Differentially Private Gaussian Processes

Michael Thomas Smith, Mauricio A. Álvarez, Max Zwiessele, Neil D. Lawrence*
Department of Computer Science, University of Sheffield
{m.t.smith,mauricio.alvarez,m.zwiessele,n.lawrence}@sheffield.ac.uk

Abstract

A major challenge for machine learning is increasing the availability of data while respecting the privacy of individuals. Here we combine the provable privacy guarantees of the Differential Privacy framework with the flexibility of Gaussian processes (GPs). We propose a method using GPs to provide Differentially Private (DP) regression. We then improve this method by crafting the DP noise covariance structure to efficiently protect the training data, while minimising the scale of the added noise. We find that, for the dataset used, this cloaking method achieves the greatest accuracy, while still providing privacy guarantees, and offers practical DP for regression over multi-dimensional inputs. Together these methods provide a starter toolkit for combining differential privacy and GPs.

1 Introduction

As machine learning algorithms are applied to an increasing range of personal data types, interest is increasing in mechanisms that allow individuals to retain their privacy while the wider population can benefit from inferences drawn through assimilation of data. Simple 'anonymisation' through removing names and addresses has been found to be insufficient[Sweeney, 1997, Ganta et al., 2008]. However, randomisation-based privacy methods (such as differential privacy, DP) provide provable protection against such attacks. A DP algorithm[Dwork et al., 2006, Dwork and Roth, 2014] allows privacy preserving queries to be performed by adding noise to the result, to mask the influence of individual data. This perturbation can be added at any of three stages in the learning process [Berlioz et al., 2015]; to the (i) training data, prior to its use in the algorithm, (ii) to components of the calculation (such as to the gradients or objective) or (iii) to the results of the algorithm. Considerable research has been undertaken looking at the second of these options, in particular fitting parameters using an objective function which has been perturbed to render it differentially private [e.g. Chaudhuri et al., 2011, Zhang et al., 2012] with respect to the training data, or more recently, Song et al. [2013] described how one might perform stochastic gradient descent with DP updates. Some attention has also been paid to non-parametric models, such as histograms [Wasserman and Zhou, 2010] and other density estimators, such as the method described in Hall et al. [2013] which performs kernel density estimation (note that there are also parametric DP density estimators, such as Wu et al. [2016] who use Gaussian mixtures to model density). For regression; besides using a perturbed objective function, one can also perform parametric DP regression by using the subsample-and-aggregate framework, as used in Dwork and Lei [2009, section 7], effectively protecting the parametric results of the regression (option iii).

In this paper we also focus on (iii) (making the result private), and investigate corrupting a Gaussian Process's (GP's) fit to the data. Importantly this paper addresses the problem of making the *outputs* of GP regression private, not its inputs. Specifically we use the results in Hall et al. [2013] to find bounds on the scale of perturbations of the mean function which allows appropriate Gaussian DP noise to be added (Section 2). The added DP noise for this initial method is too large for many problems. To ameliorate this we consider the situation in which we know *a priori* the locations of the test points, and thus can reason about the specific correlation structure in the predictions for given perturbations in the training outputs (Section 3). Assuming the Gaussian mechanism is used to provide the DP noise, we are able to find the optimal noise covariance to protect training outputs.

*Work completed while at the University of Sheffield.

Finally we compare this strategy for inducing privacy with a DP query using the Laplace mechanism on the bin means[Dwork and Roth, 2014, section 3.4], and show that it provides greater accuracy for a given privacy guarantee, for the citibike dataset.

Combining the ubiquity of GP regression with the rigorous privacy guarantees offered by DP allows us to build a toolkit to apply DP to a wide array of problems, amenable to GP regression.

2 Applying Differential Privacy to a Gaussian Process

The challenge is as follows; we have a dataset in which some variables (the inputs, $\boldsymbol{X}$) are public, for example the latitude and longitude of all homes in a country. We also have a variable we want to keep secret ($\boldsymbol{y}$, e.g. house price). We want to allow people to make a prediction about this variable at a location, while still ensuring that the dataset's secret variables remain private. In this section we fit a standard GP model to a dataset and calculate the bound on the scale of the perturbation we need to add to the posterior mean to provide a DP guarantee on the training outputs.

Hall et al. [2013] extend DP to functions. Consider a function, f, that we want to release predictions from (with privacy guarantees on these outputs). If the family of functions from which this function is sampled lies in a reproducing kernel Hilbert space (RKHS) then one can consider the function as a point in the RKHS. We also consider another function, f', that's been generated using identical data except for the perturbation of one row. The distance, $||f - f'||$, between these points is bounded by the sensitivity, Δ. The norm is defined to be $||g|| = \sqrt{\langle g, g \rangle_H}$. Specifically the sensitivity is written $\Delta \geq \sup_{D \sim D'} ||f_D - f_{D'}||_H$. Hall et al. [2013] showed that one can ensure that a version of f, function $\widetilde{f}$ is (ε, δ)-DP by adding a scaled sample from the Gaussian process G (which uses the same kernel as f). We scale the sample by $\frac{\Delta \mathrm{c}(\delta)}{\varepsilon,}$ where $\mathrm{c}(\delta) \geq \sqrt{2 \log \frac{1.25}{\delta}}$. Relating this to the definition of DP, finding Δ allows us to know how much the function f can change between neighbouring databases. We then choose the scale of the noise added by the randomised algorithm, M, to mask these changes. In the GP case we have some data (at inputs X and outputs $\boldsymbol{y}$). We assume for this paper that the *inputs* are non-private (e.g. people's ages), while the outputs are private (e.g. number of medications).

The mean function of a GP posterior lies in an RKHS. We need to add the correctly scaled sample to ensure its DP release. It will become clear that the covariance function does *not* need perturbation as it does not contain direct reference to the output values. We first need to know the possible values that the (training data's) output $\boldsymbol{y}$ can take.

Using the notation of Williams and Rasmussen [2006], the predictive distribution from a GP at a test point $\boldsymbol{x}_*$ has mean, $\bar{f}_* = \boldsymbol{k}_*^\top \left(K' + \sigma_n^2 I\right)^{-1} \boldsymbol{y}$, and covariance, $V[f_*] = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top \left(K' + \sigma_n^2 I\right)^{-1} \boldsymbol{k}_*$, where $\bar{f}_*$ is the mean of the posterior, $k(\boldsymbol{x}_*, \boldsymbol{x}_*)$ is the test point's prior variance, $\boldsymbol{k}_*$ is the covariance between the test and training points, K' is the covariance between (the latent function that describes) the training points, σ_n^2 is the variance of the iid noise added to each observation and $\boldsymbol{y}$ are the outputs observed values of the training data.

