Action Recognition Based on Kernel Machine Encoding Qualitative Prior Knowledge*

Masamichi Shimosaka Taketoshi Mori Tatsuya Harada Tomomasa Sato
Graduate School of Information Science and Technology
The University of Tokyo, Tokyo, JAPAN
{simosaka, tmori, harada, tomo}@ ics.t.u-tokyo.ac.jp

Abstract – *This paper proposes a recognition algorithm based on kernel classifier for human daily life action such as walking or lying down. The advantage of the proposed algorithm is to realize implant of qualitative human knowledge and robust recognition accuracy at the same time. The main features of the presented method are: 1)utilizing Gaussian process with latent variables for relation between recognized labels and input human motion, 2)in order to embed prior knowledge for proper recognition of novel motion dissimilar to the learned motion data, assigning probabilistic labels to virtual human motions generated in "Sparse" area of input motion feature space, 3)learning parameters of classifier by real human motion with labels and the virtual motions in Bayesian perspective. The result of cross-validation like experiment shows that the accuracy of the proposed method is as good as support vector classification based recognition methods. It is also shown that the proposed method can recognize some novel motion fit into human common sense even when the classifiers without embedded knowledge fails to recognize it.*

Keywords: Behavior Recognition, Knowledge Incorporation, Kernel Methods, Bayesian Statistics, Motion Capture

1 Introduction

Recognizing human daily life action is one of essential factors to realize smooth communication between intelligent systems and human. It is also a key technical element to realize analysis and surveillance of human activity with intelligent system for human life assistance. The authors have built action recognition systems for daily life action, such as walking and sitting [1], [2]. Roughly speaking, the recognition algorithms we developed are divided into two categories. The former type of the recognition algorithms has advantage in embedding prior knowledge of human action. The latter type is based on statistical methodology, especially memory based approach.

In the former type recognition algorithms, a designer of the system describes discriminant rules based on human knowledge about action, and implements the rules into the system. This approach has advantage in embedding qualitative prior knowledge of action but has difficulty in optimizing parameters of the rules.

In the latter type recognition algorithms, motion features relevant to the target action name are selected by human, then the system classifies the selected motion features with kernel technique [3]. In case of walking, forward motion of hips is selected as one of the relevance motion features of walking. This approach has advantage that it is easier to make system robust than the former type, but has disadvantage in not fully making use of the human qualitative prior knowledge about action. For example, if some action is described as "head to be high", the former technique recognizes motion with concept of "how high the head is", on the other hand, the latter technique utilizes height of head without concept of "head is high or not".

When volume of training data is too small and the distribution of the data is not properly scattered into motion feature space, it is hard to make the classifier understand the concept of description about action. This sometimes invokes that the latter type algorithms cannot give assurance of output proper recognition result for novelty motion dissimilar to the training data because of small diversity in the training motion data. In this context, the proper recognition result means the result fitting into human common sense.

This problem can be resolved only if the training motion data have very large size and diversity. But the preparation of such data demands very laborious work. In other pattern recognition community, some previous researches already targeted compensation technique for the poor data problem. In area of computer vision, proper prior knowledge that categorization of images remains unchanged after the images are rotated, resized, translated is often used. Schölkopf et al. [4] used virtual images rotated and rescaled from real image to enhance the performance of the system. Romdhani et al. [5] also used virtual samples by changing illumination of real images for robust face detection. In area of natural language processing, bioinformatics, and speech processing where the data can be obtained very easily, semi supervised learning technique is often used to enhance the performance because it is very hard to prepare labels of all the data. Inoue et al. [6] and Nigam et al. [7], combines little amount of labeled data with large size of unlabeled data.

*0-7803-8566-7/04/$20.00 © 2004 IEEE.

Unfortunately, it is not easy to apply the techniques mentioned above for the domain of action recognition, because it is difficult to make definition of invariant motion feature in action recognition itself, and it is also very hard to prepare large size of unlabeled data. In contrast to the area of natural speech processing, the preparation of large diversity of motion data is very hard work, because measuring human motion only for a short time makes an actor exhausted.

Therefore, the main contribution of this paper is to build recognition algorithm to prevent the recognition system from nonguarantteed performance for novelty motion dissimilar to the training motion data. The proposed recognition algorithm basically integrates both the advantages of the previous two techniques we developed, the qualitative knowledge-based and the statistical approach. This is designed not only to assure the system performance for input motion similar to the training data, but also to ensure the proper recognition result for "unseen" alien motion.

The proposed algorithm pays attention to the "sparse area" where the training data rarely happen. The virtual data generation technique is utilized in the sparse area and labeling task is automatically done probabilistically. The proposed algorithm combines the real training motion with labels and virtual motion with probabilistic labels in Bayesian perspective, and optimizes the performance.

Next section introduces the configuration of our recognition system. Then, the formulation of probabilistic model to incorporate qualitative prior knowledge with kernel technique is introduced. Section 4 explains the automatic algorithm to generate and implant virtual motion. Section 5 shows the validity of our approach. Finally, we give some brief conclusions.

