Incremental Sparse Kernel Machine

Masa-aki Sato[1] and Shigeyuki Oba[2]

[1] ATR, Human Information Science Laboratories, and CREST, JST,
masa-aki@atr.co.jp
[2] Nara Institute of Science and Technology, shige-o@is.aist-nara.ac.jp

Abstract. The Relevance Vector Machine (RVM) gives a probabilistic model for a sparse kernel representation. It achieves comparable performance to the Support Vector Machine (SVM) while using substantially fewer kernel bases. However, the computational complexity of the RVM in the training phase prohibits its application to large datasets. In order to overcome this difficulty, we propose an incremental Bayesian method for the RVM. The preliminary experiments showed the efficiency of our method for large datasets.

1 Introduction

The SVM [3] has been recognized as a state-of-the-art method for machine learning. It makes predictions based on a linear combination of kernel functions defined on a subset of the training data points called the support vectors. This sparse kernel representation avoids overfitting to the training data and increases the generalization ability. However, the main disadvantage of the SVM is its lack of probabilistic model. Therefore, it cannot estimate the uncertainty of a prediction, which gives important information in application to real-world problems. Recently, a probabilistic model for a kernel representation called the Relevance Vector Machine (RVM) [1] was proposed. The Bayes estimation of the RVM with an Automatic Relevance Determination (ARD) prior [1, 2] leads to a sparse kernel representation. The RVM achieves comparable performance to the SVM while using substantially fewer kernel bases [1]. However, the principal disadvantage of the RVM is its computational complexity in the training phase. It requires $O((\#\ of\ data)^3)$ computation time. This prohibits the application of the RVM to large datasets.

In order to overcome this difficulty, we propose an incremental Bayesian method for the RVM. The proposed method can learn a large amount of data incrementally and gives a sparse kernel representation. It requires $O((\#\ of\ bases)^2 * (\#\ of\ data))$ computation time, which is much smaller than the computation time required by the RVM for a large amount of data.

The Gaussian Process (GP) [4] also provides a probabilistic model for kernel representation. An online learning method for the GP was proposed [5] recently. However, the GP has no inherent mechanism for basis selection. In order to get a sparse kernel representation, it is necessary to introduce heuristic mechanisms [5]. The advantage of our method is that the sparse kernel representation is an

inherent property of the Bayes estimation with the ARD prior. The preliminary experiments showed the efficiency of our approach for large datasets.

2 Relevance Vector Machine

In supervised learning, a set of input-output pairs $(\boldsymbol{X}, \boldsymbol{Y}) \equiv \{\boldsymbol{x}(n), \boldsymbol{y}(n) | n = 1, \ldots, N\}$ is given to a learner, where $\boldsymbol{x}$ and $\boldsymbol{y}$ represent an L-dim. input vector and a D-dim. output vector, respectively. The task of the learner is to predict the output $\boldsymbol{y}$ for a new input data $\boldsymbol{x}$ based on the training data $(\boldsymbol{X}, \boldsymbol{Y})$ and some prior knowledge of the problem. The RVM [1] is defined as a probabilistic model for the kernel representation:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{W}, \sigma) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{W}), \sigma^{-1}), \tag{1}$$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{W}) = \sum_{n=0}^{N} \boldsymbol{w}_n \phi_n(\boldsymbol{x}) = \sum_{n=0}^{N} \boldsymbol{w}_n K(\boldsymbol{x}, \boldsymbol{x}(n)), \tag{2}$$

where $\boldsymbol{w}_n$ denotes a D-dim. weight vector, $\phi_n(\boldsymbol{x}) \equiv K(\boldsymbol{x}, \boldsymbol{x}(n))(n = 1, \ldots, N)$ is a kernel basis function defined on an input point $\boldsymbol{x}(n)$, and $\phi_0(\boldsymbol{x}) \equiv 1$. $\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \sigma^{-1})$ denotes a normal distribution over $\boldsymbol{y}$ with mean $\boldsymbol{\mu}$ and variance σ^{-1}. Accordingly, the conditional likelihood for the training data can be written as $P(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \sigma) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{y}(n)|\boldsymbol{f}(\boldsymbol{x}(n), \boldsymbol{X}, \boldsymbol{W}), \sigma^{-1})$. Applying the Bayesian method together with the ARD prior [1, 2], one can get a sparse kernel representation: $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}_B, \boldsymbol{W}) = \sum_{j=0}^{J} \boldsymbol{w}_j \phi_j(\boldsymbol{x})$, where $\boldsymbol{X}_B \equiv \{\boldsymbol{x}_B(j) | j = 1, \ldots, J\} \subset \boldsymbol{X}$ represents a reduced set of input data that defines the sparse kernel basis $\phi_j(\boldsymbol{x}) \equiv K(\boldsymbol{x}, \boldsymbol{x}_B(j))$ and is called the relevance vector [1].

