International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 26, No. 8 (2012) 1250019 (16 pages)
© World Scientific Publishing Company
DOI: 10.1142/S021800141250019X

World Scientific
www.worldscientific.com

INCREMENTAL SPARSE PSEUDO-INPUT GAUSSIAN PROCESS REGRESSION

HEUNG-IL SUK*,‡, YUZHUO WANG*,§ and SEONG-WHAN LEE†,¶

*Department of Computer Science and Engineering, Korea University
Anam-dong, Seongbuk-ku, Seoul 136-713, Korea

†Department of Brain and Cognitive Engineering, Korea University
Anam-dong, Seongbuk-ku, Seoul, 136-713, Korea

‡hisuk@image.korea.ac.kr
§yzwang@image.korea.ac.kr
¶swlee@image.korea.ac.kr

Received 18 June 2011
Accepted 5 November 2012
Published 21 January 2013

In this paper, we devise a novel method that incrementally learns pseudo-data, which represent the whole training data set for Gaussian Process (GP) regression. The method involves sparse approximation of the GP by extending the work of Snelson and Ghahramani. We call the proposed method *Incremental Sparse Pseudo-input Gaussian Process* (ISPGP) regression. Unlike the Snelson and Ghahramani's work, the proposed ISPGP algorithm allows for training from either a huge amount of training data by scanning through it only once or an online incremental training data set. We also design a likelihood weighting scheme to incrementally determine pseudo-data while maintaining the representational power. Due to the nature of the incremental learning algorithm, the proposed ISPGP algorithm can theoretically work with infinite data to which the conventional GP or Sparse Pseudo-input Gaussian Process (SPGP) algorithm is not applicable. From our experimental results on the KIN40K data set, we can see that the proposed ISPGP algorithm is comparable to the conventional GP algorithm using the same number of training data. It also significantly reduces the computational cost and memory requirement in regression and is scalable to a large training data set without significant performance degradation. Although the proposed ISPGP algorithm performs slightly worse than Snelson and Ghahramani's SPGP algorithm, the level of performance degradation is acceptable.

Keywords: Gaussian process regression; incremental learning; pseudo-data.

1. Introduction

A Gaussian Process (GP) is a Bayesian approach for the task of regression or classification under the assumption that any finite linear combination of training samples will be normally distributed. The GP associates a random variable to each input

¶Corresponding author.

and for any set of inputs the associated random variables are jointly Gaussian. It thus involves random functions characterized by mean and kernel functions. The kernel provides the covariance: each pair of random variables at input points x and x′ has a covariance $K(\mathrm{x}, \mathrm{x}')$. Training in GP thus requires the manipulation of the covariance matrix with the whole training set, and the computational complexity is cubic due to the inversion of the covariance matrix. This unfavorable complexity prevents GP from being applied to a large number of data points.

Hence, the development of a method for scaling down the computational complexity of GP is of great importance in machine learning. Many research groups have devoted their efforts to this problem and proposed methods of sparse approximation to the full GP method largely by selecting a subset of training data.[2,5,13–16] Snelson and Ghahramani proposed an elegant method of approximating GP regression with virtual data which they called 'pseudo-inputs'.[15,16] Their approximation method aims to reduce the computational burden caused by matrix operations like inversion and diagonalization with a representative subset of training data found by continuous optimization. In this manner, their method reduces the computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(M^2N)$, where N is the total number of training data and $M < N$ for the number of representative samples.[15] Furthermore the memory requirement is $\mathcal{O}(MN)$, which is also much smaller than the $\mathcal{O}(N^2)$ of the standard GP method.

The sparse approximation techniques described in the literature, however, are mainly focused on the *batch* training of GP and there are few methods which can handle sequentially arriving training data. In this paper, we propose a new method called the *Incremental Sparse Pseudo-input Gaussian Processes* (ISPGP), which extends Snelson and Ghahramani's method[15] of representing data via sparse pseudo-inputs to an incremental learning method for the application to a large data set. The proposed ISPGP algorithm can theoretically accommodate an infinite number of data points and learn incrementally, thus improving the predictive performance with a small memory requirement and a low computational cost. It can also realize high prediction accuracy for a large data set to which the conventional GP and SPGP algorithms can not be applicable. Based on the nature of incremental learning methods, the proposed ISPGP algorithm is time-efficient in the sense that there is no need for full retraining as new data points arrive.

The main contribution of the paper is two-fold. First, we propose an incremental learning method for approximation in GP regression, i.e., ISPGP, based on Snelson and Ghahramani's SPGP algorithm.[15] The proposed ISPGP algorithm represents all the previous training data with a reduced amount of pseudo-data and then co-trains the pseudo-data with the newly arriving data. Second, we develop a likelihood weighting scheme to update the pseudo-data, which represents a large amount of real training data, by assigning different weights to the pseudo-data and the new data during optimization of the likelihood.

The rest of the paper is organized as follows. The previous work on Gaussian process is reviewed in Sec. 2. In Sec. 3, we briefly explain the general concept of SPGP

algorithm.[15] In Sec. 4, we describe the proposed ISPGP and experimental results is followed in Sec. 5. We wrap this paper in Sec. 6 with conclusions and directions for future work.

2. Previous Work

Because the GPs are nonparametric and prediction is based on all training data, it is inevitable that full retraining will be performed each time new data arrive. Much of the work done over the past two decades has attempted to address the computational problem, by developing efficient sparse approximations to the full GP. Refer to Quiñonero-Candela and Rasmussen's work[9] for a unifying view and analysis of sparse approximations of GP regression. The most obvious and simple technique to reduce the computational cost in training is to ignore the large number (N) of training data points and to make predictions based only on a subset of training data (M). We call this technique Subset of Data (SD). For highly redundant data sets, where extra data points provide very little or no extra information about the regression function, there is no point in wasting computation on a sophisticated sparse approximation that achieves little performance gain.

