \sim \text{GRF}(\boldsymbol{0}, \boldsymbol{C}(\boldsymbol{\theta})) \tag{21}$$

Define (20) as in (5), and (21) shows $n \times 1$ vector of random effects $\boldsymbol{w}$ at locations $\boldsymbol{s}$, conditioned on covariance parameters $\boldsymbol{\theta}$, constitutes a GRF. Let $n \times 1$ zero vector $\boldsymbol{0}$ be the stationary GRF mean, with $n \times n$, symmetric, positive-definite covariance matrix $\boldsymbol{C}(\boldsymbol{\theta})$. Let $C(\boldsymbol{s}_i, \boldsymbol{s}_j; \boldsymbol{\theta})$ denote the covariance of random effects at locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, so that $C(\boldsymbol{\theta}) = [C(\boldsymbol{s}_i, \boldsymbol{s}_j; \boldsymbol{\theta})]_{i,j=1}^n$.

To define the PPM, start with knot locations. Let $\boldsymbol{S}^* = \{\boldsymbol{s}_1^*, \ldots, \boldsymbol{s}_m^*\}$ be a set of $m < n$ chosen knot locations, which may or may not be a subset of observed locations. We denote knot location random effects with $m \times 1$ vector $\boldsymbol{w}^* = [w(\boldsymbol{s}_i^*)]_{i=1}^m$, and the $m \times m$ knot covariance matrix and its elements as $\boldsymbol{C}^*(\boldsymbol{\theta}) = \left[C(\boldsymbol{s}_i^*, \boldsymbol{s}_j^*)\right]_{i,j=1}^m$. The knot random effects form a distinct m-dimensional GRF.

$$\boldsymbol{w}^*|\boldsymbol{\theta} \sim \text{GRF}\{\boldsymbol{0}, \boldsymbol{C}^*(\boldsymbol{\theta})\} \tag{22}$$

The predictive process modelling procedure uses the m selected knots, the covariance structure of the parent process, and kriging to interpolate w at site $\boldsymbol{s}_0$ [Schabenberger and Gotway, 2004]; See Appendix ?? for kriging details. Let $\tilde{w}(\boldsymbol{s}_0)$ represent this interpolated random effect, and let $\boldsymbol{c}(\boldsymbol{s}_0; \boldsymbol{\theta}) = \left[C(\boldsymbol{s}_0, \boldsymbol{s}_j^*; \boldsymbol{\theta})\right]_{j=1}^m$ be

an $m \times 1$ covariance vector giving the covariance of the $\boldsymbol{s}_0$ random effect with the knot random effects.

$$\tilde{w}(\boldsymbol{s}_0) = E[w(\boldsymbol{s}_0)|\boldsymbol{w}^*] \tag{23}$$

$$= \boldsymbol{c}^T(\boldsymbol{s}_0;\boldsymbol{\theta}) \cdot \boldsymbol{C}^{*-1}(\boldsymbol{\theta}) \cdot \boldsymbol{w}^* \tag{24}$$

For a GRF, the weights of linear combination (24) minimize the squared error loss function among all linear predictors [Schabenberger and Gotway, 2004]; and notice that the linear combination varies spatially. Accordingly, predictive process $\tilde{w}(\boldsymbol{s})$ defines another GRF and covariance matrix.

$$\tilde{\boldsymbol{w}}(\boldsymbol{s}) \sim \text{GRF}\{0, \tilde{C}(\cdot)\} \tag{25}$$

$$\tilde{C}(\boldsymbol{s}, \boldsymbol{s}';\boldsymbol{\theta}) = \boldsymbol{c}^T(\boldsymbol{s};\boldsymbol{\theta}) \cdot \boldsymbol{C}^{*-1}(\boldsymbol{\theta}) \cdot \boldsymbol{c}(\boldsymbol{s}';\boldsymbol{\theta}) \tag{26}$$