3 Incremental Sparse Kernel Machine

3.1 Incremental Bayesian method

In the original RVM [1, 2], the posterior parameter distribution is calculated for the entire dataset $(\boldsymbol{X}, \boldsymbol{Y})$. The incremental Bayesian method proposed here uses only a part of the dataset in one learning epoch. At the beginning of the learning, a dataset $(\boldsymbol{X}_D, \boldsymbol{Y}_D) \equiv \{\boldsymbol{x}_D(m), \boldsymbol{y}_D(m) | m = 1, \ldots, M\}$ is selected from the dataset $(\boldsymbol{X}, \boldsymbol{Y})$. The current basis set $\boldsymbol{X}_B \equiv \{\boldsymbol{x}_B(j) | j = 1, \ldots, J\}$ is also selected from the input dataset $\boldsymbol{X}$. The conditional likelihood for the current dataset $(\boldsymbol{X}_D, \boldsymbol{Y}_D)$ under the current basis set $\boldsymbol{X}_B$ is given by

$$P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B, \boldsymbol{W}, \sigma) = \prod_{m=1}^{M} \mathcal{N}(\boldsymbol{y}_D(m)|\boldsymbol{f}(\boldsymbol{x}_D(m), \boldsymbol{X}_B, \boldsymbol{W}), \sigma^{-1}), \tag{3}$$

where $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}_B, \boldsymbol{W})$ is defined in (2). We employ the hierarchical ARD prior [2] for the model parameters $\boldsymbol{W} \equiv \{\boldsymbol{w}_j | j = 0, \ldots, J\}$:

$$P_0(\boldsymbol{W}|\sigma, \boldsymbol{\alpha}) = \prod_{j=0}^{J} \mathcal{N}(\boldsymbol{w}_j | \overline{\boldsymbol{w}}_{j0}, (\sigma\alpha_j)^{-1}), \quad P_0(\boldsymbol{\alpha}) = \prod_{j=0}^{J} \Gamma(\alpha_j | a_{j0}, \bar{\alpha}_{j0}^{-1}), \tag{4}$$

where the hyperparameters $\boldsymbol{\alpha} \equiv \{\alpha_j | j = 0, \ldots, J\}$ are introduced for controlling the variance of each weight vector $\boldsymbol{w}_j$. The mean weight vector $\overline{\boldsymbol{w}}_{j0}$ of the ARD prior (4) is assumed to be zero, i.e., $\overline{\boldsymbol{w}}_{j0} \equiv 0$. This introduces a bias on $\boldsymbol{w}_j$ toward a null vector. However, we have no idea about the proper value of the hyperparameter $\boldsymbol{\alpha}$. Therefore, the hyperparameter $\boldsymbol{\alpha}$ is integrated by introducing the hierarchical prior for $\boldsymbol{\alpha}$. The hyperprior for $\boldsymbol{\alpha}$ is assumed to be a product of the Gamma distribution defined by $\Gamma(\alpha|a,b) \equiv \alpha^{-1}(ab\alpha)^a \exp(-ab\alpha)/\Gamma(a)$, which is the conjugate prior for the inverse-variance parameter. $\Gamma(a)$ is the Gamma function defined by $\Gamma(a) \equiv \int_0^\infty dt t^{a-1} e^{-at}$. We also introduce a hierarchical prior for the inverse of the output variance parameter σ:

$$P_0(\sigma|\tau) = \Gamma(\sigma|\gamma_{\sigma 0}, \tau), \qquad P_0(\tau) = \Gamma(\tau|\gamma_{\tau 0}, \overline{\tau}_0^{-1}). \tag{5}$$