2 Daily life action recognition system: HARS

2.1 Input and output

The input of the system is time series of human motion. Concretely speaking, our system utilizes articulated human body motion whose three-dimensional configuration is recovered. The output of our system contains some action names in synchronized with frame of the input motion.

2.2 Configuration of HARS

Figure 1 shows the processing flow and the configuration of the system. In order to realize the simultaneous recognition, the system contains multiple parallel recognition processes, each of which is assigned to the recognizer for one certain action. This primitive process is called as "Action Element Recognizer(AER)". One AER runs in parallel simultaneously with the others. The system collects the results of all AERs, and outputs the results of each recognition process per frame. An AER which recognizes walking discriminates whether someone is walking, for example.

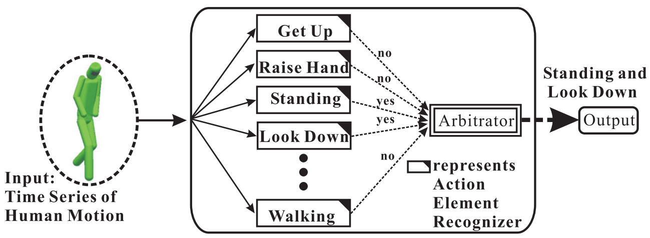

Figure 1: Configuration of recognition system

2.3 Kernel based action element recognizer

Time series of human motion is utilized as input of each AER. An AER outputs result of classification whether the assigned action occurs or not per frame in synchronized with the input motion. One category is named as "yes" which represents that the assigned action clearly occurs. Another is named as "no" which represents that the opposite meaning of "yes". Thus, the AER is equivalent to a binary classifier.

2.3.1 Classification rule in AER

This section explains formulation of classification rule of AER as binary kernel classifier. We denote by $\boldsymbol{x}$ the time series of input motion. $\{\boldsymbol{x}_i, y_i\}_{i=1}^{l}$ are the input-output pairs in total l frames, where $\boldsymbol{x}_i$ represents i th frame temporal template motion and its corresponding reference binary(e.g. "yes" or "no") signal by y_i. We can write by α_i the coefficient whose value is proportional to importance of the motion template. Similarity value between the input motion and one template motion is represented by Kernel $K(\boldsymbol{x}, \boldsymbol{x}_i)$. The mapping between the input and the output of the binary classifier in the AER f can be written as

$$f(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right), \tag{1}$$

where b depicts offset and the function $\mathrm{sgn}(\cdot)$ is a step function where the relation of input-output is represented as

$$\mathrm{sgn}(t) = \begin{cases} +1 & \text{if} \quad t > 0 \\ -1 & \text{if} \quad \text{otherwise} \end{cases}.$$

Learning process in the binary classifier of AER tunes the coefficients($\boldsymbol{\alpha}, b$) from the training data. The detailed explanation of the learning process is mentioned in section 3 and section 4.

2.3.2 Deriving kernels: combination of kernel values per expression

The proposed method derives the kernel value in the classifier as the products of all the kernel values corresponding to the similarity in each expression about target action. Concretely speaking, the kernel value which corresponds to the similarities in j th expression $K_j(\cdot, \cdot)$ can be written as

$$K_j(\varphi_j(\boldsymbol{x}_i^{(j)}), \varphi_j(\boldsymbol{x}^{(j)})),$$

where $\boldsymbol{x}^{(j)}$ denotes the selected input motion attribute, such as height of head, and bentness of hips in the j th expression, and $\varphi_j(\cdot)$ represents the converter from the selected attribute to the input feature.

When the numbers of the expressions in the target action is d, the kernel value in the target action $K(\cdot,\cdot)$ can be written as

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \prod_{j=1}^{d} K_j(\varphi_j(\boldsymbol{x}_i^{(j)}), \varphi_j(\boldsymbol{x}^{(j)})).$$

3 Statistical learning incorporating prior knowledge in kernel classifier

3.1 Classifying and learning with probabilistic models

3.1.1 Classification rule of input motion

The classification process is based on posterior conditional probability $p(y|\boldsymbol{x}, D, X_v, \mathcal{H})$, where $\boldsymbol{x}$ depicts input motion to be classified, $y = \pm 1$ represents the code as recognition result. We write by D dataset of real input output pairs. X_v denotes dataset of the virtual motion and $\mathcal{H}$ represents qualitative prior knowledge about action. Virtual motion means the artificial motion data generated and projected into the motion input feature space. $\mathcal{H}$ works as label assigner for virtual motion. Because the code of the label is binary, the classification rule can be derived as

$$\hat{y} = \mathrm{sgn}\left(\ln \frac{p(y=+1|\boldsymbol{x}, D, X_v, \mathcal{H})}{p(y=-1|\boldsymbol{x}, D, X_v, \mathcal{H})}\right), \qquad (2)$$

where $\hat{y}$ denotes the estimated label.