We note that (ignoring any previous parameter selection) the covariance does not depend on the training output values (in $\boldsymbol{y}$) so we only need to make the mean function private. We also assume for the following analysis that the kernel has a maximum value of one, and a minimum of zero. This restricts us to using a subset of stationary kernels and normalising the output values prior to prediction.

Concentrating then on the mean function, Williams and Rasmussen [2006] note that, using the representer theorem, we can rewrite the above expression as the weighted sum of n kernel functions, $\bar{f}(\boldsymbol{x}_*) = \sum_{i=1}^n \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_*)$, where $\boldsymbol{\alpha} = \left(K' + \sigma_n^2 I\right)^{-1} \boldsymbol{y}$. For simplicity in the following we replace $K' + \sigma_n^2 I$ with K. We are interested in finding,

$$||f_D(\boldsymbol{x_*}) - f_{D'}(\boldsymbol{x_*})||_H^2 = \Big\langle f_D(\boldsymbol{x_*}) - f_{D'}(\boldsymbol{x_*}), f_D(\boldsymbol{x_*}) - f_{D'}(\boldsymbol{x_*}) \Big\rangle. \tag{1}$$

In Hall et al. [2013, section 4.1], the vector $\boldsymbol{x}$ is identical to $\boldsymbol{x}'$ with the exception of the last element n. In our case the inputs are identical (we are not trying to protect this part of the data). Instead it is to the values of y (and hence $\boldsymbol{\alpha}$) that we need to offer privacy.

$$f_D(\boldsymbol{x_*}) - f_{D'}(\boldsymbol{x_*}) = \sum_{i=1}^n \alpha_i k(\boldsymbol{x_*}, \boldsymbol{x}_i) - \sum_{i=1}^n \alpha'_i k(\boldsymbol{x_*}, \boldsymbol{x}_i) = \sum_{i=1}^n k(\boldsymbol{x_*}, \boldsymbol{x}_i) \left(\alpha_i - \alpha'_i\right) \tag{2}$$

In the kernel density estimation example, in Hall et al. [2013], all but the last term in the two summations cancel as the α terms were absent. In our case however they remain and, generally, $\alpha_i \neq \alpha'_i$. We therefore need to provide a bound on the difference between the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha'}$. To reiterate, $\boldsymbol{\alpha} = K^{-1}\boldsymbol{y}$. So the difference between the two vectors is, $\boldsymbol{\alpha} - \boldsymbol{\alpha'} = K^{-1}\left(\boldsymbol{y} - \boldsymbol{y'}\right)$

As K doesn't contain private information itself (it is dependent purely on the input and the features of the kernel) we can find a value for the bound using a specific K. See the supplementary material for a weaker, general, upper bound, for when the precise value of K is unknown.

The largest change we protect against is a perturbation in *one* entry of $\boldsymbol{y}$. Therefore we assume that all the values of $\boldsymbol{y}$ and $\boldsymbol{y'}$ are equal except for the last element which differs by at most d. Equation 2 is effectively adding up the values of the last column in K^{-1}, each scaled by the kernel's value at that point, $k(\boldsymbol{x_*}, \boldsymbol{x}_i)$ and by the difference between $\boldsymbol{y}$ and $\boldsymbol{y'}$. If we assume that the kernel values are bound between -1 and 1 (not unreasonable, as many stationary kernels have this property, if we normalise our data), and also note that the *use of the last column is arbitrary*, we can bound equation 2 by the maximum possible sum of any column's absolute values in K^{-1} (i.e. the infinity norm), times d; i.e. $||K^{-1}||_\infty \times d$.

We finally introduce our additional restriction that the value of the kernel is non-negative (an optional additional bound to further reduce the noise scale, which is true for many kernels, such as the Exponentiated Quadratic (EQ) kernel). The bound is then calculated by finding the infinity norm for the following two matrices, and taking the larger. In one K^{-1} is modified so that all negative-values are ignored, in the other all values are initially inverted, before the negative-values are discarded. We shall call this bound, $\mathrm{b}(K^{-1})$. Note that the two options described are necessary to allow us to account for the uncertainty in the sign of $\boldsymbol{y} - \boldsymbol{y'}$, whose magnitude is bound by d but in an unknown direction. Returning to the calculation of the sensitivity, we can expand equation 2 substituted into 1:

$$||f_D(\boldsymbol{x}_*) - f_{D'}(\boldsymbol{x}_*)||^2 = \left\langle \sum_{i=1}^{n} (\alpha_i - \alpha'_i) k(\boldsymbol{x}_*, \boldsymbol{x}_i), \sum_{i=1}^{n} (\alpha_i - \alpha'_i) k(\boldsymbol{x}_*, \boldsymbol{x}_i) \right\rangle \tag{3}$$

To reiterate, we use our constraint that the chosen kernel has a range of 0 to 1, so the summations above will have a magnitude bounded by $d \times \mathrm{b}(K^{-1})$. This means that an upper bound on the sensitivity is,

$$||f_D(\boldsymbol{x}_*) - f'_D(\boldsymbol{x}_*)||^2 \leq d^2 \; \mathrm{b}(K^{-1})^2. \tag{4}$$

!Kung San women example

We use, as a simple demonstration, the heights and ages of 287 women from a census of the !Kung [Howell, N., 1967]. We are interested in protecting the privacy of their heights, but we are willing to release their ages. We have set the lengthscale *a priori*, to 25 years as from our prior experience of human development this is the timescale over which gradients vary[1]. We can find the empirical value of our sensitivity bound on the inverse covariance matrix, $\mathrm{b}(K^{-1})$ and the value of $c(\delta)$. Substituting in our given values we find that we should scale our GP samples by 28.53. Figure 1A illustrates that, even with large ε the DP noise overwhelms the function we want to estimate. It is worth noting before moving onto the next section that, if the sensitivity of the training data had been smaller (for example count or histogram data, with $\Delta = 1$) then this method could produce usable predictions at reasonable ε. The key advantage of this method over the following is that one need not specify the test input locations in advance. We find however we are able to considerably reduce the scale of the DP noise by insisting that we are given the test input points *a priori.*

3 The Cloaking Method

Both of the methods examined so far are limited to low-sensitivity datasets (such as histogram/count data). We now introduce an alternative we refer to as *cloaking*, that allows a considerable reduction in the DP noise added but at the cost of needing to know the test point inputs *a priori.* We approach this new method by first reasoning about the direction (across test points) noise is added by the

[1] Hyperparameters are all set *a priori*, but appear precise as there is some normalisation which has taken place. Kernel variance $\sigma^2 = 7.72^2\text{cm}^2$, Gaussian noise $\sigma_n^2 = 14^2$ cm^2, DP: $\delta = 0.01$, $\varepsilon = 50.0$, $\Delta = 100$cm (inforced by rectifying all values to lie 50cm of the mean).