When using the SD technique, there are different schemes for selecting M subset data points from the N total training points. The simplest scheme is random selection. More sophisticated schemes can be used based on various information criteria which score how much one learns about the regression function by including a point in the subset. For instance, Lawrence *et al.* proposed an informative vector machine using differential entropy scores for subset selection.[5] Seeger *et al.* developed a method of selecting patterns for sparse greedy approximation of Bayesian GP regression (Ref. 12) and demonstrated that their method was as fast as random selection and outperformed it.[13] Srinivasan and Duraiswami used an information theoretic measures for sampling the data and proposed a Rényi entropy-based subset selection algorithm.[17]

As stated in the work of Snelson and Ghahramani,[15] the common problem in these methods is that they do not have a robust method of learning the optimal kernel hyperparameters, due to pattern selection. Selecting patterns from the training data set does not guarantee convergence to a smooth solution in regression, because of the nonsmooth fluctuations in the marginal likelihood and its gradient. Focusing on the speed of optimal active pattern selections weakens the power of GP regression, resulting in low quality solutions.

Recently, Snelson and Ghahramani proposed a new method of sparse representation called the Sparse Pseudo-input Gaussian Processes (SPGP)[15] to circumvent this problem. They parameterized the covariance with M pseudo-data points, which are not constrained to be selected from the real training data points. They assume that $M < N$, where N is the number of training data points, and hence they obtain a sparse regression method which has $O(M^2N)$ computational cost in training and $O(M^2)$ in prediction per test case. The pseudo-data or active set points and

hyperparameters can be found in a single smooth joint optimization. The SPGP can be viewed as a Bayesian regression model with noises dependent on inputs. Experimental results in Ref. 15 showed that the SPGP method could match the full GP performance with a small number of pseudo-inputs, i.e., a very sparse solution, and it significantly outperformed other sparse approaches.

Although there are lots of subset selection methods for GP regression, these methods cannot be applied when the data set is large, i.e., it cannot be entirely stored in memory, or the training data are presented sequentially. In order to address this problem, Csato and Opper proposed a sparse GP method which could process data sequentially and was particularly designed to approximate non-Gaussian likelihoods.[3] The method is also based on a subset of data points, and so it is naturally related to the ones we have discussed above. Their method processes one data point at a time and consists of a Gaussian projection to deal with the non-Gaussianity of the likelihood, and a projection onto a sparse model to limit computation. Ranganathan *et al.* proposed a new GP inference algorithm, called the Online Sparse Matrix Gaussian Processes (OSMGP), and demonstrated its advantages with vision applications.[10] The OSMGP method is based on the observation that for kernels with local support, a Gram matrix is typically sparse. Maintaining and updating the sparse Cholesky factor of a Gram matrix can be efficiently achieved using Givens rotations. This leads to an exact, online algorithm for which the updating time scales linearly with the size of the Gram matrix. Nguyen-Tuong *et al.* presented a local GP regression model for real-time online model learning and control.[8] They partitioned the training data into local regions and an individual GP is trained for each partition. The prediction at an input point is performed by nearby local models. Zhao *et al.* also utilized the locality of the sample points and proposed a local mixture GP experts systems in which different local GP experts in terms of both temporal and spatial information dominate a mapping behavior with the specific covariance function adapting to a local region (Ref. 19).

3. Sparse Gaussian Process Using Pseudo-Inputs (SPGP)

In this section, we provide a concise summary of the Sparse Gaussian Process using Pseudo-inputs (SPGP) for regression. We have to note that most of the notations and explanations are based on Snelson and Ghahramani's work[15] and for more detailed reviews on the conventional GP the reader should see Refs. 1, 4, 7 and 11. The major notations used in the following sections are described in Table 1.

The mean and variance of the predictive distribution in the conventional GP can be considered as a function of a new input, hyperparameters, and N training input and target pairs, $\mathbf{X}$ and $\mathbf{y}$. In order to derive a sparse approximation, Snelson and Ghahramani proposed a method in which the predictive mean and variance are parameterized by a pseudo-data set, which consists of pseudo-targets $\bar{\mathbf{f}} = \{\bar{f}_m\}_{m=1}^{M}$ and pseudo-inputs $\bar{\mathbf{X}} = \{\bar{\mathrm{x}}_m\}_{m=1}^{M}$ of size M, instead of all N real training data points,[15] where $M < N$. The pseudo-targets $\bar{\mathbf{f}}$ are not constrained to be a subset of

Table 1. Description of the notations used in the paper.

Notation	Description	Notation	Description
D	Data dimension	$\mathbb{D}$	Data set
N	Number of input data	M	Number of pseudo-data
y	Training target	x	Training input
y_*	Predicted target	x_*	Test input
$\bar{\mathbf{f}}$	Pseudo-target	$\bar{\mathrm{x}}$	Pseudo-input
f	Latent function	$K(\cdot,\cdot)$	Covariance kernel function
$[\mathbf{K}_N]_{nn'}$	$K(\mathrm{x}_n, \mathrm{x}_{n'})$	$[\mathbf{K}_M]_{mm'}$	$K(\bar{\mathrm{x}}_m, \bar{\mathrm{x}}_{m'})$
$[\mathbf{K}_{NM}]_{nm}$	$K(\mathrm{x}_n, \bar{\mathrm{x}}_m)$	$[\mathbf{k}_\mathrm{x}]_n$	$K(\mathrm{x}_n, \mathrm{x})$
$[\mathbf{k}_\mathrm{x}]_m$	$K(\bar{\mathrm{x}}_m, \mathrm{x})$		

the real observations in the training data set, but are distributed in a very similar manner to the real data, and this is why Snelson and Gharamani denoted the pseudo-targets by $\bar{\mathbf{f}}$ instead of $\bar{\mathbf{y}}$.[15] Thus we do not need to consider the noise variance for those ones. Assuming the pseudo-data are known, the likelihood of a single data point is computed as follows

$$p(y|\mathrm{x}, \bar{\mathbf{X}}, \bar{\mathbf{f}}) = \mathbb{N}(y|\mathbf{k}_{\mathbf{X}}^T \mathbf{K}_M^{-1} \bar{\mathbf{f}}, K_{\mathrm{xx}} - \mathbf{k}_\mathrm{x}^T \mathbf{K}_M^{-1} \mathbf{k}_\mathrm{x} + \sigma^2), \tag{1}$$

where $[\mathbf{k}_M]_{mm'} = K(\bar{\mathrm{x}}_m, \bar{\mathrm{x}}_{m'})$, $[\mathbf{k}_\mathrm{x}]_m = K(\bar{\mathrm{x}}_m, \mathrm{x})$, for $m = 1, \ldots, M$. Since the target values are generated *i.i.d.*, the likelihood for the full training data set is computed as follows

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{f}}) &= \prod_{n=1}^{N} p(y_n|\mathrm{x}_n, \bar{\mathbf{X}}, \bar{\mathbf{f}}) \\ &= \mathbb{N}(\mathbf{y}|\mathbf{K}_{NM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}, \mathbf{\Lambda} + \sigma^2\mathbf{I}), \end{aligned} \tag{2}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}_n = K_{nn} - \mathbf{k}_n^T \mathbf{K}_M^{-1} \mathbf{k}_n$, and $[\mathbf{K}_{NM}]_{nm} = K(\mathrm{x}_n, \bar{\mathrm{x}}_m)$.