3.2 Incremental Sparse Kernel Machine

The posterior distribution over the model parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \sigma)$ and the hyperparameters $\boldsymbol{\xi} = (\boldsymbol{\alpha}, \tau)$ is given by

$$P(\boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{Y}_D, \boldsymbol{X}_D, \boldsymbol{X}_B) = \frac{P(\boldsymbol{Y}_D, \boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{X}_D, \boldsymbol{X}_B)}{P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B)}, \tag{6}$$

$$P(\boldsymbol{Y}_D, \boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{X}_D, \boldsymbol{X}_B) = P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B, \boldsymbol{W}, \sigma) P_0(\boldsymbol{W}|\sigma, \boldsymbol{\alpha}) P_0(\sigma|\tau) P_0(\boldsymbol{\alpha}) P_0(\tau),$$

$$P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B) = \int d\boldsymbol{\theta} d\boldsymbol{\xi} P(\boldsymbol{Y}_D, \boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{X}_D, \boldsymbol{X}_B). \tag{7}$$

Since the integration in the marginal likelihood (evidence) $P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B)$ is analytically intractable, we use the variational Bayes (VB) method developed in [6, 7, 2]. In order to approximate the posterior (6), the VB method introduces a trial (variational) posterior $Q(\boldsymbol{\theta}, \boldsymbol{\xi})$ and defines the free energy for Q:

$$\begin{aligned} F[Q] &= \int d\boldsymbol{\theta} d\boldsymbol{\xi} Q(\boldsymbol{\theta}, \boldsymbol{\xi}) \log(P(\boldsymbol{Y}_D, \boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{X}_D, \boldsymbol{X}_B)/Q(\boldsymbol{\theta}, \boldsymbol{\xi})) \\ &= \log P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B) - \int d\boldsymbol{\theta} d\boldsymbol{\xi} Q(\boldsymbol{\theta}, \boldsymbol{\xi}) \log \frac{Q(\boldsymbol{\theta}, \boldsymbol{\xi})}{P(\boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{Y}_D, \boldsymbol{X}_D, \boldsymbol{X}_B)} \end{aligned} \tag{8}$$

The second term in (8) is the Kullback-Leibler divergence between the trial posterior $Q(\boldsymbol{\theta}, \boldsymbol{\xi})$ and the true posterior $P(\boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{Y}_D, \boldsymbol{X}_D, \boldsymbol{X}_B)$ given by (6). Since the first term in (8) does not depend on $Q(\boldsymbol{\theta}, \boldsymbol{\xi})$, the free energy $F[Q]$ is maximized when $Q(\boldsymbol{\theta}, \boldsymbol{\xi})$ is equal to $P(\boldsymbol{\theta}, \boldsymbol{\xi}|\boldsymbol{Y}_D, \boldsymbol{X}_D, \boldsymbol{X}_B)$. In this case, the free energy $F[Q]$ becomes equal to the log-evidence $\log P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{X}_B)$. The VB method assumes a factorized form of $Q(\boldsymbol{\theta}, \boldsymbol{\xi})$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\xi}) = Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) Q_{\boldsymbol{\xi}}(\boldsymbol{\xi}). \tag{9}$$

This factorization assumption is weaker than the factorization assumption used in [2], where each component of the parameters and the hyperparameters is assumed to be factorized. Under the assumption of (9), the free energy is alternately maximized with respect to $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $Q_{\boldsymbol{\xi}}(\boldsymbol{\xi})$. The maximized free energy gives a lower bound on the log-evidence.

The alternate free energy maximization leads to the functional form of the posterior given by

$$
\begin{aligned}
Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) =& Q_{\boldsymbol{W}}(\boldsymbol{W}|\sigma)Q_{\sigma}(\sigma), & Q_{\boldsymbol{\xi}}(\boldsymbol{\xi}) =& Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})Q_{\tau}(\tau), \\
Q_{\boldsymbol{W}}(\boldsymbol{W}|\sigma) =& \prod_{j=1}^{J} \mathcal{N}(\boldsymbol{w}_j|\overline{\boldsymbol{w}}_j, \sigma^{-1}\Sigma^{-1}), & Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) =& \prod_{j=0}^{J} \Gamma(\alpha_j|a_j, \overline{\alpha}_j^{-1}), \\
Q_{\sigma}(\sigma) =& \Gamma(\sigma|\gamma_{\sigma}, \Delta), & Q_{\tau}(\tau) =& \Gamma(\tau|\gamma_{\tau}, \bar{\tau}^{-1}),
\end{aligned} \tag{10}
$$

where the $((J+1)\times D)$ matrix $\overline{\boldsymbol{W}} \equiv (\overline{\boldsymbol{w}}_0, \overline{\boldsymbol{w}}_1, \ldots, \overline{\boldsymbol{w}}_J)'$, the $((J+1)\times(J+1))$ inverse-covariance matrix Σ, $\gamma_{\sigma}, \Delta, \{a_j, \overline{\alpha}_j | j = 0, \ldots, J\}, \gamma_{\tau}$, and $\bar{\tau}$ are determined in the following steps.