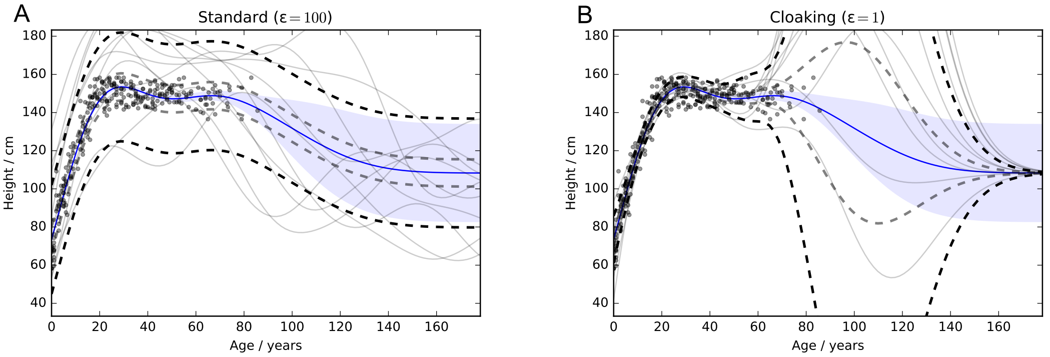

Figure 1: Heights and ages of female !Kung San. Figure A, standard GP method. Figure B, cloaking method. Solid blue lines, posterior means of the GPs; grey lines, DP samples; Black and grey dashed lines, SE and $\frac{1}{4}SE$ confidence intervals for DP noise respectively; blue area, GP posterior variance (excluding noise). $\delta = 0.01$, $\Delta = 100$cm.

earlier (Section 2) method, and comparing its effect to the effect of modifying a training point. The sensitivity in the earlier methods needed to be quite high because the noise added (sampled from the *prior* covariance) is *not necessarily in the direction a perturbation in a training output would cause*.

Consider the simple case of two training and two test points, illustrated in figure 2. Subfigures B and D illustrate (with an ellipse) the shape of the noise added to the predictions *if we sample from the prior* (as in section 2) and the changes caused by the perturbation of the training data. The figure demonstrates that the prior doesn't provide the most efficient source of noise. In particular the large amount of correlated noise that is added in this example is not necessary. Perturbations in individual training points cannot cause such correlated noise in the test outputs. To summarise; there is no perturbation in a single training point's output which could cause the predictions to move in the direction of the prior's covariance.

Differential Privacy for vectors of GP predictions

From Hall et al. [2013, proposition 3]: given a covariance matrix M and result vectors describing neighbouring databases $\mathbf{y}_*$ and $\mathbf{y}'_*$, we define the bound,

$$\sup_{D \sim D'} ||M^{-1/2}(\mathbf{y}_* - \mathbf{y}'_*)||_2 \leq \Delta \tag{5}$$

Δ is a bound on the scale of the output change, in term of its Mahalanobis distance with respect to the added noise covariance. The algorithm outputs $\tilde{\mathbf{y}}_* = \mathbf{y}_* + \frac{\mathrm{c}(\delta)\Delta}{\varepsilon} Z$ where $Z \sim \mathcal{N}_d(0, M)$. (ε, δ)-DP is achieved if $\mathrm{c}(\delta) \geq \sqrt{2 \log \frac{2}{\delta}}$. We want M to have the greatest covariance in those directions most affected by changes in training points. We are able to compute K, the covariance between all training points (incorporating sample variance) and K_{*f} the covariance between training and test points. Given the training outputs $\mathbf{y}$, we can find the predictions for all test points simultaneously, $\mathbf{y}_* = K_{*f} K^{-1} \mathbf{y}$. The cloaking matrix $C = K_{*f} K^{-1}$ describes how the test points change wrt changes in training data. We use it to write the perturbed test values as $\mathbf{y}'_* = \mathbf{y}_* + C\left(\mathbf{y}' - \mathbf{y}\right)$. We assume one training item i has been perturbed, by at most d; $y'_i = y_i + d$. As y_i is the only training output value perturbed, we can see that the change in the predictions is dependent on only one column of C, $\mathbf{c}_i$; $\mathbf{y}'_* - \mathbf{y}_* = d\mathbf{c}_i$ This can be substituted into the bound on Δ in equation 5. Rearranging the expression for the norm (and using M's symmetry);

$$||dM^{-1/2}\mathbf{c}_i||_2 = (dM^{-1/2}\mathbf{c}_i)^\top (dM^{-1/2}\mathbf{c}_i) = d\mathbf{c}_i^\top M^{-\top/2} M^{-1/2} \mathbf{c}_i d = d^2 \mathbf{c}_i^\top M^{-1} \mathbf{c}_i$$

We want to find M such that the noise sample Z is small but also that Δ is also minimised. A common way of describing that noise scale is to consider the determinant (generalised variance), the square root of the determinant (proportional to the volume of a confidence interval), or the log of the determinant (proportional to the differential entropy of a normal distribution (plus a constant) $\frac{1}{2} \ln\left((2\pi e)^k \cdot |\mathbf{\Sigma}|\right)$. We use the latter but they will all have similar results. We show in the supplementary material that the optimal $M = \sum_i \lambda_i \mathbf{c}_i \mathbf{c}_i^\top$ with the $\boldsymbol{\lambda}$ found using gradient descent, $\frac{\partial L}{\partial \lambda_j} = \mathrm{Tr}\left(M^{-1} \mathbf{c}_j \mathbf{c}_j^\top\right) + 1$.

