7. Feature Selection for Classification of Variable length Multi-attribute Motions*

Chuanjun Li, Latifur Khan, and Balakrishnan Prabhakaran

Summary. As a relatively new type of multimedia, captured motion has its specific properties. The data of motions has multiple attributes to capture movements of multiple joints of a subject, and has different lengths for even similar motions. There are no row-to-row correspondences between data matrices of two motions. To be classified and recognized, multi-attribute motion data of different lengths are reduced to feature vectors by using the properties of Singular Value Decomposition (SVD) of motion data in this Chapter. Different feature selection approaches are explored, and by applying Support Vector Machines (SVM) to the feature vectors, we can efficiently classify and recognize real world multi-attribute motion data. With our datasets of hundreds of 3D motions with different lengths and variations, classification by SVM is compared with classification by related similarity measures, in terms of accuracy and CPU time.

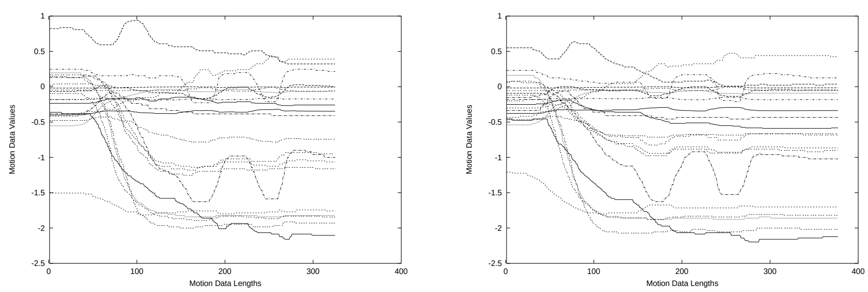
7.1 Introduction

Captured motion stream is a relatively new type of multimedia, and can have a variety of applications: gait analysis (physical medicine and rehabilitation), virtual reality, as well as in entertainment fields such as animations and video gaming industry. The motions recorded by gesture sensing devices (such as the data glove CyberGlove) and 3D human motion capture cameras (such as the Vicon cameras) have multiple attributes and multiple dimensions. For instance, a gesture sensing device such as CyberGlove has multiple sensors that transmit values to indicate motions of a hand, and a motion capture system generates multiple degrees of freedom (DOF) data for human motions. As a result, a multi-attribute motion yields a matrix over the motion duration, rather than a multidimensional vector as for a time series sequence.

Motion classification identifies a motion class to which an unknown motion most likely belongs, and poses several challenges:

* This chapter is extended from our previous work [1]

- Each of the motion dataset has dozens of attributes rather than one attribute as for a time series sequence. Motion data of multiple attributes are aggregate data, and should be considered together to make the motions meaningful.
- The matrices of motion data can be of variable lengths, even for similar motions. Motions are carried out with different speeds at different time. They can have different durations, and motion sampling rates may also be different. There are no continuous row-to-row correspondences between data of similar motions as shown in Figure 7.1.



(a) (b)

Fig. 7.1. Data for two similar motions. Similar motions can have different lengths, and different corresponding attribute pairs can have different variations at different time, hence there are no row-to-row correspondences between data of similar motions.

High accuracy classification makes the effective applications of new motions possible, and requires the extraction of highly representative feature vectors of motions. Instead of considering motion data rows/frames for motion identification, we consider the geometric structures of motion matrices in a high dimensional space. If a motion frame has dimension n, then a motion sequence of length m can be taken to be m vectors in an nD space. The distributions of the nD vectors are explored for feature extraction in this chapter. We obtain feature vectors for motion patterns using the Singular Vector Decomposition (SVD) properties of the motion matrices since SVD optimally exposes the geometric structure of a matrix. Different approaches to extracting feature vectors are explored based on how information is to be extracted from SVD. We then classify the vectors for all the motions. The learning machines can be trained for classification by using a training set of vectors each of which has a unique class label. After training, the machines can determine the class label of a new motion and thus classify the new motion during a testing phase.

There are many classification techniques for different applications. Classification by Support Vector Machines (SVM) has been proven to be computationally efficient especially when dealing with relatively large datasets [2] and has been successfully applied to solve many real world problems [3–5]. This chapter explores the feasibility of SVM in motion classification and experiments with different criteria for class decision. Experiments with hand gestures and human motions achieve 95-100% classification accuracy.

In comparison with classification by using SVM, we also computed the similarities of motion data in the testing datasets with motion data in the training datasets. The similarity measures used are the weighted-sum SVD [6], Eros [7], MAS [8] and kWAS [9], similarity measures proposed recently for capturing similarities of variable-length multi-attribute motion patterns.

