Journal of Southeast University (English Edition) Vol. 21, No. 1, pp. 44 - 47 Mar. 2005 ISSN 1003—7985

Application of thermal parameter soft sensor in power plant

Xiong Zhihua[1] Zhu Feng[2] Shao Huihe[1]

([1] Institute of Automation, Shanghai Jiaotong University, Shanghai 200030, China)

([2] Henan Electric Power Research Institute, Zhengzhou 450052, China)

Abstract: In order to solve the problem of the invalidation of thermal parameters and optimal running, we present an efficient soft sensor approach based on sparse online Gaussian processes (GP), which is based on a combination of a Bayesian online algorithm together with a sequential construction of a relevant subsample of the data to specify the prediction of the GP model. By an appealing parameterization and projection techniques that use the reproducing kernel Hilbert space (RKHS) norm, recursions for the effective parameters and a sparse Gaussian approximation of the posterior process are obtained. The sparse representation of Gaussian processes makes the GP-based soft sensor practical in a large dataset and real-time application. And the proposed thermal parameter soft sensor is of importance for the economical running of the power plant.

Key words: Gaussian process; soft sensor; sparse approximation; online learning; economical monitoring

With the development of technology and the complication of thermal processes, we need to implement real-time control or optimization for variables to the process in order to ensure the safety and efficiency. However, some important variables are difficult to detect due to the limitation of techniques and technology. Furthermore, we tend to meet the difficulty of improper parameters for economical monitoring. For the latter, we usually use the rated value to replace the value of improper parameters, but the error is evident in most peaking power units. Soft sensors[1] provide a convenient solution to eliminate the above problems.

1 GP-Based Soft Sensor

The soft sensor model can be described as follows: Given the sample $\{\boldsymbol{x}^i, t^i\}_{i=1}^n$, where n is the size of training data, $\boldsymbol{x}^i \in \mathbf{R}^d$ are inputs of the soft sensor model, and $t^i \in \mathbf{R}$ are the corresponding desired targets, suppose that the input vector for a test case is denoted as $\boldsymbol{x}$ and the inputs are d-dimensional $x_1, \dots, x_d$ and the targets are scalar. From this training set we want to learn a model of the dependency of the targets on the inputs with the objective of making accurate predictions of $\boldsymbol{t}$ for previously unseen values of $\boldsymbol{x}$.

The predictive distribution for a test case $\boldsymbol{x}$ is obtained from the $(n+1)$-dimensional joint Gaussian distribution for the outputs of the n training cases and the test case, by conditioning on the observed targets in the training set. In general, the predictive distribution is Gaussian with mean and variance[2, 3].

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{t} \tag{1}$$

$$\sigma_{\hat{y}}^2(\boldsymbol{x}) = C(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x}) \tag{2}$$

where $\boldsymbol{k}(\boldsymbol{x}) = \{C(\boldsymbol{x}, \boldsymbol{x}^{(1)}), \dots, C(\boldsymbol{x}, \boldsymbol{x}^{(n)})\}^{\mathrm{T}}$, $\boldsymbol{K}$ is the covariance matrix for the training cases $K_{ij} = C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$, $\boldsymbol{t} = \{t^{(1)}, \dots, t^{(n)}\}^{\mathrm{T}}$, and $C(\boldsymbol{x}, y)$ is the covariance function.

There are many choices of covariance functions that may be reasonable. Formally, we are required to specify functions that will generate a non-negative definite covariance matrix for any set of points $(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(k)})$. From a modeling point of view we wish to specify covariance so that points with nearby inputs will give rise to similar predictions. A better covariance function is recommended in Ref. [3]:

$$C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = v_0 \exp\left[-\frac{1}{2}\sum_{l=1}^{d}\omega_l (x_l^{(i)} - x_l^{(j)})^2\right] + a_1 \sum_{l=1}^{d} x_l^{(i)} x_l^{(j)} + v_1 \delta(i, j) + a_0 \tag{3}$$

where $\boldsymbol{\theta} \overset{\text{def}}{=} \{\log v_0, \log v_1, \log \omega_1, \dots, \log \omega_d, \log a_0, \log a_1\}$ plays the role of hyperparameters, which closely corresponds to hyperparameters in ANN (in fact the weights have been integrated out exactly). We define the hyperparameters to be the log of the variables in Eq. (4) since these are positive scalar parameters.