3.1.2 Formulation of distribution

Before detailed explanation of the posterior distribution used in this paper, some variables used in the distribution are introduced. Dataset $X = \{\boldsymbol{x}_i\}_{i=1}^n$ contains motion data in total n frames. In related to X, $Y = \{y_i\}_{i=1}^n$ represents the collection of labels. D is equivalent to $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$. Dataset $X_v = \{\tilde{\boldsymbol{x}}_j\}_{j=1}^m$ contains virtual motion in total m frames. In related to X_v, $Y_v = \{\tilde{y}_j\}_{j=1}^m$ denotes the collection of probabilistic labels for virtual motion. The variable $z \in \mathbb{R}$ depicts a latent variable corresponding to $\boldsymbol{x}$. We write by $Z = \{z_i\}_{i=1}^n$ dataset of latent variables corresponding to X. $Z_v = \{\tilde{z}_j\}_{j=1}^m$ represents latent variables corresponding to X_v. These latent variables serve as indicators of the binary output codes. The poster distribution for classification rule can be derived as

$$\begin{aligned} &p(y|\boldsymbol{x}, D, X_v, \mathcal{H}) \\ &= \int p(y, Y_v, z, Z, Z_v|\boldsymbol{x}, D, X_v, \mathcal{H}) dY_v dz dZ dZ_v \\ &= \mathcal{E}_{p(Y_v, z, Z, Z_v|\boldsymbol{x}, D, X_v, \mathcal{H})}[p(y|z)], \end{aligned} \qquad (3)$$

where operation written by $\mathcal{E}_q[f]$ represents expectation of function f with distribution q. Reason why we adopt latent variables is that this decomposes complicated probabilistic models to combination of simple distributions. This factorization technique is used in the generic Gaussian process classification [8]. Roughly speaking, output codes and input motions is independent if latent variables are conditioned. The conditional distribution of latent variables z, Z, Z_v by input motion $\boldsymbol{x}, X, X_v$ is described as Gaussian process. **Eq.**(3) can be decomposed as

$$\begin{aligned} &p(Y_v, z, Z, Z_v|\boldsymbol{x}, D, X_v, \mathcal{H}) \\ &\quad \propto p(Y|Z)p(Y_v|Z_v, X_v, \mathcal{H})p(z, Z, Z_v|\boldsymbol{x}, X, X_v). \end{aligned} \quad (4)$$

The graphical model for the proposed probabilistic modeling is shown in Figure 2. Alphabets in circle and square represent observed and latent variables, respectively. Alphabets in doubly square represent are given priori by human knowledge. Especially $\mathcal{S}$ represents the virtual motion sampler for compensating the sparse area of training motion. The detailed explanation of $\mathcal{S}$ is mentioned in section 4. $\mathcal{H}$ represents qualitative prior knowledge that serves as a part of label assigners for virtual motion. Each distribution which appeared right hand side of **Eq.**(4) is defined as follows.

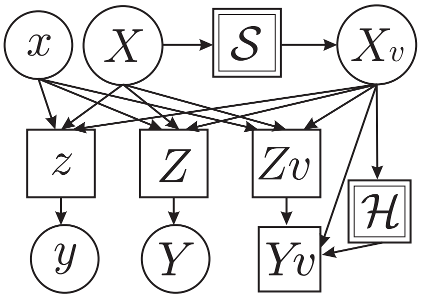

Figure 2: Graphical model of the *Gaussian process classifier with embedding prior knowledge*

Conditional distribution of output codes

- Related to real motion
  We assume samples of Z, Y generate in i.i.d. In other words, the conditional distribution of Y with Z can be decomposed as

  $$p(Y|Z) = \prod_{i=1}^{n} \mathrm{sig}(y_i, z_i),$$

  where operation written by $\mathrm{sig}(y, z)$ represents

  $$1/(1 + \exp(-\beta^{-1} yz)),$$

  and β represents any positive value.

- Related to virtual motion
  We also assume the samples of Y_v, Z_v, X_v can be acquired as i.i.d. Then the distribution can be decomposed as

  $$p(Y_v|Z_v, X_v, \mathcal{H}) = \prod_{j=1}^{m} p(\tilde{y}_j|\tilde{z}_j, \tilde{\boldsymbol{x}}_j, \mathcal{H}).$$

We design this distribution of probabilistic labels as product of belief distribution by qualitative prior knowledge $\mathcal{H}$ and distribution of sigmoid shapes $\mathrm{sig}(\cdot,\cdot)$. The embedding of prior knowledge intervenes from the belief distribution $p(\tilde{y}_j|\tilde{\boldsymbol{x}}_j,\mathcal{H})$. Thus the distribution of probabilistic label for virtual motion using prior knowledge is defined as

$$p(\tilde{y}_j|\tilde{z}_j,\tilde{\boldsymbol{x}}_j,\mathcal{H}) \propto p(\tilde{y}_j|\tilde{\boldsymbol{x}}_j,\mathcal{H})\mathrm{sig}(\tilde{y}_j,\tilde{z}_j).$$

The distribution for embedding of prior knowledge is defined as

$$p(\tilde{y}_j|\tilde{\boldsymbol{x}}_j,\mathcal{H}) = h(\tilde{\boldsymbol{x}}_j)^{\frac{1+\tilde{y}_j}{2}}\left(1-h(\tilde{\boldsymbol{x}}_j)\right)^{\frac{1-\tilde{y}_j}{2}},$$

where function $h(\tilde{\boldsymbol{x}})$ denotes belief of action occurring and is designed with prior knowledge about action. The range of output of $h(\cdot)$ is $0 \sim 1$. The brief introduction of design method of h is mentioned in section 3.2.