This Chapter extends our previous work in [1] in the following ways:

- Different feature vectors are explored. The new feature vectors try to consider more information from SVD.
- SVM with probability estimate is explored. This option provides the possibility of segmenting streams by SVM for further work.
- Various human motions captured by infrared cameras are experimented in addition to hand gestures.
- Comparison with new similarity measures is made.

The rest of the chapter is organized as follows. Section 7.2 gives a brief review of related work. Section 7.3 contains the background knowledge of SVM, SVD and dynamic time warping (DTW). Section 7.5 proposes a new approach to classifying multi-attribute motion data using SVD and SVM, and the classification is further verified by using DTW for motion directions. Section 7.6 experimentally evaluates the accuracy and CPU time of our proposed approach, followed by Section 7.7 which concludes this chapter.

7.2 Related Work

Recognition of multi-attribute sequences has obtained increasing attentions in recent years. Mostly distance measures are defined for multi-attribute data to reflect the similarities of multi-attribute data. In [10], multi-attribute sequences of equal lengths are considered. Scaling and shifting transformations are considered when defining sequence distances and an index structure is proposed for shift and scale transformations. Similarity search of multi-attribute sequences with different lengths cannot be solved by the distance definitions and the index as proposed in [10].

Multi-attribute sequences are partitioned into subsequences in [11]. Each of the partitioned subsequences is contained in a Minimum Bounding Rectangle (MBR). Every MBR is indexed and stored into a database by using an R-tree or any of its variants. Estimated MBR distances are used to speed up the searching of similar motions. If two sequences are of different lengths, the shorter sequence is compared with the other by sliding from the beginning to the end of the longer sequence. When two similar sequences with different durations or with local accelerations and decelerations are considered, other approaches would be needed.

Dynamic time warping (DTW) and longest common subsequence (LCSS) are extended for similarity measures of multi-attribute data in [12]. Before the exact LCSS or DTW is performed, sequences are segmented into MBRs to be stored in an R-tree. Based on the MBR intersections, similarity estimates are computed to prune irrelevant sequences. Both DTW and LCSS have a computational complexity of $O(wd(m+n))$, where w is a matching window size, d is the number of attributes, and m, n are the lengths of two data sequences. When w is a significant portion of m or n, the computation can be even quadratic in the length of the sequences, making it non-scalable to large databases with long multi-attribute sequences. It has been shown in [12] that the index performance significantly degrades when the warping length increases. Even for a small number of 20 MBRs per long sequence, the index space requirements can be about a quarter of the dataset size.

Hidden Markov models (HMMs) have been used to address speech and handwriting recognition [13, 14] as well as American Sign Language (ASL) recognition problems [15]. Different states should be specified for each sign or motion unit when HMMs are involved, the number of words in a sentence is required to be known beforehand, and grammar constraints should also be known beforehand for using HMMs. When the specified states are not followed, or motion variations are relatively large, recognition accuracy would decrease dramatically. This is true even when legitimate or meaningful motions are generated for HMM-based recognitions. This chapter addresses the classification of individual motions, no states or grammar constraints are involved for individual motions. Thus HMMs are not suitable for our classification purpose.

Shahabi et al. [16] applied learning techniques such as Decision Trees, Bayesian classifiers and Neural Networks to recognize static signs for a 10-sign vocabulary, and achieved 84.66% accuracy. In [6], a weighted-sum SVD is defined for measuring the similarity of two multi-attribute motion sequences. The similarity definition takes the minimum of two weighted sums of the inner products of right singular vectors. Eros as proposed in [7] computes the similarity of two motion patterns by using angular similarities of right singular vectors. The angular similarities are weighted by a different weights obtained from singular values of all available motion patterns in the database. The singular value weight vector is the same for similarity computation of all motion patterns.

Li et al. define a similarity measure for multi-attribute motion data in [8] as follows.

$$\Psi(Q, P) = |u_1 \cdot v_1| \times (\boldsymbol{\sigma} \cdot \boldsymbol{\lambda} - \eta)/(1 - \eta)$$

where u_1 and v_1 are the first singular vectors of Q and P, respectively, $\boldsymbol{\sigma} = \sigma/|\sigma|$, $\boldsymbol{\lambda} = \lambda/|\lambda|$, and σ and λ are the vectors of the singular values of Q^TQ and P^TP, respectively. Weight parameter η is to ensure that the normalized singular value vectors $\boldsymbol{\sigma}$ and $\boldsymbol{\lambda}$ and the first right singular vectors u_1 and v_1 have similar contributions to the similarity measure and is determined by experiments. η can be set to 0.9 for the multi-attribute motion data. This similarity measure captures the most important information revealed by the first right singular vectors and the singular values, and can be applied to prune most of the irrelevant motion data, and inner products of equal-length re-interpolated first left singular vectors have been used as the first attempt to consider motions with different directions or with repetitions.