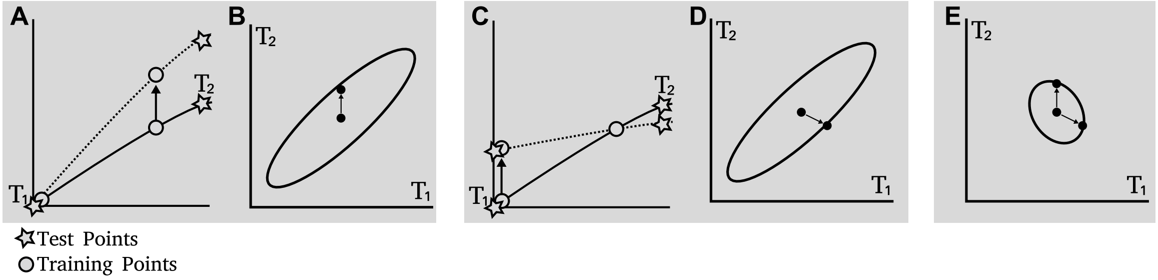

Figure 2: Figures A and C illustrate the perturbation of a pair of training points (circles) and the effect on a pair of test points T_1 and T_2 (stars). Figures B and D illustrate the associated change in the two test points, plotting them against each other (the posterior mean prediction indicated by the central dots). The ellipses represent the *shape of the DP noise added using the DP method devised in section 2*. One can see that in figure B, T_1 remains largely unaffected by the perturbation in the training point, while T_2 experiences a large change. In figure D perturbation in the other training point causes the posterior mean to move rightwards and downwards. Importantly, the covariance of the DP noise added by previous methods is mostly not in either of these directions, but instead adds correlated noise. This additional, correlated noise, isn't useful in protecting the privacy of the training data, as no perturbation in a single training data's output could cause the test points to move in a correlated way. The cloaking method instead adds noise illustrated in figure E, as small as possible while still masking the possible perturbations.

We return to the example of the !Kung San women data to demonstrate the improvement in privacy efficiency. Figure 1B illustrates the results for a reasonable value of $\varepsilon = 1$[2]. The input domain is deliberately extended to demonstrate some features of the method. First, where the data is most concentrated the scale of the added noise is small. This is quite intuitive as the effect one training point has there on the posterior prediction will be overwhelmed by its neighbours. A second and potentially less intuitive finding is that the DP noise is greatest at about 110 years, well away from the nearest data point. This, we believe, is because of the fulcrum and lever interaction between the outliers and the data's concentration. The figure 2A partly demonstrates this, with the test point T_2 being changed slightly more than the perturbation in the training point. The third observation is that the added DP noise eventually approaches zero away from the training data; the posterior mean will equal the prior mean regardless of the training data's outputs. The RMSE without DP was 6.8cm and with $(1, 0.01)$-DP, 12.2cm, suggesting that this provides quite large, but practical levels of DP perturbation. Note that the 200 test point inputs that make up the graph's predictions are exactly those specified as part of the cloaking algorithm (X_*); a large number of nearby test points doesn't degrade the quality of the predictions as they are all very highly correlated. The noise that is already added to ensure a test point does not breach DP, is almost exactly the noise that is needed by its close neighbours.

As a further example we consider a spatial dataset of 4766 property sales since 2013 from the Land Registry [2017] centred on London[3]. Figure 3 illustrates how the DP noise scale varies over the city. Note how areas of high training data concentration the DP noise added was small, while in sparse areas the noise added was high. Also areas towards the corners of the map where there are no data, return to the GP prior's mean, and have little DP noise added.

Hyperparameter optimisation

So far in this paper we have selected the values of the kernel hyperparameters *a priori*. To demonstrate a DP method for their selection we evaluate the SSE of k-fold cross-validation iterations for each hyperparameter configuration and select the optimum using the exponential mechanism. Details of the method (including bounds on the sensitivity) can be found in the supplementary material.

[2] EQ kernel, $\delta = 0.01$, lengthscale 25 years, gaussian noise 14cm

[3] The data was thresholded to lie between £100k and £500k (so the sensitivity could be bounded). $\varepsilon = 1$ and $\delta = 0.01$, lengthscale 15km, Gaussian variance £^{2}400k^2.

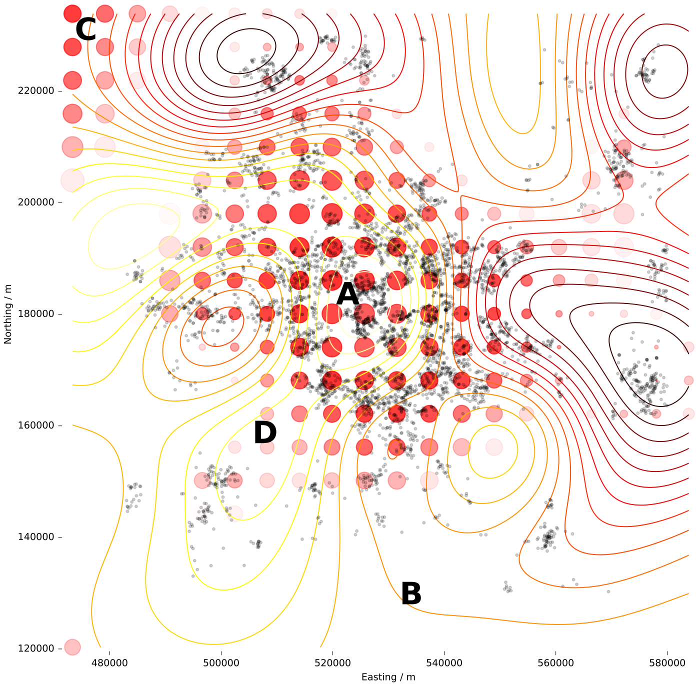

Figure 3: 4766 property prices in a $10,000\text{km}^2$ square around London (2013-16). Dots, property locations; circles, DP price predictions. Predicted price indicated by circle area. The scale of the DP noise indicated by transparency (Opaque: no DP noise. Transparent: DP noise std is at least 40% of Δ). Non-DP predictions indicated by contours from £215k to £437k. $\varepsilon = 1$ and $\delta = 0.01$. Areas (A) with high concentrations of training data provide predictions with little DP noise, areas with few data have much more DP noise (B). Areas far from any data return to the GP's prior mean, and have little DP noise added (C). Interesting 'bridging' effects between data concentrations causes the DP noise to remain low as the posterior is 'supported' at both sides of areas with low density (e.g. D).

4 Results

We finally compare cloaking with simple binning (with the bin means made DP using the Laplace Mechanism[4]). We also fitted a GP to this binned data, using an 'integral' kernel. Briefly, this kernel models a latent function which is then integrated across (between bin boundaries) to provide predictions of the model's observations. We found this often does better than simple binning when the data is noisy. We did not include the standard GP method (from section 2) as we found it not competitive.

