LETTER **Communicated by Edward Snelson**

Clustering Based on Gaussian Processes

Hyun-Chul Kim
grass@postech.ac.kr
Department of Computer Science, Yonsei University, 134 Shinchondong, Sudaimunku Seoul, 120-749, Korea

Jaewook Lee
jaewookl@postech.ac.kr
Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk 797-784, Korea

In this letter, we develop a gaussian process model for clustering. The variances of predictive values in gaussian processes learned from a training data are shown to comprise an estimate of the support of a probability density function. The constructed variance function is then applied to construct a set of contours that enclose the data points, which correspond to cluster boundaries. To perform clustering tasks of the data points, an associated dynamical system is built, and its topological invariant property is investigated. The experimental results show that the proposed method works successfully for clustering problems with arbitrary shapes.

1 Introduction

Gaussian process models are Bayesian kernel machines that have been successfully applied to regression and classification (Rasmussen & Williams, 2006). Gaussian processes for regression (GPR) model the density of the target values as a multivariate gaussian density. The covariance between the targets at two different points is usually controlled by a small set of hyperparameters that captures interpretable properties of the function, such as the length scale of autocorrelation, the overall scale of the function, and the amount of noise. GPR has some advantages over other nonparametric regression algorithms for its ability to provide a principled hyperparameter selection method and the distribution of predicted target value (Williams & Rasmussen, 1995; Neal, 1998; Gibbs & MacKay, 1997) and is shown to be an excellent method for nonlinear regression (Rasmussen, 1996).

In gaussian processes for classification, a latent function is employed whose sign determines the class label instead of modeling them using a multivariate gaussian density and integrates over both hyperparameters and latent values of the function at the data points. It has been reported in a number of works (Williams & Barber, 1998; Neal, 1998; Gibbs & MacKay,

2000; Opper & Winther, 2000; Minka, 2001; Csato, 2002) that they are effectively applied to some difficult classification problems.

So far, most of works related to gaussian processes have been on their applications to supervised learning tasks such as regression and classification, although there are some approaches to apply gaussian process (GP) models to an unsupervised learning task such as density estimation (Schmidt, 2000; Csato, 2002; Kim & Lee, 2006). Clustering, which GP models have not yet been rigorously developed and applied to, is also one of the important unsupervised learning tasks. Clustering is the task of grouping the input patterns naturally (Duda, Hart, & Stork, 2001; Jain, Murty, & Flynn, 1999). Clustering techniques are useful in several exploratory pattern analysis, grouping, decision-making, and machine learning situations, including data mining, document retrieval, image segmentation, and pattern classification.

In this letter, we make a rigorous development of a gaussian process model for clustering. To this end, we start with the observation that the variances in GPR capture regions in input space where the probability density is in some sense large and establish that the variance function in GPR represents the support of a probability density with its generalization error bound. We then take a maximum of the variance function values of the data points as the level value, which is shown to make up the cluster boundaries. A dynamical system associated with the variance function is built and applied to facilitate the cluster labeling of the data points, as well as to partition the input space into several clustered regions. We verify the performance of the proposed method through simulation.

The organization of this letter is as follows. In section 2, we review gaussian processes for regression (GPR). In section 3, we show that the constructed variance function of GPR learning from data estimates the support of an unknown probability density. In section 4, we present a clustering method utilizing the constructed variance function, and we show the simulation results in section 5.

2 Review of Gaussian Processes for Regression

In this section, we briefly review the gaussian processes for regression (O'Hagan, 1978; Williams & Rasmussen, 1995; Gibbs & MacKay, 1997; Neal, 1998). Suppose that we have a data set D of data points $\mathbf{x}_i \in \Re^d$ with continuous target values t_i: $D = \{(\mathbf{x}_i, t_i)|i = 1, 2, \ldots, N\}$, $X = \{\mathbf{x}_i|i = 1, 2, \ldots, N\}$, $T = \{t_i|i = 1, 2, \ldots, N\}$. Given this data set, we wish to find the predictive distribution of the target value $\tilde{t}$ for a new data point $\tilde{\mathbf{x}}$. Let the target function $f(\cdot)$. We put $\mathbf{f}(= \mathbf{f}_N)$ as $[f_1, f_2, \ldots, f_N] = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N)]$, and put $\mathbf{f}_{N+1}$ as $[f_1, f_2, \ldots, f_N, f_{N+1}] = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N), f(\mathbf{x}_{N+1})]$, where $\mathbf{x}_{N+1} = \tilde{\mathbf{x}}$.

Gaussian processes for regression assume that the target function has a gaussian process prior. This means that the density of any collection of target function values is modeled as a multivariate gaussian density. Writing the

dependence of $\mathbf{f}$ on X implicitly, the GP prior over functions can be written as

$$p(\mathbf{f}|X,\Theta) = \frac{1}{(2\pi)^{N/2}|\mathbf{C}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{f}-\boldsymbol{\mu})^{\top}\mathbf{C}^{-1}(\mathbf{f}-\boldsymbol{\mu})\right\}, \tag{2.1}$$

where the mean $\boldsymbol{\mu}$ is usually assumed to be zero, $\boldsymbol{\mu} = \mathbf{0}$, and each term of a covariance matrix C_{ij} is a parameterized function of $\mathbf{x}_i$ and $\mathbf{x}_j$ with hyperparameters Θ, that is, $C_{ij} = C(\mathbf{x}_i, \mathbf{x}_j; \Theta)$.

Since $\mathbf{f}$ and $\mathbf{f}_{N+1}$ have a gaussian density, we get the following predictive density of f_{N+1} as a conditional gaussian density,

$$p(f_{N+1}|D, \mathbf{x}_{N+1}, \Theta) = \frac{p(\mathbf{f}_{N+1}|\Theta, \mathbf{x}_{N+1}, X)}{p(\mathbf{f}_N|\Theta, X)} \tag{2.2}$$

$$= \frac{Z_N}{Z_{N+1}} \exp\left[-\frac{1}{2}\left(\mathbf{f}_{N+1}^T\mathbf{C}_{N+1}^{-1}\mathbf{f}_{N+1} - \mathbf{f}_N^T\mathbf{C}_N^{-1}\mathbf{f}_N\right)\right], \tag{2.3}$$

where Z_N and Z_{N+1} are appropriate normalization constants of $p(\mathbf{f}_N|\Theta, X)$ and $p(\mathbf{f}_{N+1}|\Theta, \mathbf{x}_{N+1}, X)$, respectively. The relationship between $\mathbf{C}_N$ and $\mathbf{C}_{N+1}$ is as follows,

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & \kappa \end{pmatrix}, \tag{2.4}$$

where

$$\mathbf{k} = [C(\tilde{\mathbf{x}}, \mathbf{x}_1; \Theta), C(\tilde{\mathbf{x}}, \mathbf{x}_2; \Theta), \ldots, C(\tilde{\mathbf{x}}, \mathbf{x}_n; \Theta)]^T \quad \text{and} \quad \kappa = C(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}; \Theta). \tag{2.5}$$

