Improving Gaussian Processes Classification by Spectral Data Reorganizing

Hang Zhou
Dept Elec. & Comp. Syst. Eng.
Monash University, Clayton, VIC 3800, Australia
hang.zhou@eng.monash.edu.au

David Suter
Dept Elec. & Comp. Syst. Eng.
Monash University, Clayton, VIC 3800, Australia
d.suter@eng.monash.edu.au

Abstract

We improve Gaussian processes (GP) classification by reorganizing the (non-stationary and anisotropic) data to better fit to the isotropic GP kernel. First, the data is partitioned into two parts: along the feature with the highest frequency bandwidth. Secondly, for each part of the data, only the spectrally homogeneous features are chosen and used (the rest discarded) for GP classification. In this way, anisotropy of the data is lessened from the frequency point of view. Tests on synthetic data as well as real datasets show that our approach is effective and outperforms Automatic Relevance Determination (ARD).

1. Introduction

A Gaussian process (GP) is fully specified by its mean function m(x) and kernel function k(x, x') , expressed as:

$$f \sim GP(m,k) \quad (1.1)$$

The kernel function studied in this paper is the Radial Basis Function (RBF) [1]

$$k_{RBF}(x-x') = \sigma_0^2 \exp[-\frac{1}{2}(\frac{x-x'}{l})^2] \quad (1.2)$$

where x and x' are input pairs, l is the characteristic length scale and σ_0 is the signal variance.

The widely used RBF kernel is an isotropic kernel whose big advantage is its simplicity.

One should not blindly apply a GP model to the data. Analysis of the data can reveal some anisotropic properties that can be lessened. Essentially, GP classification is the problem of how the usually anisotropic data can be better modeled by the isotropic GP kernel.

In our approach, the data anisotropy is lowered on two aspects, i.e., partitioning to make the data on each dimension have more consistent spectral measures as well as picking out those dimensions where the spectral measures are more homogeneous.

There are some existing approaches to GP data partitioning. Gramacy and Lee combined the stationary GP and linear models with tree partitioning, to model non-stationary data.[2] Kim et al divided sharp changing spatial data into disjoint stationary regions [3]. Both solutions involve inference via Markov Chain Monte Carlo (MCMC) which is computationally costly.

Work has also been done to determine the relevance of multiple inputs in supervised learning, in particular Automatic Relevance Determination (ARD) (developed by David Mackay and Radford Neal [4]). ARD is done by adapting the values of the hyperparameters associated with each input. The hyperparameters are given prior distributions, then the posterior distribution is calculated given the training data.

An issue for ARD is: what prior should be used for the hyperparameters? (as it will have a significant impact on the estimation result accuracy of the hyperparameters). Also it is costly to compute.

We present a new solution to lessen the anisotropy of the data in two steps: First, the data is partitioned into two parts along the feature with the highest frequency bandwidth value (see Section 3.1). Secondly, for each of the partitioned dataset, the input features with similar "effective dwell time" values (see Section 3.2) are chosen from the input vector (and the rest being discarded).

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

This paper is organized as follows. Non-uniform data spectral analysis is first introduced in Section 2. The data reorganizing algorithm is then described in Section 3, followed by experiment details and results in Section 4. Finally, Section 5 provides the main conclusions of the work.

2. Non-uniform data spectral analysis

In order to analyze the (usually) high dimensional and unevenly spaced training data, on each dimension individually and efficiently, we apply the non-uniform discrete Fourier transform (NDFT) [5] to each dimension separately.

Then we employ the notion of "effective dwell time"[1] from Bretthorst's work [6].

As stated in [6], for uniformly sampled data in the time domain, let N be the total number of samples, and f_k be the frequency. The locations of the uniform data samples are given by

$$t_j = j\Delta T, \quad j \in \{0,1...N-1\} \tag{2.1}$$

where ΔT is the time interval between data samples - called 'dwell time'.

The bandwidth is given as

$$-f_{N_c} \le f \le f_{N_c} \tag{2.2}$$

$$\text{where } f_{N_c} = \frac{1}{2\Delta T} \tag{2.3}$$

In non-uniform data, there is no ΔT that all the acquisition times are integer multiples of. However, there exists (see next section) an effective dwell time $\Delta T'$ where we have (approximately)

$$t_l = k_l \Delta T' \tag{2.4}$$

where k_l is an integer. $\Delta T'$ is always less than or equal to the smallest time interval between data items.

Similar to eqn (2.3), for nonuniform data, we have

$$f_{N_c} = \frac{1}{2\Delta T'} \tag{2.5}$$

3. Data reorganizing

3.1 Data partitioning

As described in Section 1, data is partitioned into two parts along the feature (dimension) with the highest frequency bandwidth (or smallest effective dwell time). Data partitioning results in new partitions each of which generally has less variation than the original..