For the !Kung dataset, we applied the hyperparameter selection technique to the cloaking mechanism's outputs and compared it to the results of binning. We found that hyperparameter selection, for one or two parameters, caused a reduction in RMSE accuracy that was noticeable but not impractical. Specifically we used the exponential mechanism with the negative-SSE of 100-fold monte-carlo cross-validation runs to select hyperparameter combinations (testing lengthscales of between 3 and 81 years, and Gaussian noise variance of between 1.1^2 cm^2 and 12.7^2 cm^2), which we tested against a validation set to provide the expected RMSE. We fixed the DP ε to 1 for both the exponential mechanism and the cloaking and binning stages. The simple binning method (depending on bin size) had a RMSE of 23.4-50.7cm, the integral method improved this to 14.2-20.8cm. With no DP on parameter selection the cloaking method's RMSE would be 14.3cm (comparable to the integral kernel's best bin size result). If we however select its hyperparameters using the exponential mechanism, the RMSE worsens to 17.4cm. Thus there is a small, but managable cost to the selection.

A final interesting result is in the effect of the level of privacy in the regression stage on the selection of the lengthscale. This is demonstrated in the distribution of probabilities over the lengthscales when we adjust ε. Figure 4 demonstrates this effect. Each column is for a different level of privacy (from none to high) and each tile shows the probability of selecting that lengthscale. For low privacy, short lengthscales are acceptable, but as the privacy increases, averaging over more data allows us to give more accurate answers.

[4] We used Laplace noise as it causes a lower RMSE than the Gaussian noise alternative (with $\delta = 0.01$).

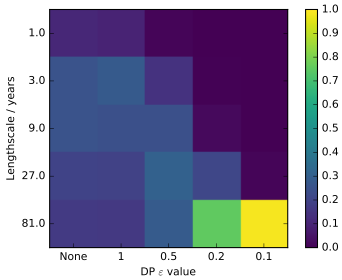

Figure 4: Effect of varying the differential privacy parameter, ε, on the likelihood of selecting each lengthscale. Colour indicates probability of parameter selection. With low privacy, a short lengthscale is appropriate which allows the GP to describe details in the data. With high privacy, a longer lengthscale is required, which will average over large numbers of individual data points.

Using data from the New York City bike sharing scheme, Citibike [2013][5] we predict journey time, give the latitude and longitude of the start and finish stations. The 4d EQ kernel had lengthscales of between 0.02 and 0.781 degrees (latitude or longitude, equivalent to roughly 1.8km to 75km. $\sigma = 1581^2\text{s}^2$, $l = 0.05°$, $\sigma_n = 1605^2\ \text{s}^2$. Durations above 2000s were thresholded.

We tested values of ε of between 0.2 and 1.0 (with δ fixed at 0.01) and with DP disabled. Monte Carlo cross-validation was used to make predictions using the DP framework (4900 training, 100 test journeys). For comparison we binned the training data into between 81 and 10,000 bins, then computed DP means for these bins. These DP values were used as predictions for the test points that fell within that bin (those bins without training data were set to the population mean). Table 1 summarises the experimental results. The new cloaking function achieves the lowest RMSE, unless both ε and the lengthscales are small. With no DP the short-lengthscale cloaking method provides the most accurate predictions, as this isn't affected by the binning and is capable of describing the most detail in the spatial structure of the dataset. The simple binning becomes less accurate with more bins, probably due to low occupancy rate (in a random training example, with 10,000 bins, only 11% were occupied, and of those 40% only had one point) and a form of overfitting. As $(1, 0.01)$-DP noise is added the simple-binning degrades quickly due to high DP noise added to low-occupancy bins. Xu et al. [2013] also describes a similar phenomenon. Predictions using the GP with an integral kernel fitted to these DP bin counts appears to provide some robustness to the addition of DP noise. As ε is further reduced, the cloaking method does better at longer lengthscales which allow more averaging over the training data. Simple binning becomes increasingly compromised by the DP noise.

5 Discussion

The cloaking method performs well, providing reasonably constrained levels of DP noise for realistic levels of privacy and provides intuitive features such as less DP-noise in those areas with the greatest concentration of training data. The earlier method, described in section 2, required much more DP perturbation, but didn't require that test point locations be known *a priori.* For the cloaking method the lengthscale provides a powerful way of trading off precision in modelling spatial structure with the costs of increasing DP-noise. We could exploit this effect by using non-stationary lengthscales [e.g. Snoek et al., 2014, Herlands et al., 2015], incorporating fine lengthscales where data is common and expansive scales where data is rarified. This could lead to DP noise which remains largely constant across the feature space. Another interesting effect of the cloaking mechanism is that the precision one gains about one location depends on where other test points are. So if you are particularly interested in predictions in one part of the input domain, one should avoid including other test points at uninteresting locations. To further reduce the DP noise, we could manipulate the sample noise for individual output values. For the initial method, by adjusting the sample noise for individual elements we can control the infinity-norm. For the cloaking method, outlying training

[5] 163,000 subscribers, 600 stations located in a box bounded between latitudes 40.6794 and 40.7872, and longitudes -74.0171 and -73.9299. Unlike the house-price data we kept the locations in these global coordinates. Each fold of the Monte Carlo cross validation sampled 5000 rows from the 1,460,317 journeys in June 2016.

	lengthscale or bins	No DP	$\varepsilon = 1$	$\varepsilon = 0.5$	$\varepsilon = 0.2$
cloaking	0.781°	490 ± 14	493 ± 13	498 ± 13	525 ± 19
	0.312°	492 ± 15	497 ± 12	502 ± 17	545 ± 26
	0.125°	402 ± 7	437 ± 21	476 ± 17	758 ± 94
	0.050°	333 ± 11	434 ± 27	612 ± 78	1163 ± 147
	0.020°	314 ± 12	478 ± 22	854 ± 54	1868 ± 106
integral binning	10^4 bins	581 ± 5	586 ± 7	597 ± 12	627 ± 23
	6^4 bins	641 ± 6	640 ± 9	658 ± 17	736 ± 41
	3^4 bins	643 ± 6	649 ± 13	677 ± 22	770 ± 51
simple binning	10^4 bins	596 ± 12	1064 ± 69	1927 ± 191	4402 ± 434
	6^4 bins	587 ± 11	768 ± 58	1202 ± 206	2373 ± 358
	3^4 bins	550 ± 12	575 ± 24	629 ± 58	809 ± 110

Table 1: RMSE (in seconds, averaged over 30-fold X-validation, ± 95% CIs) for DP predictions of citibike journey durations. Five lengthscales (in degrees latitude/longitude) and three bin resolutions were tested for the cloaking and binning experiments respectively. The cloaking method, with the right lengthscale, makes more accurate predictions than either of the binning methods. As we increase ε, cloaking needs longer lengthscales to remain competitive. These longer lengthscales effectively allow the predictions to 'average' over more training data.

points, around the 'edge' of a cluster, could have their sample noise increased. One important issue is how to make DP GP predictions if we want to protect the values of the training *inputs*. This could be approached by considering a bound on the inverse-covariance function, a suggestion is provided in the supplementary material.