The optimal pseudo-inputs which represent the real data are obtained by maximizing the likelihood given in Eq. (2) with respect to $\bar{\mathbf{X}}$ and $\bar{\mathbf{f}}$. Here it is assumed that the pseudo-inputs have a similar distribution to the real training data set, so it is possible to place a Gaussian prior over the pseudo-targets like $p(\bar{\mathbf{f}}|\bar{\mathbf{X}}) = \mathbb{N}(\bar{\mathbf{f}}|\mathbf{0}, \mathbf{K}_M)$ and to integrate those pseudo-targets in Eq. (2). Using the Bayes rule, we can compute the posterior distribution over pseudo-targets $\bar{\mathbf{f}}$ as follows

$$p(\bar{\mathbf{f}}|\mathbb{D}, \bar{\mathbf{X}}) = \mathbb{N}(\bar{\mathbf{f}}|\mathbf{K}_M \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_M \mathbf{Q}_M^{-1} \mathbf{K}_M), \tag{3}$$

where $\mathbf{Q}_M = \mathbf{K}_M + \mathbf{K}_{MN}(\mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{NM}$. The predictive distribution over a new input x_* is then obtained by integrating the likelihood in Eq. (1) with the posterior in Eq. (3)

$$\begin{aligned} p(y_*|\mathrm{x}_*, \mathbb{D}, \bar{\mathbf{X}}) &= \int p(y_*|\mathrm{x}_*, \bar{\mathbf{X}}, \bar{\mathbf{f}}) p(\bar{\mathbf{f}}|\mathbb{D}, \bar{\mathbf{X}}) d\bar{\mathbf{f}} \\ &= \mathbb{N}(y_*|\mu_*, \sigma_*^2), \end{aligned} \tag{4}$$

where $\mu_* = \mathbf{k}_*^{\mathrm{T}} \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$ and $\sigma_*^2 = K_{**} - \mathbf{k}_*^{\mathrm{T}}(\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1})\mathbf{k}_* + \sigma^2$.

The computational cost in training is $O(M^2N)$ and is dominated by the matrix multiplication for the calculation of $\mathbf{Q}_M$, where inversion of the diagonal matrix $\mathbf{\Lambda} + \sigma^2\mathbf{I}$ is trivial. By computing the necessary values in advance, the prediction cost for the mean and the variance per test case is $O(M)$ and $O(M^2)$, respectively.

In order to find the pseudo-inputs from the training data set, we maximize the marginal likelihood, which can be computed by integrating out the pseudo-targets from the full data likelihood, with respect to both pseudo-inputs $\bar{\mathbf{X}}$ and hyper-parameters $\mathbf{\Theta} = \{\theta, \sigma^2\}$

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \mathbf{\Theta}) &= \int p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{f}}) p(\bar{\mathbf{f}}|\bar{\mathbf{X}}) \\ &= \mathbb{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \mathbf{\Lambda} + \sigma^2\mathbf{I}) d\bar{\mathbf{f}}. \end{aligned} \tag{5}$$

In SPGP, we can best utilize the limited resources of M pseudo-data points by allowing them to vary flexibly and continuously away from the real training data points. Thus the pseudo-data set, pseudo-inputs and pseudo-targets, can provide a better representation of all the training data and a better solution than other subset-based methods. Contrary to other subset selection techniques in which subset selection and gradient-based hyperparameter learning are interleaved, the hyper-parameters and pseudo-inputs in SPGP can be optimized smoothly.

4. Proposed Incremental Learning Algorithm in SPGP

When the total amount of training data is too large, it can become computationally infeasible to apply standard GP or SPGP regression methods to the data set. In this section, we propose a new method for handling a huge amount of training data in regression by extending Snelson and Ghahramani's method.[15] Instead of considering the whole training data set all at once, we can divide the data set into multiple small-sized sets, which can be either a single data point or any portion of the whole training data set. Then we can consider each of the small-sized sets arrives sequentially, i.e., one data set after another. Hereafter, we call the newly arrived data *incremental data*.

In our framework, we first apply the SPGP algorithm[15] to our initial data, which can be the first training data set given to us in an online system or the first small-sized data set among a huge training data set in a batch system. As the next incremental data arrive, we combine them with the pseudo-data derived from the previous training round and then apply the proposed novel incremental learning algorithm, which we call the ISPGP.

Suppose that in a certain training round we have L pseudo-data points obtained from the previous training round and we want to set M new pseudo-data points to be optimized in this round with new data points. We can obtain the single-point likelihood of new data points $\{\mathrm{x}_*, y_*\}$ which are parameterized by the M new pseudo-data points as follows

$$p(y_*|\mathrm{x}_*, \bar{\mathbf{X}}, \bar{\mathbf{f}}) = \mathbb{N}(y_*|\mathbf{k}_{\bar{\mathrm{x}}*}^T\mathbf{K}_M^{-1}\bar{\mathbf{f}}, K_{**} - \mathbf{k}_{\bar{\mathrm{x}}*}^T\mathbf{K}_M^{-1}\mathbf{k}_{\bar{\mathrm{x}}*} + \sigma^2), \tag{6}$$

where $[\mathbf{K}_M]_{mm'} = K(\bar{\mathrm{x}}_m, \bar{\mathrm{x}}_{m'})$ and $[\mathbf{k}_{\bar{\mathrm{x}}^*}]_m = K(\bar{\mathrm{x}}_m, \mathrm{x}_*)$ for $m = 1, 2, \ldots, M$. The mean of the pseudo-target is used for the prediction of y_*. We can also compute the single-point likelihood of the old pseudo-data $\{\mathrm{x}_o, y_o\}$ parameterized by the M new pseudo-data points:

