Efficient Margin-Based Query Learning on Action Classification

Masamichi Shimosaka, Taketoshi Mori and Tomomasa Sato
The University of Tokyo
Email: {simosaka,tmori,tomo}@ics.t.u-tokyo.ac.jp

***Abstract*— In this paper, we propose a margin-based query learning algorithm for action recognition to reduce a laborious work on annotating action labels of time-series motion. The annotation is an inevitable task for designers of recognition systems with supervised learning techniques. Query learning is a kind of compensation approach for this, and can also be categorized into interactive learning. Our algorithm is a natural extension of maximum margin learning; a.k.a. support vector machines. Thanks to the theoretical analysis of the optimal condition of the maximum margin learning, the algorithm runs with a single and simple criterion. To prevent poor performance of the classifier learned with very few size of labeled motion data set, the algorithm exploits cluster information of massive unlabeled motion dataset. In contrast to the previous margin-based query learning methods, the algorithm has superiority in terms of stability. The empirical evaluation using real motion and synthetic dataset shows that our algorithm can achieve both drastic reduction of annotation cost and making robust classifiers.**

I. Introduction

Recognizing human action is one of essential foundations to achieve smooth communication between intelligent systems, especially robots, and human. It is also a key technical element in achieving analysis and surveillance of human activity by intelligent systems. Traditionally, researchers have tried to build robust and feasible action recognition systems by borrowing statistical machine learning techniques. One of the pioneers of this research field is proposed by Yamato et al. [1]. This trend continues in the current century [2], [3]. We also built a recognition algorithm and system for human daily life action [4] based on kernel methods [5]: an approach of statistical learning. Kernel methods are known as not only theoretically robust but also robust techniques in recent machine learning community. A well known supervised kernel-based learning called maximum margin learning [6] is applied to various fields of pattern recognition communities [7], [2]. We have also adopted the maximum margin learning to obtain high accurate action classifiers [4].

However, there exists a critical problem of the maximum margin learning. It is a well known fact that the supervised learning requires fully annotated dataset. Thanks to this result, a designer of recognition systems must annotate action labels to whole motion sequences when using supervised learning. This is very laborious when massive instances are available. Ironically, the accuracy tends to be higher than poor dataset. This is a dilemma in supervised learning. Hence, it is a critical challenge for researchers to reduce cost of annotation. Specifically, researchers in the field of action recognition must pay attention to this topic because it is much easier to capture human motions than to annotate action labels to the motions. In other words, someone must assign labels to time-series motion at a couple of thousands times even if a couple of minutes of motions are captured.

In machine learning community other than action recognition research, there already exists an idea for effective learning called query learning [8]. The process of the query learning is as follows (see Fig. 1): As a preparation a classifier should be learned with small size dataset a priori. Then the classifier iterates active sampling, querying, and re-learning. Specifically, the active sampling is a process to select actively a instance which would effectively makes the classifier smart. In the querying process, the classifier queries the selected sample to human, and then the sample is annotated by human. In re-learning process, the classifier is re-optimized with the annotated dataset that includes the newly annotated sample.

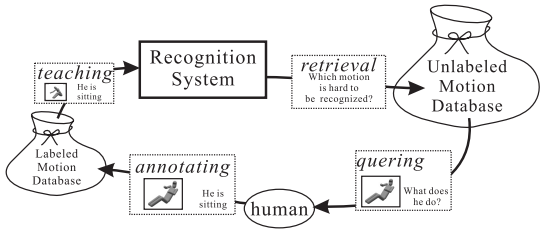

Fig. 1. Process of query Learning is shown.

In this paper, we design a novel query learning method, which is related to the maximum margin learning. Specifically, we propose a new learning method that leverages the property of margin. There already exists margin-based query learning [9], however, this algorithm retains a critical problem on stability. The proposed algorithm solves this problem.

The rest of this paper proceeds as follows. Next section outlines a scheme of action recognition and, describes its configuration and formulation. Section III introduces a concept of query learning and a property of the margin, which is important for the proposed query learning. Next, section IV outlines the proposed algorithm and describes the superiority to the previous research works. Section V explains the experimental result to validate the proposed method. We conclude in the last section with some directions for future research.

II. Online Action Recognition and Maximum Margin Learning

A. Scheme of Action Recognition and its Configuration

As a basis of our action recognition, we introduce a scheme of online action recognition proposed by Mori et al. [4]. This scheme can tackle the multi-label [10] problem of action recognition with simple approach. The multi-label problem is also known as the problem of simultaneous recognition. A phenomenon that some action often occurs with another action at the same time: e.g. *waving hand while standing* is an example of this. This must be firstly considered to develop a recognition system for daily actions. Mori et al.'s proposed the parallel and independent framework of multiple binary classifiers. Because this is very simple and empirically works well, we also take on quite similar scheme for action recognition (see Fig. 2). In other words, each binary classification discriminates input motion to annotate (*yes*) or (*no*).

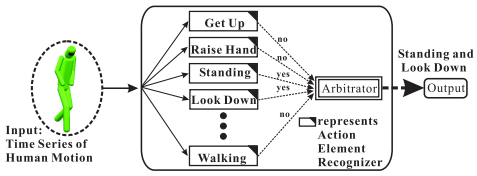

Fig. 2. Configuration of our action recognition scheme is shown.