Future work is also needed to address how one optimises the hyperparameters of these models in a DP way. The method described in section 3 works but is far from optimal. It may be possible to extend methods that use DP Bayesian optimisation to estimate values for hyperparameters [Kusner et al., 2015], or approximate the likelihood function to work with the method described in Han et al. [2014]. It would also be interesting to investigate how to modify the objective function to incorporate the cost of DP noise.

The actual method for releasing the corrupted mean function for the output-noise methods has not been discussed. Options include releasing a set of samples from the mean and covariance functions (a necessary step for the cloaking method, as the test-points need specifying in advance), or providing a server which responds with precise predictions given arbitrary input queries, sampling from the same Gaussian. The examples given here all use the EQ kernel but the cloaking method works with arbitrarily complex kernel structures and it has no requirement that the covariance function be stationary. GP Classification is also an obvious next step, using a logistic link function, or similar.

Finally, in unpublished work, we have found evidence that the use of inducing inputs can significantly reduce the sensitivity, and thus DP noise required, for the initial DP GP model. Future work should be undertaken to investigate the potential for further reducing the DP noise through the use of inducing inputs.

We have presented novel methods for combining DP and GPs. GPs are a highly flexible approach for a range of challenges in machine learning. In the longer term we believe a comprehensive set of methodologies could be developed to enhance their applicability in privacy preserving learning. We have applied DP for functions to GPs and given a set of known test points we were able to massively reduce the scale of perturbation for these points by considering the structure of the perturbation sensitivity across these points. In particular we found that the cloaking method performed considerably more accurately than the binning alternatives.

References

Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 107–114. ACM, 2015.

Kamalika Chaudhuri and Staal A Vinterbo. A stability-based validation procedure for differentially private machine learning. In *Advances in Neural Information Processing Systems*, pages 2652–2660, 2013.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Citibike. Citibike system data. `https://www.citibikenyc.com/system-data`, 2013.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380. ACM, 2009.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273. ACM, 2008.

Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *The Journal of Machine Learning Research*, 14(1):703–727, 2013.

Shuo Han, Ufuk Topcu, and George J Pappas. Differentially private convex optimization with piecewise affine objectives. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 2160–2166. IEEE, 2014.

William Herlands, Andrew Wilson, Hannes Nickisch, Seth Flaxman, Daniel Neill, Wilbert Van Panhuis, and Eric Xing. Scalable gaussian processes for characterizing multidimensional change surfaces. *arXiv preprint arXiv:1511.04408*, 2015.

Howell, N. Data from a partial census of the !kung san, dobe. 1967-1969. `https://public.tableau.com/profile/john.marriott#!/vizhome/kung-san/Attributes`, 1967.

Matt J Kusner, Jacob R Gardner, Roman Garnett, and Kilian Q Weinberger. Differentially private bayesian optimization. *arXiv preprint arXiv:1501.04080*, 2015.

Land Registry. Hm land registry price paid data. `https://data.gov.uk/dataset/land-registry-monthly-price-paid-data`, 2017.

Jasper Snoek, Kevin Swersky, Richard S Zemel, and Ryan P Adams. Input warping for bayesian optimization of non-stationary functions. In *ICML*, pages 1674–1682, 2014.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 245–248. IEEE, 2013.

Latanya Sweeney. Weaving technology and policy together to maintain confidentiality. *The Journal of Law, Medicine & Ethics*, 25(2-3):98–110, 1997.

James M Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra and its Applications*, 11(1):3–5, 1975.

Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

Yuncheng Wu, Yao Wu, Hui Peng, Juru Zeng, Hong Chen, and Cuiping Li. Differentially private density estimation via gaussian mixtures model. In *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*, pages 1–6. IEEE, 2016.

Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB Journal*, 22(6):797–822, 2013.

Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

Supplementary Material

Deriving and optimising the cloaking variables

Note that when solving for M, we put an arbitrary (positive) bound on Δ of 1, as any scaling of Δ caused by manipulating M, will scale the determinant of M by the inverse amount[6].

We will express our problem as trying to *maximise* the entropy of a k-dimensional Gaussian distribution with covariance $P = M^{-1}$;

$$\text{Maximise } \ln(|P|), \text{ subject to } n \text{ constraints, } 0 \leq \mathbf{c}_i^\top P \mathbf{c}_i \leq 1.$$

Considering just the upper bounds and expressing this as a constraint satisfaction problem using Lagrange multipliers we have $L = \ln|P| + \sum_i \lambda_i (1 - \mathbf{c}_i^\top P \mathbf{c}_i)$. Differentiating (and setting to zero), $\frac{\partial L}{\partial P} = P^{-1} - \sum_i \lambda_i \mathbf{c}_i \mathbf{c}_i^\top = 0$. We also have the slackness conditions (for all i), $\lambda_i(\mathbf{c}_i^\top P \mathbf{c}_i - 1) = 0$ and we also require that $\lambda_i \geq 0$. Rearranging the derivative, we have, $P^{-1} = \sum_i \lambda_i \mathbf{c}_i \mathbf{c}_i^\top$. Note that as $\lambda_i \geq 0$, P^{-1} is positive semi-definite (psd)[7], thus the initial lower bound (that $\mathbf{c}_i^\top P \mathbf{c}_i \geq 0$) is met. The upper bound (that $\mathbf{c}_i^\top P \mathbf{c}_i \leq 1$) is achieved if the λ_i are correctly chosen, such that the Lagrange and slackness conditions are met.