To reiterate, $m \times 1$ vector $\boldsymbol{c}(\boldsymbol{s};\boldsymbol{\theta}) = \left[C(\boldsymbol{s}, \boldsymbol{s}_j^*)\right]_{j=1}^m$ gives the covariance of the random effect at $\boldsymbol{s}$ with knot random effects. Finally, the predictive process model:

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\beta}_j + \tilde{w}(\boldsymbol{s}) \tag{27}$$

4.3.2 Improved Predictive Process Models

4.4 Approximation; SPDE and INLA

Integrated Nested Laplace Approximation (INLA), a mathematically intensive and computationally geared approximation technique, works well for Bayesian hierarchical models with latent Gaussian **markov** random fields (GMRFs) [Rue and Martino, 2007]. A GMRF, by virtue of its sparse precision matrix, enables INLA's orders of magnitude faster approximation method. However, to use INLA for continuous domain spatial models with latent Gaussian random fields (GRFs), one must represent the GRF as a GMRF; and stochastic calculus provides a link. A particular stochastic partial derivative of a Matern GRF equals a Gaussian random field white noise process. This identity provides a platform on which to approximate a Matern GRF with a GMRF, in the form of a piecewise linear basis representation. The discrete representation consists of deterministic basis functions, defined by a triangulation of the domain, and GMRF weights. The basis representation has a sparse precision matrix, and thus qualifies for the INLA approximation and its computational advantages. Scientists use this GRF to GMRF translation process—projecting the SPDE onto the basis representation—known as Finite Element Method, extensively in other fields.

4.4.1 Gaussian Markov Random Fields

4.4.2 Stochastic Partial Differential Equation (SPDE)

The exponential convariance function is a member of the larger Matern family, defined by the following covariance function.

$$\mathrm{C}(h) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\kappa h)^{\nu} K_{\nu}(\kappa h)$$

This parameterization includes range parameter $\kappa > 0$, smoothness parameter $\nu > 0$, scale parameter σ^2, and modified Bessel function $K_{\nu}(\cdot)$ [Schabenberger and Gotway, 2004]. While $\nu = 1/2$ defines the exponential covariance function, Whittle [1954] declared Matern($\nu = 1$) the "elementary correlation" function in two dimensions. Both functions yield a similarly decaying-with-distance spatial covariance, but a Matern random field solves the following SPDE [Whittle, 1954].

$$(\kappa^2 - \Delta)^{\alpha/2} x(\boldsymbol{s}) = \mathcal{W}(\boldsymbol{s}),$$

This includes $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$, the Laplace operator; spatial scale parameter κ, as in the Matern; smoothness parameter α; and Gaussian spatial white noise process $\mathcal{W}(\boldsymbol{s})$. The particular SPDE and Matern coupling dictates $\alpha = \nu + d/2$, where d = 2 for $\mathbb{R}^2$; and

$$\sigma^2 = \frac{\Gamma(\nu)}{\Gamma(\alpha)(4\pi)^{d/2}\kappa^{2\nu}}.$$

Based on Whittle [1954] and Mondal [2017] we use $\nu = 1$, which implies $\alpha = 2$. This specification simplifies the SPDE to

$$(\kappa^2 - \Delta) x(\boldsymbol{s}) = \mathcal{W}(\boldsymbol{s});$$

the Matern covariance to

$$\mathrm{C}(h) = \sigma^2(\kappa h) K_1(\kappa h);$$

and the variance to $\sigma^2 = \frac{1}{4\pi\kappa}$.

Piecewise Linear Basis Representation

The next step, known as the Finite Element Method, projects the SPDE onto a piecewise linear basis representation [Simpson et al., 2012]. The basis represenation,

$$x(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s}) x_k,$$

contains deterministic basis functions, $\psi_k(\cdot)$; and weights $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, which constitute a GMRF. The two combine so that the distribution of $x(\boldsymbol{s})$ approximates the Matern GRF that solves the SPDE, but retains a sparse precision matrix and its accordant computational advantages.