By rearranging the terms in equation 2.3, we get a gaussian density of $f_{N+1}(= \tilde{f})$ for a new data point $\mathbf{x}_{N+1}(= \tilde{\mathbf{x}})$ given by

$$p(f_{N+1}|D, \Theta, \mathbf{x}_{N+1}) = \mathcal{N}(\mathbf{k}^T\mathbf{C}^{-1}\mathbf{t}, \kappa - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}). \tag{2.6}$$

The covariance function is the kernel, which defines how data points generalize to nearby data points. The commonly used covariance function, adopted in this letter, is

$$C(\mathbf{x}_i, \mathbf{x}_j; \Theta) = v_0 \exp\left\{-\frac{1}{2}\sum_{m=1}^{d} l_m\left(x_i^m - x_j^m\right)^2\right\} + v_1 + \delta_{ij}v_2. \tag{2.7}$$

The covariance function is parameterized by the hyperparameters $\Theta = \{l_m|m = 1, \ldots, d\} \cup \{v_0, v_1, v_2\}$, which we can learn from the data. The covariance between the targets at two different points is a decreasing function

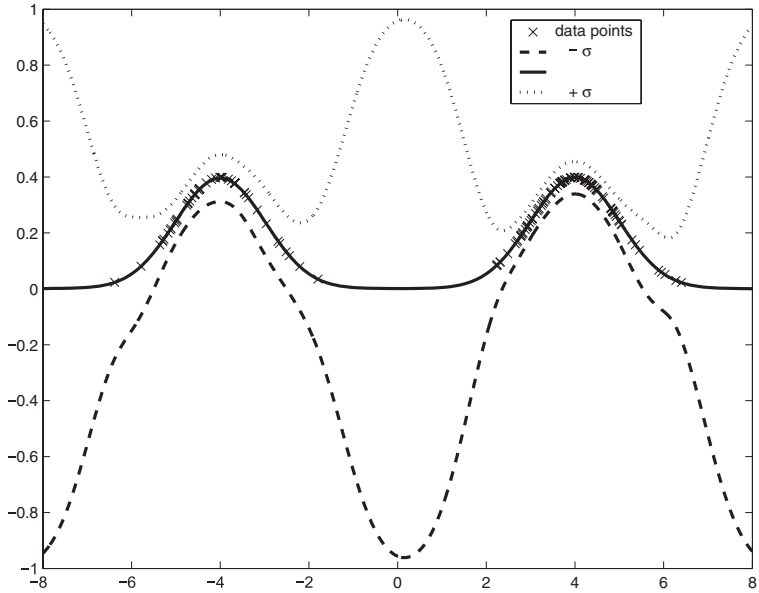

Figure 1: Examples of gaussian process regression. Variance of the predictive value is related to the density.

of their distance in input space. This decreasing function is controlled by a small set of hyperparameters that capture interpretable properties of the function, such as the length scale of autocorrelation, the overall scale of the function, and the amount of noise.

The posterior distributions of the hyperparameters given the data can be inferred in a Bayesian way via Markov chain Monte Carlo (MCMC) methods (Williams & Rasmussen, 1995; Neal, 1998) or they can be selected by maximizing the marginal likelihood (the evidence) (Gibbs & MacKay, 1997). A Bayesian treatment of multilayer perceptrons for regression has been shown to converge to a gaussian process as the number of hidden nodes approaches infinity if the priors on input-to-hidden weights and hidden unit biases are independent and identically distributed (Neal, 1996). Empirically, gaussian processes have been shown to be an excellent method for nonlinear regression (Rasmussen, 1996).

3 Support Estimate from Gaussian Processes

One key observation of gaussian process regression (GPR), as illustrated in Figure 1, is that the variances of predictive values have smaller values in a denser area and larger values in a sparse area (Rasmussen & Williams, 2006;

Kim & Lee, 2006). This means that the variances in GPR capture regions in input space where the probability density is in some sense large, thereby motivating us to consider the variances as a good estimate of the support of a probability density function. A support of a function $f : \Re^d \mapsto Y$ is mathematically defined as the closure of the set of arguments of a function f for which f is not zero. The support of a probability density function is often applied to clustering problems since each separate support domain can be considered a cluster. So clustering problems can be solved in some sense by estimating support of a probability distribution generating given unlabeled data.

This key observation can be explained utilizing the following property of GPR: the variance function of a predictive distribution of GPR in equation 2.6, given by

$$\sigma^2(\mathbf{x}) = \kappa - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k}, \tag{3.1}$$

is nondegenerate; that is, the variance becomes large (small) in a sparse (dense) region since the second term on the right-hand side becomes small in a sparse region of input space wherein points lie far away from the sample data points.

Another good property is that the variance function in equation 3.1 does not depend on the target values, $\mathbf{t}$. This feature enables us to apply the GPR to unlabeled sample data,

$$\mathcal{X} = \{\mathbf{x}_i | i = 1, 2, \ldots, N\},$$

by taking all target values of zero with noise allowance, that is, $t_i = 0$ for all $i = 1, \ldots, N$, without changing the variance behavior.

With this machinery, the sample functions from the GP posterior would fluctuate around zero. In a denser area, the degree of fluctuation of the sample functions is low with more probability, because the sample function values should be smooth and close to the target values, that is, zero, near the data points. So $E[f(\mathbf{x})^2]$ should be closer to zero in a denser area and bigger in a less dense area, which is exactly the same as the variance function $\sigma^2(\mathbf{x})$, that is,

$$E[f(\mathbf{x})^2] = V[f(\mathbf{x})] + \{E[f(\mathbf{x})]\}^2 \tag{3.2}$$

$$= \sigma^2(\mathbf{x}). \tag{3.3}$$

Figure 2 shows the variance of the sampled functions given the 1D data points sampled from a bimodal distribution. Intuitively, we see that there are two separate support domains in a denser area where each support is in some sense related to a cluster or a subcluster. Variances clearly show that the data points are sampled from a bimodal distribution.