We refer to the similarity measure as defined in [8] which captures the main angular similarity of two motions as MAS hereafter. Furthermore, kWAS as proposed in [9] considers the angular similarities of the first k right singular pairs weighted by the associated singular values.

In contrast, this chapter addresses motion classification by utilizing SVD, SVM, and DTW. Different feature extraction approaches are explored, and different class decision criteria are experimented with motion data generated for both hand gestures and human motions.

7.3 Background

This section gives some background knowledge of singular value decomposition and support vector machines for feature extraction and motion classification.

7.3.1 Support Vector Machines

Support Vector Machines are a class of learning machines that aim at finding optimal hyperplanes, the boundaries with the maximal margin of separation between every two classes, among different classes of input data or training data in a high dimensional feature space $\mathcal{F}$, and new test data can be classified using the separating hyperplanes. The optimal hyperplanes, obtained during a *training* phase, make the smallest number of training errors. Figure 7.2 illustrates an optimal hyperplane for two classes of training data.

Let $\{x_i, y_i\}$, $i = 1, 2, \ldots, L$ be L training data vectors x_i with class labels y_i, and $y_i \in \{-1, +1\}$ for binary classification. Given an input vector x, an SVM constructs a classifier of the form

$$g(x) = sign(\sum_{i=1}^{L} \alpha_i y_i K(x_i, x) + b)$$

where $\{\alpha_i\}$ are non-negative Lagrange multipliers each of which corresponds to an example from the training data, b is a bias constant, and $K(\cdot, \cdot)$ is a kernel satisfying the conditions of Mercer's theorem [4]. Frequently used kernel

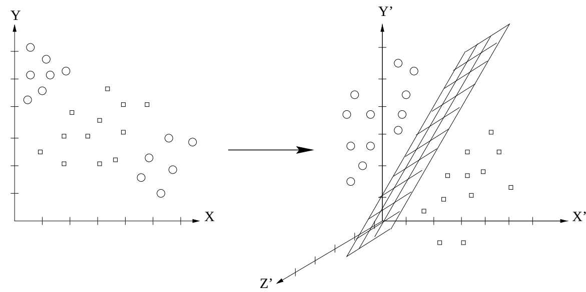


Fig. 7.2. Optimal hyperplane illustration

functions are the polynomial kernel $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$ and Gaussian Radial Basis Function (RBF) $K(x_i, x_j) = e^{-|x_i - x_j|^2/2\sigma^2}$.

The above decision function does not produce a probability. In many applications, a posterior probability, rather than an un-calibrated decision value, is needed for capturing the classification uncertainty. Efforts of mapping SVM outputs to posterior probabilities have been made in [4, 17]. Platt [17] uses a sigmoid function to estimate the binary class probability which is monotonic in g:

$$p(y = +1|g) = \frac{1}{1 + \exp{(Ag + B)}}$$

The parameters A and B can be fitted by using maximum likelihood estimation.

For multi-class classification, class probabilities can be estimated from binary class probabilities by pairwise coupling. Wu et al. [18] propose a multi-class probability approach which is more stable than other popular existing methods by using the following optimization:
Optimization:

$$\min_p \sum_{i=1}^{k} \sum_{j:j \neq i} (r_{ij} p_j - r_{ji} p_i)^2$$

under the constraints:

$$\sum_{i=1}^{k} p_i = 1, p_i \geq 0, \forall i$$

where r_{ij} are the binary class probability estimates of $\mu_{ij} \equiv P(y = i|y = i$ or $j, x)$ as obtained in [17].

The above optimization problem can be solved using Gaussian elimination after some algebra as shown in [18]. The vectors for which $\alpha_i > 0$ after optimization are called *support vectors*. Support vectors lie closest to the optimal hyperplane. After training, only the support vectors of the training data are used to represent the classifiers, and other training vectors have no influences.

For multi-class classification, two commonly used methods are one-versus-rest and one-versus-one approaches. The one-versus-rest method constructs k

classifiers for k classes, each of which separates that class from the rest of the data, and a test data point will be classified in the class with the highest probability estimate. The one-versus-one method constructs a classifier for each pair of classes. The probability of a test data vector belongs to one class is estimated from binary class probabilities, and the class with the largest posterior is the winning class for the test vector: $\arg\max_i[p_i]$. The one-versus-one method will be used for this work due to its simplicity and high classification accuracy [19].