Given a covariance function, the log likelihood $l = \log P(D \mid \boldsymbol{\theta})$ of the training data is calculated by

$$l = -\frac{1}{2}\log \det \boldsymbol{K} - \frac{1}{2}\boldsymbol{t}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{t} - \frac{n}{2}\log 2\pi \tag{4}$$

Usually, we have three methods to adapt the hy-

Received 2004-04-20.

Foundation item: The National High Technology Research and Development Program of China (863 Program) (No. 2002AA412010).

Biographies: Xiong Zhihua (1979—), male, graduate; Shao Huihe (corresponding author), male, professor, hhshao@sjtu.edu.cn.

perparameters, such as maximum likelihood[3], generalized cross-validation (GCV)[4] and Markov chain Monte Carlo (MCMC)[5] method. The partial derivative of the log likelihood of the training data l with respect to all the hyperparameters can be computed using matrix operations, and takes time $O(n^3)$. Next, we will set our model via iterative sparse approximation algorithm[6-9], whose computational time is $O(nd^2)$.

2 Sparse GP-Based Soft Sensor

In the GP framework, the parameters are functions and the GP priors specify a Gaussian distribution over a function space. The posterior process is entirely specified by all its finite dimensional marginals. Hence, let $\boldsymbol{f} = \{f(x_1), \ldots, f(x_M)\}$ be a set of function values such that $\boldsymbol{f}_D \subseteq \boldsymbol{f}$, where $\boldsymbol{f}_D$ is the set of $f(x_i) = f_i$ with x_i in the observed set of inputs, we compute the distribution using the data likelihood together with the prior $p_0(\boldsymbol{f})$ as

$$p_{\text{post}}(\boldsymbol{f}) = \frac{P(D \mid \boldsymbol{f}) p_0(\boldsymbol{f})}{\langle P(D \mid \boldsymbol{f}_D) \rangle_0} \tag{5}$$

where $\langle P(D \mid \boldsymbol{f}_D) \rangle_0$ is the average of the likelihood with respect to the prior GP (GP at time 0). The posterior distribution form can be used to express posterior expectations as typically high dimensional integrals. For prediction, one is especially interested in expectations of functions of the process in inputs, which are not contained in the training set. At first glance, one might assume that every prediction on a novel input would require the computation of a new integral. Even if we had good methods for approximate integration, this would make the predictions at new inputs a rather tedious task. Luckily, the following lemma shows that simple but important predictive quantities like the posterior covariance of the process at arbitrary inputs can be expressed as a combination of a finite set of parameters which depend on the training data only. For arbitrary likelihood we can show the following lemma.

Lemma[6,7] The result of the Bayesian update Eq. (5) using a GP prior with mean function $\langle f_x \rangle_0$ and kernel $K_0(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{T}})$ and data $D = \{(x_n, y_n) \mid n = 1, 2, \ldots, N\}$ is process with mean and kernel functions given by

$$\langle f_x \rangle_{\text{post}} = \langle f_x \rangle_0 + \sum_{i=1}^{N} K_0(\boldsymbol{x}, x_i) q(i)$$

$$K_{\text{post}}(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{T}}) = K_0(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{T}}) + \sum_{i,j=1}^{N} K_0(\boldsymbol{x}, x_i) R(ij) K_0(x_j, \boldsymbol{x}^{\mathrm{T}})$$

The parameters $q(i)$ and $R(ij)$ are given by

$$q(i) = \frac{1}{Z} \int \mathrm{d}\boldsymbol{f} p_0(\boldsymbol{f}) \frac{\partial P(D \mid \boldsymbol{f})}{\partial f(x_i)}$$

$$R(ij) = \frac{1}{Z} \int \mathrm{d}\boldsymbol{f} p_0(\boldsymbol{f}) \frac{\partial^2 P(D \mid \boldsymbol{f})}{\partial f(x_i) \partial f(x_j)} - q(i) q(j)$$

where $\boldsymbol{f} = \{f(x_1), \ldots, f(x_N)\}^{\mathrm{T}}$ and $Z = \int \mathrm{d}\boldsymbol{f} p_0(\boldsymbol{f}) P(D \mid \boldsymbol{f})$ is a normalized constant.

The parameters $q(i)$ and $R(ij)$ have to be computed only once during the training of the model, and are fixed when we make predictions. The parametric form of the posterior mean (assuming a zero mean for the prior) resembles the representations for the predictors in other kernel methods (such as SVM) that are obtained by minimizing certain cost functions.