$$p(y_o|\mathrm{x}_o, \bar{\mathbf{X}}, \bar{\mathbf{f}}) = \mathbb{N}(y_o|\mathbf{k}_{\bar{\mathrm{x}}o}^T \mathbf{K}_M^{-1}\bar{\mathbf{f}}, K_{oo} - \mathbf{k}_{\bar{\mathrm{x}}o}^T \mathbf{K}_M^{-1}\mathbf{k}_{\bar{\mathrm{x}}o} + \sigma^2), \tag{7}$$

where $[\mathbf{K}_M]_{mm'} = K(\bar{\mathrm{x}}_m, \bar{\mathrm{x}}_{m'})$ and $[\mathbf{k}_{\bar{\mathrm{x}}o}]_m = K(\bar{\mathrm{x}}_m, \mathrm{x}_o)$ for $m = 1, 2, \ldots, M$. In Eqs. (6) and (7), $\bar{\mathbf{X}}$ and $\bar{\mathrm{x}}$ denote the new pseudo-inputs and $\bar{\mathbf{f}}$ represents the new pseudo-targets to be optimized in a training round. Since the incremental data and the old pseudo-data are generated *i.i.d.* we can deduce a formulation of the likelihood for the full data set $\{\mathbf{y}, \mathbf{X}\}$, i.e., L old pseudo-data points and Q incremental data points:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{f}}) &= \prod_{q=1}^{Q} p(y_q|\mathrm{x}_q, \bar{\mathbf{X}}, \bar{\mathbf{f}}) \prod_{l=1}^{L} p(y_l|\mathrm{x}_l, \bar{\mathbf{X}}, \bar{\mathbf{f}}) \\ &= \prod_{q=1}^{Q} \mathbb{N}(y_q|\mathbf{k}_{\bar{\mathrm{x}}q}^T \mathbf{K}_M^{-1}\bar{\mathbf{f}}, K_{qq} - \mathbf{k}_{\bar{\mathrm{x}}q}^T \mathbf{K}_M^{-1}\mathbf{k}_{\bar{\mathrm{x}}q} + \sigma^2) \\ &\quad \times \prod_{l=1}^{L} \mathbb{N}(y_l|\mathbf{k}_{\bar{\mathrm{x}}l}^T \mathbf{K}_M^{-1}\bar{\mathbf{f}}, K_{ll} - \mathbf{k}_{\bar{\mathrm{x}}l}^T \mathbf{K}_M^{-1}\mathbf{k}_{\bar{\mathrm{x}}l} + \sigma^2) \\ &= \mathbb{N}(\mathbf{y}_Q|\mathbf{K}_{QM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}, \mathbf{\Lambda}_Q + \sigma^2\mathbf{I})\mathbb{N}(\mathbf{y}_L|\mathbf{K}_{LM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}, \Lambda_{(L)} + \sigma^2\mathbf{I}), \end{aligned}$$

where $\mathbf{\Lambda}_Q = \mathrm{diag}(\boldsymbol{\lambda}_q)$, $\lambda_q = K_{qq} - \mathbf{k}_{\bar{\mathrm{x}}q}^T \mathbf{K}_M^{-1}\mathbf{k}_{\bar{\mathrm{x}}q}$, $[\mathbf{k}_{QM}]_{qm} = K(\mathrm{x}_q, \bar{\mathrm{x}}_m)$, $m \in \{1, 2, \ldots, M\}$, and $\mathbf{\Lambda}_L = \mathrm{diag}(\boldsymbol{\lambda}_l), \lambda_l = K_{ll} - \mathbf{k}_{\bar{\mathrm{x}}l}^T \mathbf{K}_M^{-1}\mathbf{k}_{\bar{\mathrm{x}}l}, [\mathbf{k}_{LM}]_{lm} = K(\mathrm{x}_l, \bar{\mathrm{x}}_m), l \in \{1, 2, \ldots, L\}$.

We can further integrate the new pseudo-targets $\bar{\mathbf{f}}$ by applying a Gaussian prior on the new pseudo-targets

$$p(\bar{\mathbf{f}}|\bar{\mathbf{X}}) = \mathbb{N}(\bar{\mathbf{f}}|\mathbf{0}, \mathbf{K}_M). \tag{8}$$

Here we should note that in some sense, the pseudo-data may not exist in the real universe. The prior of Eq. (8) constraints, however, the pseudo-data to be distributed in a very similar manner to the real ones. So it is reasonable to use the pseudo-data for training and prediction. The marginal likelihood of the combined data $\{\mathbf{y}, \mathbf{X}\}$, i.e., the old pseudo-data points and the incremental data points, is computed as follows

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \mathbf{\Theta}) &= \int p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{f}}) p(\bar{\mathbf{f}}|\bar{\mathbf{X}}) d\bar{\mathbf{f}} \\ &= \int \mathbb{N}(\mathbf{y}_Q|\mu_Q, \Sigma_Q)\mathbb{N}(\mathbf{y}_L|\mu_L, \Sigma_L) p(\bar{\mathbf{f}}|\bar{\mathbf{X}}) d\bar{\mathbf{f}} \\ &= \mathbb{N}(\mathbf{y}_Q|\mathbf{0}, \Sigma_{QQ})\mathbb{N}(\mathbf{y}_L|\mathbf{0}, \Sigma_{LL}), \end{aligned} \tag{9}$$

where $\mu_Q = \mathbf{K}_{QM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}$, $\Sigma_Q = \mathbf{\Lambda}_Q + \sigma^2\mathbf{I}$, $\mu_L = \mathbf{K}_{LM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}$, $\Sigma_L = \mathbf{\Lambda}_L + \sigma^2\mathbf{I}$, $\Sigma_{QQ} = \mathbf{K}_{QM}\mathbf{K}_M^{-1}\mathbf{K}_{MQ} + \mathbf{\Lambda}_Q + \sigma^2\mathbf{I}$, and $\Sigma_{LL} = \mathbf{K}_{LM}\mathbf{K}_M^{-1}\mathbf{K}_{ML} + \mathbf{\Lambda}_L + \sigma^2\mathbf{I}$. We can see that

the mathematical form of the marginal likelihood is exactly the same as that of the real training data in Snelson's SPGP model except that a Gaussian distribution over N real data points is substituted with the multiplication of two independent Gaussian distributions over L old pseudo-data points and Q incremental data points.

