# Accurate and Scalable Gaussian Processes for Fine-grained Air Quality Inference

**Anonymous Authors**

Anonymous affiliation
publications22@aaai.org

**Abstract**

Air pollution is a global problem and has a severe impact on human health. Fine-grained air quality (AQ) monitoring is important in mitigating air pollution. However, existing AQ station deployments are sparse. Conventional interpolation techniques fail to learn the complex AQ phenomena. Physics-based models require domain knowledge and pollution source data for AQ modeling. In this work, we propose a Gaussian processes based approach for estimating AQ. The important features of our approach are: a) a non-stationary (NS) kernel to allow input depended smoothness of fit; b) a Hamming distance-based kernel for categorical features; and c) a locally periodic kernel to capture temporal periodicity. We leverage batch-wise training to scale our approach to a large amount of data. Our approach outperforms the conventional baselines as well as a state-of-the-art neural attention-based approach.

## A Computation Time and Resources

In all our experiments, we have used a GPU system with configurations mentioned in Table 1. Time taken to run each stationary and non-stationary kernel experiment for 40 epochs without early stopping is around 30 and 50 seconds respectively. However, due to use of early stopping based on marginal log likelihood value, actual time for the experiments can vary accordingly. Maximum GPU RAM consumption was found around 4 GB and system RAM consumption was nearly 2 GB in batched setting.

| | |
|---|---|
| CPU | Intel Xeon Silver 4208 @ 2.10GHz |
| No. of CPUs | 32 |
| System RAM | 128 GB |
| GPU | NVIDIA TITAN Xp |
| No. of GPUs | 1 |
| GPU RAM | 12 GB |

Table 1: Configuration of the server on which all the experiments were executed.

## B Mathematical Notation

We define the mathematical notations used in the paper in table 2.



## C Derivation of Negative Log Marginal Likelihood

We now derive the loss function used for stationary GPs in our work, known as negative log marginal likelihood (Eq. 4 in the paper) as the following,

We assume a prior distribution $p(\mathbf{f}_n|X_n) = \mathcal{N}(\mathbf{0}, K)$ and likelihood $p(\mathbf{y}_n|\mathbf{f}_n, X_n) = \mathcal{N}(\mathbf{f}_n, \sigma^2 I_n)$. Now, marginal likelihood is given as:

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} \tag{1}$$

$$= \int \mathcal{N}(\mathbf{0}, K)\mathcal{N}(\mathbf{f}, \sigma^2 I)d\mathbf{f} \tag{2}$$

$$= \mathcal{N}(\mathbf{0}, K + \sigma^2 I) \tag{3}$$

Transition from Eq. 2 to Eq. 3 is as per Eq. 2.115 in PRML (Bishop 2006). Now, taking $K_y = K + \sigma^2 I_n$:

$$p(\mathbf{y}|X) = \frac{1}{\sqrt{(2\pi)^n |K_y|}} \exp\left(-\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y}\right) \tag{4}$$

$$-\log p(\mathbf{y}|X) = \frac{1}{2}\left(\mathbf{y}^T K_y^{-1}\mathbf{y} + \log|K_y| + n\log(2\pi)\right) \tag{5}$$

Eq. 5 is the closed form of negative log marginal likelihood.

## D Derivation of Objective Function for Non-stationary GP

We can minimize negative log marginal likelihood which has the following form:

$$p(\mathbf{y}|X) = \int p\left(\mathbf{y}|X, \boldsymbol{\ell}\right) \cdot p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}_m, \overline{X}_m\right) d\boldsymbol{\ell} \tag{6}$$

However, Eq. 6 is intractable (closed form does not exist), thus, we can maximize $p(\boldsymbol{\ell}|\mathbf{y}, X)$ leveraging the Bayes rule.