Making an immediate use of this representation is usually not possible because the posterior process is in general not Gaussian and the integrals cannot be computed exactly. Hence, we need approximation to keep the inference tractable. One popular method is to approximate the posterior by a Gaussian process[10]. This may be formulated within a variational approach, where a certain dissimilarity measure between the true and the approximate distribution is minimized. The most popular choice is the Kullback-Leibler divergence between distributions defined as

$$\mathrm{KL}(p \mid q) = \int \mathrm{d}\boldsymbol{\theta} p(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \tag{6}$$

where $\boldsymbol{\theta}$ denotes the set of arguments of the densities. If $\hat{p}$ denotes the approximating Gaussian distribution, one usually tries to minimize $\mathrm{KL}(\hat{p} \| p_{\text{post}})$, with respect to $\hat{p}$ which in contrast to $\mathrm{KL}(p_{\text{post}} \| \hat{p})$ requires only the computation of expectation over tractable distributions.

In this paper, we use a different approach. To speed up the learning process in order to allow for the learning of large datasets, we aim at learning the data by a single sequential sweep through the examples. Let $\hat{p}_t$ denote the Gaussian approximation after processing t examples, we use Bayesian rule

$$p_{\text{post}}(\boldsymbol{f}) = \frac{P(y_{t+1} \mid \boldsymbol{f}) \hat{p}_t(\boldsymbol{f})}{\langle P(y_{t+1} \mid \boldsymbol{f}_D) \rangle_t} \tag{7}$$

to derive the updated posterior. Since p_{post} is no longer Gaussian, we use a variational technique in order to project it to the closest Gaussian process $\hat{p}_{t+1}$. Unlike the usual variational method, we then minimize the divergence $\mathrm{KL}(p_{\text{post}} \| \hat{p})$. This is possible, because in our online method the posterior (7) contains only the likelihood for a single example and the corresponding non-Gaussian integral is one-dimensional, which can be performed analytically for many relevant cases.

3 Simulation and Result

Some parameters are required to be computed or monitored for real-time use in thermal power systems. However, we tend to meet problems proposed in the beginning of this paper. Before setting the soft sensor, we should first choose the corresponding secondary variables. Limited to the space, we only give some important primary variables in the power plant as examples (see Tab. 1). One can do the similar job for other thermal parameters according to the on-spot demand.

Tab. 1 Secondary variables for primary variables concerned

Primary variables	O_2 content in flue gas	Feed water temperature	Vacuum of condenser	Flux of main steam	Load of ball mill
Secondary variables	Flux of main steam	Pressure of main steam	Pressure of main steam	Temperature of main steam	Coal rate
	Flux of feed water	Power of generator	Temperature of main steam	Temperature of reheat steam	Flux of hot air
	Charge of fuel	Temperature of steam extraction from #1 heater	Inlet temperature of circulating water	Pressure of main steam	Flux of recirculation air
	Temperature of exhaust fume	Pressure of steam extraction from #1 heater	Outlet temperature of circulating water	Temperature of feed water	Outlet temperature of ball mill
	Flux of input air	Flux of main steam	Temperature of condensation water	Power of generator	Differential pressure between inlet and outlet of ball mill
	Current of fan blower		Flux of main steam	Pressure of governing stage	Current of ball mill
	Flux of induced air		Power of generator	Vacuum	
	Current of induced fan				

Next, we will propose a soft sensor for predicting O_2 content in flue gas compared with zirconia analyzer as an example. One can implement the others according to Tab. 1. We obtain the samples from a peaking power unit, which first runs steadily at 125 MW, then drops down to 80 MW and finally runs at 80 MW steadily.

To illustrate the performance, we begin with a zero-mean Gaussian noise with variance 0.01. The results for the predictive output of the soft sensor and the actual output of the zirconia analyzer are shown in Fig. 1. The generalization mean square error (MSE) is 0.035 51 and the basis vector set is {5.478 5, 6.446, 5.994 9, 4.917 6, 7.454 3, 5.799 5, 7.366 3, 4.390 4, 7.731 4, 7.694 8, 7.775 7, 7.841 9, 7.629 9, 6.195 6, 3.941 2, 6.811 1, 7.548, 8.099 6, 7.571 4, 5.876 2, 4.817 8, 3.932 3, 4.695, 4.962 1, 6.870 9, 5.072 5, 5.802, 7.505 3, 6.021, 7.590 2, 6.720 4, 7.706 4, 4.985 2, 5.740 9, 5.994 2, 7.575 4, 5.950 7, 7.546 4, 5.003 4, 7.621 4, 7.578 4, 5.664 8}.