B. Kernel-Based Action Classification

In each binary action classifier, we leverage kernel methods [5]. The kernel in this context represents a similarity metric between motions. Specifically, the kernel given for the action classifier K can be denoted as $K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ denote input motion features and $\mathcal{X}$ is an arbitrary collection of the motion features that can be computed from a time-series measured motion data. Height of head is an example of the motion feature. The classifiers do not have any concerns with what the variable $\boldsymbol{x}$ should be. The classifier works to discriminate the motion only via similarity score K.

In this paragraph, we define the formulation of the kernel-based action classification. Specifically, the classifier can be categorized into the non-parametric classification algorithm. The action classifier can be formulated as the following discriminative function $g(\cdot)$:

$$g(\boldsymbol{x}) = \sum_{n=1}^{N} \{\boldsymbol{\alpha}\}_n y^{(n)} K(\boldsymbol{x}, \boldsymbol{x}^{(n)}) + b \tag{1}$$

where $D = \{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1}^{N}$ represents a collection of the annotated motion dataset with N frames. $y^{(n)} \in \{\pm 1\}$ denotes an annotation for the n-th motion samples $\boldsymbol{x}^{(n)}$. If y is equal to $+1$, then corresponding motion $\boldsymbol{x}$ is judged as target action occurs. The N dimensional vector $\{\boldsymbol{\alpha}\}_n \geq 0,\ (n = 1, \ldots, N)$ depicts the weighting parameter of the classifier. The classification rule with this function is $\hat{y} = \mathrm{sgn}(g(\boldsymbol{x}))$. The operator $\mathrm{sgn}(\cdot)$ represents a step function as

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

C. Maximum Margin Learning for Making Action Classifier

We employ an optimization method called maximum margin learning [6] for adjusting the weighting parameter $\boldsymbol{\alpha}$. This is also known as support vector machines (SVMs). This adjusts $\boldsymbol{\alpha}$ to maximize the degree of the separation between two classes in *feature* space. The degree of separation is called *margin*. As a result, the maximum margin learning results in the following quadratic programming (QP) with linear constraints:

$$\begin{aligned} &\text{minimize}: \quad W(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{G}\boldsymbol{\alpha} - \mathbf{1}_N^{\mathsf{T}}\boldsymbol{\alpha} \\ &\text{subject to}: \quad \begin{cases} \sum_{n=1}^{N} \{\boldsymbol{\alpha}\}_n y^{(n)} = 0 \\ 0 \leq \{\boldsymbol{\alpha}\}_n \leq C,\ n = 1, \ldots, N \end{cases}, \end{aligned} \tag{2}$$

where square matrix $\{\boldsymbol{G}\}_{i,j} = y^{(i)} y^{(j)} K(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$ contains a product by label and similarity between motion samples. The positive constant $C > 0$ is a regularization factor to smooth the classification boundary. The benefit of the maximum margin learning is that the learning can avoid local minima which often occurs in traditional back propagation neural networks [11].

Thanks to the Karush-Kuhn-Tucker (KKT) condition [12], which is the necessary condition for the optimality in the QP problem, we can find an interesting and important property of the classifier optimized by the margin based learning. From (2), all the motion samples must satisfy the following relations under optimality:

$$y_i f(\boldsymbol{x}_i) \begin{cases} > 1, & \alpha_i = 0 \\ = 1, & \alpha_i \in (0, C) \\ < 1, & \alpha_i = C \end{cases}. \tag{3}$$

This implies that a part of the motion samples contributes discrimination rule, whereas the rest motion samples that satisfy $y^{(n)} g(\boldsymbol{x}^{(n)}) > 1$ never contribute the classification. The relevant motion samples in the discrimination are called *support vectors*.

III. Problem Formulation of Query Learning and Property of the Margin-Based Learning

This section formulates a problem setting of the query learning, and then explains the property of the margin-based learning that is important for the query learning.

A. Formulation of Query Learning

Before mentioning the detail explanation, we introduce notation utilized in the proposed algorithm. Let N be the size of the dataset about time-series motion $X = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$. In the variable X, the annotations themselves are not contained. In dataset X, there exists L ($L \ll N$) annotated samples and the rest $N - L$ un-annotated dataset. Let I be the set for the IDs of the annotated dataset, D_I be the collection of the annotated samples. This means $D_I = \{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i \in I}$, where y represents the labels. Another variable with the suffix $\setminus I$, $X_{\setminus I}$ represents the un-annotated motion samples, and X_I depicts

the annotated motion[1]. If the number L, which is size of the collection I, is equal to N, we give alias of this variable for D_I as $D_{\mathrm{ALL}} = D_I$. The query learning in this paper is a process with the following procedures:

1) Initial phase: First, a part of motions in X is selected, and then annotated to build the annotated dataset D_I and the collection of index I. In this phase, a classifier is learned with this dataset D_I.

2) Main query phase: The following 3 steps are iterated to reach some conditions. First, some motion instances are actively sampled from $X_{\setminus I}$, and I are updated. Second, the selected motion are queried to human and given the annotation to update D_I, and $X_{\setminus I}$. Third, the classifier re-learns with the updated D_I.