SVM basically applies to classification of vectors, or uni-attribute time series data. To classify multi-attribute data, which are matrices rather than vectors, we need to transform or reduce multi-attribute matrices into vectors. We propose to use SVD to reduce multi-attribute motion data to feature vectors. Before showing how to extracting feature vectors, we present a brief introduction of SVD in the following subsection.

7.3.2 Singular Value Decomposition

As proven in [20], for any real $m \times n$ matrix A, there exist orthogonal matrices

$$U = [u_1, u_2, \ldots, u_m] \in R^{m \times m}, V = [v_1, v_2, \ldots, v_n] \in R^{n \times n}$$

such that

$$A = U \Sigma V^T$$

where $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}) \in R^{m \times n}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)} \geq 0$. The σ_i is the i^{th} singular value of A in non-increasing order and the vectors u_i and v_i are the i^{th} left and right singular vectors of A for $i = \min(m,n)$, respectively. The singular values of a matrix A are unique, and the singular vectors corresponding to distinct singular values are uniquely determined up to the sign [21]. Figure 7.3 shows an SVD example for a 6×4 matrix A.

The ith largest singular value σ_i of A is actually the 2-norm or Euclidean length of the ith largest projected vector Ax which is orthogonal to all the $i-1$ larger orthogonal vectors as shown by

$$\sigma_i = \max_{U} \min_{x \in U, \|x\|_2 = 1} \| Ax \|_2$$

where the maximum is taken over all i-dimensional subspaces $U \subseteq \Re^n$ [20]. Note that σ_1 is the largest 2-norm of A projections onto any x directions:

$$\sigma_1 = \max_{\|x\|_2 = 1} \| Ax \|_2$$

Hence the right singular vectors are the corresponding projection directions of the associated singular values, and the singular values account for the Euclidean lengths of different vectors projected by the row vectors in A onto different right singular vectors.

$$A = \begin{pmatrix} 1.1 & 2.2 & 3.3 & 4.4 & 5.5 \\ 1.5 & 2.6 & 3.7 & 4.8 & 5.9 \\ 2.3 & 3.4 & 4.5 & 5.6 & 6.7 \\ 3.6 & 4.7 & 5.8 & 6.9 & 7.1 \\ 4.5 & 5.7 & 6.9 & 7.2 & 8.3 \\ 5.8 & 7.3 & 9.2 & 11.6 & 13.8 \end{pmatrix}$$

$$= \begin{pmatrix} -0.2388 & -0.5360 & -0.3535 & 0.1605 & -0.4683 & -0.5345 \\ -0.2642 & -0.4301 & -0.2866 & 0.1400 & -0.0238 & 0.8018 \\ -0.3151 & -0.2184 & -0.1529 & 0.0990 & 0.8651 & -0.2673 \\ -0.3817 & 0.3366 & -0.4920 & -0.7038 & -0.0604 & 0.0000 \\ -0.4386 & 0.5938 & -0.1623 & 0.6450 & -0.1124 & 0.0000 \\ -0.6602 & -0.1188 & 0.7079 & -0.1830 & -0.1244 & 0.0000 \end{pmatrix} \begin{pmatrix} 33.7448 & 0 & 0 & 0 & 0 \\ 0 & 2.3277 & 0 & 0 & 0 \\ 0 & 0 & 0.6587 & 0 & 0 \\ 0 & 0 & 0 & 0.4985 & 0 \\ 0 & 0 & 0 & 0 & 0.0207 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -0.2537 & 0.6262 & 0.6581 & -0.1571 & -0.2928 \\ -0.3377 & 0.4551 & -0.1716 & 0.1735 & 0.7869 \\ -0.4296 & 0.2635 & -0.5715 & 0.3573 & -0.5402 \\ -0.5196 & -0.1832 & -0.2122 & -0.8070 & 0.0146 \\ -0.6058 & -0.5457 & 0.4073 & 0.4078 & 0.0545 \end{pmatrix}^T$$

Fig. 7.3. SVD of a 6×5 example matrix A.

When A is subtracted by the its respective column means as for the covariance matrix used for principal component analysis (PCA), the first right singular vector v_1 gives the direction along which the multi-dimensional row vectors or points contained in A have the largest variance, and the second right singular vector v_2 is the direction with the second largest variance, and so on. The singular values σ_i reflect the variance along the corresponding i^{th} singular vectors. Figure 7.4 shows the data in an 18×2 matrix and its first singular vector v_1 and its second singular vector v_2. Along the first singular vector v_1, data points have the largest variation as shown in Figure 7.4. If A has non-zero means, column means contribute to both singular values and right singular vectors.