$$p(\boldsymbol{\ell}|\mathbf{y}, X) = \frac{p\left(\mathbf{y}|X, \exp(\boldsymbol{\ell})\right) p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right)}{p(\mathbf{y}|X)} \tag{7}$$

$$p(\boldsymbol{\ell}|\mathbf{y}, X) \propto p\left(\mathbf{y}|X, \exp(\boldsymbol{\ell})\right) p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right) \tag{8}$$

$$\log p(\boldsymbol{\ell}|\mathbf{y}, X) \propto \log p\left(\mathbf{y}|X, \exp(\boldsymbol{\ell})\right) + \log p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right) \tag{9}$$

| Symbol | Meaning |
|---|---|
| $\mathcal{S}$ | a set of air quality stations |
| $\mathcal{S}^*$ | a set of unmonitored locations |
| $\mathcal{T}$ | a set of time-stamps |
| $X, X_n$ | training data (features) of size $n$ |
| $\mathbf{y}, \mathbf{y}_n$ | training data (observations) |
| $\sigma^2$ | noise variance in data |
| $X^*, X_t$ | test data (features) of size $t$ |
| $\mathcal{GP}_y$ | a Gaussian process over $PM_{2.5}$ observations |
| $f(\mathbf{x}), \mathbf{f}$ | $\mathcal{GP}_y$ latent function values |
| $m(\mathbf{x})$ | mean function of $\mathcal{GP}_y$ |
| $k(\mathbf{x}, \mathbf{x}')$ | covariance function or kernel evaluated at $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{GP}_y$ |
| $K, K_f$ | $n \times n$ covariance matrix for $\mathbf{f}$ values |
| $K_y = K_f + \sigma^2 I$ | $n \times n$ noisy covariance matrix for $\mathbf{y}$ values |
| $K^*$ | $n \times t$ covariance matrix between train and test inputsin $\mathcal{GP}_y$ |
| $K^{**}$ | $t \times t$ covariance matrix among test inputs in $\mathcal{GP}_y$ |
| $\sigma_f^2$ | kernel variance in $\mathcal{GP}_y$ |
| $\boldsymbol{\ell}$ | length scale vector at $\mathbf{x} \in X$ in $\mathcal{GP}_y$ |
| $\mathcal{L}$ | Loss function |
| $\mathbf{f}^*$ | $\mathcal{GP}_y$ posterior function values |
| $\mathcal{GP}_\ell$ | a Gaussian process over the length scales |
| $f_\ell(\mathbf{x}), \mathbf{f}_\ell$ | $\mathcal{GP}_\ell$ latent function values |
| $m_\ell(\mathbf{x})$ | mean function of $\mathcal{GP}_\ell$ |
| $k_\ell(\mathbf{x}, \mathbf{x}')$ | covariance function or kernel evaluated at $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{GP}_\ell$ |
| $\overline{K}, \overline{K}_{f_\ell}$ | $m \times m$ covariance matrix for $\mathbf{f}_\ell$ values |
| $\overline{\sigma}_f^2$ | kernel variance in $\mathcal{GP}_\ell$ |
| $\overline{\sigma}^2$ | noise variance in $\mathcal{GP}_\ell$ |
| $\overline{\sigma}_l^2$ | kernel length scale in $\mathcal{GP}_\ell$ |
| $\overline{X}$ | Inducing points |
| $\overline{\mathbf{x}}_j^T$ | Inducing points for $j^{th}$ feature |
| $\overline{K}_\ell = \overline{K}_{f_\ell} + \overline{\sigma}^2 I$ | $m \times m$ noisy covariance matrix among inducing points in $\mathcal{GP}_\ell$ |
| $\overline{K}^*$ | $m \times n$ covariance matrix between train and inducing points in $\mathcal{GP}_\ell$ |
| $\overline{K}^{**}$ | $n \times n$ noisy covariance matrix for $\boldsymbol{\ell}$ values |

Table 2: Mathematical notations used in the paper.