3) Validation phase: After a certain number of the iteration in the main query process, the performance validation and checks for the termination of the query learning process are executed utilizing a part of motions in $X_{\setminus I}$.

The main goal of the query learning processes is to build $g_{D_I}(\cdot) \approx g_{D_{\mathrm{ALL}}}(\cdot)$ so as to keep the size $|I|$ to N small where a function $g_{D_I}(\cdot)$ represents a discriminative function learned with D_I.

B. Property of Classifiers Optimized by Maximum Margin Learning

The KKT condition of the margin-based learning provides us cues to make a simple query criterion in the maximum margin learning. Here we show a simple example of this. Let $g_{D_I}(\cdot)$ be the discriminative function learned from D_I, let $D_{I\setminus\dagger}$ be a collection where $\boldsymbol{x}^\dagger$ are removed from the collection D_I, and $D_{I\cup *} \leftarrow \{D \cup [\boldsymbol{x}^*, y^*]\}$ be a collection in which the newly data $\boldsymbol{x}^*, y^*$ are added to the collection D_I. Given an assumption: the removed sample $\boldsymbol{x}^\dagger$ is not a support vector in $g_{D_I}(\cdot)$ and $g_{D_I}(\boldsymbol{x}^*)y^* > 1$, it satisfies

$$g_{D_{I\setminus\dagger}}(\boldsymbol{x}) = g_{D_I}(\boldsymbol{x}) = g_{D_{I\cup *}}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \mathcal{X}. \tag{4}$$

This result implies that a discriminative function $g_{D_I}(\cdot)$ would not change when a sample in the outside of the margin area is added to or removed from the dataset D_I. This means that there is room to improve the discriminative function $g_{D_I}(\cdot)$ only if there exist samples in the margin area. In other words, the prospect of the query learning: making D_{ALL} with small L would gain high, if instances in the margin area are actively sampled. This is because the selected motions should be support vectors in the final form of the discriminative function and the score of the QP problem would improve when such kind of samples are added.

IV. Margin-Based Query Learning Exploiting Global Similarity of Motion

A. Outline of the Algorithm

The proposed algorithm can be divided into the following three phases. The way of deviation is the same to the process in the previous section.

[1]The variable X_I itself does not contain the annotations

Initial phase : The algorithm requires at least one motion sample annotated positive label ($|I| = 1$). Next, the algorithm searches the index that satisfies $i^* = \mathrm{argmin}_{i \notin I}\{\boldsymbol{r}^*\}_i$, then selects i^* and $\boldsymbol{x}^{(i^*)}$ is annotated by human. i^* is added into the I. In the above equation, $\boldsymbol{r}^*$ represents a *global similarity* vector. This can be calculated from the following query vector $\boldsymbol{q}$:

$$\{\boldsymbol{q}\}_{i \in I} = 1, \quad \{\boldsymbol{q}\}_{i \notin I} = 0. \tag{5}$$

The vector $\boldsymbol{r}^*$ depends on the indices of the annotated data I, and represents similarity from the annotated dataset. This means that higher the element in $\boldsymbol{r}^*$ is, higher the similarity between the corresponds motion sample and X_I is. The detail explanation of the computation of *global similarity* will be in the next subsection. This procedure is executed iteratively to reach the size of indices $|I|$ be n_b. This procedure reduces the risk of D_I be biased: D_I contains a couple of motions extremely similar to the other data.

Main query phase : By borrowing the property of the margin and the idea mentioned in the previous section, the criterion of the margin-based query learning is to utilize the absolute value of the output of the discriminative function $g_{D_I}(\cdot)$. Specifically, the algorithm have to sample a motion $\boldsymbol{x}^{(j)}$ in $X_{\setminus I}$ that satisfies $|g_{D_I}\left(\boldsymbol{x}^{(j)}\right)| \leq 1$. The proposed algorithm utilizes the following retrieval criterion to improve the efficiency of the query:

$$j^* = \underset{j \notin I}{\mathrm{argmin}} \left| g_{D_I}(\boldsymbol{x}^{(j)}) \right|. \tag{6}$$

This strategy is adopted from our heuristic. Intuitive interpretation of this retrieval provides that the selected motion sample $\boldsymbol{x}^{(j^*)}$ would stabilize and improve the classification boundary because $\boldsymbol{x}^{(j^*)}$ is closest to the decision boundary. The motion sample closest to the decision boundary can be assumed as a hardest motion to be recognized.

This procedure iterates until there are no samples that satisfies $|g_{D_I}(\boldsymbol{x})| \leq 1$, where $\boldsymbol{x} \in X_{\setminus I}$. Then the validation phase starts in the algorithm. It is because it is probable that there exists a sample $\boldsymbol{x}$ that satisfies s $|g_{D_I}(\boldsymbol{x})| > 1$ even if incorrect classification result $yg_{D_I}(\boldsymbol{x}) < 1$ occurs.

Validation phase : Similar to the sampling idea in the initial phase, the algorithm selects a motion sample to validate from $X_{\setminus I}$ by leveraging the concept of the *global* similarity and the query vector $\boldsymbol{q}$ defined in (5).