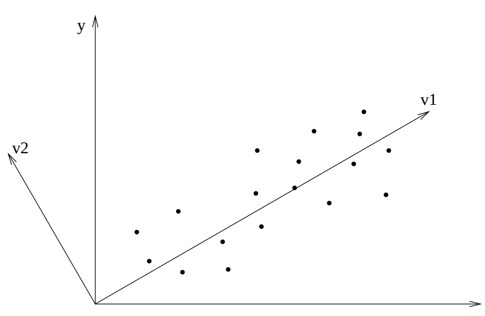


Fig. 7.4. Geometric structure of a matrix exposed by its SVD

The right singular vectors of a matrix A can be proven to be actually the corresponding eigenvectors of $M = A^T A$, and the singular values of A are the square roots of the corresponding eigenvalues of M. Hence the computations of right singular vectors and singular values can be done by computing the eigenvectors and eigenvalues of the symmetric matrix $M = A^T A$. For symbolic convenience, we will use u_1 and σ to represent the respective first singular vector and singular value vector of $M = A^T A$, and use v_1 and λ to represent those of another motion B with corresponding $M' = B^T B$.

7.4 Feature Vector Extraction Based on SVD

7.4.1 SVD Properties of Motion Data

When two motions are similar, the row vectors in the motion matrices should cover similar trajectories in the n-dimensional space, hence the geometric structures of the motion data matrices are similar. Identically, for two similar motions, all corresponding singular vectors should be close to each other, and the corresponding singular values should also be proportional to each other. For realistic motions with variations, singular vectors associated with different singular values have different sensitivities to the motion variations. If a singular value is large and well separated from its neighbors, the associated singular vector would be relatively insensitive to small motion variations. On the other hand, if a singular value is among a poorly separated cluster, its associated singular vector would be highly sensitive to motion variations.

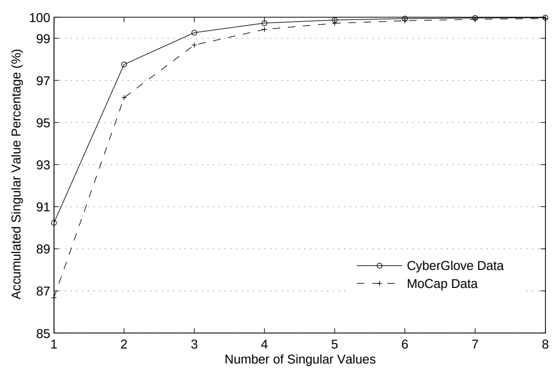


Fig. 7.5. Accumulated singular value percentages in singular value sums for two data sources: CyberGlove data and captured human body motion data. There are 22 singular values for one CyberGlove motion, and 54 singular values for one captured motion.

Figure 7.5 shows the accumulative singular values for hand gestures captured by a data glove CyberGlove and human body motions captured by multiple digital cameras as described in section 7.6. It shows that the first

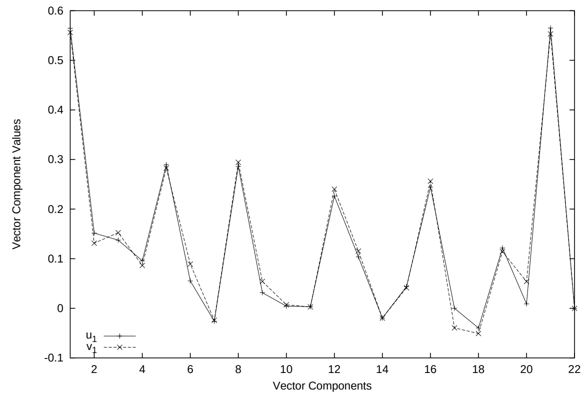


Fig. 7.6. First singular vectors u_1 and v_1 for two similar motions

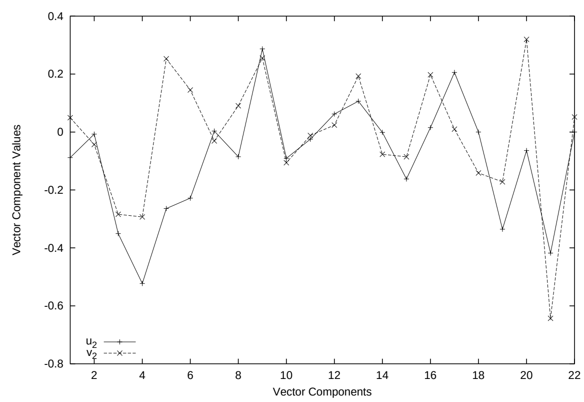


Fig. 7.7. Second singular vectors u_2 and v_2 for two similar motions. The large differences in the second singular vectors show that second singular vectors might not be close to each other due to the variations in the similar motions.

two singular values account for more than 95% of the sum of singular values, while the others might be very small. If the column means are small, the variance along the first singular vector would be the largest and the first singular value accounts for most of the variances. The first singular vectors of similar motions would be close to each other as shown in Figure 7.6 while others might not be close to each other as shown in Figure 7.7. If the column means are large, the first singular value would mainly account for column means and the associated variance might not be the largest. In this case the first singular vector would be close to the column mean vector if the direction of largest variance is different from the direction of the large column mean vector, and the second singular vector would be close to the direction with the largest variance as illustrated in Figure 7.8.