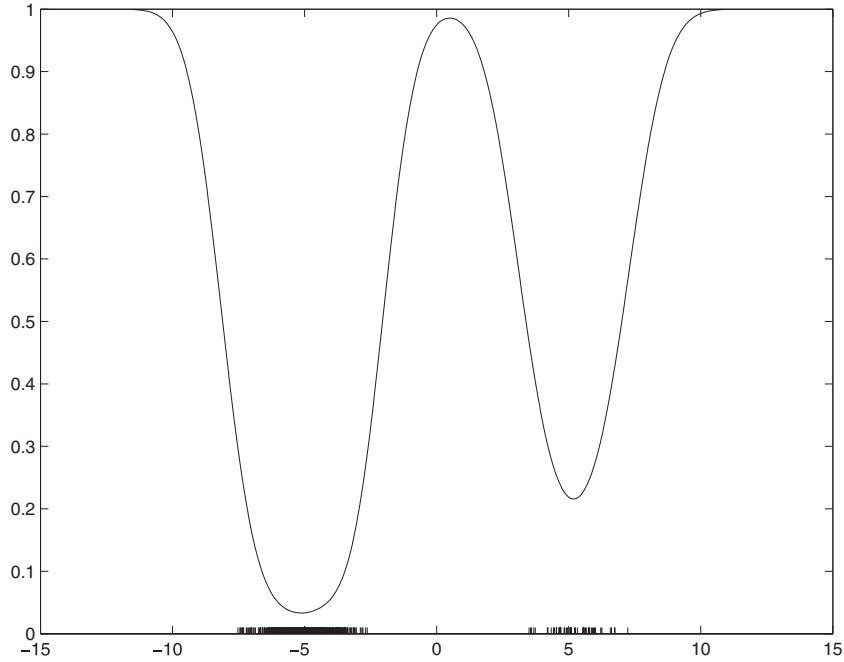

Figure 2: Variance of $f(\cdot)$ for the 1D data set of 500 points sampled from $\mathcal{N}(-5, 1)$ and 50 points sampled from $\mathcal{N}(5, 1)$.

To be more rigorous, we next give a theoretical result showing that the variance function can indeed represent the support of a probability density with its generalization error bound. We begin with the definition for $D(\mathbf{X}, f, \theta)$, where $\mathbf{X} \equiv (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ (Schölkopf & Smola, 2002).

Definition. *Let f be a real-valued function on a space X, Fix $\theta \in \Re$. For $\mathbf{x} \in X$ let $d(\mathbf{x}, f, \theta) = \max\{0, \theta - f(\mathbf{x})\}$. Similarly for a training sequence $\mathbf{X} \equiv (\mathbf{x}_1, \ldots, \mathbf{x}_l)$, we define*

$$\mathcal{D}(\mathbf{X}, f, \theta) = \sum_{\mathbf{x} \in X} d(\mathbf{x}, f, \theta). \tag{3.4}$$

The variance function estimates the support region by $\sigma^2(\mathbf{x}) \leq \theta$, where $\theta = \max_{x \in X} \sigma^2(\mathbf{x})$.

The following theorem provides a bound on the probability that a novel point lies outside the region slightly larger than the estimated support region by the variance function of gaussian processes, similar to the result

obtained in Schölkopf, Platt, Shawe-Taylor, Smola, and Williamson (1999) for support vector machines. In the following, log and ln denotes logarithms to base 2 and natural logarithms, respectively.

Theorem 1. *Consider a fixed but unknown probability distribution P with no atomic components on the space F with support contained in a ball of radius 1 about the origin and $\sigma^2(\mathbf{x}) = \kappa - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}$. Assume that $\theta = \max_{\mathbf{x}\in X} \sigma^2(\mathbf{x})$. Then with probability $1-\delta$ over randomly drawn training sequences X of size N, for all $\gamma > 0$, $\sigma^2(\mathbf{x}) = \kappa - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}$, and,*

$$P(\mathbf{x} : \sigma^2(\mathbf{x}) > \theta + 2\gamma) \le \frac{2}{N}\left(k + \log\frac{N^2}{2\delta}\right), \tag{3.5}$$

where

$$k = \frac{c_1 \log(c_2\hat{\gamma}^2 N)}{\hat{\gamma}^2} + \mathcal{D}\hat{\gamma}\log\left(e\left(\frac{(2N-1)\hat{\gamma}}{\mathcal{D}} + 1\right)\right) + 2, \tag{3.6}$$

$c_1 = 4c^2$, $c_2 = ln(2)/c^2$, $c = 103$, $\hat{\gamma} = \gamma/\|w\|$, *and* $\mathcal{D} = \mathcal{D}(X, g, \kappa - \theta)$.

Proof. First, we show that $h(\mathbf{x})(= \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k})$ is a linear function in a kernel defined feature space. $\mathbf{k}$ can be represented as $\Phi_X\Phi(\mathbf{x})$, where $\Phi_X = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_l)]^T$. Then we get

$$h(\mathbf{x}) = \Phi(\mathbf{x})^T\Phi_X^T\mathbf{C}^{-1}\Phi_X\Phi(\mathbf{x}) \tag{3.7}$$

$$= \mathrm{tr}(\Phi(\mathbf{x})^T\Phi_X^T\mathbf{C}^{-1}\Phi_X\Phi(\mathbf{x})) \tag{3.8}$$

$$= \mathrm{tr}(\Phi(\mathbf{x})\Phi(\mathbf{x})^T\Phi_X^T\mathbf{C}^{-1}\Phi_X). \tag{3.9}$$

If we denote with $w = \Phi_X^T\mathbf{C}^{-1}\Phi_X$ and $\Phi_2(\mathbf{x}) = \mathrm{Vec}(\Phi(\mathbf{x})\Phi(\mathbf{x})^T)$ (meaning $\Phi_2(\mathbf{x})[l*(i-1)+j] = \Phi(\mathbf{x})[i]\Phi(\mathbf{x})[j]$), then $h(\mathbf{x})$ is a linear function $f_w(\Phi_2(\mathbf{x}))(= w\cdot\Phi_2(\mathbf{x}))$ in a kernel feature space. So $\sigma^2(\mathbf{x}) = \kappa - h(\mathbf{x})$ is a linear function.

By theorem 17 in Schölkopf et al. (1999), with probability $1-\delta$, for $\hat{\theta} = \min_{\mathbf{x}\in X} h(\mathbf{x})(= \kappa - \theta)$ and all $\gamma > 0$,

$$P\{\Phi_2(\mathbf{x}) : f_w(\Phi_2(\mathbf{x})) < \hat{\theta} - 2\gamma\} = P\{\mathbf{x} : \sigma^2(\mathbf{x}) > \kappa - \hat{\theta} + 2\gamma\} \tag{3.10}$$

$$= P\{\mathbf{x} : \sigma^2(\mathbf{x}) > \theta + 2\gamma\} \tag{3.11}$$

$$\le \frac{2}{N}\left(k + \log\frac{N^2}{2\delta}\right), \tag{3.12}$$

where

$$k = \frac{c_1 \log(c_2 \hat{\gamma}^2 N)}{\hat{\gamma}^2} + \mathcal{D}\hat{\gamma} \log\left(e\left(\frac{(2N-1)\hat{\gamma}}{\mathcal{D}} + 1\right)\right) + 2, \tag{3.13}$$

$c_1 = 4c^2$, $c_2 = ln(2)/c^2$, $c = 103$, $\hat{\gamma} = \gamma/\|w\|$, and $\mathcal{D} = \mathcal{D}(X, f_w, \hat{\theta})$.

This result shows that the variance function of a gaussian process characterizes the support of a high-dimensional distribution of a given data set and the estimated support set by GP has tractable complexity even in high-dimensional cases.