Specifically, the algorithm samples $\boldsymbol{r}^*$ corresponding global similarity is the smallest in $X_{\setminus I}$ and validates the performance of the discriminative function with human annotation. Then the selected motion is added to I and $\boldsymbol{q}$ is updated. The procedure in which the selection, annotation, validation, and updating the indices are executed iterates n_v times, only if the discriminative function correctly classifies the selected motions, otherwise, the algorithm returns to the main query process.

The idea of this phase is that the algorithm should validate the performance with unlabeled data unrelated to the annotated or *known* dataset, because it is natural that the classifier gains

TABLE I

PROCESS OF THE MARGIN-BASED QUERY LEARNING

	Setting: Preparing motion dataset D that includes $\boldsymbol{x}$ with positive label (the number of this kind of samples is a) and $N-a$ motions without labels, kernel $\mathcal{K}$, parameters α, then making index I, calculating transition matrix S in (9), and setting initial query: $\boldsymbol{q}$ in (5)
1	**Initial phase:** Calculating global similarity $\boldsymbol{r}^*$ in (12), then selecting $i^* = \mathrm{argmin}_i\{\boldsymbol{r}^*\}_i$, then updating ($I \leftarrow I \cup i^*$, $\boldsymbol{q}$ by (5), and D_I) until $\lvert I\rvert < n_b$
2	**Main loop :** Making action recognizer by SVM from D_I and then selecting most significant data $\boldsymbol{x}_{i^*}$ that satisfies (6) until unlabeled samples exist in margin area of $g_{D_I}(\cdot)$. Finally, updating I and D_I and returning to **2**
3	**Validation phase :** Calculating global similarity $\boldsymbol{r}^*$ in (12), then select $i^* = \mathrm{argmin}_i\{\boldsymbol{r}\}_i$, then updating ($I \leftarrow I \cup i^*$ and D_I), then returning to **2** if the classifier fails to classify, else if the classifier never fails to classify n_v times, terminating the algorithm

high classification accuracy for the validation data extremely similar to the *known* dataset D_I. The proposed algorithm can be summarized in TABLE I.

B. Computing Global Similarity

In this subsection, the procedure of computing the *global* similarity utilized in the initial and the validation phase of the proposed algorithm is described. The vector $\boldsymbol{r}^*$ is a tuple of similarities between samples in X and the dataset X_I. It is noteworthy that the vector $\boldsymbol{r}^*$ does not represent the tuple of similarity between each motion sample in X and a motion sample in X_I. This leads to that this procedure requires a cluster assumption of the motion dataset. Hence this value requires an alternate procedure other than the computing kernel, which represents a *local* similarity between two samples.

Considering cluster information of the dataset is similar to the estimation of density information of the motion dataset. The simplest way to compute the density estimation is to utilize some parametric probability densities such as Gaussian mixture distributions [13]. But they require the model selection problem; what the number of the clusters should be. Furthermore, this strategy collapses if $\boldsymbol{x}$ can not be written in a vector data but be a time-series data. Hence, the procedure of computing the global similarity in this research should be alternate to the traditional parametric approaches and should be useful for the arbitrary data types $\boldsymbol{x}$.

The propose algorithm adopts a non-parametric similarity metric computation related to the Google PageRank algorithms [14] and the spectral learning methods [15]. The PageRank algorithm is utilized in search engines for WWW. The proposed global similarity computation requires the following processes: 1) making a *local* similarity metric matrix and its normalization, 2) the computation of the *global* similarity with the query vector $\boldsymbol{q}$ and the normalized local similarity matrix. These procedures are described in the following paragraphs.

1) Making Local Similarity Matrix and Normalization: Let $\mathcal{A}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ be the *local* similarity between two motion data $\boldsymbol{x}, \tilde{\boldsymbol{x}}$. This can be written as

$$\mathcal{A}\left(\boldsymbol{x}, \tilde{\boldsymbol{x}}\right) = \exp\left(-\lambda^2 d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right), \tag{7}$$

where $d(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \geq 0$ denotes some distance metric and λ is a positive value. This definition leads to the *local* similarity should be in $[0, 1]$ and high value of $\mathcal{A}(\cdot, \cdot)$ represents that the data $\boldsymbol{x}$ is similar to the other data $\tilde{\boldsymbol{x}}$. The distance metric $d(\cdot, \cdot)$ can be derived from the kernel defined priori,

$$d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \left| \frac{\boldsymbol{\phi}(\boldsymbol{x})}{|\boldsymbol{\phi}(\boldsymbol{x})|} - \frac{\boldsymbol{\phi}(\tilde{\boldsymbol{x}})}{|\boldsymbol{\phi}(\tilde{\boldsymbol{x}})|} \right|^2 = 2 - \frac{2K\left(\boldsymbol{x}, \tilde{\boldsymbol{x}}\right)}{\sqrt{K(\boldsymbol{x}, \boldsymbol{x}) K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}})}},$$

where a function $\boldsymbol{\phi}(\cdot)$ should satisfy $K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \boldsymbol{\phi}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x})$. This definition leads to the different value with respect to the types of the kernels; however, we can utilize kernel function directly as the *local* similarity value $\mathcal{A}(\cdot, \cdot)$ if the kernel leverages exponential function $\exp(\cdot)$, such as radial basis functions kernels and Mahalanobis kernels.

