Improved prediction accuracy for disease risk mapping using Gaussian Process stacked generalisation - Supplementary information

Samir Bhatt[1,*], Ewan Cameron[2], Seth R Flaxman[4], Daniel J Weiss[2], David L Smith[3], and Peter W Gething[2]

[1] *Department of Infectious Disease Epidemiology, Imperial College London, London W2 1PG, UK*
[2] *Oxford Big Data Institute, Nuffield Department of Medicine, University of Oxford, Oxford OX3 7BN, UK*
[3] *Institute for Health Metrics and Evaluation, University of Washington, Seattle, Washington 98121, USA*
[4] *Department of Statistics, University of Oxford, 24-29 St Giles, Oxford OX1 3LB, UK*
[*] *Corresponding author: bhattsamir@gmail.com*

1 Supplementary information

1.1 Generalisation methods

We choose 5 level 0 generalisers to feed into the level 1 generaliser. We chose these 5 due to (a) ease of implementation through existing software packages, (b) the differences in their approaches and (c) a proven track record in predictive accuracy.

1.1.1 Gradient boosted trees and Random forests

Both Gradient boosted trees and random forests produce ensembles of regression trees [1]. Regression trees partition the space of all joint covariate variable values into disjoint regions R_j ($j = \{1, 2, ..., J\}$) which are represented as terminal nodes in a decision tree. A constant γ_j is assigned to each region such that the predictive rule is $x \in R_j \rightarrow f(x) = \gamma_j$ [2]. A tree is

therefore formally expressed as $T(x,\theta) = \sum_{j=1}^{J} \gamma_j \mathbb{I}(x \in R_j)$.

Gradient boosted trees model the target function by a sum of such trees induced by forward stagewise regression

$$f_M(x) = \sum_{m=1}^{M} T(x,\theta_m) \quad (1)$$

$$\hat{\theta}_m = \arg\min_{\theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i,\theta_m)) \quad (2)$$

Solving equation 1 is done by functional gradient decent with regularisation performed via shrinkage and cross validation [3]. In our implementations we also stochastically sampled covariates and samples in each stagewise step [4].

Random forests combine trees by bagging [2] where bootstrap samples (B) of covariates and data are used to create an average ensemble of trees:

$$f_M(x) = \frac{1}{B} \sum_{b=1}^{B} T(x,\theta_b) \quad (3)$$

The optimal number of bootstraped trees B is found by cross validation [5].

We used the H2O package in R to fit both the gradient boosted models and random forest models and meta-parameters such as tree depth and samples per node were evaluated using a coarse grid search.

1.1.2 Elastic net regularised regression

Elastic net regularised regression [6] is a penalised linear regression where the coefficients of the regression are found by

$$f_M(x) = X^T \hat{\gamma} \quad (4)$$

$$\hat{\gamma} = \arg\min_{\gamma} ||y - f_M(x)||^2 + \lambda_2 ||\gamma||^2 + \lambda_1 ||\gamma||_1 \quad (5)$$

Where the subscripts on the penalty terms represent the ℓ_1 (i.e. lasso) and ℓ_2 (i.e. ridge) norms. These norms induce shrinkage in the coefficients of the linear model allowing for better generalisation.

Equation 4 is fitted by numerically maximising the likelihood and optimal parameters for λ_1, λ_2 are computed by cross validation over the full regularisation path. Fitting was done using the H2O package in R.

1.1.3 Generalised additive splines

Generalised additive splines extend standard linear models by allowing the linear terms to be modelled as nonlinear flexible splines [7].

$$f_M(x) = \gamma_0 + f_1(x_1) + , .. , + f_m(x_m) \tag{6}$$

$$s.t. \quad \arg\min_{\beta} ||y - f_M(x)||^2 + \lambda \int (f_M^{''}(x))^2 dx \tag{7}$$

Where $f^{''}$ denotes the second derivative and penalises non smooth functions (that can potentially overfit). Fitting was done using generalised cross validation with smoothness selection done via restricted maximum likelihood. Fitting was done using the mgcv package in R.