4 Clustering Based on the Variance Function

In the previous section, we showed that the variance function of GPR can represent an estimate of the support of a probability density function. As a by-product of the constructed variance function, $\sigma^2(\cdot)$ in equation 3.1, cluster boundaries can be obtained by a set of contours that enclose the points in data space, which is given by $\{\mathbf{x} : \sigma^2(\mathbf{x}) \leq r^*\}$, where $r^* = \max_k \sigma^2(\mathbf{x}_k)$. (See Figure 4.) Specifically, the level set of $\sigma^2(\cdot)$ is decomposed into several disjoint connected sets,

$$L(r^*) := \{\mathbf{x} : \sigma^2(\mathbf{x}) \leq r^*\} = C_1 \cup \cdots \cup C_p, \tag{4.1}$$

where the C_i, $i = 1, \ldots, p$ are disjoint connected sets corresponding to different clusters and p is the number of clusters determined by $\sigma^2(\cdot)$.

4.1 Cluster Characterization from a Dynamical Systems Viewpoint.

In this section we describe a way to assign to each data point a cluster index determined by the level set in equation 4.1. To do this, we build the following dynamical system associated with the variance function, $\sigma^2(\cdot)$:

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}) := -\nabla\sigma^2(\mathbf{x}). \tag{4.2}$$

The existence of a unique time evolution solution (or trajectory) $\mathbf{x}(t) : \Re \to \Re^n$ for each initial condition $\mathbf{x}(0) = \mathbf{x}_0$ is guaranteed by the smoothness of the function F. (For more details on dynamical system theory, see Guckenheimer & Homes, 1986; Lee & Chiang, 2004). A state vector $\bar{\mathbf{x}} \in \Re^n$ satisfying the equation $F(\bar{\mathbf{x}}) = 0$ is called an equilibrium point of equation 4.2 and called an (asymptotically) stable equilibrium point if all the eigenvalues of its derivative are positive. Geometrically, for each $\mathbf{x}$, the vector field $F(\mathbf{x})$ in equation 4.2 is orthogonal to the hypersurface

$\{\mathbf{y} : \sigma^2(\mathbf{y}) = r^*\}$, where $\sigma^2(\mathbf{x}) = r^*$ and points inward the surface. This property makes each trajectory flow inward and remain in one of the clusters described by the level sets of the variance function σ^2, which is rigorously proved below.

The basin of attraction of a stable equilibrium point $\mathbf{s}$ is defined as a set of all the points converging to $\mathbf{s}$ when the process in equation 4.2 is applied,

$$A(\mathbf{s}) := \left\{ \mathbf{x}(0) \in \Re^n : \lim_{t \to \infty} \mathbf{x}(t) = \mathbf{s} \right\}.$$

One distinguished feature of the constructed system in equation 4.2 is that it preserves a topological property (i.e., connectedness) of the level set in equation 4.1, as is shown in the next lemma.

Lemma 1. *For any given level value $r > 0$, each connected component of the level set $L(r) = \{\mathbf{x} : \sigma^2(\mathbf{x}) \le r\}$ is positively invariant, that is, if a point is on a connected component of $L(r)$, then its entire positive trajectory lies on the same component.*

Proof. Let $\mathbf{x}(0) = \mathbf{x}_0$ be a point in a connected component, say C, of $L(r)$ and $\mathbf{x}(t)$ be the trajectory starting at $\mathbf{x}(0) = \mathbf{x}_0$. Since

$$\frac{d}{dt}\sigma^2(\mathbf{x}) = -\nabla\sigma^2(\mathbf{x})^T \frac{d\mathbf{x}}{dt} = -\|\nabla\sigma^2(\mathbf{x})\|^2 \le 0,$$

$\sigma^2(\mathbf{x}(t))$ is a decreasing function of $t \in \Re$, and so we have $\sigma^2(\mathbf{x}(t)) \le \sigma^2(\mathbf{x}_0)$ for all $t \ge 0$, or equivalently, $\{\mathbf{x}(t) : t \ge 0\} \in L(r)$. Since $\{\mathbf{x}(t) : t \ge 0\}$ is connected, we should have $\mathbf{x}(t) \in C$ for all $t \ge 0$.

This lemma implies that when the dynamic process, equation 4.2, is applied, the generated trajectory $\mathbf{x}(\cdot)$ of a data point $\mathbf{x}(0) = \mathbf{x}_0$ never exits the connected component of the level set in equation 4.1 where it starts. The next theorem, which serves as a theoretical basis of clustering based on the variances of predictive values, implies that the generated trajectory must terminate at one of the stable equilibrium points in the same connected component.

Theorem 2. *The trajectory of process 4.2 approaches one of the equilibrium points of the equation. In particular, almost all the trajectory approaches one of stable equilibrium points of equation 4.2.*

Proof. To prove this, we first show that every trajectory is bounded. For any given $\mathbf{x}_0 \in \Re^n$, we consider the set $L(r) = \{\mathbf{x} : \sigma^2(\mathbf{x}) \le r\}$ where $r = \sigma^2(\mathbf{x}_0)$. $\mathbf{x} \in L(r)$ implies that $\mathbf{k}_x^T \mathbf{C}^{-1} \mathbf{k}_x \ge \alpha$, where $\alpha = \mathbf{k}_{x_0}^T \mathbf{C}^{-1} \mathbf{k}_{x_0} > 0$. Since $\mathbf{C}^{-1}$ is positive definite, we can define a norm $\|\cdot\|_C$ on $\Re^N$ by $\|\mathbf{v}\|_C = \mathbf{v}^T \mathbf{C}^{-1} \mathbf{v}$ for a

vector $\mathbf{v} \in \Re^N$. Then by the equivalence of a norm, there exists a $c > 0$ such that

$$\|\mathbf{k}_x\|_2^2 := \mathbf{k}_x^T \mathbf{k}_x \geq c\|\mathbf{k}_x\|_C^2 = c\mathbf{k}_x^T \mathbf{C}^{-1}\mathbf{k}_x \geq c\alpha.$$

In other words, we have

$$\|\mathbf{k}_x\|_2^2 = \sum_{k=1}^{N} C(\mathbf{x}, \mathbf{x}_k)^2 \geq c\alpha.$$

From the form of $C(\mathbf{x}, \mathbf{x}_k)$, we should have that for some $M_1 > 0$, $\|\mathbf{x} - \mathbf{x}_k\| \leq M_1$ for all k. If we set $M_2 = \max_k \|\mathbf{x}_k\|$, then

$$\|\mathbf{x}\| \leq \|\mathbf{x} - \mathbf{x}_k\| + \|\mathbf{x}_k\| \leq M_1 + M_2,$$

which means that $\|\mathbf{x}\|$ is bounded. Therefore, the set $L(r) = \{\mathbf{x} : \sigma^2(\mathbf{x}) \leq r\}$ is bounded. Now let $\mathbf{x}(0) = \mathbf{x}_0$ be a point in $L(r)$ where $r = \sigma^2(\mathbf{x}_0)$. Since $L(r)$ is bounded and positive invariant by lemma 1, we should have that $\{\mathbf{x}(t) : t \geq 0\}$ is bounded.