Deterministic Basis Function

A triangulation of the domain defines the deterministic component, $\psi_k(\boldsymbol{s})$, of the basis representation. Figure 12 [Simpson et al., 2012] illustrates how domain triangulation and determines a basis function.

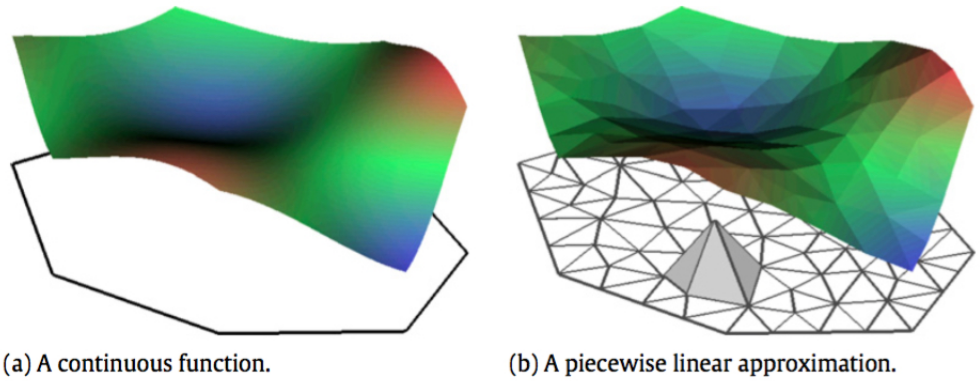

Figure 12: A Gaussian Markov random field, defined as the piecewise linear basis function $x(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s}) x_k$, approximates a Matern GRF. This image illustrates how a triangular mesh over the domain determines basis functions $\psi_k(\boldsymbol{s})$ [Simpson et al., 2012].

Keep in mind that a realization of a GRF is essentially a function, and Figure 12 shows how a discrete piecewise linear basis function can approximate a continuous function [Simpson et al., 2012]. We have $\psi_k(\boldsymbol{s}) = 1$ at the kth vertex, 0 at all other vertices, and surface function values for triangle interior points are linear combinations of the three home triangle vertices.

GMRF Weights

Stochastic calculus identities provide a way to calculate the weights in the basis representation. Define $\langle f, g\rangle = \int f(\boldsymbol{u})g(\boldsymbol{u})d\boldsymbol{u}$, and find weights $\boldsymbol{x}$ such that

$$\left[\left\langle \phi_k, (\kappa^2 - \Delta)^{\alpha/2}\boldsymbol{x}\right\rangle\right]_{k=1,\ldots,n} \stackrel{D}{=} \left[\langle \phi_k, \mathcal{W}\rangle\right]_{k=1,\ldots,n},$$

for a set of test functions ϕ_k [Lindgren et al., 2011]. The appropriate weight vector $\boldsymbol{x}$ gives the stochastic weak solution solution to the SPDE [Mao, 2007], Lindstrom [2014]; we use the Galerkin solution, with $\alpha = 2$ and $\phi_i = \psi_i$ [Lindgren et al., 2011]. Replace $\boldsymbol{x}$ with basis function representation $\Sigma_k \psi_k w_k$,

$$\left[\left\langle \phi_i, (\kappa^2 - \Delta)^{\alpha/2}\psi_j\right\rangle\right]_{i,j} \boldsymbol{w} \stackrel{D}{=} \left[\langle \phi_k, \mathcal{W}\rangle\right]_{k},$$

and let $\alpha = 2$ and $\phi_i = \psi_i$ as in the Galerkin solution:

$$\Big(\kappa^2[\langle \psi_i, \psi_j\rangle] + [\langle \psi_i, -\Delta\psi_j\rangle]\Big)\boldsymbol{w} \stackrel{D}{=} \Big[\langle \psi_k, \mathcal{W}\rangle\Big].$$