Multivariate adaptive regression splines

Multivariate adaptive regression splines [8] build a model of the form

$$f_M(x) = \sum_{i=m}^{M} \gamma_i V_i(x) \tag{8}$$

Where β_is are coefficients and V_is are basis functions that can either be constant, be a hinge function of the form $max(0, x - const), max(0, const - x)$, or the product of multiple hinge functions. Fitting is done using a two stage approach with a forward pass adding functions and a backward pruning pass via generalised cross validation. Fitting was done using the earth package in R.

1.2 Details of the Gaussian Process

For the Gaussian process spatial component the covariance function is chosen to be Matérn of smoothness $\nu = 1$:

$$k_\theta(s_i, s_j) = \kappa/\tau \|s_i - s_j\| \mathcal{K}_1^{(2)}(\kappa \|s_i - s_j\|), \tag{9}$$

with $\kappa = \sqrt{2}/\rho$ an inverse range parameter (for range, ρ), τ a precision (i.e., inverse variance) parameter, and $\mathcal{K}_1^{(2)}$ the modified Bessel function of the second kind and order 1. Typically, θ is defined with elements $\{\log \kappa,\ \log \tau\}$ to ensure positivity via the exponential transformation. For computational efficiency and scalability we follow the stochastic partial differential equation (SPDE) approach [9] to approximate the continuous Gaussian process in Equation 1 (main manuscript) with a discrete Gauss-Markov random field (GRMF) of sparse precision matrix, Q_θ $[= \Sigma_\theta^{-1}]$, allowing for fast computational matrix inversion [10]. To find the appropriate GMRF structure, the Matérn [11] spatial component is parametrised as the finite element solution to the SPDE, $(k^2 - \triangle)(\tau f(s)) = \mathcal{W}(s)$ defined on a spherical manifold, $\mathbb{S}^2 \in \mathbb{R}^3$, where $\triangle = \frac{\partial}{\partial s_{(1)}^2} + \frac{\partial}{\partial s_{(2)}^2} + \frac{\partial}{\partial s_{(3)}^2}$ is the Laplacian (for Cartesian coordinates $s_{(1)}, s_{(2)}, s_{(3)}$) and $\mathcal{W}(s)$ is a spatial white noise process [9].

To extend this spatial process to a spatio-temporal process, temporal innovations are modelled by first order autoregressive dynamics:

$$f(s_i, t) = \phi f(s_i, t-1) + \omega(s_i, t), \tag{10}$$

where $|\phi| < 1$ is the autoregressive coefficient and $\omega(s_i, t)$ is iid Normal. Practically the spatio-temporal process is achieved through a Kronecker product of the (sparse) spatial SPDE precision matrix and the (sparse) autoregressive temporal precision matrix.

As specified above in our GMRF implementation we instead use the precision (inverse covariance) matrix. Using the precision matrix the conditional predictive distribution takes the form

$$z|y, \theta \sim N(\mu^*, Q^{-1*}) \tag{11}$$

$$\mu^* = \mu_{(s',t')|\theta} + Q^{-1}_{(s',t'),(s^\circ,t^\circ)|\theta} A^T Q_{y|(s^\circ,t^\circ),\theta} \left(y - A\mu_{(s^\circ,t^\circ)|\theta}\right) \tag{12}$$

$$Q^{-1*} = Q^{-1}_{(s',t'),(s^\circ,t^\circ)|\theta} \tag{13}$$

Where $Q_{(s',t'),(s^\circ,t^\circ)|\theta} = Q_{(s',t')|\theta} + A^T Q_{y|(s^\circ,t^\circ),\theta} A$. In Equation 12 A is introduced as a sparse observation matrix that maps the finite dimensional GMRF at locations (s°, t°) to functional

evaluations at any spatio-temporal locations e.g prediction locations (s', t') or data locations (s, t), provided these locations are within a local domain.