Next, we show that every bounded trajectory converges to one of the equilibrium points. Since $\{\mathbf{x}(t) : t \geq 0\}$ is nonempty and compact and $\phi(t) = \sigma^2(\mathbf{x}(t))$ is a nonincreasing function of t, ϕ is bounded from below because f is continuous. Hence, $\phi(t)$ has a limit a as $t \to \infty$. Let $\omega(\mathbf{x}_0)$ be the ω-limit set of $\mathbf{x}_0$. Then for any $p \in \omega(\mathbf{x}_0)$, there exists a sequence $\{t_n\}$ with $t_n \to \infty$ and $\mathbf{x}(t_n) \to p$ as $n \to \infty$. By the continuity of g, $\sigma^2(p) = \lim_{n\to\infty} \sigma^2(\mathbf{x}(t_n)) = a$. Hence, $\sigma^2(p) = a$, for all $p \in \omega(\mathbf{x}_0)$. Since $\omega(\mathbf{x}_0)$ is an invariant set, for all $\mathbf{x} \in \omega(\mathbf{x}_0)$,

$$\frac{\partial}{\partial t}\sigma^2(\mathbf{x}) = -\|\nabla\sigma^2(\mathbf{x})\| = 0,$$

or equivalently, $F(\omega(\mathbf{x}_0)) = 0$. Since every bounded trajectory converges to its ω-limit set and $\mathbf{x}(t)$ is bounded, $\mathbf{x}(t)$ approaches $\omega(\mathbf{x}_0)$ as $t \to \infty$. Hence, it follows that every bounded trajectory of system 4.2 converges to one of the equilibrium points.

Therefore, the trajectory of $\mathbf{x}(0) = \mathbf{x}_0$ under the process 4.2 approaches one of its equilibrium points, say $\bar{\mathbf{x}}$. If $\bar{\mathbf{x}}$ is not a stable equilibrium point, then the region of attraction of $\bar{\mathbf{x}}$ has a dimension less than or equal to $n - 1$. Therefore, we have

$$\Re^n = \bigcup_{i}^{M} \overline{A(\mathbf{s}_i)}$$

where $\{\mathbf{s}_i : i = 1, \ldots, M\}$ is the set of the stable equilibrium points of system 4.2.

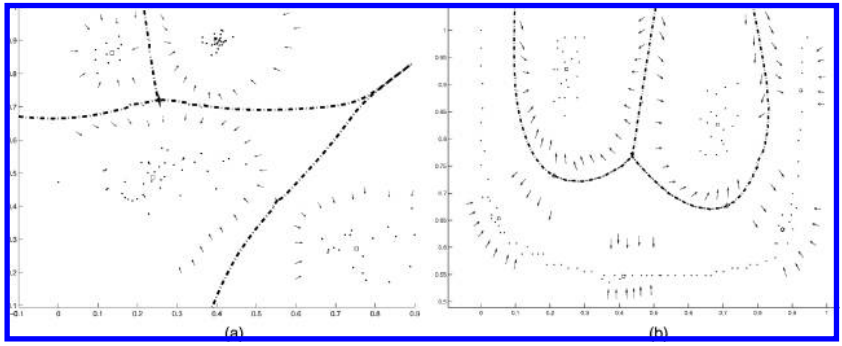

Figure 3: The partitioning property of the proposed clustering algorithm. The dashed lines represent the basin boundary separating each clusters, and the arrows represent the direction of the system trajectories.

By this theorem, each in-sample data point converges to one of the stable equilibrium points in the same connected component of that point when equation 4.2 is applied. Hence, we can assign a cluster label to each in-sample data point by identifying the cluster label of its corresponding stable equilibrium point (SEP) $\mathbf{s}_i$. Also this theorem enables us to partition the data space into a small number of separate clustered regions (i.e., basins of attraction), where each region is represented by the corresponding SEP as its candidate point. From a computational point of view, this result implies that we do not need to determine the exact basins to partition the whole data space. Since almost all the data points converge to one of the SEPs, we can identify a basin to which a data point belongs by locating a stable equilibrium point that the data point converges to. From a clustering point of view, this result implies that we can assign a cluster label to even out-of-sample data points by identifying the cluster label of its corresponding SEP, thereby extending the clustering ability to out-of-sample points, as does the dynamic-based support vector clustering in Lee and Lee (2006). This viewpoint is illustrated in Figure 3 and is described in the following corollary.

Corollary 1. *Let the level set of $\sigma^2(\cdot)$ be decomposed into several disjoint connected sets,*

$$L(r^*) := \{\mathbf{x} \in \Re^n : \sigma^2(\mathbf{x}) \leq r^*\} = C_1 \cup \cdots \cup C_p,$$

where the C_i, $i = 1, \ldots, p$ are disjoint connected sets corresponding to different clusters and p is the number of clusters determined by $\sigma^2(\cdot)$. Then almost all the (in-sample or out-of-the sample) data points in the input space approach one of the clusters, C_i, when process 4.2 is applied.

4.2 Clustering Algorithm. To assign cluster labels to the stable equilibrium points, we can employ a complete graph strategy as in Ben-Hur, Horn, Siegelmann, and Vapnik (2001) that is based on the following observation: given a pair of stable equilibrium points located in different clusters, a straight path connecting them must exit from the contour boundary. That is, there should be a segment of points $\mathbf{y}$ in the path such that $\sigma^2(\mathbf{y}) > r^*$. In this letter, we adopted a reduced complete graph strategy applied to the set of the stable equilibrium points as in Lee and Lee (2005) to improve labeling speed. When this strategy is applied, we can build an adjacency matrix A whose elements are given by

$$A_{ij} = \begin{cases} 1 & \text{if} \sigma^2(\mathbf{y}) > r^* \text{ for any point } \mathbf{y} \text{ on the segment between} \\ & \mathbf{s}_i \text{ and } \mathbf{s}_j; \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Clusters are then defined as the connected components of the graph induced by $\mathbf{A}$.

The conceptual procedure of the clustering algorithm can be summarized as follows:

Step 1: For given unlabeled training data $X = \{\mathbf{x}_i | i = 1, 2, \ldots, N\}$, construct a variance function, $\sigma^2(\cdot)$, in equation 3.1, and compute the level value $r^* = \max_{\mathbf{x}_i \in X} \sigma^2(\mathbf{x}_i)$.

Step 2: Using each data point as an initial value, we apply the process in equation 4.2 to find its corresponding stable equilibrium point, denoted by $\mathbf{s}_i$, $i = 1, \ldots, p$. Let X_i be the set of training data points that converge to a stable equilibrium point $\mathbf{s}_i$ for each $i = 1, \ldots, N$.