Let $\boldsymbol{C}_{i,j} = \langle \psi_i, \psi_j\rangle$, and $\boldsymbol{G}_{i,j} = \langle \psi_i, -\Delta\psi_j\rangle$, so that

$$\left(\kappa^2\boldsymbol{C} + \boldsymbol{G}\right)\boldsymbol{w} \stackrel{D}{=} N(\boldsymbol{0}, \boldsymbol{C}).$$

For $\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{Q}^{-1})$, we have then

$$\boldsymbol{Q}_\kappa = \left(\kappa^2\boldsymbol{C} + \boldsymbol{G}\right)^T \boldsymbol{C}^{-1}\left(\kappa^2\boldsymbol{C} + \boldsymbol{G}\right).$$

However, $\boldsymbol{C}_{ij}^{-1}$ has a sparse precision matrix, so replace $\boldsymbol{C}$ with diagonal matrix $\widetilde{\boldsymbol{C}}$,

$$\widetilde{\boldsymbol{C}}_{i,i} = \langle \psi_i, \boldsymbol{1}\rangle = \int \psi_i(\boldsymbol{s})d\boldsymbol{s}.$$

Note that this solution provides the distribution of the weights x_k, not $x(\boldsymbol{s})$ itself, in the basis representation

$$x(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s})x_k.$$

With this representation achieved, via the SPDE link, we move on to the approximation procedure that we aim for.

4.4.3 Integrated Nested Laplace Approximations (INLA)

INLA proceeds through a carefully constructed and calibrated series of calculations and approximations, to acheive estimates of key quantites. In this section, we define $\boldsymbol{x}(\boldsymbol{s}) = (\boldsymbol{\beta}, \boldsymbol{w}(\boldsymbol{s}))$, where $\boldsymbol{\beta}$ has a Normal prior distributiona as required by INLA methods. The key quantites, then, include the maginal posterior distributions for latent field parameters, $p(x_i|\boldsymbol{y})$; and covariance hyperparameter posterior $p(\theta|\boldsymbol{y})$. This means we never obtain posteriors $p(\boldsymbol{x}|\boldsymbol{y})$ and $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. As a basic INLA roadmap, this section includes calculating:

1. $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$, directly.
2. $\tilde{p}(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$, as a Gaussian approximation of $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$.
3. $\boldsymbol{x}_0 = \text{argmax}_x \tilde{p}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}_0)$, or the posterior mode of $\tilde{p}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta})$ for a given $\boldsymbol{\theta} = \boldsymbol{\theta}_0$.
4. $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$, using a basic probability identity, Gaussian approximation $\tilde{p}(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$, evaluated at posterior mode $\boldsymbol{x}_0$.
5. $p(x_i|\boldsymbol{y})$, by way of numerical integration of $\tilde{p}(x_i|\boldsymbol{\theta}, \boldsymbol{y})\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$ over all $\boldsymbol{\theta}$.

Note again:

- $p(\boldsymbol{x}|\boldsymbol{y})$ — not estimated
- $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$ — not estimated

Step 1, Gaussian Approximation

The basic approach to estimating $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ is "matching the mode and curvature at the mode" of estimator $\tilde{p}(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ to that of $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ [Rue and Held, 2005]. As mentioned, INLA requires Gaussian priors for all paramters except covariance hyperparameters; but, INLA also requires conditional independence, whereby $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \prod_i p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\theta})$. Our analysis satisfies this necessity, and therefore

$$p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \sum_i \log\, p(y_i|x_i)\right).$$