Since the old pseudo-data set is a sparse representation of all the previous training data points, it is relatively more informative than a new small incremental data set. Based on this fact, we apply different weights for the likelihoods of the pseudo-data and the incremental data. We scale up the contribution of the pseudo-data set in incremental learning by taking its likelihood to the power of w. So the weighted marginal likelihood of the combined data is formulated as

$$p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \boldsymbol{\Theta}) = \mathbb{N}(\mathbf{y}_Q|\mathbf{0}, \mathbf{K}_{QM}\mathbf{K}_M^{-1}\mathbf{K}_{MQ} + \boldsymbol{\Lambda}_Q + \sigma^2\mathbf{I}) \left[\mathbb{N}(\mathbf{y}_L|\mathbf{0}, \mathbf{K}_{LM}\mathbf{K}_M^{-1}\mathbf{K}_{ML} + \boldsymbol{\Lambda}_L + \sigma^2\mathbf{I})\right]^w. \tag{10}$$

Equation (10) can be represented in a simplified form as follows

$$\begin{aligned} p(\bar{\mathbf{X}}, \boldsymbol{\Theta}) &= p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \boldsymbol{\Theta}) \\ &= p_* p_o^w, \end{aligned} \tag{11}$$

where p_* and p_o denote the likelihood of the incremental data and the old pseudo-inputs, respectively. In our experiments, we apply two weighting schemes for the likelihood of the old pseudo-data by setting $w = 1$, which means equally weighting both likelihoods, and $w = \frac{N_C}{N_I}$, which is the ratio of the number of training data points N_C used to derive the current pseudo-data points and the number of incremental data points N_I.

It is parameterized by M new pseudo-inputs $\bar{\mathbf{X}}$ and hyperparameters $\boldsymbol{\Theta}$. In order to find the pseudo-inputs from the incremental data points and the old pseudo-inputs, we maximize the log marginal likelihood of Eq. (10) with respect to $\bar{\mathbf{X}}$ and $\boldsymbol{\Theta}$. We perform the optimization by taking the derivative of the log marginal likelihood with respect to a variable x as given by

$$\frac{\partial \log(p)}{\partial \bar{\mathbf{X}}} = \frac{\partial \log(p_*)}{\partial \bar{\mathbf{X}}} + w\frac{\partial \log(p_o)}{\partial \bar{\mathbf{X}}}. \tag{12}$$

Since p_* and p_o have the same form as the marginal likelihood in SPGP, their derivatives are also in the same form. This method can be applied to any covariance function that is differentiable with respect to the input points. Once we obtain M optimal new pseudo-inputs and hyperparameters by gradient ascent, the predictions for new inputs can be performed by Eq. (4).

A complete algorithm for learning in the proposed ISPGP is given in Fig. 1. The role of pseudo-data in each ISPGP training round can be two-fold. It can be used for prediction of new inputs and for further incremental learning without any increase in the computational burden.

Algorithm 1: Learning in the Proposed ISPGP

Input:

Incremental learning data $\{\mathbf{X}, \mathbf{y}\}$ and current pseudo-inputs $\bar{\mathbf{X}}_{old}$

Output:

Pseudo-inputs $\bar{\mathbf{X}}_{new} = \{\bar{\mathbf{x}}_m\}_{m=1}^{M}$ and hyperparameters $\boldsymbol{\Theta}_{new}$

Learning

if $\bar{\mathbf{X}}_{old}$ *is empty* **then**
  Initialize hyperparameters $\boldsymbol{\Theta}_{init}$
  $\bar{\mathbf{X}}_{init} = M$ points randomly selected from $\mathbf{X}$
  $[\bar{\mathbf{X}}_{new}, \boldsymbol{\Theta}_{new}]$ = maximize the marginal likelihood of Eq. (5)
end
else
  $\boldsymbol{\Theta}_{init} = \boldsymbol{\Theta}_{old}$
  $\bar{\mathbf{X}}_{init} = M$ points randomly selected from $\{\mathbf{X} \cup \bar{\mathbf{X}}_{old}\}$
  if *a weighting scheme* **then**
    $[\bar{\mathbf{X}}_{new}, \boldsymbol{\Theta}_{new}]$ = maximize the marginal likelihood of Eq. (9)
  end
  else
    $[\bar{\mathbf{X}}_{new}, \boldsymbol{\Theta}_{new}]$ = maximize the marginal likelihood of Eq. (10)
  end
  $\bar{\mathbf{X}}_{old} = \bar{\mathbf{X}}_{new}$
  $\boldsymbol{\Theta}_{old} = \boldsymbol{\Theta}_{new}$
end

Fig. 1. A pseudo learning algorithm for the proposed ISPGP method.

5. Experimental Results and Analysis

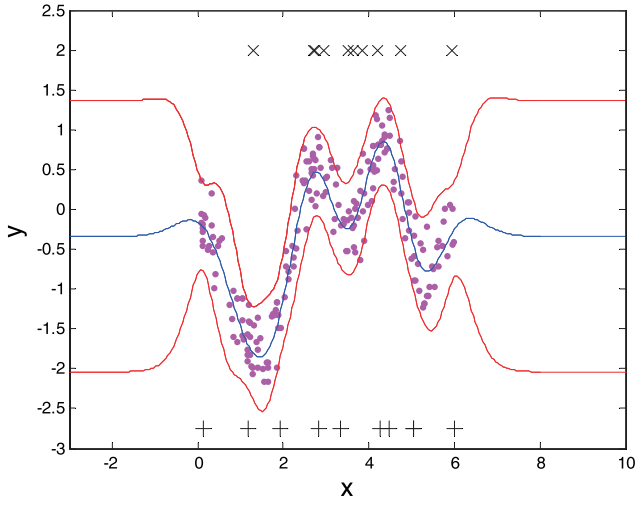
In this section, we first simulate and compare the predictive distribution of SPGP and the proposed ISPGP algorithms with a one-dimensional data set, and then evaluate the performance of the proposed ISPGP algorithm on a real data set comparing with that of the conventional GP and SPGP methods.

We built our system with MATLAB 2009b on a 2.8 GHz PC. We use the squared exponential covariance function[11] with Automatic Relevance Determination (ARD).[6] For maximization of the marginal likelihood of Eqs. (5), (9), and (10) in Algorithm 1, the optimization code by Rasmussen[a,11] was used for our experiments.