Conditional distribution of latent variables

As mentioned above, Gaussian process is used as the distribution for input and latent variables

$$p(z,Z,Z_v|\boldsymbol{x},X,X_v) = \mathcal{N}(\boldsymbol{0},G+\sigma^2 I),$$

where $\mathcal{N}(\boldsymbol{\mu},\Sigma)$ denotes Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. G represents gram matrix [3] as

$$\begin{aligned} G &= \left(\begin{array}{ccc} \multicolumn{2}{c}{\Psi} & \begin{array}{c}\boldsymbol{k}_r\\ \boldsymbol{k}_v\end{array} \\ \boldsymbol{k}_r^t & \boldsymbol{k}_v^t & K(\boldsymbol{x},\boldsymbol{x}) \end{array}\right), \begin{array}{l} \{\boldsymbol{k}_r\}_i = K(\boldsymbol{x}_i,\boldsymbol{x}) \\ \{\boldsymbol{k}_v\}_i = K(\tilde{\boldsymbol{x}}_i,\boldsymbol{x}) \end{array} \\ \Psi &= \left(\begin{array}{cc} \boldsymbol{K}_{rr} & \boldsymbol{K}_{rv} \\ \boldsymbol{K}_{rv}^t & \boldsymbol{K}_{vv} \end{array}\right), \begin{array}{l} \{\boldsymbol{K}_{rr}\}_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ \{\boldsymbol{K}_{vv}\}_{ij} = K(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{x}}_j) \\ \{\boldsymbol{K}_{rv}\}_{ij} = K(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_j) \end{array}. \end{aligned}$$

The positive parameter σ represents a scatter parameter that keeps the covariance matrix positive definitive.

3.1.3 Learning with optimized latent variables

Learning to determine optimized parameters can be derived as follows. Because we use distribution of output label with latent variables as $p(y|z) = \mathrm{sig}(y,z)$, the classification rule based on posterior probability is equal to

$$z\left(p(y=+1|\boldsymbol{x},D,X_v,\mathcal{H})-0.5\right) > 0.$$

Herbrich [9] proved that the optimized latent variable is derived from

$$\hat{z} = \left(\hat{Z}^t,\hat{Z}_v^t\right)\Psi^{-1}\left(\boldsymbol{k}_r,\boldsymbol{k}_v\right),$$

where $\hat{Z},\hat{Z}_v$ denotes the optimized latent variables. When function $g(\cdot)$ can be written with weighing parameters

$$\hat{\boldsymbol{\nu}}^t = \left(\hat{Z}^t,\hat{Z}_v^t\right)\Psi^{-1}$$

as

$$g(\boldsymbol{x}) = \sum_{i=1}^{n}\hat{\nu}_i K(\boldsymbol{x}_i,\boldsymbol{x}) + \sum_{j=1}^{m}\hat{\nu}_{j+n}K(\tilde{\boldsymbol{x}}_j,\boldsymbol{x}), \quad (5)$$

the classification rule in **Eq.**(2) is equivalent to $\hat{y} = \mathrm{sgn}(g(\boldsymbol{x}))$. This clearly specifies the proposed method is a kind of kernel classification algorithm defined as **Eq.**(1). The derivation of latent variables $\hat{Z},\hat{Z}_v$, the essential parameters for estimating weighting coefficients $\hat{\boldsymbol{\nu}}$, is based on the right hand side of **Eq.**(4). Markov chain Monte Carlo [10] or Laplace's approximation [8] can be applied for estimating $\hat{Z},\hat{Z}_v$. This paper uses latter technique because of its efficiency. Laplace's method approximates the distribution at peak point with second order Taylor's expansion. The Newton-Raphson technique is applied for searching the peak point. We call the proposed Gaussian-based probabilistic model incorporating qualitative prior knowledge via virtual motion as VPK_GPC(embedding Virtual Motion with Prior Knowledge Gaussian Process Classifier). VPK_GPC allows any parameterization of prior knowledge.

3.2 Belief computation

This subsection describes the design method of belief computation $h(\cdot)$ for virtual motion $\tilde{\boldsymbol{x}}$. The design methodology in this paper is based on Mori et al.'s technique [1]. It utilizes fuzzy inference technique. The inference rules are derived from the description about action hand-written by human. In this technique, the belief is the production of each fuzzy membership function per description. Each fuzzy membership function output the value from $0 \sim 1$ as the belief that the target action occurs.

For example, we show the procedure of belief computation for standing. When standing can be described as "head to be high" and "hips are not bent", then the height of head and angle between upper and lower body are selected as the gazed relevance motion features. As a fuzzy membership function, the belief "head to be high" corresponding to height of head is designed as monotonic increase function. The belief "hips not to be bent" is also described by monotonic decrease function. The calculation flow of belief computation is shown in Figure 3.