1) In the maximization step with respect to the posterior parameter distribution $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, the parameters in $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ are updated:

$$
\begin{aligned}
&\Sigma = \Phi'\Phi + \overline{\boldsymbol{A}}, \quad \overline{\boldsymbol{W}} = \Sigma^{-1}(\Phi'\boldsymbol{Y}_D + \overline{\boldsymbol{A}}\,\overline{\boldsymbol{W}}_0), \quad \gamma_{\sigma} = \gamma_{\sigma 0} + DM/2, \\
&\gamma_{\sigma}\Delta = \frac{1}{2}\mathrm{Tr}(\boldsymbol{Y}_D - \Phi\overline{\boldsymbol{W}})'(\boldsymbol{Y}_D - \Phi\overline{\boldsymbol{W}}) + \frac{1}{2}\mathrm{Tr}(\overline{\boldsymbol{W}} - \overline{\boldsymbol{W}}_0)'\overline{\boldsymbol{A}}(\overline{\boldsymbol{W}} - \overline{\boldsymbol{W}}_0) + \gamma_{\sigma 0}\bar{\tau},
\end{aligned} \tag{11}
$$

where the $(M\times(J+1))$ matrix Φ, the $((J+1)\times(J+1))$ matrix $\overline{\boldsymbol{A}}$, and the $(M\times D)$ matrix $\boldsymbol{Y}_D$ are defined as $(\Phi)_{mj} = \phi_j(\boldsymbol{x}_D(m)) = K(\boldsymbol{x}_D(m), \boldsymbol{x}_B(j))$, $\overline{\boldsymbol{A}} = \mathbf{diag}(\overline{\alpha}_0, \overline{\alpha}_1, \ldots, \overline{\alpha}_J)$, and $\boldsymbol{Y}_D = (\boldsymbol{y}(1), \ldots, \boldsymbol{y}(M))'$, respectively. The mean weight of the posterior is given by $\langle \boldsymbol{W} \rangle_{\boldsymbol{\theta}} \equiv \int d\boldsymbol{\theta} Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\boldsymbol{W} = \overline{\boldsymbol{W}}$, and the mean of the inverse-variance parameter σ is given by $\langle \sigma \rangle_{\boldsymbol{\theta}} = \Delta^{-1}$. The covariance matrix for the weight $\boldsymbol{W}$ is given by $\sigma^{-1}\Sigma^{-1}$. γ_{σ} is a degree of freedom of the Gamma distribution $Q_{\sigma}(\sigma)$ and the variance of σ is proportional to γ_{σ}^{-1}.

2) In the maximization step with respect to the posterior hyperparameter distribution $Q_{\boldsymbol{\xi}}(\boldsymbol{\xi})$, the parameters in $Q_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ are updated:

$$
\begin{aligned}
a_j =& a_{j0} + \frac{1}{2}D, \quad a_j\overline{\alpha}_j^{-1} = a_{j0}\overline{\alpha}_{j0}^{-1} + \frac{1}{2}\left(\Delta^{-1}||\overline{\boldsymbol{w}}_j - \overline{\boldsymbol{w}}_{j0}||^2 + D(\Sigma^{-1})_{jj}\right), \\
\gamma_{\tau} =& \gamma_{\tau 0} + \gamma_{\sigma 0}, \quad \gamma_{\tau}\bar{\tau}^{-1} = \gamma_{\tau 0}\bar{\tau}_0^{-1} + \gamma_{\sigma 0}\Delta^{-1}.
\end{aligned} \tag{12}
$$

The means of the hyperparameters are given by $\langle \alpha_j \rangle_{\boldsymbol{\xi}} \equiv \int d\boldsymbol{\xi} Q_{\boldsymbol{\xi}}(\boldsymbol{\xi})\alpha_j = \overline{\alpha}_j$ and $\langle \tau \rangle_{\boldsymbol{\xi}} = \bar{\tau}$.