Next, the procedure makes a matrix depicted by $\boldsymbol{W}$ that contains where each element is calculated from $\mathcal{A}(\cdot, \cdot)$. This matrix is called *local* similarity matrix. Specifically, this matrix can be derived as

$$\{\boldsymbol{W}\}_{i,j} = (1 - \delta_{ij}) \mathcal{A}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right), \tag{8}$$

by using $X = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$. The algorithm normalizes this matrix as

$$\boldsymbol{S} = \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2}, \tag{9}$$

where $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix where i-th row and i-th column element represents sum of i-th row vector of $\boldsymbol{W}$. This is equivalent to $\{\boldsymbol{D}\}_{i,i} = \sum_{j=1}^{N} \{\boldsymbol{W}\}_{i,j}$.

An element of $\boldsymbol{S}$ defined at i-th row and j-th column represents the probability of transition from the state in i-th vertex to the j-th vertex, when it assumes that all the points are connected as a graph [16]. Hence, we can interpret each element of $\boldsymbol{S}$ represents similarity value in probabilistic form.

2) Computing Global Similarity with Query Vector: Here, the procedure of computing $\boldsymbol{r}^*$ exploiting $\boldsymbol{S}$ and $\boldsymbol{q}$ is described. In the algorithm, the following linear dynamics associating to the *global* similarity $\boldsymbol{r}$ is designed.

$$\boldsymbol{r}_t = \alpha \boldsymbol{S} \boldsymbol{r}_{t-1} + (1 - \alpha) \boldsymbol{q} \tag{10}$$

where $\boldsymbol{r}_t$ represents the state vector at "time" t and the constant $\alpha \in [0, 1)$. At the initial setting, the state vector at time $t = 0$ is set as $\boldsymbol{r}_0 = \boldsymbol{0}$. This dynamical systems affect the state vector $\boldsymbol{r}$ with respect to the similarity between a motion sample and the collection X_I. This can be seen as a diffusion operation of the particles on X_I. An element of $\boldsymbol{r}$ corresponding to $X_{\setminus I}$ would be smaller than that in X_I. Given that the dynamics comes to stable at time $t \to \infty$ and $\boldsymbol{r}$ converges, the following equation satisfies:

$$\boldsymbol{r}_\infty = \alpha \boldsymbol{S} \boldsymbol{r}_\infty + (1 - \alpha) \boldsymbol{q}. \tag{11}$$

This assumption makes us solve $\boldsymbol{r}^*$ analytically as

$$\boldsymbol{r}^* \propto (\boldsymbol{I} - \alpha \boldsymbol{S})^{-1} \boldsymbol{q}. \tag{12}$$

We utilize the above equation as the implementation of the algorithm.

3) Improving Efficiency of Computation for Global Similarity: The procedure of obtaining the global similarity in (12) takes the cost in the order of $\mathcal{O}(N^2)$ in terms of the memory and $\mathcal{O}(N^3)$ in the issue of the computation thanks to the inversion of $(I - \alpha S)$. In the case of action recognition, the size of dataset for the training is assumed to be from 10^3 to 10^4, hence, it is a critical matter for us to run the algorithm in the standard PCs, even when their computational power and their resource grows dramatically in recent years. Hence, the algorithm utilizes some approximation to calculate the global similarity vector.

The idea of the approximation is based on the incomplete Cholesky factorization technique [17]. This technique assumes the factorized matrix that contains the local similarity $\mathcal{A}(\cdot, \cdot)$ as a Gram matrix of radial basis function kernels. Each element of the Gram matrix contains a kernel value of the pairs. Specifically, the algorithm approximates a square matrix $\hat{\boldsymbol{A}}$ with orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{N \times M}$ $(M \ll N)$ as

$$\hat{\boldsymbol{A}} \approx \boldsymbol{R}\boldsymbol{R}^{\mathsf{T}}, \tag{13}$$

where $\{\hat{\boldsymbol{A}}\}_{i,j} = \mathcal{A}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)$. Hence the inverse computation of $I - \alpha S$ can be derived as

$$\begin{aligned}
(\boldsymbol{I} - \alpha \boldsymbol{S})^{-1} &= (\boldsymbol{I} - \alpha \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})^{-1} && (14)\\
&= \boldsymbol{D}^{1/2}(\boldsymbol{D} - \alpha \boldsymbol{W})^{-1}\boldsymbol{D}^{1/2} && (15)\\
&= \boldsymbol{D}^{1/2}(\boldsymbol{D} + \alpha \boldsymbol{L} - \alpha \hat{\boldsymbol{A}})^{-1}\boldsymbol{D}^{1/2} && (16)\\
&\approx \boldsymbol{D}^{1/2}(\boldsymbol{C} - \alpha \boldsymbol{R}\boldsymbol{R}^{\mathsf{T}})\boldsymbol{D}^{1/2}. && (17)
\end{aligned}$$