4 Generating / implanting virtual motion

Because virtual motions are assumed to be given priori and used in VPK_GPC, it is important to design how to generate and implant virtual motion into VPK_GPC. This section explains the procedure of generating and implanting virtual motion data. As mentioned above, the motivation using virtual motion is to compensate sparseness of the training data. In order to build proper virtual motion technique, 1) definition of the sparse area, 2) selection and implant of the generated virtual motion in the sparse area are important problems.

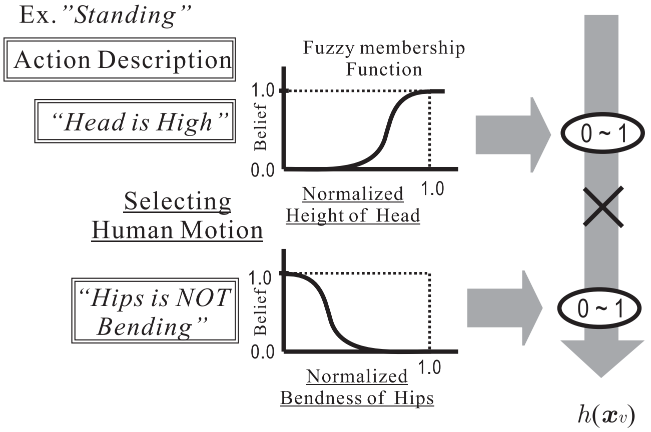

Figure 3: Processing flow of calculating belief with fuzzy membership function as prior knowledge

4.1 Definition of "*sparse*" feature space

In machine learning community, there are some interesting techniques similar to the concept of sparseness of the data. One is outlier detection framework [11]. Another is one-class classification framework [12] which classifies the input datum whether the known category draws or not.

In this paper, the sparse concept is set as extension to outlier detection framework, then we define the sparse area with probabilistic density as follows. As first step of defining the sparseness, the density of the training data $X = \{\boldsymbol{x}_i\}_{i=1}^{n}$ is estimated with mixture of Gaussian distributions as $\hat{p}_{\mathcal{N}}(\boldsymbol{x})$. The estimation procedure is done with Expectation Maximization(EM) algorithm [13]. Then we define the sparse feature space S as

$$\{\forall \boldsymbol{x}_u \in S | \ln \hat{p}_{\mathcal{N}}(\boldsymbol{x}_u) \leq r\}, \tag{6}$$

where $\boldsymbol{x}_u$ denotes any input motion data. Threshold r is defined as

$$r = \mathcal{E}_{\hat{p}_{\mathcal{N}}}\left[\ln \hat{p}_{\mathcal{N}}(\boldsymbol{x})\right] - k \cdot \mathcal{V}_{\hat{p}_{\mathcal{N}}}\left[\ln \hat{p}_{\mathcal{N}}(\boldsymbol{x})\right], \tag{7}$$

where $\mathcal{V}_q[f]$ represent variation of f with distribution q, positive parameter k denotes the adjustable parameter for deciding the sparseness. Tendency that the new motion is discriminated as motion in sparse feature space will increase when we set k small.

4.2 Strategy for implanting virtual motion

This subsection introduces how to generate and implant virtual motion in the sparse area mentioned in 4.1. We think virtual motion worth to be implanted is based on correction or clarity of belief computed with prior knowledge. We also have to care computational efficiency of the implanting. As consideration of clarity of belief, $\boldsymbol{s}$, the candidate virtual motion to implant, must satisfy

$$\min\{h(\boldsymbol{s}), 1 - h(\boldsymbol{s})\} < c, \tag{8}$$

where $c > 0$ denotes the adjustable parameters to determine whether $h(\boldsymbol{s})$ is clear or not.

In order to realize efficient implanting of virtual motion, we adopt the following strategies. In this context, the computational efficiency means that the correction power into VPK_GPC via virtual motion. In other words, the correction power means the power of repair of improper classification boundary of recognizer in sparse area via a virtual motion with prior knowledge. Thus we pay attention to the feature space that satisfies following properties.

- Unclarity of posterior probability is large.
- Difference between posterior probability and belief computation by prior knowledge is large.

In other words, the former area represents neighborhood of classification boundary and the latter means the recognizer output misclassified result which does not fit into human common sense. As an implementation step, we define the unclarity with entropy concept as

$$\mathcal{A}(\boldsymbol{s}) = \sum_{y=\pm 1} -p(y|f(\boldsymbol{s})) \ln p(y|f(\boldsymbol{s})), \tag{9}$$

where $\boldsymbol{s}$ denotes candidate of virtual motion to be implanted, function f represent the function of AER classifier appeared in **Eq.**(1), and we set $p(y|f(\boldsymbol{s}))$ as $\mathrm{sig}(y, f(\boldsymbol{s}))$. We also define the difference $\mathcal{D}$ with Kullback-Leibler Divergence as

$$\mathcal{D} = \frac{\mathrm{KL}(h||q) + \mathrm{KL}(q||h)}{2}, \tag{10}$$

where distribution q represents $\mathrm{sig}(y, f(\boldsymbol{s}))$.