3) After the convergence, this step selects the weights that have norm $||\overline{\boldsymbol{w}}_j||^2$ larger than the threshold value $W_{\min}$. The corresponding basis points $\boldsymbol{x}_B(j)$ form an old basis set $\boldsymbol{X}_{\mathrm{old}}$. In the next epoch, we select a new dataset $(\boldsymbol{X}_{\mathrm{new}}, \boldsymbol{Y}_{\mathrm{new}})$ from the entire dataset $(\boldsymbol{X}, \boldsymbol{Y})$. The basis set and the dataset in the next epoch are given by $\boldsymbol{X}_B = \{\boldsymbol{X}_{\mathrm{new}}, \boldsymbol{X}_{\mathrm{old}}\}$ and $(\boldsymbol{X}_D, \boldsymbol{Y}_D) = (\{\boldsymbol{X}_{\mathrm{new}}, \boldsymbol{X}_{\mathrm{old}}\}, \{\boldsymbol{Y}_{\mathrm{new}}, \boldsymbol{Y}_{\mathrm{old}}\})$, respectively.

4) We have no specific information on the weights $\boldsymbol{W}_{\mathrm{new}}$ that correspond to the new basis points $\boldsymbol{X}_{\mathrm{new}}$. Therefore, we use the ARD prior defined in (4) with a small a_{j0} and $\overline{\boldsymbol{w}}_{j0} = 0$. On the other hand, we do have information on the weights $\boldsymbol{W}_{\mathrm{old}}$ that correspond to the old basis points $\boldsymbol{X}_{\mathrm{old}}$. Accordingly, we use the obtained value of $\overline{\boldsymbol{w}}_j, \overline{\alpha}_j$, and a_j as the prior parameter in the next

epoch, i.e., $\overline{\boldsymbol{w}}_{j0} = \overline{\boldsymbol{w}}_j, \overline{\alpha}_{j0} = \overline{\alpha}_j$, and $a_{j0} = a_j$. An alternative method is to use the obtained posterior $Q_{\boldsymbol{W}}(\boldsymbol{W}_{\rm old}|\sigma)$ as the new prior for $\boldsymbol{W}_{\rm old}$. In this case, the matrix $\overline{\boldsymbol{A}}$ in (11) becomes a block diagonal matrix and the block matrices are given by $\overline{\boldsymbol{A}}_{\rm old} = \Sigma_{\rm old}$ and $\overline{\boldsymbol{A}}_{\rm new} = \mathbf{diag}(\overline{\boldsymbol{\alpha}}_{\rm new})$ in an obvious notation. We also have information on the inverse-variance parameter σ, so we reset the prior parameter as $\bar{\tau}_0 = \bar{\tau}, \gamma_{\sigma 0} = \gamma_\sigma$, and $\gamma_{\tau 0} = \gamma_\tau$ or use the obtained posterior $Q_\sigma(\sigma)$ as the new prior for σ.

We repeat the above process until all data $(\boldsymbol{X}, \boldsymbol{Y})$ are processed. The predictive distribution after the learning can be calculated by using the posterior parameter distribution: $\hat{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{X}_B) = \int d\boldsymbol{W} d\sigma P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{X}_B, \boldsymbol{W}, \sigma) Q_{\boldsymbol{W}}(\boldsymbol{W}|\sigma) Q_\sigma(\sigma)$ $= \mathcal{T}(\boldsymbol{y}|\overline{\boldsymbol{W}}' \cdot \boldsymbol{\phi}(\boldsymbol{x}), \Delta(1 + \boldsymbol{\phi}(\boldsymbol{x})' \Sigma^{-1} \boldsymbol{\phi}(\boldsymbol{x})), 2\gamma_\sigma)$, where $\mathcal{T}(\boldsymbol{y}|\boldsymbol{\mu}, C, \gamma)$ denotes the t-distribution with mean $\boldsymbol{\mu}$, variance $C/(1 - 2\gamma^{-1})$, and degree of freedom γ. $\boldsymbol{\phi}(\boldsymbol{x})$ denotes the $(J+1)$-dim. vector defined by $\phi_j(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}_B(j))(j = 1, \ldots, J)$ and $\phi_0(\boldsymbol{x}) \equiv 1$.