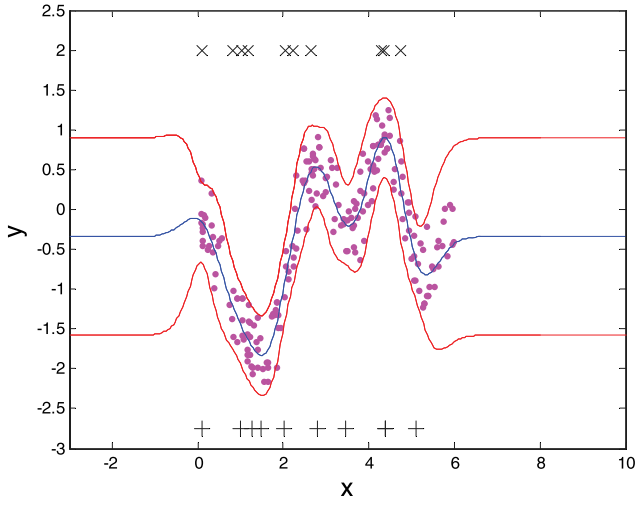
5.1. *Simulation on a 1D example*

Figure 2 presents simulations of the SPGP and the proposed ISPGP algorithms. We used the same number of pseudo-data points in both algorithms and randomly placed the initial pseudo-inputs fixing the hyperparameters to their true values for this demonstration. In the figure, the black '+'s (bottom) represent the updated values of the pseudo-inputs. Figure 2(a) shows the predicted values using 10 pseudo-data points after two rounds of the incremental learning with 200 data points, 100 data points per round. Figure 2(b) represents the batch training of SPGP for 200 data points using 10 pseudo-data points. We can see that prediction of the proposed ISPGP is very similar to that of SPGP.

[a] Available at http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize.



(a) A result obtained by the proposed ISPGP algorithm after two rounds, each of which we used 100 data points.



(b) A result obtained by SPGP with 200 data points.

Fig. 2. Predictive distributions (mean and two standard deviation lines) with 10 pseudo-inputs in 1D.

In order for the quantitative comparison, we also presented a sum of the squared difference and a statistical significance on the mean values and the standard deviations of the methods in Table 2. From the table, we can see that the difference between two methods in terms of the mean and the standard deviation are not statistically significant. That is, we can reject the hypothesis that the methods are different from each other beyond the 99% confidence level.

Table 2. A quantitative comparison of the results between SPGP and ISPGP. For the statistical significance test, the paired t-test is considered.

	Mean	Standard Deviation
Sum of the squared difference	0.9173	2.6144
p-Value	4.06e-18	1.89e-24

5.2. *Evaluation on a real data set*

In this experiment, we use the KIN40K[b] data set, which consists of 40,000 points, aiming to predict the distance of a robot arm head from a target based on an eight-dimensional input feature consisting of joint positions and twist angles. The data was generated with maximum nonlinearity and little noise, resulting in a very difficult regression task.[18]

5.2.1. *Experimental settings*

We randomly selected 10,000 data for training and 500 data for test from the KIN40K data set. For incremental learning, we first divided the training data into groups, 200 data points per group. We empirically set the number of pseudo-data points in both SPGP and ISPGP, 20% of the total N data points. Although we may need a larger number of pseudo-data in training from a larger number of training data, the greater the amount of training data, the greater the redundancy contained therein. Based on this fact, instead of a linear increase of the number of pseudo-data points M of ISPGP according to the number of training data, we set it in a certain round as $M = \sqrt{\frac{N}{1000}} \times 200$, where N is the total number of data points trained up to the end of the round.

In the proposed ISPGP, the first round is the same as Snelson and Ghahramani's SPGP batch training.[15] In the following rounds, the pseudo-inputs are initialized by randomly selected points among the combination of the incremental data and the current pseudo-inputs derived from the previous round, and the hyperparameters are initialized by those derived from the previous round.

We evaluate the predictive accuracy with the root-mean-square error (RMSE) metric, which is defined as follows

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_i - \mu_i)^2}, \tag{13}$$

where T is the number of test points, and y_i and μ_i are the target and the predicted mean of the ith test point, respectively.

5.2.2. *Likelihood weighting scheme in learning*

As explained in Sec. 4, we apply a likelihood weighting scheme with different weights for the likelihoods of the pseudo-data and the incremental data. The feasibility of the

[b]Available at http://ida.first.fraunhofer.de/∼anton/data.html.

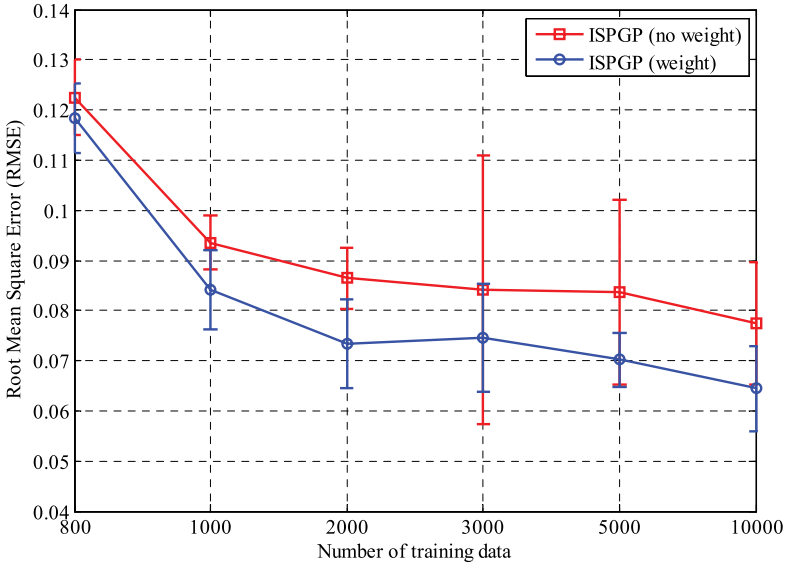


Fig. 3. Means (markers) and variances (vertical bars) of the RMSE of the proposed ISPGP with and without a likelihood weighting factor from our 10 repeated experiments.

likelihood weighting scheme of the pseudo-data in the proposed ISPGP algorithm is evaluated by the comparison of the performances of two weighting factors $w = 1$ and $w = \sqrt{\frac{N_C}{N_I}}$ where N_C and N_I, respectively, denote the number of training data points used to derive the current pseudo-data and the number of incremental data points. The latter weighting factor reflects the ratio of the information between the current pseudo-data and the incremental data.