Effective dwell time $\Delta T'$ can be estimated using eqn (2.4) where $\Delta T'$ can be regarded as the maximum common divisor of all the times t_l.

In our application, t_l is equivalent to the feature data locations (i.e. the projected high dimensional input data on each dimension) and $\Delta T'$ is related to the location difference. t_l is typically not a integer, therefore, a practical and approximate way to implement eqn (2.4) for non-integer data is to

1) Multiply each t_l by a factor c (set to 10000) and truncate to an integer.
2) Let $d_{\min}$ be the smallest interval between data points. Multiply $d_{\min}$ by c and truncate as well.
3) Calculate how well all t_l can be divided by $d_{\min}$

$$R = \frac{1}{N}\sum_{l=1}^{N}\frac{\text{mod}(t_l, d_{\min})}{t_l} \tag{3.1}$$

where N is the total number of data point and mod is modulus after division..

4) If $d_{\min}$ is effectively a common divisor of all t_l then $R < R_{thres}$ for, say, $R_{thres} = 0.01$. If this is not currently true then we need to decrease $d_{\min}$ further i.e.,

$$\text{If} \quad R < R_{thres} \tag{3.2}$$

$$\text{Then} \quad \Delta T' = d_{\min} \tag{3.3}$$

$$\text{Else} \quad d_{\min} = d_{\min} - 1 \tag{3.4}$$

5) Repeat step 3) and 4) until eqn (3.2) is satisfied.

As a result, effective dwell time $\Delta T'$ has been estimated following eqn (3.3).

The data is partitioned as follows: On the dimension with the highest bandwidth, locate the position where the number of data points on either side of that are equal. Although dividing the data into two equal parts may not be the optimized way of partitioning, it is effective in reducing the data variation, and therefore the data in each segment is more isotropic.

3.2 Feature pruning

For the two partitioned datasets created in the previous section, the most relevant features are chosen and the rest removed. We define this relevancy as having similar "effective dwell time".

[1] In our application, data is collected in the (feature) space domain. Therefore, simply substitute time with feature data location values.

Assume $\Delta T_i^{'}, i = 1...d$ is the effective dwell time on each dimension where d is the number of features. Feature pruning is done in the following way:

1) Cluster $\Delta T_i^{'}$ into two parts using fuzzy c-means (i.e., degree of membership are given).
2) In each of the two clusters, pick out the features with the fuzzy membership function values greater than the preset threshold which is 0.5 in this paper.
3) Retain the features that are in the largest cluster, and discard the rest.

As a result of this feature selection and the previous data partitioning, we now have two sets of data, each with individually reduced features: in classification each part is used completely separately.

4. Experiments and results

We run Lawrence's program[2] [7] for GP classification.

4.1 Synthetic data

The 3D synthetic data file has 300 data points with the bandwidth set to 59, 29 and 1 on each dimension respectively (a big variation on bandwidth among features). Its frequency content and the signal wave are shown in Figure 1 on top and bottom respectively.

Effective dwell time values $\Delta T'$ and the corresponding bandwidths are first calculated as shown in Table 1 where the largest bandwidth is on "dimension 1". So the synthetic data is partitioned along this dimension into two equal parts whose effective dwell time and bandwidth are again calculated and listed in Table 2.

Then, apply a fuzzy c-means clustering (into 2 clusters) to the dwell time values of each of the partitioned dataset (the membership function matrixes are shown in Table 3). It can be seen that both the partitioned datasets and the original data choose dimension 1 and 2 on feature pruning.

For comparison, GP classification is applied to datasets in four situations, i.e. the original data, the feature pruned (only) data, partitioned (only) data and the partitioned –feature pruned data.

Table 4 shows the GP classification results. It can be seen that the by either pruning the features or partitioning the data; better classification results can be obtained. When applying both partitioning and pruning, the GP classification results outperform all the other situations

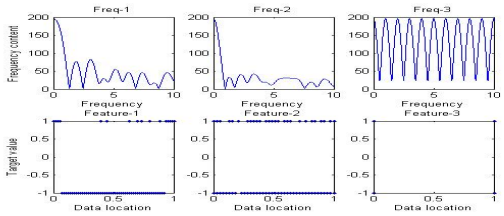

Figure 1. Synthetic data on frequency and signal domain.

Table 1. Dwell time (DT) and bandwidth (BW) of the synthetic data.

	DT	BW
Dim1	0.0169	59.0
Dim2	0.0256	39.0
Dim3	1.0	1.0

Table 2. Dwell time (DT) and bandwidth (BW) of the two partitioned synthetic data.

	DT-1	BW-1	DT-2	BW-2
Dim1	0.0357	28.0	0.0333	30.0
Dim2	0.0256	39.0	0.0256	39.0
Dim3	1.0	1.0	1.0	1.0

Table 3. Dwell time membership function matrix of the original and partitioned synthetic data.

