



HAL
open science

Distributed Minimum Temperature Prediction Using Mixtures of Gaussian Processes

Sergio Hernández, Philip Sallis

► **To cite this version:**

Sergio Hernández, Philip Sallis. Distributed Minimum Temperature Prediction Using Mixtures of Gaussian Processes. 11th International Symposium on Environmental Software Systems (ISESS), Mar 2015, Melbourne, Australia. pp.484-491, 10.1007/978-3-319-15994-2_49 . hal-01328596

HAL Id: hal-01328596

<https://inria.hal.science/hal-01328596v1>

Submitted on 8 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Distributed Minimum Temperature Prediction using Mixtures of Gaussian Processes

Sergio Hernández¹ and Philip Sallis²

¹ Laboratorio de Procesamiento de Información Geoespacial.
Universidad Católica del Maule, Talca, Chile

² Geoinformatics Research Centre.
Auckland University of Technology, Auckland, New Zealand.

Abstract. Minimum temperature predictions are required for agricultural producers in order to assess the magnitude of potential frost events. Several regression models can be used for the estimation problem at a single location but one common problem is the amount of required data for training, testing and validation. Nowadays, sensor networks can be used to gather environmental data from multiple locations. In order to alleviate the amount of data needed to model a single site, we can combine information from the different sources and then estimate the performance of the estimator using hold-out test sites. A mixture of Gaussian Processes (MGP) model is proposed for the distributed estimation problem and an efficient Hybrid Monte Carlo approach is also proposed for the estimation of the model parameters.

1 Introduction

Crop damages due to frost events is an important issue in many agricultural areas. There are several empirical models that attempt to predict or anticipate the occurrence of the minimum temperature at a given time and location. These methods deliver descriptive results, however calibration is required for the model parameters in order to account for the time of the year and the local conditions [13]. Artificial neural networks (ANN) models have been previously proposed as an alternative to the empirical methods for the estimation task [5]. For instance, ANN models were used to predict air temperature at hourly intervals from one to 12 hours ahead [12] using meteorological data such as temperature, relative humidity, solar radiation, wind speed and direction, as well as seasonal variables to account for the day of the year.

More recently, an ensemble of twelve ANN models was used to achieve temperature predictions throughout an entire year [11]. The ANN models were able to achieve low prediction error when using a large dataset for training consisting of 1.5 million records. To alleviate the issues related to the high dimensionality estimation problem, a support vector regression model was proposed for the prediction task using a reduced training set (300,000 records) [3].

In this paper we present a Bayesian framework for minimum temperature prediction using multiple sources of environmental data. A Mixture of Gaussian

Process (MGP) model is then proposed as an alternative to the ANN model for dealing with a reduced amount of data. The idea behind the MGP approach is to use local Gaussian Process regression models on different subsets of the input data and a gating network that is responsible of assigning input data points to one of the local experts [14].

The parameters of the MGP model includes a set of hyper-parameters for each Gaussian process model and the gating network. A Bayesian approach for estimating the model hyper-parameters from data is also presented [10]. Nevertheless, even for the reduced amount of data, the running time of the algorithm can be prohibitively slow. Therefore, we propose an efficient Hybrid Monte Carlo implementation in a shared memory machine that may overcome some of the estimation issues in high dimensional problems.

As a result, our framework is able to deal with deal with a large number of instances and deliver several predictions at test locations.

2 Gaussian Processes Regression

Regression problems deal with datasets in the form $\mathcal{D} = \{\mathbf{x}, y\}_{i=1}^n$, where \mathbf{x}_i is an input vector and y_i is a continuous variable representing output data. In our case, the input data contains several environmental variables such as temperature, humidity and solar radiation measured at different times of the day (e.g. 18pm and 15pm) and the output data is the minimum temperature measured at any time of the following next day.

A Gaussian process regression model consists of a non-linear mapping between the input and the output data, such that:

$$y_i = f(\mathbf{x}_i) + \epsilon_i \quad (1)$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ represents a zero-mean noise term. More formally, a Gaussian process defines a prior over the function $f(\mathbf{x}) \sim GP(0, \kappa(\mathbf{x}, \mathbf{x}'))$ and \mathbf{K} is a positive-definite kernel or covariance matrix with elements $[K_{ij}] = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. Now, we can write the marginal distribution $p(y|\mathbf{x})$ as:

$$p(y|\mathbf{x}) = \int p(y|f, \mathbf{x})p(f|\mathbf{x})df \quad (2)$$

Integrating f in Equation 2 also leaves a normal distribution with covariance $\mathbf{K}_x = \mathbf{K} + \sigma^2\mathbf{I}$. If we now observe new data \mathbf{x}_* , we can write the predictive distribution as follows:

$$p(f_*|\mathbf{x}_*, \mathcal{D}) = \mathcal{N}(\mathbf{k}_*^T \mathbf{K}_x^{-1} y, \mathbf{k}_{**} - \mathbf{k}_*^T \mathbf{K}_x^{-1} \mathbf{k}_*) \quad (3)$$

where \mathbf{k}_* denotes the vector of covariances between the test point and the training data. Similarly, $\mathbf{k}_{**} = \kappa(\mathbf{x}_*, \mathbf{x}_*)$ is the covariance function of the test data.