Step 3: For each pair of stable equilibrium point $\mathbf{s}_i$ and $\mathbf{s}_j$, $i, j = 1, \ldots, p$, define an adjacency matrix A with its elements:

$$A_{ij} = \begin{cases} 1 \text{ if } \sigma^2(\mathbf{y}) > r^* \text{ for any point } \mathbf{y} \text{ on the segment between} \\ \quad \mathbf{s}_i \text{ and } \mathbf{s}_j; \\ 0 \text{ otherwise.} \end{cases} \tag{4.4}$$

and assign the same cluster index to the stable equilibrium point in the same connected components of the graph induced by $\mathbf{A}$.

Step 4: For each $i = 1, ..., p$, assign the same cluster label to all the training data points in X_i.

Figure 4 illustrates the proposed algorithm. In this example, a sample data set with two concentric circles is given, as shown in Figure 4a. Step 1 of the algorithm applied to this data set provides a variance function, $\sigma^2 : \Re^2 \to \Re$, as shown in Figure 4b. After applying step 2, we obtain multiple stable equilibrium points corresponding to each data point, as shown in

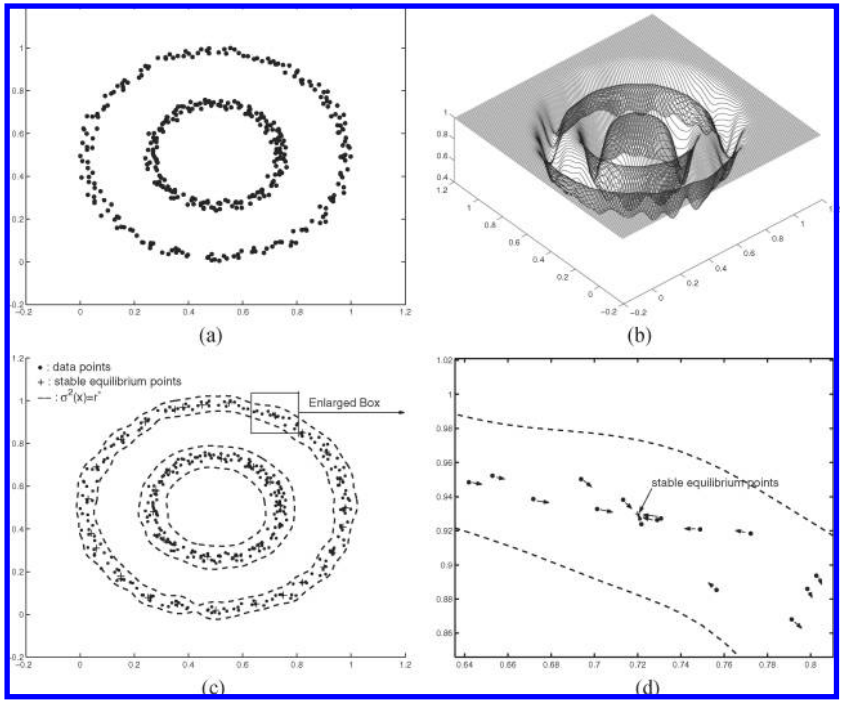

Figure 4: Clustering of the proposed algorithm applied to the circle-shaped data points.

Figure 4d. By applying step 3 to the obtained stable equilibrium points, we obtain an induced graph with two connected components with the cluster labels of the stable equilibrium points being identified according to its belonging. Finally by applying step 4, we assign the cluster label of each data point to the same cluster label of its corresponding stable equilibrium point.

Remarks. (1) Our method is methodologically similar to support vector clustering (SVC) (Ben-Hur et al., 2001; Lee & Lee, 2005) in that separate connected components of a level set of a support function (i.e., a variance function in our method and a trained kernel radius function in SVC) play an important role in identifying their clusters. Our method differs, however, from the SVC in that when the parameters are changed, the variance function in our method can be immediately obtained and changes continuously with parameters, whereas the trained kernel radius function in SVC should be retrained by invoking a quadratic programming solver and may change irregularly with parameters, which makes it hard to choose the appropriate

parameters of the SVC from its geometry. (2) Our method requires the matrix inversion computation, which has the complexity $O(n^3)$. We could speed up the computation by using the Nyström method (Williams & Seeger, 2001). (3) There could be two nearest-neighbor stable equilibrium points that are in the same cluster defined by the contours but with a joining straight line segment that crosses over the threshold contour. The graph-based clustering (Lee & Lee, 2005) would then separate these two stable equilibrium points and all the data points associated with them. In our experiments, this does not happen because of a rather densely located stable equilibrium points, but it could potentially happen more often in some practical high-dimensional cases and would require more additional cluster characterizations as in Lee & Lee (2006), which deserves a future research topic.

5 Simulation Results

We applied the proposed clustering method to the six two-dimensional data sets with highly complex shapes, which is shown in Figures 5a to 5f. The data set in Figure 5a is from Camastra and Verri (2005); the data sets in Figures 5b to 5d are from the spectral clustering site (http://www.ms.washington.edu/~spectral/), and the data sets in Figures 5e and 5f are synthetically made. Before the proposed method was used for clustering, all data sets were scaled so that the elements of the data points were between 0 and 1. Numerical gradients were used to find the stable equilibrium points. To check the connectedness between two stable equilibrium points, we checked 10 to 20 points along the straight line between two stable equilibrium points. (When the variance function value at any one among the checked 10 points is higher than $\sigma^2 = r^*$, we set $A_{ij} = 0$.)

Figures 5a′ to 5f′ show the clustering results of the propose method applied to the six data sets. For all six data sets, $v_0 = 1$ and $v_1 = 10$ were used. The solid lines represent the cluster boundaries and the cross mark represents stable equilibrium points. For the data set in Figure 5c, $l_i = 200$ was used. For the rest of the data sets, $l_i = 400$ was used. Separate regions enclosed by solid lines represent different clusters. The data sets in Figures 5a to 5f are not clustered properly by the conventional clustering methods such as K-means. The proposed clustering method clusters all six data sets perfectly.

We also applied the proposed clustering method to the three high-dimensional data sets. One is the 5-dimensional data set, another is the 10-dimensional data set, and the other is the 20-dimensional data set. The 5-dimensional data set has five clusters whose data points have gaussian distributions. The variance of all the five clusters is 1, and their means are $(7, 0, 0, 0, 0)$, $(0, 7, 0, 0, 0)$, $\ldots$, $(0, 0, 0, 0, 7)$. The 10-dimensional data set has five clusters whose data points have gaussian distributions. The variance of all five clusters is 1, and their means are $(7, 0, 0, 0, 0, 0, 0, \ldots, 0)$, $(0, 7, 0, 0, 0, 0, 0, \ldots, 0)$, $\ldots$, $(0, 0, 0, 0, 7, 0, 0, \ldots, 0)$. The 20-dimensional