In the above equations, diagonal matrices $\boldsymbol{L}, \boldsymbol{C} \in \mathbb{R}^{N \times N}$ satisfy $\boldsymbol{L} = \hat{\boldsymbol{A}} - \boldsymbol{W}$ and $\boldsymbol{C} = \boldsymbol{D} + \alpha \boldsymbol{L}$. Thanks to the Woodbury's formula [18], this approximated result can be rewritten as

$$\begin{aligned}
(\boldsymbol{I} - \alpha \boldsymbol{S})^{-1} &\approx \boldsymbol{D}^{1/2}\boldsymbol{C}^{-1}\boldsymbol{D}^{1/2} + \alpha \boldsymbol{D}^{1/2}\boldsymbol{C}^{-1}\boldsymbol{R} && (18)\\
&\quad (\boldsymbol{I} - \alpha \boldsymbol{R}^{\mathsf{T}}\boldsymbol{C}^{-1}\boldsymbol{R})^{-1}\boldsymbol{R}^{\mathsf{T}}\boldsymbol{C}^{-1}\boldsymbol{D}^{1/2} && (19)
\end{aligned}$$

Due to the fact that $\boldsymbol{D}$ and $\boldsymbol{C}$ are diagonal matrices, and the size of the multiple of the matrices satisfies $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{C}^{-1}\boldsymbol{R} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{C}^{-1}\boldsymbol{D}^{1/2} \in \mathbb{R}^{M \times N}$, this approximation reduces the memorial cost to the level of $\mathcal{O}(MN)$. This approximation technique allows us to adjust the accuracy on factorizing $\hat{\boldsymbol{A}}$. The relation between the accuracy of the approximation and size M depends on the local similarity metric $\mathcal{A}(\cdot, \cdot)$, however, the empirical result implies the algorithm requires only $M = 10^2$ to relative to the $N = 10^4$, when we set $|\hat{\boldsymbol{A}} - \boldsymbol{R}\boldsymbol{R}^{\mathsf{T}}|_2 \approx 10^{-3}$.

C. Related works on Margin-Based Query Learning

There already exists several margin-based query learning methods. One is proposed by Campbell et al. [9] where the query criterion is quite the same with our heuristic. However, there is a critical difference between this method and our algorithm. The method of Campbell et al. sometimes fails to obtain globally optimal classifiers or the efficiency of the main query process is not good thanks to the quality of D_I. This is because the method of Campbell et al. makes sometimes biased D_I due to the fact that the algorithm randomly selects the motion samples. This critical problem is also reported in a recent research [19].

The superiority of the proposed methods relative to the method proposed by Campbell et al. is that leveraging the density information of X stabilizes the algorithm at the initial phase and the validation phase.

V. Experimental Results

As for the validation of the proposed method in this paper, we execute two kinds of experiments. One is to validate the effectiveness thanks to the query learning. Another is to evaluate an impact of leveraging the global similarity.

A. Effectiveness of the Query Learning

Because the main objective utilizing query learning scheme is to reduce the annotation cost for the motion dataset, we evaluate the efficiency of the proposed query learning. Specifically, we calculate the performance of the classifier with a couple of dozens of queries and re-learning, and then estimate how many queries $L = |I|$ would make the classifier $g_{D_I}(\cdot)$ reach the performance of the classifier with fully annotated dataset $g_{D_{\mathrm{ALL}}}(\cdot)$. In this experiment, we ignore the impact of exploiting the global similarity at the initial and the validation phases. This means we evaluate the performance of the classifier with respect to the number of the queries.

Action Dataset: In this experiment, a motion dataset: ICS Action Database [20] are utilized. This is an annotated motion capture dataset designed for evaluating the performance of action recognition. This contains over 100 motion capture files where an actor behaves dozens of daily actions. Specifically, in this dataset, human motions are annotated with 25 action names such as sitting, lying and folding arms in frame-wise. Annotations for each action category are executed separately. An annotation file for some target action contains human's judgment for of the corresponding action per frame by three degrees (*yes*, *neutral* and *no*). The label *neutral* represents ambiguous recognition result by human.

In this experiment, the label of each action symbol is re-assigned as $y = +1$ if the label is *yes* and $y = -1$ when the label is *neutral* or *no*. In this experiment three action symbols, such as *lying*, *standing*, and *sitting* are selected as target actions to be recognized.

Evaluation Method: As a performance criterion in this experiment, F-measure [21] is utilized. F-measure is a kind of performance criteria often utilized in the researches of information retrieval (IR). This indicates the performance with a single value that combines the performance of *recall* and *precision*. The *recall* is a ratio of counting $\hat{y}^{(n)} = +1$ when $y^{(n)} = +1$ with respect to the total size of motions where $y^{(n)} = +1$. Similar to the measure of *recall*, the *precision* is a ratio of counting $y^{(n)} = +1$ when $\hat{y}^{(n)} = 1$ with respect to the counts of $\hat{y}^{(n)} = +1$. The definition of F measure can be denoted as

$$F = \frac{1}{\frac{1}{(\beta+1)R} + \frac{\beta}{(\beta+1)P}}, \tag{20}$$

where R represents the recall, P depicts the precision, and the positive constant $\beta > 0$ is an adjustable parameter. The parameter β weights relative importance of the precision to the recall. When the constant β is equal to 1, this is a harmonic average of the recall and the precision. In this experiment, we set the parameter β as $\beta = 1$. From the definition of the F-measure, higher F-measures is, higher the performance of the classifier is. We calculate F-measures of $g_{D_I}(\cdot)$ at each query step, by comparing the performance of $g_{D_{\mathrm{ALL}}}(\cdot)$ and evaluate the number of the queries where $g_{D_I}(\cdot)$ reaches and saturates. In this experiment, the initial dataset D_I is prepared according to the initial phase of our algorithm.