	Original data		Data-1		Data-2	
	Cluster-1	Cluster-2	Cluster-1	Cluster-2	Cluster-1	Cluster-2
Dim1	1.0	0.0	1.0	0.0	1.0	0.0
Dim2	1.0	0.0	1.0	0.0	1.0	0.0
Dim3	0.0	1.0	0.0	1.0	0.0	1.0

Table 4. Classification performance comparison of different processed synthetic data.

	Detection rate	False positives	Total points
Original data	0.8462	1	300
Feature pruned only data	0.8654	1	300
Partitioned data-1	0.8333	0	141
Partitioned data-2	0.8824	1	159
Merged partitioned data	0.8593	1	300
Partitioned data-1 (feature pruned)	0.9444	0	141
Partitioned data-2 (feature pruned)	0.8824	1	159
Merged partitioned data (feature pruned)	0.9115	1	300

4.2 Real datasets

32 data files are (randomly) chosen from 'The UCI Repository of Machine Learning Databases and

[2] http://www.cs.man.ac.uk/~neill/ivm/downloadFiles/

Domain Theories' [3], the Pattern Recognition and Neural Networks Datasets'[4] and the 'Delve Datasets'[5]. The number of features in each file ranges from 2 to 60.

The overall GP classification performance of the 32 data files using RBF GP, RBFARD GP, feature pruned only GP and the data reorganized (partition-feature pruned) GP are compared in Figure 2 (detection rate) and Figure 3 (false positives).

It can be seen that RBFARD GP, feature pruned GP and the data reorganized GP have achieved better classification results compared with the standard RBF GP. The "data reorganized" GP further outperforms RBFARD and "feature pruned" GP on both detection rate and false positives values.

Therefore, the partition-feature pruning approach is effective and works well on diversified datasets with varied feature (dimension) numbers.

5. Conclusions

An effective new way is proposed in this paper to improve the GP classification performance by data reorganizing based on the analysis in frequency domain. The data reorganizing not only partitions the data, but also chooses the more homogeneous features: so as to be better represented by an isotropic kernel. Tests on both synthetic data and real datasets show that the approach enhances the GP classification performance. Comparing with ARD, it achieves better results.

Our approach is intuitive but a little ad hoc. It will be interesting to seek an optimized way of pruning the features so as to *ensure* that the most informative features are extracted and kept.

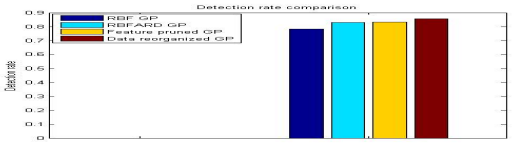

Figure 2. Detection rate comparison of 32 data files between RBF GP, RBFARD GP, feature pruned GP and data reorganized GP.

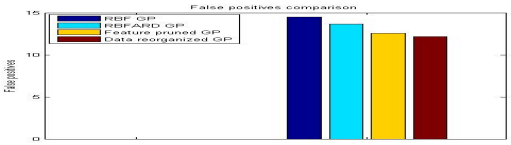

Figure 3. False positives comparison of 32 data files between RBF GP, RBFARD GP, feature pruned GP and data reorganized GP.

References

[1] E. Snelson, "Tutorial: Gaussian Process Models for Machine Learning," Gatsby Computational Neuroscience Unit, UCL 2006.

[2] R. B. Gramacy, "Bayesian Treed Gaussian Process Models," in *Department of Applied Math & Statistics* Santa Cruz: University of California, Santa Cruz, 2005.

[3] H.-M. Kim, B. K. Mallick, and C. C. Holmes, "Analyzing Nonstationary Spatial Data Using Piecewise Gaussian Processes," *Journal of American Statistical Association,* vol. 100, pp. 653-668, 2005.

[4] R. M. Neal, *Bayesian Learning for Neural Networks*: Springer-Verlag New York, Inc, 1996.

[5] S. Bagchi and S. K. Mitra, *The Nonuniform Discrete Fourier Transform and Its Applications in Signal Processing*: Kluwer Academic Publishers, 1999.

[6] G. L. Bretthorst, "Nonuniform sampling: bandwidth and aliasing," *Maximum Entropy and Bayesian Methods in Science and Engineering,* pp. 1-28, 2000.

[7] N. D. Lawrence, J. C. Platt, and M. I. Jordan, "Extensions of the Informative Vector Machine," in *Deterministic and Statistical Methods in Machine Learning*, 2004.

[3] http://mlearn.ics.uci.edu/databases/

[4] http://www.stats.ox.ac.uk/pub/PRNN/

[5] http://www.cs.toronto.edu/~delve/data/datasets/html