4.3 Sequential learning algorithm

Inconveniently, some recognition classifier including VPK_GPC is assumed to be priori given in the strategy mentioned in 4.2. Thus we create a sequential learning algorithm in order to build VPK_GPC with proper virtual motion with probabilistic label automatically.

Concretely speaking, the sequential algorithm starts with null virtual motion and builds VPK_GPC without virtual motion, i.e. $X_v = \phi$. Then, the algorithm automatically generates and selects virtual motion as mentioned in 4.2. Next, the algorithm groups together the selected virtual motion into X_v, then the algorithm refines VPK_GPC classifier. This procedure is iteratively done until the change of classification boundary is small or the volume of sparse feature space to be worth generating proper virtual motion is very small. **Table** 1 shows the procedure of the sequential learning algorithm. We set the range of the parameters in **Table** 1 as γ, ϵ is $0 < \gamma, \epsilon \ll 1$.

5 Performance evaluation

We validate the proposed algorithm in two perspective: 1) the recognition performance for motion similar to the training data, 2) proper recognition result for novel motion dissimilar to the training data. As the former aspect, we evaluate

Table 1: Procedure of sequential learning algorithm

	Setting: Motion Dataset with labels D, Kernel K, Prior Knowledge $h(\cdot)$
0	Setting virtual motion data set X_v as null, then normalizing and estimating density of the real motion data set$X$
1	Generating f in **Eq.**(5) by VPK_GPC after integrating D and X_v
2	Generating uniform random vectors $\{\boldsymbol{s}\}_{i=1}^{n}$: in input space
3	Selecting virtual samples in *"Sparse"* area(**Eq.**(6), **Eq.**(7)) as $\boldsymbol{s}_s$ from $\boldsymbol{s}$
4	Selecting $\boldsymbol{s}_c$ from $\boldsymbol{s}_s$ which satisfies **Eq.**(8)
5	Selecting $\boldsymbol{s}_v$ from $\boldsymbol{s}_c$ which satisfy $\mathcal{A}(\boldsymbol{s}_c) > a$, $\mathcal{D}(h(\boldsymbol{s}_c), p(y \mid f(\boldsymbol{s}_c))) > e, (a, e > 0)$
6	Terminating if num. of $\boldsymbol{s}_v$ is less than γn
7	Selecting ϵn samples from $\boldsymbol{s}_v$ based on $\mathcal{A}, \mathcal{D}$
8	Adding selected samples to X_v and Returning to **1**

the interference of the VPK_GPC by virtual motion and prior knowledge. As the latter aspect, we evaluate advantage of incorporating prior knowledge via virtual motion by observing behavior of the proposed recognizer for novel motion. In the following sentences, motion used in the experiments are explained.

5.1 Motion used in the experiments

In the experiments of this paper, we use ICS action database [14] containing 25 actions such as lying, sitting, or running. This is the collections of motion data with reference action name labels. The 25 actions are stored in the database at least five times. Each motion data in this database contains a motion capture data and its reference files per each target action. One reference file contains human's judgment for the a certain assigned action per frame by three degrees ("yes", "neutral" and "no"). The label "neutral" represent ambiguous recognition result by human. In this experiment, we transform the label as $y = +1$ if the reference in the database is labeled as "yes", else $y = -1$.

The specification of the motion and labels of the database is listed as **Table** 2. BVH [15], the format of motion capture data, is a de-facto standard in computer graphics by the Biovision Corporation. A BVH file contains the structure of a human as a linked joint model(figure) and the motion of the figure per frame.

Table 2: Specification of ICS action database

D.O.F.	36(11 Articulation)
Measurement	Magnetic(Ascention MotionStar) Posture and Position
Actor	a male in 20s
Format	Biovision BVH and its label
Num. of Files	125 (Avg. 3.2[sec.])

5.2 Performance for motion similar to the training data

In order to verify the proposed algorithm ensure the robust performance for recognizing motion data similar to the training data, we compare the performance of the proposed algorithm, SVM classifier, Gaussian process classifier, and Mori et al.'s knowledge-based algorithm [1]. The last one is also served as belief calculator in VPK_GPC.

In this experiment, we select three basic action category lying, sitting, and standing as the target action names. The performance is evaluated by using like cross validation scheme. Concretely speaking, we run over the calculation of the performance by 1000 frames training motion data and the other 11000 frames testing motion data in recall/ precision aspect. The number of the repetition for training with 1000 frames and evaluating with 11000 frames is 20. Finally we acquire the mean and variance of accuracy in each action name by each recognition method.

5.2.1 Parameter settings of learning algorithms

As preparation, we adjust the parameters of knowledge-based recognition algorithm for fitting the input motion and the recognition result. All the kernel based technique, SVM, Gaussian process classifier, and the proposed method uses same type of kernel and exact same parameters. The used type is Gaussian kernel. When dimension of the total feature input into the kernel function is f_d, we tried to set the kernel parameters σ in the Gaussian kernel as

$$\sigma/\sqrt{f_d} = \{0.25, 0.5, 1.0, 1.5\}.$$

The penalty factor C in SVM which also represents the maximum value of the Lagrange variables is selected from $10, 100, 1000$ and is determined through 5-fold cross validation.