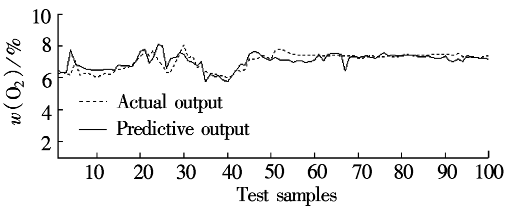

Fig. 1 Results for predictive output and actual output

From the above result, we can conclude that the proposed soft sensor is hopeful to replace the zirconia analyzer or to be used as a backup since the zirconia analyzer is expensive and short-life-span. And it is easy to realize the online optimization and economical running via the soft sensor.

4 Discussion

Recently, kernel methods (KM)[11] transcend the border between the two most important paradigms — GP and SVM, which is perhaps one of the major reasons why branches like GP and SVM have long been developed in isolation from each other. And GP is becoming popular in the community of kernel machines[12]. The main drawback of GP is heavy computational scaling, which is being alleviated by the introduction of sparse approximation as shown in this paper.

We discuss the thermal parameter soft sensor based on sparse Gaussian processes, which is of importance to industrial processes. And it can be implemented via online computation which can be employed in thermal power systems for optimal running and economical monitoring. Furthermore, we believe that the proposed method opens new possibilities in applying kernel methods to potential fields.

References

[1] McAvoy T J. Contemplative stance for chemical process [J]. *Automatica*, 1992, **28**(2): 441 – 442.

[2] MacKay D J C. Introduction to Gaussian processes[R]. Cambridge: Cambridge University, 1998.

[3] Williams C K I. Prediction with Gaussian processes: from the linear regression to linear prediction and beyond [A]. In: Jordan M I, ed. *Learning and Inference in Graphical Models*[C]. Kluwer Academic Press, 1998. 599 – 621.

[4] Wahba G. *Spline models for observational data* [M]. Philadelphia: SIAM, 1990.

[5] Anderieu C, de Freitas N, Doucet A, et al. An introduction to MCMC for machine learning[J]. *Machine Learning*, 2003, **50**(1): 5 – 43.

[6] Csato L. Gaussian processes — iterative sparse approximations[D]. Birmingham: Department of Computer Science and Applied Mathematics of Aston University, 2002.

[7] Csato L, Opper M. Sparse online Gaussian processes [J]. *Neural Computation*, 2002, **14**(3): 641 – 668.

[8] Smola A J, Scholkopf B. Sparse greedy matrix approximation for machine learning[A]. In: *Proceedings of the 17th International Conference on Machine Learning*[C]. San Francisco, 2000. 911 – 918.

[9] Williams C K I, Seeger M. Using the Nystrom method to speed up kernel machines[A]. In: Leen T K, Diettrich T G, Tresp V, eds. *Advances in Neural Information Processing Systems*[C]. Cambridge: MIT Press, 2001, **13**: 682 – 688.

[10] Seeger M. Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers [A]. In: Solla S A, Leen T K, Muller K R, eds. *Advances in Neural Information Processing Systems* [C]. Cambridge: MIT Press, 2000, **12**. 603 – 609.

[11] Scholkopf B, Smola A J. *Learning with kernels*[M]. Cambridge: MIT Press, 2002.

[12] Seeger M. Gaussian processes for machine learning[R]. Berkeley: University of California, 2004.

热力参数软仪表在电厂中的应用

熊志化[1] 朱 峰[2] 邵惠鹤[1]

([1] 上海交通大学自动化研究所, 上海 200030)

([2] 河南电力试验研究院, 郑州 450052)

摘要: 为了解决电厂中热力参数失效和优化运行的问题, 提出了一种基于稀疏高斯过程的软测量建模方法, 它基于 Bayes 在线学习算法, 通过构造序列的相关子样本来给出高斯过程的预测输出. 通过利用参数化和再生核 Hilbert 空间范数的投影技巧, 得到优化后的参数和后验过程的稀疏高斯逼近. 高斯过程的稀疏表达使得基于高斯过程的软仪表能够满足大规模数据集的实时应用需要, 所提出的热力参数软仪表对于电厂经济运行有着重要的意义.

关键词: 高斯过程; 软仪表; 稀疏逼近; 在线学习; 经济运行

中图分类号: TK39