Condition and Parameters: In this experiment, the dataset from ICS Action Database are divided into 5 subsets, then each subset is utilized as the training dataset ($N \approx 2000$). We calculate maximum, average, and minimum of the F-measure with respect to the size $|I|$. The kernel used in this experiment is radial basis function. Even though motion feature selection problem and the adjustment of the kernel parameters is one of the open problem of kernel-based machine learning techniques for the pattern recognition researchers, we borrow the idea of the heuristic but practical approach of this problem from our past work [4]. Even though the performance of the classification cannot achieve highest quality, our decision-making in this experiment is valid. This is because the motivation of this experiment is to clarify the usefulness of the query learning with respect to the standard maximum margin learning. The constants required in the algorithm in section IV are set as $\alpha = 0.99$, $n_b = 10$, $n_v = 10$.

Result: The experimental result shows that the classifier in every action, *lying*, *standing*, and *sitting*, reaches the performance of the classifier $g_{D_{\mathrm{ALL}}}(\cdot)$ within about 100 times queries. Fig. 3 shows that the F-measure of the classifier of *sitting* with respect to the size of I. In this figure, the symbol plus represents the average value of F-measure. The dotted lines covering the average F-measure represents the maximum and the minimum of the F-measure, respectively. From this result, it can be found that the classifier obtains as high classification performance as the $g_{D_{\mathrm{ALL}}}(\cdot)$ within only 50 queries. This result implies the algorithm be beneficial in terms of the cost of the annotation process because the total number of motions samples: N is over a thousand.

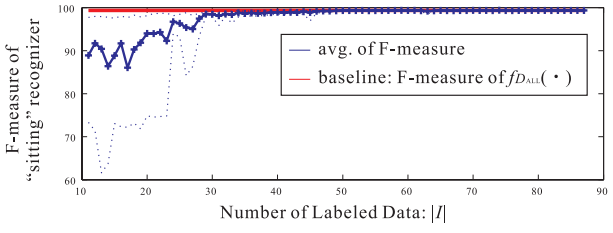

Fig. 3. Annotated size $|I|$ vs. F-measure of recognizer

B. Impact of Leveraging Global Similarity

In this experiment, we validate the impact of leveraging the global similarity $\boldsymbol{r}^*$ at initial phase of the learning. Specifically, this experiment clarifies the impact of the global similarity by comparing the performance of the method of Campbell et al. In this experiment, we obtain the performance of the classifier learned from D_I of both methods and then show the difference of the two methods. Next, we discriminate the two methods to visualize the difference of the random sampling technique adopted in [9] and the sampling technique using the global similarity by using 2 dimensional synthetic dataset.

1) Issues on Classification Performance:

Action Dataset, Evaluation Method, and Condition : This experiment also utilizes the same dataset as in the previous experiment and also utilizes F-measure for the evaluation criterion. As a condition of the experiment, we set $n_b = 10$, which is the same in the previous experiment. As for the procedure of random sampling for the initial phase of the query learning, the algorithm is given 1 positive sample selected priori and randomly extract $n_b - 1$ instances. The target actions are the same as the previous experiment.

Result: The experiment result shows that the performance of the proposed method at the stage $|I| = 10$ achieves 88.9%, meanwhile the compared method utilizing the random sampling method achieves only 85.9%. This result implies that leveraging the global similarity makes the algorithm stable. In addition, the random sampling method sometimes fails to obtain the initial classifier, because the random sampler fails to draw any negative instances within $n_b - 1$ times hence the learning of SVM cannot be executed. We neglected these failures to calculate the F-measure. The proposed algorithm cannot hedge completely the risk to occur these failures, however, the probability is much smaller than the random sampling. This is because the algorithm tends to select the negative samples on $|I| = 2$, thanks to incorporating the density information of the dataset.

2) Visualization to Impact of Global Similarity on 2-dimensional Synthetic Dataset::

Dataset: A mixture of Gaussian distributions whose component is 5 is utilized and draws 200 points in 2-dimensional feature space. The drawn 200 points are utilized as X.

Result: As the first step, the global similarity in case of $|I| = 1$ is shown in Fig. 4. The circle represents the annotated sample, i.e. X_I. The square represents the unannotated samples whose size is proportional to the global similarity. Larger the square is, larger the global similarity is. Fig. 4 shows that the global similarity declines gradually with respect to the distance between sample and the annotated sample. The more important things shown in this figure is that the global similarity in the isolated cluster (the right-bottom of this figure) is much smaller than the others. This result implies the algorithm tends to select the negative instance within $|I| = 2$.