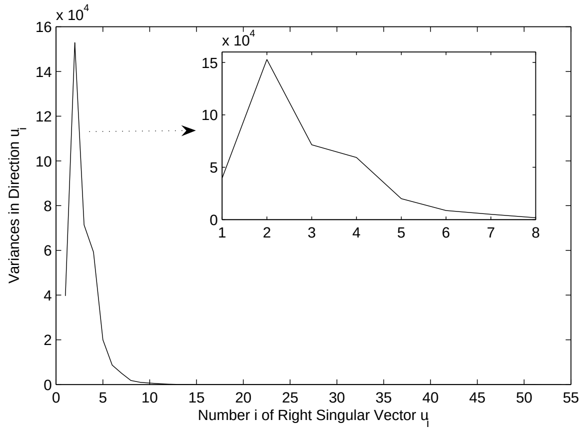


Fig. 7.8. Variances in the directions of different singular vectors. Due to large column means, variance in the direction of the first singular vector might not be the largest.

7.4.2 Feature Vector Extraction

We can extract the feature vectors from the singular vectors and singular values according to how they can capture the similarity of motions. The first singular vectors are the most dominating factors contributing to the similarity of two motions due to their associated large singular values. Other singular vectors are less reliable in capturing the similarities due to their associated singular values which might be small and approach zero. Hence we can use singular values as weights to reflect the reliability of the associated singular vectors. On the other hand, since the singular values are the Euclidean lengths or 2-norms of projections of the motion matrices A on their associated singular vectors, they reflect the shapes of the hyper-ellipsoids Ax with x being a unit 2-norm vector. The hyper-ellipsoid shapes should be similar for similar motions, hence the normalized singular values should be proportional. In other words, if the normalized singular value vectors $\boldsymbol{\sigma} = \sigma/|\sigma|$ is for one motion and $\boldsymbol{\lambda} = \lambda/|\lambda|$ is for another, then $\boldsymbol{\sigma}$ should be close to $\boldsymbol{\lambda}$ if the two motions are similar to each other. Two different feature vectors can thus be extracted as follows by using only dominating information from singular vectors and singular values.

1. The first singular vector u_1 concatenated by the normalized singular value vector $\boldsymbol{\sigma}$, or
2. The weighted first singular vector $w_1 u_1$ followed by the weighted second singular vector $w_2 u_2$, with $w_i = \sigma_i / \sum_{i=1}^{n} \sigma_i$.

Since the right singular vector u_1 can have opposite signs, the following steps can be taken to obtain consistent signs for u_1 of similar patterns.

1. Generate a matrix S with rows being the first right singular vectors u_1 of all known patterns.
2. Subtract the elements of S by their corresponding column means, and update S to be the resulting matrix with zero column means.
3. Compute the SVD of S and let its first right singular vector be s_1.
4. Project the first right singular vector u_1 of patterns (or pattern candidates) onto s_1 by computing $u_1 \cdot s_1$.
5. Negate all components of any u_1 if the corresponding the inner product $u_1 \cdot s_1 < 0$, and let u_1 be the negated vector.

As $|u_1| = 1$ and $|s_1| = 1$, the inner product $u_1 \cdot s_1 = |u_1||s_1| \cos(\alpha)$ ranges over [-1, 1], where α is the angle between the two vectors. Since the projections of u_1 onto s_1 have the largest variances among projections on any unit vectors, we can expect that $u_1 \cdot s_1$ will not cluster around zero. Our experiments with hundreds of patterns of different sources show that no pattern has $|u_1 \cdot s_1| <$ 0.3. Because similar patterns should have close projections $|u_1 \cdot s_1|$, reasonable variations in similar patterns would not result in $u_1 \cdot s_1$ projections of opposite signs if their u_1 signs are the same. That is, only if the u_1 signs of similar motions are opposite can their $u_1 \cdot s_1$ projections have different signs. Hence, u_1 of similar motions would have the same sign by requesting $u_1 \cdot s_1 > 0$.