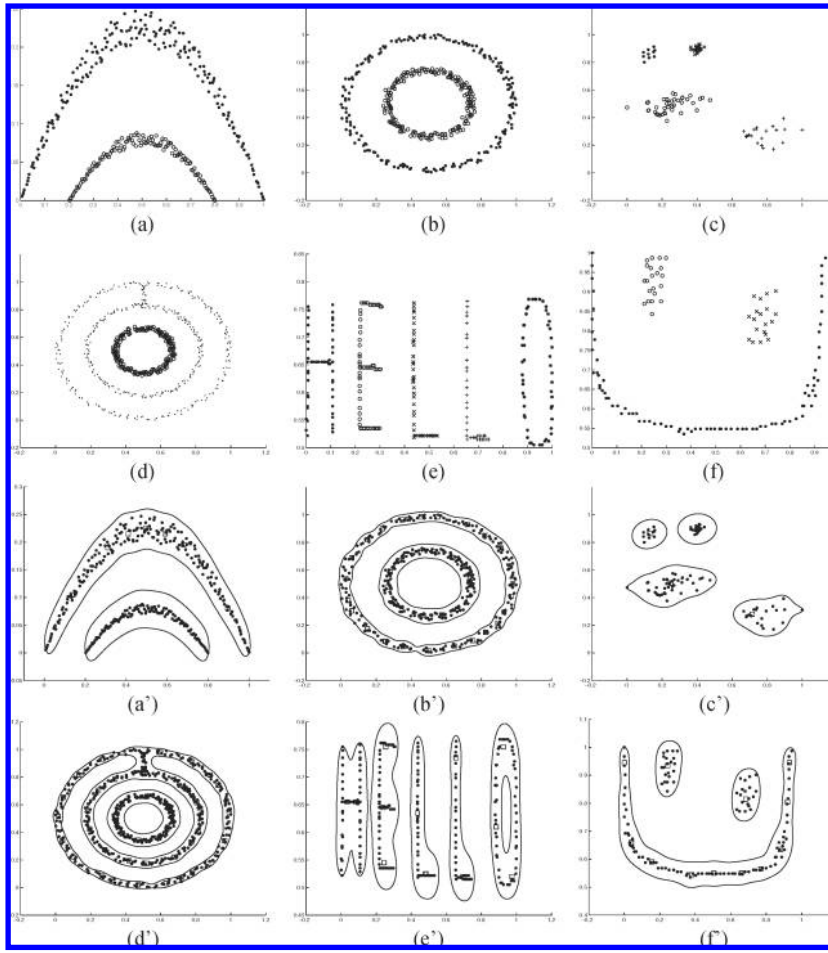

Figure 5: (a– f) Six original data sets. (a′–f′) Clustering results of the proposed method applied to the data set.

data set has five clusters whose data points have gaussian distributions. The variance of all five clusters is 1, and their means are $(5, 0, 0, 0, 0, 0, 0, \ldots, 0)$, $(0, 5, 0, 0, 0, 0, 0, \ldots, 0)$, $\ldots$, $(0, 0, 0, 0, 0, 5, 0, 0, \ldots, 0)$. For all three data sets, $v_0 = 1$ and $v_1 = 10$ were used. For the 5-dimensional data set, $l_i = 60$ was used. For the 10-dimensional data set, $l_i = 40$ was used. For the 20-dimensional data set, $l_i = 60$ was used. The proposed clustering method clusters all the three data sets perfectly.

We also compared our method with the spectral clustering method. We applied this method to all the data sets used in the experiments. With the

proper parameters, the spectral clustering method clusters all nine data sets perfectly. To measure the clustering performance quantitatively, we use the three measures such as the reference error rate RE, cluster error rate CE, and F Score $FScore$. Let $n_{r,c}$ be the number of data points that belong to reference cluster r and are assigned to cluster c by a clustering algorithm. Let n, n_r, and n_c be the number of all the data points, the number of the data points in the reference cluster r, and the number of the data points in the cluster c obtained by a clustering algorithm. Then the measures RE and CE are defined as

$$RE = 1 - \frac{\sum_r \max_c n_{r,c}}{n}, \quad CE = 1 - \frac{\sum_c \max_r n_{r,c}}{n}. \tag{5.1}$$

The reference error rate RE is zero when all the data points belonging to every cluster obtained by a clustering algorithm belong to one reference cluster. The cluster error rate CE is zero when all the data points belonging to every reference cluster belong to one cluster obtained by a clustering algorithm. When both RE and CE are zero, it means that the clusters exactly match the reference clusters.

Let $L_{r,c}$, $P_{r,c}$ be the recall and the precision defined as $\frac{n_{r,c}}{n_c}$, $\frac{n_{r,c}}{n_r}$, respectively. Then the FScore of the reference cluster r and cluster c is defined as

$$F_{r,c} = \frac{2R_{r,c}P_{r,c}}{R_{r,c} + P_{r,c}}. \tag{5.2}$$

The FScore of the reference cluster r is the maximum FScore value over all the clusters as

$$F_r = \max_c F_{r,c}. \tag{5.3}$$

The overall FScore is defined as

$$FScore = \sum_r \frac{n_r}{n} F_r. \tag{5.4}$$

In general, the higher the FScore, the better the clustering result. The FScore is 1 when the clusters exactly match the reference clusters. Both the spectral clustering method and our method show the reference error rate (RE) [0] and the cluster error rate (RE) [0] and Fscore ($FScore$) [1] for all the data sets tested. In terms of all the measures, such as reference error rate, cluster error rate, and FScore, our method gives comparable results to the spectral clustering method.

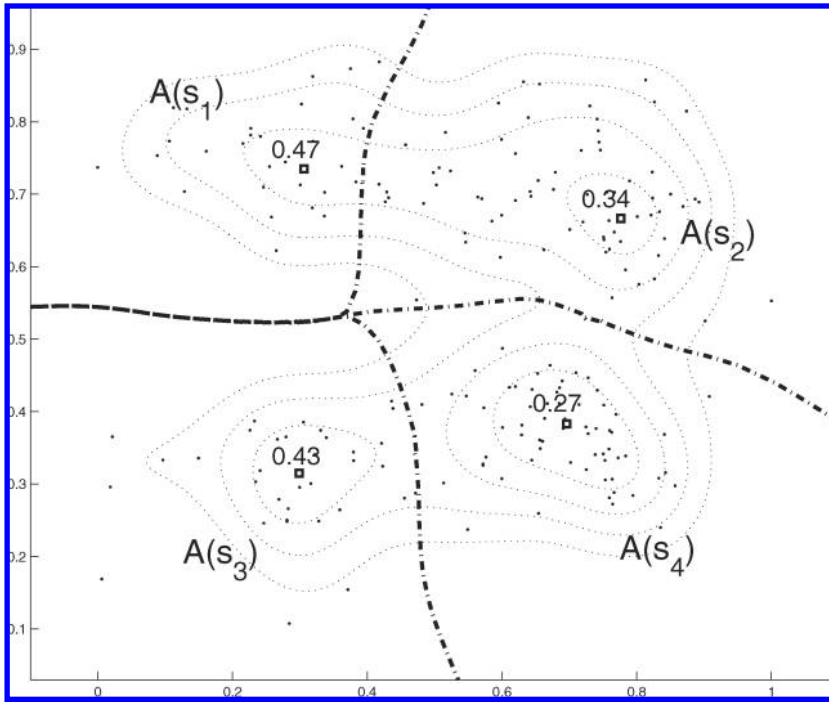

Figure 6: Clustering of the Leptograpsus crabs data set with overlapped clusters.