2.1 Mixtures of Gaussian Processes

Wireless sensor networks make use of sensor nodes to collect environmental data from multiple locations [15, 9]. Instead of having one single dataset, our interest is to gather data from different farming sites and perform centralized learning using a horizontally partitioned dataset $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \dots \cup \mathcal{D}_n$. Figure 1 depicts the data collection mechanism.

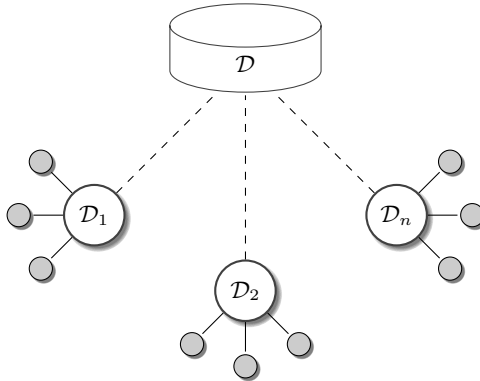


Fig. 1: Distributed environmental data from wireless sensor networks. Data is collected from multiples sites, each one having multiple sensor nodes. Every node reports the same information collected at different spots.

Compared to the cost of analyzing single datasets, the computational complexity of learning from a central dataset is increased [8]. This is specially problematic in Gaussian processes calculations that uses inversion of matrices and linear solvers that require approximation methods for large datasets [6]. One of such approaches is the Mixture of Gaussian Processes (MGP), that builds several regression or classification models on different subsets of the data. In this case, the complexity of performing large-scale training is reduced since each model can be trained independently and because of the centralized nature of the data collection mechanism, there is no requirement on the number of models to be trained.

In this case, we consider a set of models which are responsible of different groups of measurements $\mathcal{S} = \{S_1, \dots, S_M\}$ [10]. A gating network parametrized by a discrete latent variable \mathbf{z} is used to assign data points to any of the M local experts with prior probability π_m , such that $p(z_i = m) = \pi_m$. The MGP prior can be now written as follows:

$$f(\mathbf{x}) \sim \sum_i^M \pi_i GP(0, \kappa_i(\mathbf{x}, \mathbf{x}')) \quad (4)$$

3 Fast Inference for Mixtures of Gaussian Processes

The parameters of the MGP model include a set of hyper-parameters for each model $\Theta = (\theta_1, \dots, \theta_M)$ and the gating network $\pi = (\pi_1, \dots, \pi_M)$. In this paper we consider kernels based on the squared exponential function:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_d \nu_d^2 (x_{id} - x_{jd})^2\right) + \sigma_y^2 \delta_{ij} \quad (5)$$

where $\theta = (\sigma_f, \sigma_y, \nu_1, \dots, \nu_D)$ and D being the dimensionality of the input vector \mathbf{x} . The posterior distribution of the model and the gating network parameters takes the form:

$$p(\Theta, \pi | \mathcal{D}) \propto p(\Theta, \pi) p(\mathcal{D} | \Theta, \pi) \quad (6)$$

Unlike Equation 3, in the MGP model we cannot compute the marginal posterior analytically. Instead, we could use the E-M algorithm to iteratively maximize the complete data log-likelihood $\log p(\mathcal{D} | \Theta, \pi)$ [16]. Instead, we sample from the marginal posterior distribution (see Equation 6) using a Hybrid Monte Carlo approach. The iterative procedure is given as follows:

1. Given the current kernel hyper-parameters Θ , sample the gating network parameters from the posterior distribution $p(z_i = m | \Theta, \mathcal{D})$
2. Sample π from a Dirichlet distribution $p(\pi_1, \dots, \pi_M) \sim \text{Dir}(\delta + c_1, \dots, \delta + c_M)$, where δ is a concentration parameter and c_m is the number of times $z_i = m$ for all i .
3. Update Θ using hybrid Monte Carlo sampling.

Distributed Gibbs sampling for latent variable models were discussed in [1]. In this case, we can partition the dataset according to the expert assignments, such that a different processor performs computations on different data subsets S_m . GP computations require several linear algebra operations, however authors have noticed that good performance on multi-core machines can be achieved by using threaded BLAS libraries [7, 2].