The Gaussian approximation takes the following form,

$$p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{Q}+\text{diag}(\boldsymbol{c}))(\boldsymbol{x}-\boldsymbol{\mu})\right),$$

where $\boldsymbol{c}$ and $\boldsymbol{\mu}$ depend on a second order Taylor expansion of $f(\boldsymbol{x}) = \sum_i \log p(y_i|x_i)$, about some $\boldsymbol{x}_0$ [Rue and Martino, 2007], [Lindstrom, 2014]. Various methods exist for determining the form of $\boldsymbol{c}$ and $\boldsymbol{\mu}$ [Rue et al., 2009]. For example, for latent random field $\boldsymbol{x} \sim \text{N}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})$, Lindstrom [2014] gives the following mean and variance for the Gaussian approximation, :

$$\text{E}_{\boldsymbol{x}_0}(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) \approx \big(\boldsymbol{Q} - \text{diag}(f''(\boldsymbol{x}_0))\big)^{-1}\big(\boldsymbol{Q}\boldsymbol{x}_0 + f'(\boldsymbol{x}_0) - f''(\boldsymbol{x}_0)\boldsymbol{x}_0\big) \tag{28}$$

$$\text{V}_{\boldsymbol{x}_0}(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) \approx \big(\boldsymbol{Q} - \text{diag}(f''(\boldsymbol{x}_0))\big)^{-1} \tag{29}$$

Step 2 uses this Gaussian approximation, $p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$.

Step 2, Laplace Approximation

This step begins with two sides of a familiar identity, and its subsequent rearrangement.

$$p(\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{30}$$

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{31}$$

$$\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})} = p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{32}$$

We use this formulation of $p(\boldsymbol{y}|\boldsymbol{\theta})$ next, in the familiar Bayesianian proporionality.

$$p(\theta|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{33}$$

$$\propto \frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})} \cdot p(\boldsymbol{\theta}) \tag{34}$$

For a given $\boldsymbol{\theta}$, let $\boldsymbol{x}_0 = \text{argmax}_x p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$. Then,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \approx \tilde{p}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{p(\boldsymbol{y}|\boldsymbol{x}_0,\boldsymbol{\theta})p(\boldsymbol{x}_0|\boldsymbol{\theta})}{p_G(\boldsymbol{x}_0|\boldsymbol{y},\boldsymbol{\theta})} \cdot p(\boldsymbol{\theta}),$$

where the Taylor approximation of $f(\boldsymbol{x}) = \sum_i \log p(y_i|x_i)$, in $p_G(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$, expands about $\boldsymbol{x}_0$. This approximation matches Tierney and Kadane [1986] Laplace approximation. Then, we have approximate maximum

likelihood estimate $\hat{\boldsymbol{\theta}}_{\text{ML}} \approx \text{argmax}_{\theta}\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$, a

Step 3, Numerical Integration

Numerical Integration over $\boldsymbol{\theta}$, or elements of $\boldsymbol{\theta}$, gives $p(x_i|\boldsymbol{y})$ and $p(\theta_i|\boldsymbol{y})$.

$$p(x_j|\boldsymbol{y}) \approx \int p_{\text{G}}(x_j|\boldsymbol{\theta}, \boldsymbol{y})\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}$$

$$p(\theta_k|\boldsymbol{y}) \approx \int \tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}_{-k}$$

In summary, from a continuous domain the SPDE-INLA approxmation produces provide, with improved speed, posterior estimates for $p(\boldsymbol{\theta}|\boldsymbol{y})$, $p(\theta_i|\boldsymbol{y})$, and $p(x_i|\boldsymbol{y})$. The R package `INLA` implements this procedure, with flexible SPDE specifications for domain triangulation.