Similarly, the above steps can be repeated for u_2 with all u_1 replaced by u_2, resulting in consistent signs for the second singular vectors of similar motions. We refer to any of the above extracted feature vectors as f if not stated otherwise.

7.5 Classification of Feature Vectors Using SVM

After a feature vector is extracted from SVD of a motion matrix, it can be classified by SVM classifiers. Before classification, the SVM classifiers need to be trained in order to have support vectors and optimal hyperplanes. All the feature vectors of training motion datasets are used as inputs to SVM classifiers for training. This training phase can be done offline (See the top portion of Figure 7.9). Similarly, we generate testing datasets and these testing data can be classified by the SVMs that have already been trained offline with training datasets.

Classification by SVM classifiers with decision values and by SVM classifiers with probability estimates in the SVM software package [19] are done for accuracy comparison, and the RBF kernel function is used for training. The type of kernel utilized by the SVM is inconsequential as long as the capacity is appropriate for the amount of training data and complexity of the classification boundary [4]. Both the training vectors and the testing vectors have the following format:

$c \quad 1{:}f_1 \; 2{:}f_{k2} \; \ldots \; 2n{:}f_{2n}$

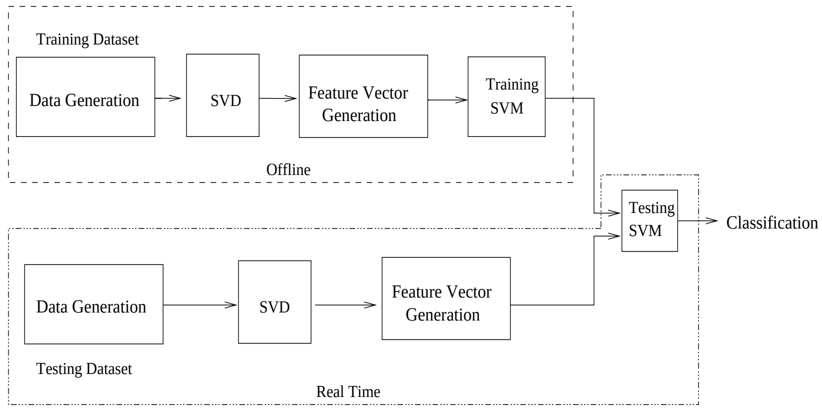


Fig. 7.9. Multi-attribute motion data classification flowchart

where c is an integer label identifying the class of the training vector f, and f_i is the i^{th} component of f.

Feature vectors for similar motions are given the same class label c, and one-versus-one multi-class SVM is employed for classification. The class label c in the testing data is used just for the purpose of obtaining classification accuracy.

For motions following similar trajectories but in different directions, it can be proved that the extracted feature vectors would be similar [1]. Further recognition of motions in different directions can be done by applying dynamic time warping distance to motion data projections on the associated first right singular vectors as shown in [1].

7.6 Performance Evaluation

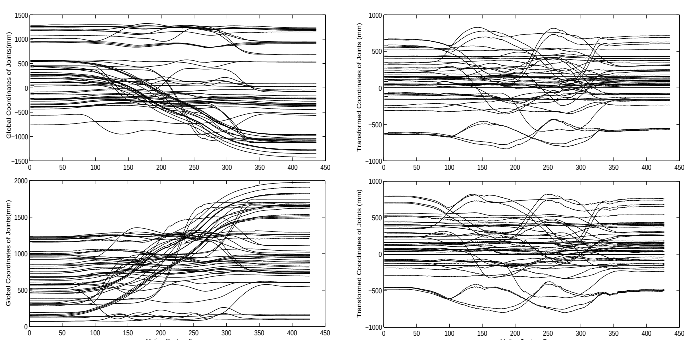
This section evaluates the extracted feature vectors by classifying them with SVM classifiers with both decision values and probability estimates. Hand gestures and various dances were generated for classification. Classification accuracy and CPU time are compared with those obtained by similarity computation using the weighted-sum SVD similarity measure [6], Eros [7], the MAS [8] and kWAS [9].

7.6.1 Hand Gesture Data Generation

We generated motion data for different hand motions using a data glove CyberGlove. Motions of a hand are captured by 22 sensors located at different positions of the glove, and one sensor generates one angular value at about 120 times/second. One hundred and ten different motions were carried out, and 3 similar motions were generated for each of them. Each motion has a different duration from all the others, and all the resultant motion data matrices have different lengths ranging from about 200 to about 1500 rows. The data matrix of one motion can have more than two times the length of the data matrix of a similar motion. Each data matrix of different motions is given a unique class label, and similar motions have the same class label.