4 Experimental Results

4.1 Dataset

In this section we provide experimental results for the distributed minimum temperature prediction using the MGP approach. The data consists of agro-meteorological variables collected using Wireless Sensor Networks from 5 different locations in the region of Maule in south central Chile³. The data was

³ <http://www.agrosense.cl>

sampled every 10 minutes but only the mean temperature, mean humidity and mean solar radiation values at 15pm and 18pm were used to predict the minimum temperature for next the day. Table 1 describes the variables used in the predictive model. In order to create the training and test patterns, each node is used independently. Therefore, in the training data set we incorporate sensor nodes from different locations. Table 1 also shows the locations of the sensor nodes used in the training and test datasets.

x_1	Temp15	Mean Temperature at 15hrs	Lat	Lon	Site ID
x_2	Hum15	Mean Humidity at 15hrs	-35.463882	-71.612818	donoso_1
x_3	Rad15	Mean Solar Radiation at 15hrs	-35.466391	-71.617502	donoso_2
x_4	Temp18	Mean Temperature 18hrs	-35.588004	-71.910842	gillmore_2
x_5	Hum18	Mean Humidity 18hrs	-35.857414	-71.602135	niceblue_1
x_6	Rad18	Mean Solar Radiation at 18hrs	-35.858761	-71.601540	niceblue_2
			-35.858845	-71.601852	niceblue_3
y	<i>MinNext</i>	Next day minimum temperature	-35.013470	-71.432918	canepa_1

Table 1: Variables used for minimum temperature prediction and sensor nodes locations

4.2 Model

The priors used for the gating network and the kernel hyper-parameters follow the settings suggested in [6, 10]. In order to get the initial values of Θ , we partition the data using k-means and ran 100 iterations of Hybrid Monte Carlo with 10 leapfrog steps. The Hybrid Monte Carlo sampler was done in C++ and the data partitioning and parallel execution was performed in R. In order to ensure that valid random numbers were generated in parallel, the TNRG random number generator was used. This approach combines the flexibility of the R environment for data processing with the fast execution of C++ shared libraries [4].

A number of 1000 iterations and 3 leapfrog sets were used to train the MGP model. Figure 2 compares the predicted and observed minimum temperatures at the training sites.

As shown in Figure 2, each model produces different predictions. Figure 3 shows the predicted minimum temperatures at the test site. Figure 3 shows the predicted minimum temperatures at a single test site.

Table 2 summarizes results using the Root Mean Square (RMS) error for the 3 test sites. It is important to notice that the ensemble RMS error is lower for all test locations.

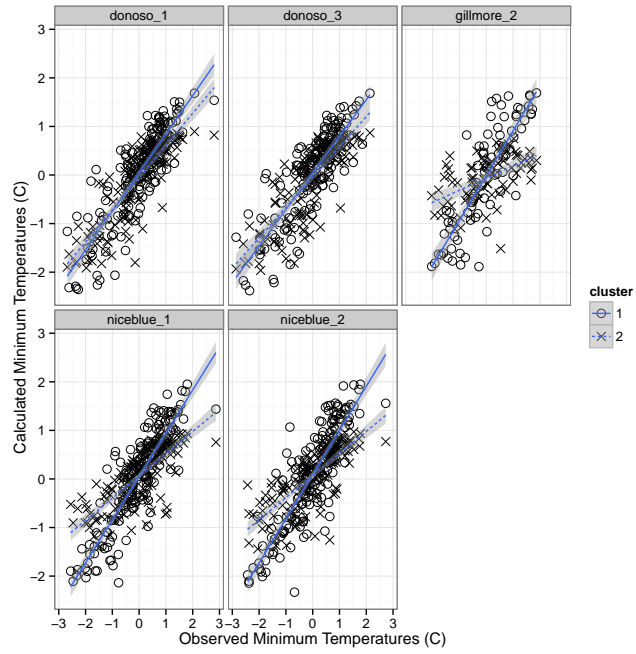


Fig. 2: Comparison between observed and calculated next day minimum temperatures at training locations

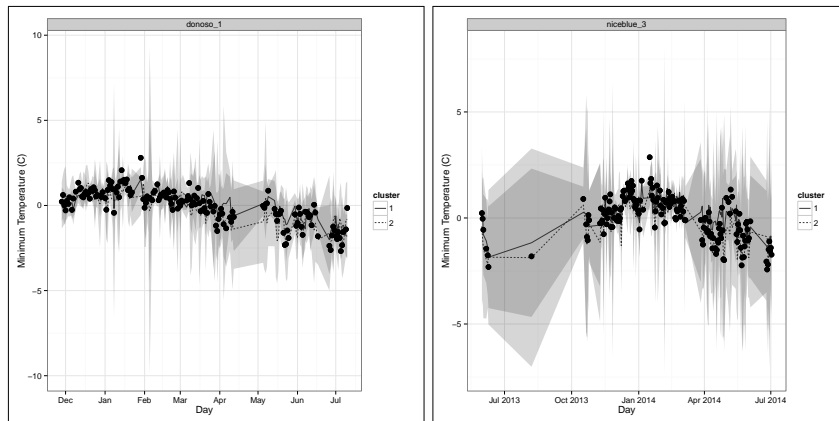


Fig. 3: Minimum temperatures at a training and test locations

5 Conclusions

In this paper we combine data from multiple and distributed sources and we are able to predict at previously unseen test locations. A mixture of Gaussian