The parameters used in the proposed algorithm is set 90% hypothesis testing. Concretely speaking, the parameters appeared in **Table** 1, **Eq.**(6), **Eq.**(7), **Eq.**(8) is set as $k = 1.645$, $c = 0.05$, $a = 0.18$, $e = 2.49$, $\epsilon = \gamma = 0.01$.

5.2.2 Result

Table 3 shows the result of the performance for all the recognition algorithms when we give the condition mentioned above. In the table, SVM, GPC, Hyp, and Prop represents the performance by support vector classification, Gaussian process classification, knowledge based recognition, and the proposed algorithm, respectively. In kernel based classification, the performance written in the table represents average performance of all over the kernel parameters mentioned above. The r with brackets represents classification error in recall perspective. Concretely speaking, (r) represents the percentage of samples which are misclassified as -1, meanwhile the true labels for these samples are $+1$. On the other hand, (p) represents error rate in precision aspect. In particular, (p) shows the percentage of the samples with label -1 when the classifier estimates the codes of them as $+1$.

Table 3: Error recognition rate in each method

Action	SVM	GPC	Prop	Hyp
Lying (r)	0.5 ± 0.1	0.7 ± 0.2	1.2 ± 1.0	4.5 ± 0
(p)	1.3 ± 0.1	1.3 ± 0.1	1.3 ± 0.1	1.9 ± 0
Sitting (r)	1.3 ± 0.3	1.2 ± 0.3	1.3 ± 0.3	1.6 ± 0
(p)	1.6 ± 0.4	1.4 ± 0.3	1.4 ± 0.3	3.8 ± 0
Standing (r)	0.5 ± 0.0	0.6 ± 0.1	0.8 ± 0.1	3.0 ± 0
(p)	1.2 ± 0.1	1.0 ± 0.2	0.7 ± 0.1	0.2 ± 0

This result implies that the performance of the proposed algorithm for motion similar to input motion is guaranteed even if the knowledge based algorithm is roughly appropriate but is subtly different exact classification boundary.

The demonstration of sequential learning algorithm for lying conditioned as $\sigma = 1.0\sqrt{f_d}$ (in this case, $f_d = 2$) is shown in Figure 4. Horizontal and vertical axis represents normalized height of head and hips. In Figure 4, the algorithm selectively implants virtual motions in feature space of "head to be low and hips to be high". This area satisfies the sparseness of the training data, the unclarity of classification output without prior knowledge, and difference between prior knowledge and classifier's result.

By taking the strategy mentioned in 4.2, the sequential learning algorithm needs small virtual motion data to be implanted. In this example, the proposed sequential training algorithm terminate with 4 th iteration. In almost case, the number of iteration of the proposed sequential algorithm is about only 3 to 5 times.

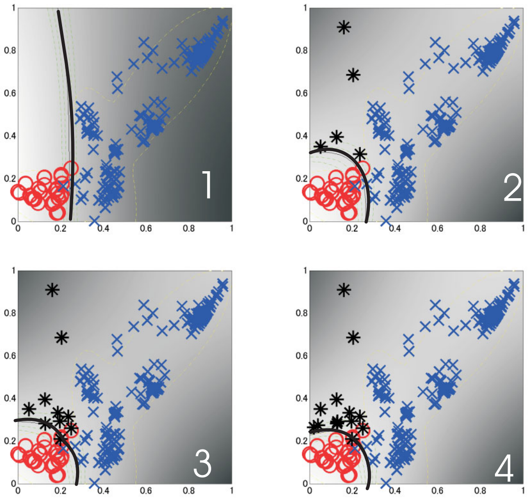

Figure 4: Process of iterative learning and infusing virtual samples for lying recognizer is shown. "Lying" and "not lying" motions are represented as circle and cross points. $*$ points represents virtual motion. The color map and curves of these thumbnails represent output of **Eq.**(5) and its zero points. The color gets lighter, the value is higher. The figure in each thumbnail represents the number of the iteration.

5.3 Application for novel motion

We observe the behavior of the recognizer for lying with or without prior knowledge for hand-standing motion, in order to verify the proposed algorithm can embed proper prior knowledge about action.

5.3.1 Setting

Because the action database used for training does not contain a kind of hand-standing action, we utilized BVH file(measured at 30 Hertz, 7 sec.) named CELEB2 which contains hand-standing like motion from Biovision Motion Collection's CD-ROM. The thumbnails of the motion in this BVH file is shown in Figure 5.