We now must maximise the expression for L wrt λ_j. To assist with this we rewrite our expression for $P^{-1} = \sum_i \lambda_i \mathbf{c}_i \mathbf{c}_i^\top = C \Lambda C^\top$, where $C = [\boldsymbol{c}_1, \boldsymbol{c}_2 ... \boldsymbol{c}_n] = K_{*f} K^{-1}$ and Λ is a diagonal matrix of the values of λ_i. The summation in the expression for L, $\sum_i \lambda_i (1 - \mathbf{c}_i^\top P \mathbf{c}_i)$ can be written as $\text{Tr}\left(\Lambda + C^\top P C \Lambda\right)$. Substituting in our definition of P, we can write the summation as: $\text{Tr}\left(\Lambda + C^\top (C \Lambda C^\top)^{-1} C \Lambda\right) = \text{Tr}\left(\Lambda + C^\top C^{-\top} \Lambda^{-1} C^{-1} C \Lambda\right)$. Assuming C is invertible, the summation becomes $\text{Tr}\left(\Lambda + \Lambda^{-1} \Lambda\right)$. Differentiating this wrt λ_j equals one. We can use this result to find the gradient of L wrt λ_j:

$$\begin{aligned}
\frac{\partial L}{\partial \lambda_j} &= \frac{\partial L}{\partial |P|} \times \frac{\partial |P|}{\partial \lambda_j} + \frac{\partial}{\partial \lambda_j} \sum_i \lambda_i (1 - \mathbf{c}_i^\top P \mathbf{c}_i) \\
&= |P|^{-1} \times |P| \,\text{Tr}\left(P^{-1} P^2 \boldsymbol{c}_j \boldsymbol{c}_j^\top\right) + 1 \\
&= \text{Tr}\left(P \boldsymbol{c}_j \boldsymbol{c}_j^\top\right) + 1
\end{aligned} \tag{6}$$

This can be solved using a gradient descent method, to give us those values of λ_i which minimise $\log|M|$ while ensuring $\Delta \leq 1$.

Citibike duration distribution

Figure 5 illustrates the distribution of journey times and the effect of thresholding.

Algorithms

Algorithm 1 describes the Cloaking method.

Hyperparamter selection

So far in this paper we have selected the values of the kernel hyperparameters *a priori*. Normally one may maximise the marginal likelihood to select the values or potentially integrates over the hyperparameters. In differential privacy we must take care when using private data to make these choices. Previous work exists to perform this selection, for example Kusner et al. [2015] describes a method for performing Differentially Private Bayesian optimisation, however their method assumes the training data isn't private. Kusner et al. [2015] do suggest that the work of Chaudhuri and Vinterbo [2013] may allow Bayesian Optimisation to work in the situation in which the training data also needs to be private.

[6] For example if we write a new Δ' with M as mM, we see that we can take out the m term from Δ''s inequality, leaving $\Delta' = m^{-1/2}\Delta$. When we scale Z, which has covariance mM by this new value of Δ' the covariance of the scaled Z equals $(\Delta')^2 mM = (m^{-1/2}\Delta)^2 mM = \Delta^2 M$, the magnitude change cancels, so any positive value of Δ (e.g. 1) will suffice for the optimisation. Also: in the following we ignore d and reintroduce it at the end of the derivation by defining $\Delta = d$ instead of it equalling 1.

[7] the summation is of a list of positive semi-definite rank-one matrices. One can see that such a sum is also positive semi-definite (to see this, consider distributing $\boldsymbol{z}^\top$ and $\boldsymbol{z}$ over the summation).

Algorithm 1 Using the cloaking algorithm

Require: $\mathcal{M}$, the GP model (the kernel, hyperparameters and training inputs and normalised[*] outputs)
Require: $X_* \in R^{P \times D}$, (the matrix of test inputs)
Require: $d > 0$, data sensitivity (maximum change possible)
Require: $\varepsilon > 0, \delta \geq 0$, the DP parameters

```
function DifferentiallyPrivateCloakingRegression($X_*$, $M$, $d$, $\varepsilon$, $\delta$)
    $C \leftarrow \mathcal{M}$.get_C($X_*$)    ▷ Compute the value of the cloaking matrix ($K_{*f}K^{-1}$)
    $\boldsymbol{\lambda} \leftarrow$ findLambdas($C$)
    $M \leftarrow$ calcM($\boldsymbol{\lambda}, C$)    ▷ Calculate the DP noise covariance matrix
    $\Delta \leftarrow$ calcDelta($\boldsymbol{\lambda}, C$)[†]
    $c \leftarrow \sqrt{2log\frac{2}{\delta}}$
    $\boldsymbol{y}_*, \sigma_*^2 \leftarrow \mathcal{M}$.get_predictions($X_*$)    ▷ Calculate non-DP predictions
    $\boldsymbol{z} \sim \mathcal{N}(0, M)$
    $\tilde{\boldsymbol{y}}_* \leftarrow \boldsymbol{y}_* + (\Delta dc/\varepsilon)\boldsymbol{z}$
    return $\tilde{\boldsymbol{y}}_*$, $\sigma_*^2$
end function

function $\mathcal{M}$.get_C($X_*$)
    From $\mathcal{M}$ compute $K_{*f}$ and $K^{-1}$    ▷ Compute covariances between training and test points
    $C \leftarrow K_{*f}K^{-1}$
    return $C$
end function

function findLambdas($C$)
    $\boldsymbol{\lambda} \leftarrow$ Uniform($0.1, 0.9$)    ▷ Initialise randomly[‡]
    $\alpha \leftarrow 0.05$    ▷ Learning rate
    do
        $\frac{dL}{d\boldsymbol{\lambda}} \leftarrow$ CalcGradient($\boldsymbol{\lambda}, C$)
        $\Delta_{\boldsymbol{\lambda}} \leftarrow -\frac{dL}{d\boldsymbol{\lambda}}\alpha$
        $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta_{\boldsymbol{\lambda}}$
    while $\Delta_{\boldsymbol{\lambda}} > 10^{-5}$
    return $\boldsymbol{\lambda}$
end function

function calcGradient($\boldsymbol{\lambda}, C$)
    M $\leftarrow$ calcM($\boldsymbol{\lambda}, C$)
    for $0 \leq j < N$ do    ▷ N, number of columns in cloaking matrix, C.
        $[\frac{dL}{d\boldsymbol{\lambda}}]_j \leftarrow -\operatorname{Tr}\left(M^+ C_{:j} C_{:j}^\top\right) + 1$
    end for
    return $\frac{dL}{d\boldsymbol{\lambda}}$
end function

function calcM($\boldsymbol{\lambda}, C$)
    $M \leftarrow \sum_i \lambda_i C_{:i} C_{:i}^\top$
    return $M$
end function

function calcDelta($\boldsymbol{\lambda}, C$)
    $M \leftarrow$ calcM($\boldsymbol{\lambda}, C$)
    $\Delta \leftarrow \mathbf{max}_j\ C_{:j}^\top M^+ C_{:j}$
    return $\Delta$
end function
```

[*] We assume the user will handle normalisation.
[†] Although we should have optimised M such that $\Delta \leq 1$, it may not have completely converged, so we compute the Δ bound for the value of M we have actually achieved.
[‡] We have found that occasionally the algorithm fails to converge. To confirm convergence we have found it useful to reinitialise and run the algorithm a few times.