As per Eq. 9, maximizing $\log p(\boldsymbol{\ell}|\mathbf{y}, X)$ is same as maximizing $\log p\left(\mathbf{y}|X, \exp(\boldsymbol{\ell})\right) + \log p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right)$. Now, $\log p\left(\mathbf{y}|X, \exp(\boldsymbol{\ell})\right)$ has the standard GP marginal likelihood form:

$$\log p\left(\mathbf{y}|X, \exp(\boldsymbol{\ell})\right) = -\frac{1}{2}\left[\mathbf{y}^T K_y^{-1}\mathbf{y} + \log|K_y| + \log(2\pi)\right] \tag{10}$$

Furthermore, $\log p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right)$ is log posterior density of $\boldsymbol{\ell}$.

$$\log p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right) = \log \mathcal{N}(\boldsymbol{\mu}_\ell, \Sigma_\ell) \tag{11}$$

$$\boldsymbol{\mu}_\ell = \overline{K}^{*T}\overline{K}_\ell^{-1}\overline{\boldsymbol{\ell}} \tag{12}$$

$$\Sigma_\ell = \overline{K}^{**} - \overline{K}^{*T}\overline{K}_\ell^{-1}\overline{K}^* \tag{13}$$

Plugging Eq. 12 and Eq. 13 in Eq. 11 following a similar procedure as in Section C, we get the following form:

$$\log p\left(\boldsymbol{\ell}|X, \overline{\boldsymbol{\ell}}, \overline{X}\right) = -\frac{1}{2}\left[\log|\overline{K}^{**} - \overline{K}^{*T}\overline{K}_\ell^{-1}\overline{K}^*| + n\log 2\pi\right] \tag{14}$$

Now, we can optimize the joint loss given in Eq. 9 as a summation of Eq 14 and Eq 10.

# E Automatic Relevance Determination (ARD)

We have briefly described ARD in Section 4.3 in the paper. Here, we provide further mathematical details. We formulated RBF kernel as the following in the paper:

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||_2^2}{2\ell^2}\right) \tag{15}$$

We can write a set of stationary kernels with a general notation having a single input $\tau$ (scaled distance) (Garg, Singh,

and Ramos 2012).

$$\tau = \frac{||\mathbf{x} - \mathbf{x}'||_2^2}{\ell^2} \tag{16}$$

$$K_{RBF}(\tau) = \sigma_f^2 \exp\left(-\frac{1}{2}\tau\right) \tag{17}$$

$$K_{Matern12}(\tau) = \sigma_f^2 \exp\left(-\sqrt{\tau}\right) \tag{18}$$

In Eq. 16, $\ell$ (length scale) is a scalar and thus each feature is scaled by the same value. This is non-ideal for datasets where features have diverse ranges and/or several features are not informative in modeling the desired phenomenon. Intuitively, we should have separate length scales for each feature to adapt the variable smoothness in each feature. Thus, a more general formulation of $\tau$ is as the following (Rasmussen and Williams 2005) (Section 5.1):

$$\tau = (\mathbf{x} - \mathbf{x}')^T M^{-1} (\mathbf{x} - \mathbf{x}') \tag{19}$$

For $\tau$ in Eq. 16, We should have $M = \ell^2 I$. To enable ARD behaviour, we simply choose $M = \text{diag}(\boldsymbol{\ell})^2$ in our work. Note that this version of ARD is commonly used in well-known Gaussian process libraries (GPy since 2012; Gardner et al. 2018). One more way of choosing $M$ is $M = \Lambda\Lambda^T + \text{diag}(\boldsymbol{\ell}_n)^2$, where $\Lambda$ is a $n \times k$ matrix.

To explore ARD in further detail, we would like to redirect reader to Section 5.1 of (Rasmussen and Williams 2005).

## References


Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN 0387310738.

Gardner, J. R.; Pleiss, G.; Bindel, D.; Weinberger, K. Q.; and Wilson, A. G. 2018. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *arXiv preprint arXiv:1809.11165*.

Garg, S.; Singh, A.; and Ramos, F. 2012. Learning Non-Stationary Space-Time Models for Environmental Monitoring. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, 288–294. AAAI Press.

GPy. since 2012. GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy.

Rasmussen, C. E.; and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN 026218253X.