4.4.4 Bayesian Inference in R-INLA

A Stan Code

```
data {
  int<lower=0> N;                 // N observations
  int<lower=0> p;                 // p predictors
  matrix[N,p] Q;                  // QR decomp - Q
  matrix[p,p] R;                  // QR decomp - R
  int<lower=0, upper=1> hit[N];   // 0/1 outcomes; array of integers
  vector[2] px_pz[N];             // N-dim array of 2-dim vectors
  vector[p] theta_SDs;            // theta prior SDs
}
transformed data{
  matrix[p,p] R_inv;
  R_inv = inverse(R);
}
parameters {
  real<lower=0> l;                  // length-scale parameter
  real<lower = 0> sigma;            // scale parameter
  real beta0;                       // intercept
  vector[p] theta;
  vector[N] Z;                      // location random effect
}
transformed parameters {
      vector[p] beta;
      beta = R_inv*theta;
}
model {
  matrix[N, N] Sigma;
  matrix[N, N] L;                     // Lwr triangular Cholesky decomp
```

```
  vector[N] Z_mod;

      l ~ lognormal(-2,1);            // E[l] = 0.223
      sigma ~ lognormal(-1.5, 1.5);   // E[sigma] = 0.687
      beta0 ~ normal(0,5);

      theta[1] ~ normal(0, theta_SDs[1]);
      theta[2] ~ normal(0, theta_SDs[2]);
      theta[3] ~ normal(0, theta_SDs[3]);
      theta[4] ~ normal(0, theta_SDs[4]);
      theta[5] ~ normal(0, theta_SDs[5]);
      theta[6] ~ normal(0, theta_SDs[6]);

      Sigma = cov_exp_quad(px_pz, sigma, l);
      for (n in 1:N)
        Sigma[n, n] = Sigma[n, n] + 1e-6;
      L = cholesky_decompose(Sigma); // Sigma = LL'

      Z ~ normal(0, 1);  // Each element is N(0,1)
      Z_mod = L * Z; // (Cov matrix Cholesky)*MVN(0,1)

      hit ~ bernoulli_logit(beta0 + Q*theta + Z_mod);
}
```

B R Code, spBayes

C R-INLA Code

D Kriging

From: "Statistical Methods for Spatial Data Analysis" [Schabenberger and Gotway, 2004]

The mean is known—Simple Kriging

1. Spatial data: $\mathbf{Z}(\mathbf{s}) = [Z(s_1), Z(s_2), \ldots, Z(s_n)]'$
2. Assume: $\mathbf{Z}(\mathbf{s}) = \mu(\mathbf{s}) + \mathbf{e}(\mathbf{s})$, $\mathbf{e}(\mathbf{s}) \sim (\mathbf{0}, \boldsymbol{\Sigma})$, where $\mathbf{0}, \boldsymbol{\Sigma}$ known.
3. Goal: find predictor $p(\mathbf{Z}; s_0)$, of $Z(s_0)$, that minimizes $E\left[\left(p(\mathbf{Z}; s_0) - Z(s_0)\right)^2\right]$
4. Consider only linear predictors of the form: $p(\mathbf{Z}; s_0) = \lambda_0 + \lambda'\mathbf{Z}(\mathbf{s})$
5. Expand, simplify, set derivative equal to zero.
6. Note: $\text{Var}[Z(s_0)] = \sigma^2$, and $\tilde{\sigma}$ =$\text{Cov}[\mathbf{Z}(\mathbf{s}), Z(s_0)]$.

7. $\tilde{\sigma}$ =Cov$[\mathbf{Z}(\mathbf{s}), Z(s_0)]$

8. Estimators for unknown λs:

$$\lambda_0 = \mu(s_0) - \lambda'\mu(\mathbf{s})$$

$$\lambda = \mathbf{\Sigma^{-1}}\tilde{\sigma}$$

9. Optimal predictor:

$p(\mathbf{Z}; s_0) = \mu(s_0) - \lambda'\mu(\mathbf{s}) + \lambda'\mathbf{Z}(\mathbf{s})$

$p(\mathbf{Z}; s_0) = \mu(s_0) - (\mathbf{\Sigma^{-1}}\tilde{\sigma})'\mu(\mathbf{s}) + (\mathbf{\Sigma^{-1}}\tilde{\sigma})'\mathbf{Z}(\mathbf{s})$

$p(\mathbf{Z}; s_0) = \mu(s_0) + \tilde{\sigma}'\mathbf{\Sigma^{-1}}(\mathbf{Z}(\mathbf{s}) - \mu(\mathbf{s}))$

10. Pay special attention to: $\tilde{\sigma}'\mathbf{\Sigma^{-1}}\mathbf{Z}(\mathbf{s})$

    - Recall: $\tilde{\sigma}$ =Cov$[\mathbf{Z}(\mathbf{s}), Z(s_0)]$
    - Data: $\mathbf{Z}(\mathbf{s}) = [Z(s_1), Z(s_2), \ldots, Z(s_n)]'$
    - Random effect: $\mathbf{e}(\mathbf{s}) \sim (\mathbf{0}, \mathbf{\Sigma})$

11. Look familiar?

12. **Best** predictor under squared error loss *if* $\mathbf{Z}(\mathbf{s})$ is GRF .