1.3 Bias variance derivation for a Gaussian process stacked generaliser

Theorem 1. *Consider a function* $\{f : \mathbb{R}^N \rightarrow \mathbb{R}\}$ *for which a sample* $D = \{x_i, y_i\}$ *exists, where* $y_i = f(x_i)$ *and* $i = \{1, ..., n\}$. *Fit* $\mathcal{L}$ *level 0 generalisers,* $M_1(x), .., M_{\mathcal{L}}(x)$, *trained on data* D. *Next define two ensembles of the* $\mathcal{L}$ *models, the first using a weighted mean,* $\bar{M}_{cwm}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i M_i(x)$, *and the second as the mean of a Gaussian process,* $\bar{M}_{gp}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i M_i(x) + \Sigma_2 \Sigma_1^{-1} \left(f(x) - \sum_{i=1}^{\mathcal{L}} \beta_i M_i(x) \right)$, *with* β *subject to convex combinations for both models. If the squared error is taken for both ensembles i.e.* $e_{cwm}(x) = (f(x) - \bar{M}_{cwm}(x))^2$ *and* $e_{gp}(x) = (f(x) - \bar{M}_{gp}(x))^2$, *then from the contribution of the covariance* $(\mathbb{I} - \Sigma_2 \Sigma_1) e_{gp}(x) \leq e_{cwm}(x)\ \forall x$

Proof. Consider a function f from $\mathbb{R}^N$ to $\mathbb{R}$ for which a sample $D = \{x_i, y_i\}$ exists, where $y_i = f(x_i)$ and $i = \{1, ..., n\}$. fit $\mathcal{L}$ level 0 generalisers, $M_1(x), .., M_L(x)$ (see Algorithm 1 in main manuscript).

Consider two ensembles, $\bar{M}$, of the $M_1(x), .., M_{\mathcal{L}}(x)$ models

$$\bar{M}_{cwm}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i M_i(x) \tag{14}$$

$$\bar{M}_{gp}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i M_i(x) + \Sigma_2 \Sigma_1^{-1} \left(f(x) - \sum_{i=1}^{L} \beta_i M_i(x) \right) \tag{15}$$

With convex combination constraints $\beta_i \geq 0 \forall i$ and $\sum_{i=1}^{L} \beta_i = 1$.

Model 1 (Equation 14), referred too here as a constrained weighted mean, is the predominant ensemble approach taken previously [12–15]. Model 2 (equation 15), is the conditional expectation of ensembling via Gaussian process regression see (main manuscript and [11]). To simplify notation here $\Sigma_2 = \Sigma_{(s',t'),(s,t)|\theta}$ and $\Sigma_1 = \Sigma_{y|(s,t),\theta}$.

Define the squared error between the target function and each individual level 0 generaliser as $\epsilon_i(x) = (f(x) - M_i(x))^2$. Define the squared error between the target function and the

ensemble as $e(x) = (f(x) - \bar{M}(x))^2$. Following from [14] define the ambiguity of a given model as $a_i(x) = (\bar{M}(x) - M_i(x))^2$.

As derived in [14] the ensemble ambiguity from using a constrained weighted mean ensemble (model 1 above) subject to convex combinations is defined as the weighted sum of the individual ambiguities.

$$\bar{a}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i a_i(x) = \sum_{i=1}^{\mathcal{L}} \beta_i (\bar{M}(x) - M_i(x))^2 \tag{16}$$

$$\bar{a}(x) = \bar{\epsilon}(x) - e(x) \tag{17}$$

where $\bar{\epsilon}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i \epsilon_i(x)$. Therefore the ensemble squared error when using a constrained weighted mean is $e_{cwm}(x) = \bar{\epsilon}(x) - \bar{a}(x)$.

Using the Gaussian process (model 2 above) the ensemble ambiguity is defined as.

$$\begin{aligned} \bar{a}(x) &= \sum_{i=1}^{\mathcal{L}} \beta_i (\bar{M}(x) - M_i(x))^2 \\ &+ \Sigma_{2^*} \Sigma_{1^*}^{-1} \left(\sum_{i=1}^{\mathcal{L}} \beta_i \big(f(x) - M_i(x)\big)^2 - \sum_{i=1}^{\mathcal{L}} \beta_i \big(\bar{M}(x) - M_i(x)\big)^2 \right) \end{aligned} \tag{18}$$

$$\bar{a}(x) = \sum_{i=1}^{\mathcal{L}} \beta_i a_i(x) + \Sigma_{2^*} \Sigma_{1^*}^{-1} \left(\sum_{i=1}^{\mathcal{L}} \beta_i e_i(x) - \sum_{i=1}^{\mathcal{L}} \beta_i a_i(x) \right) \tag{19}$$