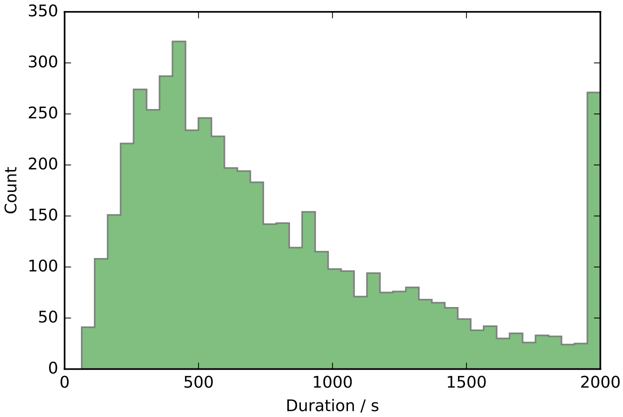

Figure 5: The duration of 5000 citibike journeys. Note the effect caused by thresholding at 2000s.

We decided instead that, due to the low-dimensionality of many hyperparameter problems, a simple grid search, combined with the exponential mechanism may allow the selection of an acceptable set of hyperparameters. For the utility function we considered using the log marginal likelihood, with additional noise in the data-fit term to capture the effect of the DP noise. However for simplicity in estimating the bound and to avoid overfitting we simply used the sum square error (SSE) over a series of K-fold cross-validation runs, which for a given fold is $\sum_{i=1}^{N} \left(y_{*i} - y_{\mathrm{t}i}\right)^2$, with predictions $\boldsymbol{y}_*$ and test values $\boldsymbol{y}_{\mathrm{t}}$.

Before proceeding we need to compute a bound on the sensitivity of the SSE. To briefly recap, the DP assumption is that one data point has been perturbed by at most d. We need to bound the effect of this perturbation on the SSE. First we realise that this data point will only be in the *test set* in one of the K folds. In the remaining folds it will be in the training data.

If the perturbed data point is in the training data ($\boldsymbol{y}$), then we can compute the sensitivity of the SSE. The perturbation this would cause to the predictions ($\boldsymbol{y}_*$) is described using standard GP regression (and the cloaking matrix). Specifically a change of d in training point j will cause a $d\boldsymbol{c}_{jk}$ change in the test point predictions, where $\boldsymbol{c}_{jk}$ is the jth column of the cloaking matrix for the kth fold.

To compute the perturbation caused by the change in the training data, we note that the SSE is effectively the square of the euclidean distance between the prediction and the test data. We are moving the prediction by $d\boldsymbol{c}_{jk}$. The largest effect that this movement of the prediction point could have on the distance between prediction and test locations is if it moves the prediction in the opposite direction to the test points. Thus it can increase (or decrease) the distance between the test and predictions by the largest length of $d\boldsymbol{c}_{jk}$ over training points. Hence for one of the folds, the largest change in the SSE is $d^2 \max_j |\boldsymbol{c}_{jk}|_2^2$.

If the perturbed data point, j, was in the test data then the SSE will change by $\left(y_{*j} + d - y_{\mathrm{t}j}\right)^2 - \left(y_{*j} - y_{\mathrm{t}j}\right)^2 = d^2 + 2d(y_{*j} - y_{\mathrm{t}j})$. The last part of the expression (the error in the prediction for point j) is unbounded. To allow us to constrain the sensitivity we enforce a completely arbitrary bound of being no larger than $\pm 4d$, thresholding the value if it exceeds this. Thus a bound on the effect of the perturbation is $d^2 + 2d \times 4d = d^2 + 8d^2 = 9d^2$.

The SSE of each fold is added together to give an overall SSE for the cross-validation exercise. We sum the $K-1$ largest sensitivities and add $9d^2$ to account for the effect of the single fold in which the perturbing datapoint, j, will be in the test set. The perturbation could have been in the test data in any of the folds. We assumed it was in the fold with the smallest training-data sensitivity to allow us to make the result a lower bound on the sensitivity of the SSE to the perturbation. If it had been in any other fold the sensitivity would have been less (as more sensitivity would be contributed by the sum over the training sensitivities). Thus the sensitivity of the SSE over the K folds is; $9d^2 + \sum_{k=1}^{K-1} d^2 \max_j |\boldsymbol{c}_{jk}|_2^2$ (where the k folds are ordered by decreasing sensitivity)

We compute the SSE and the SSE's sensitivity for each of the hyperparameter combinations we want to test. We then use the computed sensitivity bound with the exponential mechanism to select the hyperparameters. To summarise, to use the exponential mechanism one evaluates the utility

$u(x, r)$ for a given database x and for elements r, from a range. One also computes the sensitivity of this utility function by picking the highest sensitivity of any of the utilities; in this case each utility corresponds to a negative SSE, and each sensitivity corresponds to the sum described above.

$$\Delta_u \triangleq \max_{r \in R} \max_{x,y} |u(x, r) - u(y, r)|$$

(where x and y are neighbouring databases).

The exponential mechanism selects an element r with probability proportional to:

$$\exp\left(\frac{\varepsilon u(x, r)}{2\Delta_u}\right)$$

Note that for a given privacy budget, some will need to be expended on this selection problem, and the rest expended on the actual regression.

Privacy on the training inputs

To release the mean function such that the training inputs remain private, we need a general bound on the infinity norm of the covariance function, that doesn't depend explicitly on the values of X.

Varah [1975] show that if J is *strictly diagonally dominant*[8] then:

$$||J^{-1}||_\infty \leq \max_{1 \leq i \leq n} \frac{1}{\Delta_i(J)} = b(J)$$

where we've defined this bound as $b(J^{-1})$. We also define $\Delta_i(J) = |J_{ii}| - \sum_{j \neq i} |J_{ij}|$, i.e. the sum of the off diagonal elements in row i subtracted from the diagonal element.

So if K is strictly diagonally dominant (which is achieved if the inputs are sufficiently far apart, and/or if sufficient uncertainty exists on the outputs), then we have a bound on the sums of its rows. The above bound means that,

$$\sum_{i=1}^{n} \alpha_i - \alpha'_i \leq \Delta_y b(J^{-1}) \tag{7}$$

To ensure sufficient distance between inputs, we could use inducing variables, which can be arbitrarily placed, so that the above constraints on the covariance matrix are observed.

[8] A matrix, J, is strictly diagonally dominant if $\Delta_i(J) > 0$ for all $1 \leq i \leq n$.