In Fig. 3, we present the means and variances of the RMSE of the proposed ISPGP with and without a likelihood weighting factor. The results were obtained from 10 repeated experiments, in each of which we randomly selected the training and test data. The performances are identical for the first round because no previous pseudo-data are used. From the second round on, the weighting scheme obviously outperforms the other one. we should note that the weighting scheme used in this paper is based on a simple intuition, so it may not be optimal. However, the experimental results effectively proved that the weighting scheme is effective in improving the regression accuracy for the KIN40K data set. Hence we use the likelihood weighted ISPGP method in the following experiments.

5.2.3. *Performance comparison*

We compare the performance of GP, SPGP, and ISPGP. In order to learn the parameters in each method a standard conjugate gradient optimization scheme is adopted with 100 iterations. The resulting mean and variance of the RMSE

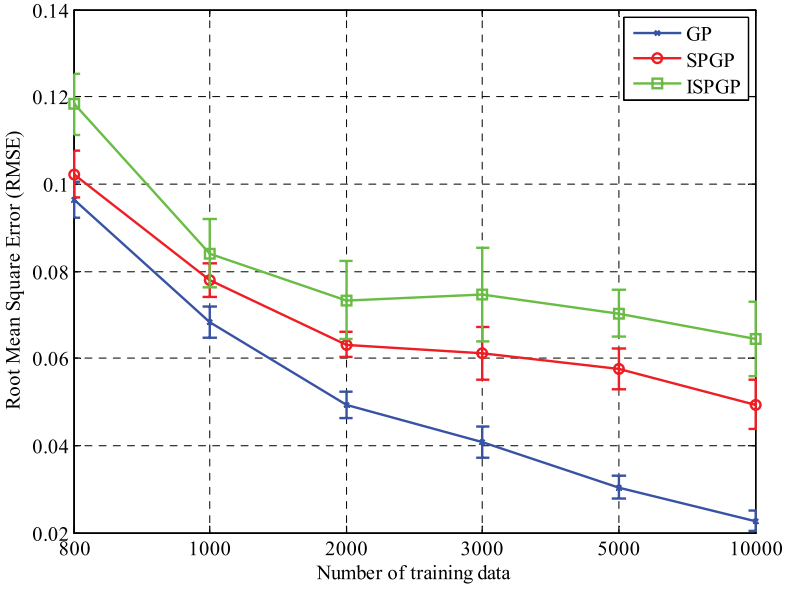


Fig. 4. Comparison of the RMSE of GP, SPGP, and ISPGP for various data sizes.

according to the number of training data are shown in Fig. 4. The results were obtained from 10 repetition, in each of which the training data was chosen in random. While the performance efficiency is in the order of GP, SPGP, and ISPGP, the advantage and usefulness of the proposed method is clear in the ensuing experiment with massive training data.

Due to the favorable memory requirement of ISPGP, i.e., $\mathcal{O}(\alpha N)$, where $\alpha = \frac{N}{R}$ and R denotes the number of rounds in ISPGP, the proposed ISPGP algorithm can process massive amounts of training data even for the case of $N \to \infty$, as long as the number of batches R is large enough.

5.3. *Training/prediction time and memory requirement*

The regression performance depends heavily on the size of the training set. Therefore, a good regression method should be able to handle a large amount of training data. We compared the computational complexity and the memory requirement of the GP, SPGP and the proposed ISPGP algorithms. Table 3 summarizes the time

Table 3. Training and prediction complexity and the memory requirement for a data set size of N in the GP, SPGP and proposed ISPGP algorithms. Here, $\alpha = \frac{N}{R}$ and R is the number of training rounds of ISPGP.

	Training	Mean Prediction	Variance Prediction	Memory Requirement
GP	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
SPGP	$\mathcal{O}(M^2N)$	$\mathcal{O}(M)$	$\mathcal{O}(M^2)$	$\mathcal{O}(MN)$
ISPGP	$\mathcal{O}(M^2N)$	$\mathcal{O}(M)$	$\mathcal{O}(M^2)$	$\mathcal{O}(\alpha M)$

complexities in terms of training, prediction and memory requirement for a data set size of N in the GP, SPGP and ISPGP algorithms.

The conventional GP method has a computational problem for a large training data set, due to its unfavorable complexity of $\mathcal{O}(N^3)$. It needs much more time than SPGP and the proposed ISPGP algorithm with the same amount of training data, indicating that the conventional GP method is not scalable to a large data set. The SPGP and proposed ISPGP algorithms reduce the computation complexity to $\mathcal{O}(M^2N)$. Since the number of pseudo-data points M is much smaller than N, the computation complexity remains low.

In terms of prediction, the time complexity in SPGP and ISPGP is identical to $\mathcal{O}(M^2)$ under the assumption that we use the same number of pseudo-data points while the time complexity of prediction in GP is $\mathcal{O}(N^2)$.

The memory requirement of GP is $\mathcal{O}(N^2)$, since the kernel matrix of all data points is applied, which is not scalable to a large data set. While the memory requirement of SPGP is $\mathcal{O}(MN)$ the proposed algorithm requires a memory size equal to $\mathcal{O}(\alpha M)$ size of memory, where $\alpha = \frac{N}{R}$ and R denotes the number of rounds in ISPGP.

6. Conclusion and Further Research

In this paper, we propose a new incremental learning method, ISPGP, which combines the idea of a sparse pseudo-input representation of GP[15] with an incremental algorithm allowing for learning from a huge training data set or an online incremental data set. The contribution of this paper is two-fold. First, we apply a pseudo-data-based representation to incremental learning. The proposed ISPGP algorithm first represents the previous training data using a greatly reduced amount of pseudo-data and then co-trains the pseudo-data with incremental data. Second, we also proposed to apply a likelihood weighting scheme for the pseudo-data, which represents a large amount of real training data, thus further contributing to optimization of the likelihood.

Experimental results on the KIN40K data set demonstrate that the prediction performance of the proposed ISPGP algorithm is comparable to the conventional GP method and Snelson and Ghahramani's SPGP algorithm using the same number of training data points. Moreover, ISPGP has learned from a large size data set for which the conventional GP and SPGP algorithms are not applicable. The proposed ISPGP algorithm also significantly reduces the computational and memory costs of GP regression, and is scalable to a large training data set.