In equations 18 and 19 above the covariance matrices operate on the ambiguities, and as such are suffixed with asterixes to distinguish those from equation 15. Additionally β in equations 18 and 19 are also subject to convex combination constraints. Substituting 17 into equation 19 yields:

$$\bar{a}(x) = \bar{\epsilon}(x) - e(x) + \Sigma_{2^*} \Sigma_{1^*}^{-1} \Big(\bar{\epsilon}(x) - \bar{\epsilon}(x) + e(x) \Big) \tag{20}$$

$$\bar{a}(x) = \bar{\epsilon}(x) - e(x) + \Sigma_{2^*} \Sigma_{1^*}^{-1} e(x) \tag{21}$$

$$\Big(\mathbb{I} - \Sigma_{2^*} \Sigma_{1^*}^{-1} \Big) e_{gp}(x) = \bar{\epsilon}(x) - \bar{a}(x) \tag{22}$$

The right hand side of equation 22 is identical to that derived using a constrained weighted mean in equation 17, but the left hand side error term, $e_{gp}(x)$ is augmented by $(\mathbb{I} - \Sigma_{2^*} \Sigma_{1^*}^{-1})$. Clearly $\bar{a}(x) \leq \bar{\epsilon}(x)\ \forall x$, with $\bar{a}(x) = \bar{\epsilon}(x)$ only when the ensemble equals the true target function, $\bar{M} = f(x)$, and $e(x) = 0$. It follows that the left hand side of equation 22 $\Big(\mathbb{I} - \Sigma_{2^*} \Sigma_{1^*} \Big) e(x) \geq 0\ \forall x$.

Therefore from the contribution of the precision or covariance stacking using a Gaussian process approach always has a lower error than stacking via a constrained weighted mean, with the error terms being equal when the contribution of the covariance is zero. That is $(\mathbb{I} - \Sigma_{2^*}\Sigma_{1^*})e_{gp}(x) \leq e_{cwm}(x)\ \forall x$ ■

We note the main purpose of this proof is to bring to light intuitions about why stacking using Gaussian processes can improve predictive performance. We highlight that the proof is based on the assertion that the data generating process is contained within the hypothesis set of functions we are using i.e, $y = f(x)$. This restrictive condition is required and generality beyond this condition will lead to "no free lunches" [16]. Our main goal in this proof is to firstly restate the results of [14] - that improved generalisation accuracy can result from an ensemble of variable and accurate generalisers. Secondly we aimed to show further prediction accuracy can be achieved by modelling the residual variation left over from the ensemble.

1.4 Alternative stacking designs

We have presented the rational for stacking and introduced a basic design (design 1 in figure 1) where multiple level 0 generalisers are stacked through a single level 1 generaliser. For this design we proposed a Gaussian process or constrained weighted mean as the level 1 generaliser. In figure 1 we suggest two alternative stacking designs.

In design 2, multiple level 0 generalisers are fitted and then passed through individual level 1 generalisers before being finally combined in a level 2 generaliser. An example of this scheme would be to fit multiple level 0 generalisers using different algorithmic methods (see Appendix 1.1) and then feed each of these into a Gaussian process regression. This design allows for the Gaussian processes to learn individual covariance structures for each level 0 algorithmic method (as opposed to a joint structure as in design 1). These level 1 Gaussian processes can then be combined through a constrained weighted mean level 2 generaliser.

In design 3, a single level 0 generaliser is used and fed into multiple level 1 generalisers before being combined via a level 2 generaliser. An example of this scheme would be to fit a single level 0 method, such as a linear mean or random forest, and then feed this single generaliser

into multiple level 1 Gaussian processes. These multiple Gaussian processes can learn different aspects of the covariance structure such as long range, short range or seasonal interactions. These level 1 generalisers can then be combined, as in design 2, through a constrained weighted mean level 2 generaliser.