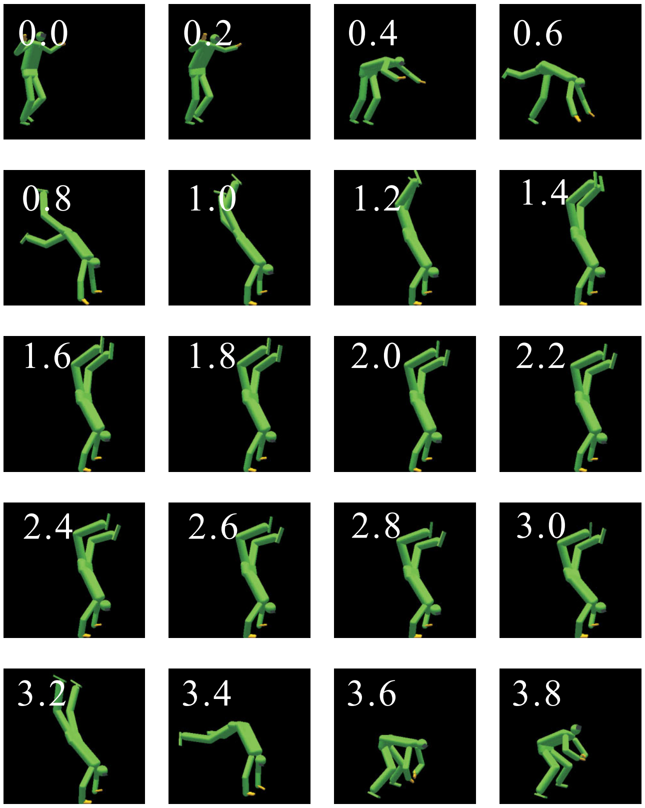

Figure 5: Thumbnails of "CELEB2.BVH" from *Biovision Motion Collection CD* where a male stands by hand are shown. The figure shown in the left-top side of each thumbnail indicates seconds from the starting time.

In this experiment, we observe the recognizer learned by the same conditions mentioned in section 5.2. The learning data is selected randomly from all the motion data and we iterate this process 30 times.

5.3.2 Observation result

In the case of Gaussian process classification without prior knowledge, misclassification occurs in 8 times of 30 trials. In this context, misclassification means the recognizer classifies input hand-standing motion as lying. The longest frames of misclassification is 12 frames, i.e. 0.4 sec.

On the other hand, the misclassification for hand-standing motion never happens in the proposed recognizer with prior knowledge. This result implies guarantee of proper embedding of prior knowledge for the recognizer.

6 Conclusion

The main contribution of this paper is to make a solution to prevent the recognition system from the improper performance for novel motion dissimilar to the training data. This paper proposes new learning and classification algorithms which incorporates prior knowledge via virtual motion in sparse area in order to compensate inaccuracy classification for novel motion in Bayesian kernel methods. The algorithm is called as VPK_GPC. We also developed new sequential learning algorithm. The sequential algorithm automatically and efficiently generate and select virtual motion into VPK_GPC iteratively.

The experimental result shows that the proposed recognition algorithm does not face excessive interference by the virtual motion data and prior knowledge even if the qualitative prior knowledge is roughly appropriate but is subtly different from exact classification boundary. We also validated the proper embedding of the prior knowledge by applying hand-standing motion as novel motion to recognizer for lying. Even if simple Gaussian process classifier fails to recognize hand-standing motion, the proposed method never fails to recognize.

The future work of this research is to make new robust algorithm to determine sparse area, because the mixture of Gaussian and its EM algorithm is depended on initial parameters and the boundary of the sparse area is imbalance. We also plan to make new algorithm that automatically determines some priori given parameters in the proposed sequential learning algorithm.

References

[1] T. Mori et al. Human-like Action Recognition System on Whole Body Motion-captured File. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2066–2073, 2001.

[2] M. Shimosaka et al. Recognition of Human Daily Life Action and Its Performance Adjustment based on Support Vector Learning. In *Proceedings of the Third IEEE International Conference on Humanoid Robots*, 2003.

[3] B. Schölkopf et al. *Learning with Kernels*. MIT Press, 2002.

[4] B. Schölkopf et al. Incorporating Invariances in Support Vector Learning Machines. In *Proceedings of the Artificial Neural Networks — ICANN'96*, pp. 47–52, 1996.

[5] S. Romdhani et al. Computationally Efficient Face Detection. In *Proceedings of the eighth International Conference on Computer Vision 2001*, Vol. 2, pp. 695–700, 2001.

[6] M. Inoue et al. Exploitation of Unlabeled Sequence in Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp. 1570–1581, 2003.

[7] K. Nigam et al. Text Classification from Labelled and Unlabeled Documents using EM. *Machine Learning*, Vol. 39, pp. 103–134, 2000.

[8] C.K.I. Williams et al. Bayesian Classification With Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, pp. 1342–1351, December 1998.

[9] R. Herbrich. *Learning Kernel Classifiers*. MIT Press, 2002.

[10] D. Gamerman. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference (Texts in Statistical Science)*. CRC PrI Llc, 1997.

[11] D. Tax et al. Outlier Detection using Classifier Instability. In *Proceedings of the workshop Statistical Pattern Recognition*, 1998.

[12] B. Scholkopf et al. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems 12*, pp. 582–588, 2000.

[13] A. Dempster et al. Maximum likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B(Methodological)*, Vol. 39, No. 1, pp. 1–38, 1977.

[14] T. Mori et al. ICS Action Database. http://www.ics.t.u-tokyo.ac.jp/action/, 2003.

[15] Biovision. Motion Capture Data Format. http://www.biovision.com/bvh.html.