The ISPGP algorithm uses only the mean information of the pseudo-target for incremental learning. Our future work will include developing a statistical weighting scheme for the pseudo-data in incremental learning.

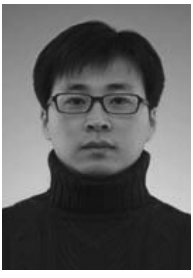
Acknowledgment

This work was supported by the National Research Foundation (NRF) Grant funded by the Ministry of Education, Science, and Technology, under Grant 2012-005741.

The authors would like to thank Dr. Edward Snelson, with whom we had fruitful discussion.

References

1. C. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
2. L. Csató, Gaussian processes — iterative sparse approximations, Ph.D. thesis, Aston University (2002).
3. L. Csato and M. Opper, Sparse online Gaussian processes, *Neural Comput.* **14** (2002), pp. 641−668.
4. M. Gibbs, *Bayesian Gaussian Processes for Regression and Classification*, Ph.D. thesis, Cambridge University (1997).
5. N. Lawrence, M. Seeger and R. Herbrich, Fast sparse Gaussian process methods: The informative vector machine, in *Advances in Neural Information Processing Systems*, Vol. 15 (MIT Press, 2003), pp. 609−616.
6. D. MacKay, Bayesian interpolation, *Neural Comput.* **4** (1992), pp. 415−417.
7. D. MacKay, *Introduction to Gaussian Processes, Neural Networks and Machine Learning*, NATO ASI Series (Kluwer Academic Press, 1998).
8. D. Nguyen-Tuong, M. Seeger and J. Peters, Local Gaussian process regression for real time online model learning and control, *Advances in Neural Information Processing Systems*, Vol. 22 ((MIT Press, 2009), pp. 1−8.
9. J. Quiñonero-Candela and C. Rasmussen, A unifying view of sparse approximate Gaussian process regression, *J. Mach. Learn. Res.* **6** (2005), pp. 1939−1959.
10. A. Ranganathan, Y. Ming-Hsuan and J. Ho, Online sparse Gaussian process regression and its applications, *IEEE Trans. Image Process.* **20**(2) (2010), pp. 391−404.
11. C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2006).
12. M. Seeger, Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximation, Ph.D. thesis, University of Edinburgh (2003).
13. M. Seeger, C. Williams and N. Lawrence, Fast forward selection to speed up sparse Gaussian process regression, in *Proc. Int. Workshop on Artificial Intelligence and Statistics*, Key West, USA (2003), pp. 205−212.
14. A. Smola and P. Bartlett, Sparse greedy Gaussian process regression, in *Advances in Neural Information Processing Systems*, Vol. 13 (MIT Press, 2001), pp. 619−625.
15. E. Snelson and Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in *Advances in Neural Information Processing Systems*, Vol. 18 (MIT Press, 2006), pp. 1257−1264.
16. E. Snelson and Z. Ghahramani, Local and global sparse Gaussian process approximations, in *Proc. Int. Conf. Artificial Intelligence and Statistics*, San Juan, Puerto Rico (2007), pp. 524−531.
17. B. Srinivasan and R. Duraiswami, Efficient subset selection via the kernelized Rènyi distance, in *Proc. Int. Conf. Computer Vision*, Kyoto, Japan (2009), pp. 1081−1088.
18. P. Sun and X. Yao, Greedy forward selection algorithms to sparse Gaussian process regression, in *Proc. Int. Joint Conf. Neural Network*, Vancouver, Canada (2006), pp. 159−165.
19. X. Zhao, Y. Fu and Y. Liu, Human motion tracking by temporal-spatial local Gaussian process experts, *IEEE Trans. Image Process.* **20**(4) (2011), pp. 1141−1151.



Heung-Il Suk received the B.S. and M.S. degrees in Computer Engineering from Pukyong National University, Busan, Korea, in 2004 and 2007, respectively, and his Ph.D. in Computer Science and Engineering from Korea University, Seoul, Korea, in 2012. He is currently a postdoctoral research associate in the Department of Radiology and the Biomedical Research Imaging Center (BRIC) at the University of North Carolina, Chapel Hill, USA. From 2004 to 2005, he was a visiting researcher at the Computer and Vision Research Center, University of Texas at Austin, USA. His current research interests include machine learning, computer vision, brain-computer interfaces, and neuroimaging analysis. He received the Outstanding Paper Award at the Korea Computer Congress in 2007. He also received the Silver Award at the 18th Samsung Human-Tech Thesis Prize in 2012.

Yuzhuo Wang received his M.S. degree in Computer Science and Engineering from Korea University, Seoul, Korea, in 2010. Her research interests include pattern recognition and computer vision.

Author photo is not available.

Seong-Whan Lee received his B.S. degree in Computer Science and Statistics from Seoul National University, Seoul, Korea, in 1984, and his M.S. degree and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Seoul, Korea, in 1986 and 1989, respectively. He is currently the Hyundai-Kia Motor Chair Professor at Korea University, Seoul, where he is the Head of the Department of Brain and Cognitive Engineering and the Director of the Institute for Brain and Cognitive Engineering. He is the Principal Investigator of the World Class University (WCU) project on "Brain and Cognitive Engineering" research, which is funded by the Ministry of Education, Science and Technology of Korea. His current research interests include pattern recognition, computer vision, and brain informatics. He has authored more than 250 publications for international journals and conference proceedings, and authored 10 books.

Dr. Lee was the winner of the Annual Best Student Paper Award of the Korea Information Science Society in 1986. He received the First Outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Award from Chungbuk National University in 1994. He received the Outstanding Research Award from the Korea Information Science Society in 1996. He received the Lotfi Zadeh Best Paper Award at the IEEE International Conference on Machine Learning and Cybernetics in 2011. He also received the Scientist of the Month Award from the Ministry of Education, Science and Technology of Korea in 2012.

A Fellow of the IEEE, IAPR, and Korean Academy of Science and Technology, he has served several professional societies as a chairman or governing board member. He was the founding Co-Editor-in-Chief of the International Journal of Document Analysis and Recognition. He has been an Associate Editor of several international journals including Pattern Recognition, ACM Transactions on Applied Perception, IEEE Transactions on Affective Computing, Image and Vision Computing, International Journal of Pattern Recognition and Artificial Intelligence, and International Journal of Image and Graphics. He was a General or Program Chair of many international conferences and workshops and was also on the program committees of numerous conferences and workshops.