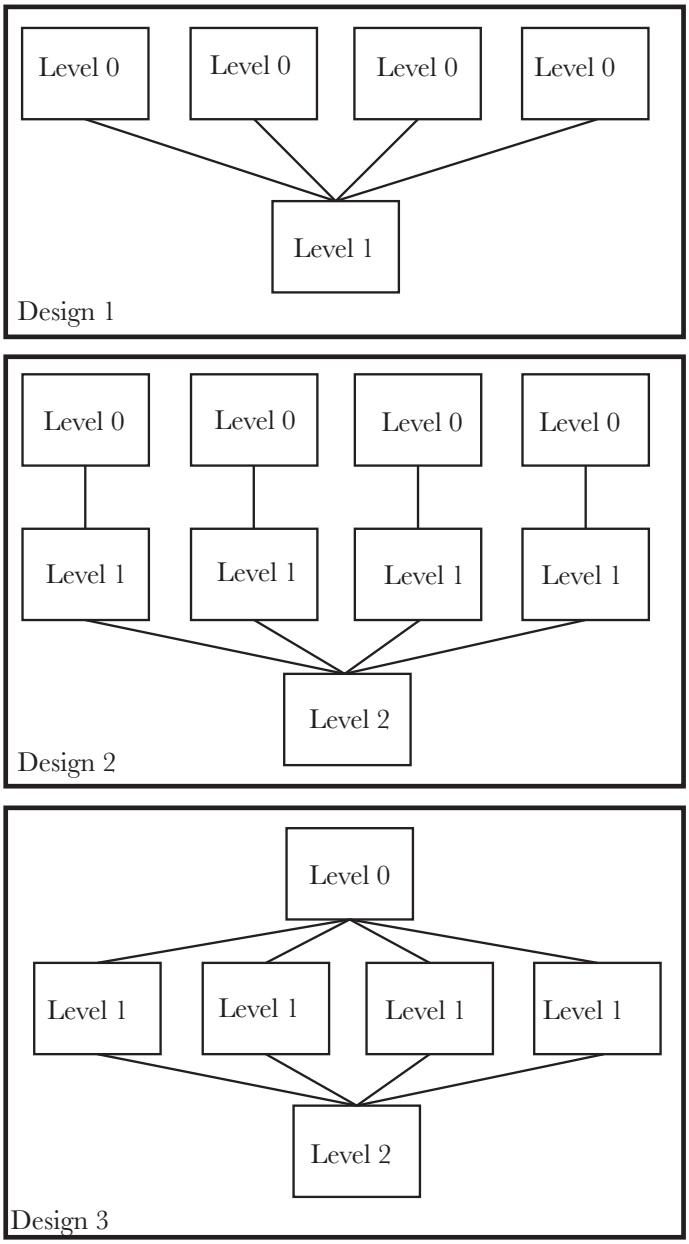

Figure 1. Three suggested stacking designs. In this paper we have exclusively used design 1, where multiple level 0 generalisers are combined through a level 1 generaliser. An alternative (design 2) is to feed each level 0 generaliser into a unique level 1 generaliser and then combine them together through a level 2 generaliser. Another alternative (design 3) is to have a single level 0 generaliser feeding into multiple level 1 generalisers that are then combined through a level 2 generaliser

References

1. L Breiman. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

2. Trevor Hastie, Robert Tibshirani, and J H Friedman. *The elements of statistical learning*. Springer, 2009.

3. J H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

4. J H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

5. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

6. Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 4 2005.

7. T. J. Hastie and R. Tibshirani. *Generalized additive models*, volume 1. 1990.

8. Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, 3 1991.

9. 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.

10. H Rue, S Martino, and N Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 2009.

11. C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*, volume 14. 2006.

12. Leo Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996.

13. Mark J. van der Laan, Eric C Polley, and Alan E. Hubbard. Super Learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1):Article25, 1 2007.

14. Anders Krogh and Jesper Vedelsby. Neural Network Ensembles, Cross Validation, and Active Learning. In *Advances in Neural Information Processing Systems*, pages 231–238. MIT Press, 1995.

15. Joseph Sill, Gabor Takacs, Lester Mackey, and David Lin. Feature-Weighted Linear Stacking. 11 2009.

16. David H Wolpert and William G Macready. No Free Lunch Theorems for Optimization. *IEEE transactions on evolutionary computation*, 1(1), 1997.