4 Experiments

The applicability of our incremental Sparse Kernel Machine (SKM) was investigated by using large datasets. The Gaussian kernel is used in all experiments. The problem is a prediction task for the chaotic Mackey-Glass (M-G) model, which is defined by the differential equation

$$ds(t)/dt = -bs(t) + as(t - \tau)/(1 + s(t - \tau)^{10}), \tag{13}$$

where $a = 0.2, b = 0.1$, and $\tau = 17$ [8]. The task was to predict $y = s(t + 85)$ from the delay coordinate $\boldsymbol{x} = (s(t), s(t - 6), s(t - 12), s(t - 18))$. The training data were prepared by adding the Gaussian noise with different levels to the generated time-series. The N/S (noise-to-signal) ratios of the added noise were 0.0, 0.11, and 0.22 in terms of standard deviation. Three datasets with 500, 1000, and 10000 data points were used to train the incremental SKM, the ν-SVM [9, 10] and the Kernel Principal Components Regression (KPCR) method [8]. The NRMSE (normalized root mean squared error) for the test dataset and the corresponding number of bases are listed in Table 1. The SKM showed better performance for the data with noise than did the ν-SVM or the KPCR, while the ν-SVM and the KPCR showed better performance for noiseless data. It should be noted that the SKM achieved good performance with a much fewer bases than the ν-SVM. The SKM selected moderate number of bases even when a large amount of data were given. Fig. 1 shows the basis points (relevance vector) obtained by the SKM for 10000 training data together with the M-G attractor.

5 Discussion

In this paper, we formulated the incremental Sparse Kernel Machine (SKM) and showed the efficiency of this method for large datasets. The Bayes estimation using the ARD prior automatically eliminated insignificant bases and gave a sparse

kernel representation for large datasets. We also examined another implementation of SKM, where the dataset in one epoch was the entire dataset and the basis set was given by $\boldsymbol{X}_B = \{\boldsymbol{X}_{\rm new}, \boldsymbol{X}_{\rm old}\}$. However, this implementation did not significantly improve performance, while it required $O((\#\ of\ bases)*(\#\ of\ data)^2)$ computation time. In this paper, we only considered regression problems. However, our method could be easily extended to classification problems by using the same method developed in [2]. Application of our method to real-world problems remains a task for future research.

Table 1. NRMSE for SKM, ν-SVM and KPCR. The number of bases are presented in parentheses.

	n/s=0.0			n/s=0.11			n/s=0.22		
	500	1000	10000	500	1000	10000	500	1000	10000
SKM	0.130	0.109	0.088	0.238	0.201	0.110	0.333	0.322	0.146
	(145)	(148)	(169)	(142)	(250)	(290)	(223)	(413)	(334)
ν-SVM	0.037	0.013	0.004	0.275	0.197	0.148	0.467	0.395	0.355
	(462)	(704)	(3177)	(333)	(507)	(2201)	(334)	(590)	(2769)
KPCR	0.038	0.008	***	0.307	0.280	***	0.443	0.414	***

References

1. Tipping, M. E. (2000). The Relevance Vector Machine. *NIPS* **12**, pp. 652-658.
2. Bishop, C. M. & Tipping, M. E. (2000). Variational relevance vector machines. *Proc. of 16th Conf. UAI,* pp. 46-53.
3. V. N. Vapnik. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag.
4. Williams, C. K. I. & Rasmussen, C. E. (1996). Gaussian Process for Regression. *NIPS* **8**. pp. 514-520.
5. Csatò, L. & Opper, M. (2001). Sparse Representation for Gaussian Process Models. *NIPS* **13**.
6. Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. *Proc. of 15th Conf. UAI,* pp. 21-30.
7. Sato, M. (2001). On-line Model Selection Based on the Variational Bayes. *Neural Computation*, **13**. 1649-1681.
8. Rosipal, R. et al. (2001). Kernel PCA for Feature Extraction and De-Noising in Non-linear Regression. *Neural Computing & Applications.* 10(3).
9. Chang, C. and Lin, C., (2001), LIBSVM: a library for SVM. (available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.)
10. Scholkopf, B. et al. (2000). New support vector algorithms. *Neural Computation,* **12**, 1207-1245.

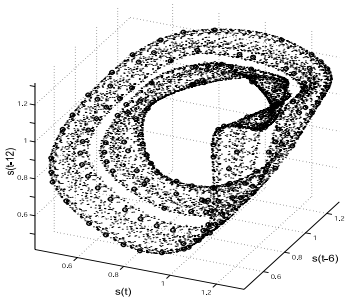

Fig. 1. Relevance vector

[0] **Acknowledgement:** This research was supported in part by the Telecommunicatios Advancement Organization of Japan.