Site ID	RMS Cluster 1	RMS Cluster 2	RMS Ensemble Mean
niceblue_3	0.4314506	0.4521971	0.3712023
canepa_1	0.5649653	0.6597936	0.5737617
canepa_2	0.7197888	0.6106701	0.5600180
canepa_3	0.5906937	0.6441202	0.5677543

Table 2: RMS values for the MGP model at test locations

process model is introduced as a method to handle the distributed learning problem. The model has the appealing property of being able to handle multi-modal regression or classification tasks. This is a new approach for distributed learning, since most methods used in the literature use locally trained models and then combine the output. In our case, we overcome the problem of having the same number of models and training locations by imposing the number of mixture components. However, this is also one of the limitations of this approach since we need to choose beforehand the number of components. Future work will involve automatic methods for model selection and model averaging procedures for the MGP model.

References

1. Asuncion, A., Smyth, P., Welling, M., Newman, D., Porteous, I., Triglia, S.: Distributed Gibbs sampling for latent variable models. In: *Scaling up Machine Learning* (2012)
2. Chandola, V., Vatsavai, R.R.: Implementing a gaussian process learning algorithm in mixed parallel environment. In: *Proceedings of the Second Workshop on Scalable Algorithms for Large-scale Systems*. pp. 3–6. *Scala '11*, ACM, New York, NY, USA (2011)
3. Chevalier, R.F., Hoogenboom, G., McClendon, R.W., Paz, J.A.: Support vector regression with reduced training sets for air temperature prediction: a comparison with artificial neural networks. *Neural Comput. Appl.* 20(1), 151–159 (feb 2011)
4. Eddelbuettel, D.: *Seamless R and C++ Integration with Rcpp*. Springer Publishing Company, Incorporated (2013)
5. Jain, A., McClendon, R., Hoogenboom, G., Ramyaa, R.: Prediction of frost for fruit protection using artificial neural networks. *American Society of Agricultural Engineers, St. Joseph, MI, ASAE Paper* pp. 03–3075 (2003)
6. Neal, R.: Regression and classification using Gaussian process priors. In: *Bayesian Statistics 6: Proceedings of the sixth Valencia international meeting*. vol. 6, p. 475 (1998)
7. Nguyen, T., Bonilla, E.: Fast allocation of gaussian process experts. In: *Proceedings of The 31st International Conference on Machine Learning*. pp. 145–153 (2014)
8. Peteiro-Barral, D., Guijarro-Berdias, B.: A survey of methods for distributed machine learning. *Progress in Artificial Intelligence* 2(1), 1–11 (2013)
9. Ruiz-Garcia, L., Lunadei, L., Barreiro, P., Robla, I.: A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends. *Sensors* 9(6), 4728–4750 (2009)

10. Shi, J.Q., Murray-Smith, R., Titterton, D.: Hierarchical gaussian process mixtures for regression. *Statistics and Computing* 15(1), 31–41 (2005)
11. Smith, B.A., Hoogenboom, G., McClendon, R.W.: Artificial neural networks for automated year-round temperature prediction. *Computers and Electronics in Agriculture* 68(1), 52 – 61 (2009)
12. Smith, B.A., McClendon, R.W., Hoogenboom, G.: Improving air temperature prediction with artificial neural networks. *International Journal of Computational Intelligence* 3(3), 179–186 (2006)
13. Snyder, R.L., de Melo-Abreu, J.: Frost forecasting and monitoring. *Frost Protection: Fundamentals, Practice, and Economics* 1, 91–112 (2005)
14. Tresp, V.: Mixtures of Gaussian processes. In: *Advances in Neural Information Processing Systems*. pp. 654–660 (2001)
15. Wang, N., Zhang, N., Wang, M.: Wireless sensors in agriculture and food industry—recent development and future perspective. *Computers and Electronics in Agriculture* 50(1), 1 – 14 (2006)
16. Yang, Y., Ma, J.: An efficient EM approach to parameter learning of the mixture of gaussian processes. In: *Advances in Neural Networks—ISNN 2011*, pp. 165–174. Springer (2011)