Next, the result of the random sampling on $n_b = 5$ is shown in the right side of Fig. 5. The circles correspond

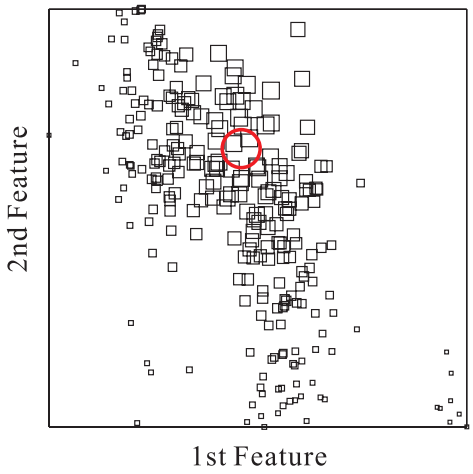

Fig. 4. Visualization of global similarity on two dimensional synthetic dataset when $|I| = 1$

to the D_I in the initial phase. Due to the fact that random sampling ignores the cluster assumption of the dataset, it often occurs that the sampler extract instances where each of them is extremely similar position. In contrast to the naive method, the proposed method extracts instances where each of them distributes broadly and the result of the sampling is relatively stable (see the left side of Fig. 5). This result also shows that the impact of using the global similarity is quite valuable to stabilize the learning at the initial phase.

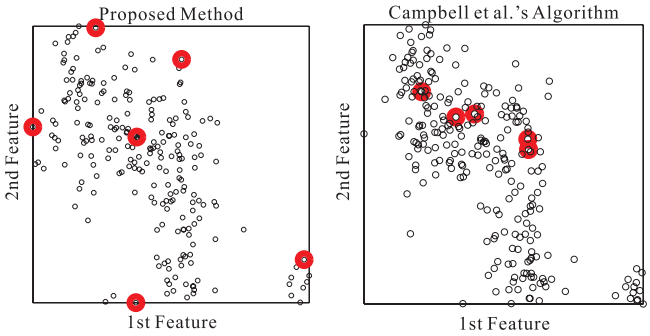

Fig. 5. Sampling results on the initial phase of the query learning for two dimensional synthetic dataset via our method (left) and Campbell et al. algorithm (right) is shown. In each figure, small size circles represent X, (red) large size and bold circles represent X_I, respectively.

VI. Conclusion

This paper introduces a margin-based query learning algorithm for making action classification to reduce laborious cost on annotating action label to motion samples, which is inevitable for making robust action classifier with supervised learning. The proposed algorithm provides us tremendous benefits in terms of practicality. Firstly, an user has only to design a kernel and prepare at least one annotated motion whose label is positive ($y = +1$). Second, the proposed algorithm leverages the single and simple criterion to search instance to be queried. In contrast to the traditional query learning, the algorithm leverages *global similarity* of motion dataset to improve the stability and accuracy of the method. The *global* similarity is a kind of graph-driven similarity metric of the local similarity matrix. Empirical evaluation results in that our algorithm is proven to reduce the annotation cost drastically for practical action recognition learning. Exploiting information of the *global* similarity helps the algorithm to be stable and effective on real and synthetic data classification.

Our suggestion for future work is to unify the criteria used in the proposed algorithm; query heuristic, maximum-margin learning, sample selection on initial and validation phase based on *global* similarity computation.

References

[1] J. Yamato et al. Recognizing human action in time-sequential images using hidden Markov model. In *Proceedings of the 1992 CVPR*, pages 379–385, 1992.

[2] D. Cao et al. Online motion classification using support vector machines. In *Proceedings of the 2004 IEEE International Conference on Robotics and Autometaion*, volume 3, pages 2291–2296, 2004.

[3] T. Inamura et al. From stochastic motion generation and recognition to geometric symbol development and manipulation. In *CD-ROM of the 3rd International Conference on Humanoid Robots*, 2003.

[4] T. Mori et al. Recognition of actions in daily life and its performance adjustment based on support vector learning . *International Journal of Humanoid Robotics*, 1(4):565–583, 2004.

[5] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] S. Romdhani et al. Computationally efficient face detection. In *Proceedings of the 8th ICCV*, volume 2, pages 695–700, 2001.

[8] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[9] C. Campbell et al. Query learning with large margin classifiers. In *Proceedings of the 17th ICML*, pages 111–118, 2000.

[10] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in NIPS 15*, pages 721–728. MIT Press, 2003.

[11] C. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[12] J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[13] R. Duda et al. *Pattern Classification 2nd Edition*. John Wiley & Sons, 2000.

[14] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International Conference on World Wide Web*, pages 107–117, 1998.

[15] D. Zhou et al. Ranking on data manifolds. In *Advances in NIPS 16*. MIT Press, 2004.

[16] F. Chung. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[17] F. Bach and M. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[18] H. Lütkepohl. *Handbook of Matrices*. Wiley & Sons, 1996.

[19] L. Wang et al. Bootstrapping SVM active learning by incorporating unlabeled images for image retrieval. In *Proceedings of the 2003 CVPR*, volume 1, pages 629–634, 2003.

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

[21] C. Rijsbergen. *Information Retrieval*. Butterworth, 1976.