References

Christian M Welch, Scott A Banks, Frank F Cook, and Pete Draovitch. Hitting a baseball: A biomechanical description. Journal of Orthopaedic & Sports Physical Therapy, 22(5):193–201, 1995.

Ross Sheldon et al. A first course in probability. Pearson Education India, 2002.

Raymond H Myers, Douglas C Montgomery, G Geoffrey Vining, and Timothy J Robinson. Generalized linear models: with applications in engineering and the sciences, volume 791. John Wiley & Sons, 2012.

Alastair J Cochran and John Stobbs. Search for the perfect swing. Triumph, 2005.

B Elliott. Biomechanics and tennis. British Journal of Sports Medicine, 40(5):392–396, 2006.

Waldo R Tobler. A computer movie simulating urban growth in the detroit region. Economic geography, 46 (sup1):234–240, 1970.

Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70 (4):825–848, 2008.

Alan E Gelfand, Peter Diggle, Peter Guttorp, and Montserrat Fuentes. Handbook of spatial statistics. CRC press, 2010.

Murali Haran. Gaussian random field models for spatial data. Handbook of Markov Chain Monte Carlo, pages 449–478, 2011.

Sudipto Banerjee, Bradley P Carlin, and Alan E Gelfand. Hierarchical modeling and analysis for spatial data. Crc Press, 2014.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(4):423–498, 2011.

Radford M Neal et al. Mcmc using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 2: 113–162, 2011.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087–1092, 1953.

Berni J Alder and T.E. Wainwright. Studies in molecular dynamics. i. general method. The Journal of Chemical Physics, 31(2):459–466, 1959.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. Physics letters B, 195(2):216–222, 1987.

Radford M Neal. Priors for infinite networks. In Bayesian Learning for Neural Networks, pages 29–53. Springer, 1996.

Andrew Gelman. personal communication, November 2016.

Rob Trangucci. personal communication, November 2016.

Stan Development Team. Stan Modeling Language User's Guide and Reference Manual, Version 2.14.1, 2016. URL `http://mc-stan.org/`.

Rob Trangucci. Hierarchical gaussian processes in stan. StanCon Contributed Talks, 2017. Conference.

Bob Carpenter. personal communication, November 2016.

Noel Cressie and Gardar Johannesson. Fixed rank kriging for very large spatial data sets. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(1):209–226, 2008.

Oliver Schabenberger and Carol A Gotway. Statistical methods for spatial data analysis. CRC press, 2004.

HÅvard Rue and Sara Martino. Approximate bayesian inference for hierarchical gaussian markov random field models. Journal of statistical planning and inference, 137(10):3177–3192, 2007.

Peter Whittle. On stationary processes in the plane. Biometrika, pages 434–449, 1954.

Debashis Mondal. Personal communication, February 2017.

Daniel Simpson, Finn Lindgren, and Håvard Rue. Think continuous: Markovian gaussian models in spatial statistics. Spatial Statistics, 1:16–29, 2012.

Xuerong Mao. Stochastic differential equations and applications. Elsevier, 2007.

Johan Lindstrom. Gaussian markov random fields. Lecture Slides, 2014. Pan-American Advanced Study Institute, Buzios.

Havard Rue and Leonhard Held. Gaussian Markov random fields: theory and applications. CRC Press, 2005.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. Journal of the royal statistical society: Series b (statistical methodology), 71(2):319–392, 2009.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. Journal of the american statistical association, 81(393):82–86, 1986.