So far, we have shown the experiment results for the data sets with separate clusters. We next perform experiments for the Leptograpsus crab data set with overlapped clusters from the PRNN site. (The Leptograpsus crabs data set is from http://www.stats.ox.ac.uk/pub/PRNN/.) It consists of sizes of five parts of the crabs. Each crab is either male or female and has either of two color forms. There are 50 crabs of each sex of each of two color forms.

To visualize the experiment result on the plane, we reduced the 5-dimensional data set by principal component analysis and used second and third principal components. In Figure 6 we plotted the data points of the crabs data set, the local minima, and saddle points. The lines going through saddle points are potential decision boundaries. We have four basins of attraction: $A(\mathbf{s}_1)$, $A(\mathbf{s}_2)$, $A(\mathbf{s}_3)$, and $A(\mathbf{s}_4)$. The values near the local minima in Figure 6 are the variance function values on the local minima.

By controlling the cutting level, which is related to r^* in equation 4.1, we can control the number of clusters from 1 to 4. When the cutting level is greater than 0.27 and smaller than 0.34, the number of clusters becomes

1 ($A(\mathbf{s}_1) \cup A(\mathbf{s}_2) \cup A(\mathbf{s}_3) \cup A(\mathbf{s}_4)$). When the cutting level is greater than 0.34 and smaller than 0.43, the number of clusters becomes 2 ($A(\mathbf{s}_1) \cup A(\mathbf{s}_2)$ and $A(\mathbf{s}_3) \cup A(\mathbf{s}_4)$). When the cutting level is greater than 0.43 and smaller than 0.47, the number of clusters becomes 3 ($A(\mathbf{s}_1) \cup A(\mathbf{s}_2)$, $A(\mathbf{s}_3)$ and $A(\mathbf{s}_4)$. When the cutting level is greater than 0.47, the number of clusters becomes 4 ($A(\mathbf{s}_1)$, $A(\mathbf{s}_2)$, $A(\mathbf{s}_3)$ and $A(\mathbf{s}_4)$). This is how we can cluster the data set with the overlapped clusters and can control the number of clusters.

6 Conclusion

We have proposed a clustering algorithm with gaussian process models. The variance function of GPR learned from a training data is shown to represent an estimate of the support of a probability density function. The variance function is then applied to construct a set of contours that enclose the data points, which correspond to cluster boundaries. A dynamic process associated with the variance function is built and applied to cluster labeling of the data points. The simulation results have shown that the method can detect clusters with arbitrarily complex shapes and high-dimensional clusters. We also showed how we can cluster the data set with overlapped clusters. One of the main implications of this letter is that a gaussian processes regression model providing predictive distribution with nondegenerate variance functions can be applied to develop a clustering algorithm with the aid of dynamical systems machinery, which shows an interesting connection between a Bayesian regression task and a clustering task.

In the simulation results, we had to have a proper hyperparameter to get a proper result. It is important to estimate a proper hyperparameter. Based on our experiment on the crabs data set, where we showed how to control the number of clusters, we could try a hyperparameter-robust approach with the number of clusters given. These are potential future research topics.

Acknowledgments

This work was supported partially by the Korea Research Foundation under the Grant number KRF-2005-041-D00708 and partially by the KOSEF under the grant number R01-2005-000-10746-0, and partially by the BK21 Research Center for Intelligent Mobile Software at Yonsei University.

References

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research*, *2*, 125–137.

Camastra, F., & Verri, A. (2005). A novel kernel method for clustering. In *IEEE Transactions on PAMI* (pp. 801–805). Piscataway, NJ: IEEE Press.

Csato, L. (2002). *Gaussian processes—Iterative sparse approximation*. Unpublished doctoral dissertation, Aston University.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. New York: Wiley.

Gibbs, M., & MacKay, D. J. C. (1997). *Efficient implementation of gaussian processes*. Available online at http://citeseer.nj.nec.com/6641.html.

Gibbs, M., & MacKay, D. J. C. (2000). Variational gaussian process classifiers. *IEEE Transactions on NN, 11*(6), 1458.

Guckenheimer, J., & Homes, P. (1986). *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. New York: Springer.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 11*(3), 564–323.

Kim, H.-C., & Lee, J. (2006). Pseudo-density estimation for clustering with gaussian processes. *Lecture Notes in Computer Science, 3971*, 1238–1243.

Lee, J., & Chiang, H.-D. (2004). A dynamical trajectory-based methodology for systematically computing multiple optimal solutions of general nonlinear programming problems. *IEEE Transactions on Automatic Control, 49*(6), 888–899.

Lee, J., & Lee, D. (2005). An improved cluster labelling method for support vector clustering. *IEEE Transactions on PAMI, 27*(3), 461–464.

Lee, J., & Lee, D. (2006). Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(11), 1869–1874.

Minka, T. (2001). *A family of algorithms for approximate Bayesian inference*. Unpublished doctoral dissertation, Massachusetts Institute of Technology.

Neal, R. M. (1996). *Bayesian learning for neural networks: Lecture notes in statistics no. 113*. New York: Springer-Verlag.

Neal, R. (1998). Regression and classification using gaussian process priors. *Bayesian Statistics 6*, 465–501.

O'Hagan, A., (1978). On curve fitting and optimal design for regression. *Journal of the Royal Statistical Society, 40*, 1–42.

Opper, M., & Winther, O. (2000). Gaussian processes for classification: Mean field algorithms. *Neural Computation, 12*, 2655–2684.

Rasmussen, C. E. (1996). *Evaluation of gaussian processes and other methods for non-linear regression*. Unpublished doctoral dissertation, University of Toronto.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Schmidt, D. (2000). Continuous probability distributions from finite data. *Phys. Rev. E, 61*(2), 1052–1055.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Scholkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (1999). *Estimating the support of a high-dimensional distribution* (Tech. Rep. 99-87). Redmond, WA: Microsoft Research. Available online at http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-99-87.

Williams, C. K. I., & Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on PAMI, 20*, 1342–1351.

Williams, C. K. I., & Rasmussen, C. E. (1995). Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural in formation processing systems,* 8. Cambridge, MA: MIT Press.

Williams, C. K. I., & Seeger, M. (2001). Using the nystrom method to speed up kernel machines. In T. G. Diettrich, T. K. Leen, & V. Tresp (Eds.), *Advances in neural information processing systems, 13*. Cambridge, MA: MIT Press.

Received January 17, 2006; accepted February 2, 2007.

This article has been cited by:

1. Daewon Lee, Kyu-Hwan Jung, Jaewook Lee. 2009. Constructing Sparse Kernel Machines Using Attractors. *IEEE Transactions on Neural Networks* **20**:4, 721-729. [CrossRef]