7.6.2 Motion Capture Data Generation

The motion capture data come from various motions captured collectively by using 16 Vicon cameras and the Vicon iQ Workstation software. A dancer wears a suit of non-reflective material and 44 markers are attached to the body suit. After system calibration and subject calibration, global coordinates and rotation angles of 19 joints/segments can be obtained at about 120 frames per second for any motion. Patterns with global 3D positional data can be disguised by different locations, orientations or different paths of motion execution as illustrated in Figure 10(a). Since two patterns are similar to each other because of similar relative positions of corresponding body segments at corresponding time, and the relative positions of different segments are independent of locations or orientations of the body, we can transform the global position data into local position data.



(a) (b)

Fig. 7.10. 3D motion capture data for similar motions executed at different locations and in different orientations: (a) before transformation; (b) after transformation.

The transformed local data are positions of different segments relative to a moving coordinate system with the origin at some fixed point of the body, for example the pelvis. The moving coordinate system is not necessarily aligned with the global system, and it can rotate with the body. So data transformation includes both translation and rotation, and the transformed data would be translation and rotation invariant as shown in Figure 10(b). The coordinates of the origin pelvis are not included, thus the transformed matrices have 54 columns.

Sixty different motions including Taiqi, Indian dances, and western dances were performed for generating motion capture data, and each motion was repeated 5 times, yielding 60 classes of 300 total human motions. Every repeated motion has a different location and different durations, and can face different orientations.

7.6.3 Performance Evaluation

For hand gesture data, we divide the motions into three datasets. Each dataset has data for 110 different motions, and includes data of one of the three similar motions. Sequentially, we use two datasets for training, and the third dataset for testing. Three test cases have been run, with one different dataset used for testing in each test case. For motion capture data, five datasets are obtained by having each datasets to include 60 different motions, one for each class. All the experiments were performed on a 3.0 GHz Intel processor of a Genuine Intel Linux box, and the code was implemented in C/C++.

The performance has been validated using K-fold validation (with $K = 3$ for hand gesture motions and $K = 5$ for the captured motions). Figure 7.11 illustrates the partitioning of the hand gesture datasets for the K-fold validation (with K=3) and (a), (b) and (c) correspond to the cases with the datasets 1, 2 and 3 being the respective testing datasets. We define the accuracy as the percentage of motions classified or recognized correctly.

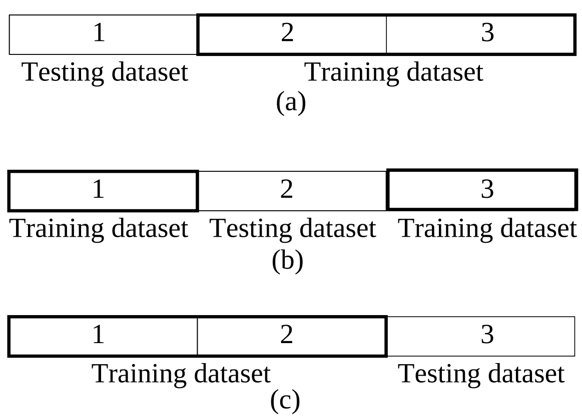


Fig. 7.11. K-fold cross validation datasets for hand gesture data, where $K = 3$.

Figures 7.12 and 7.13 show that classifiers with decision values outperform classifiers with probability estimates, while classification of the two different feature vectors can give 95-100% accuracy using both classifiers. This observation implies that although classifiers with probability estimates are more suitable than classifiers with decision values for some other applications, such as motion stream segmentation, the accuracy might decrease due to probability estimation. Classifiers with decision values are considered below for further comparison with similarity measures.

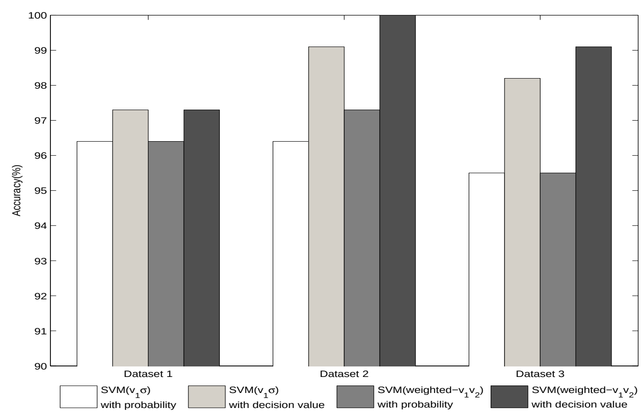


Fig. 7.12. Hand gesture classification accuracy. SVM classifiers with decision values and with probability estimates are compared, and two different feature vectors are also compared.

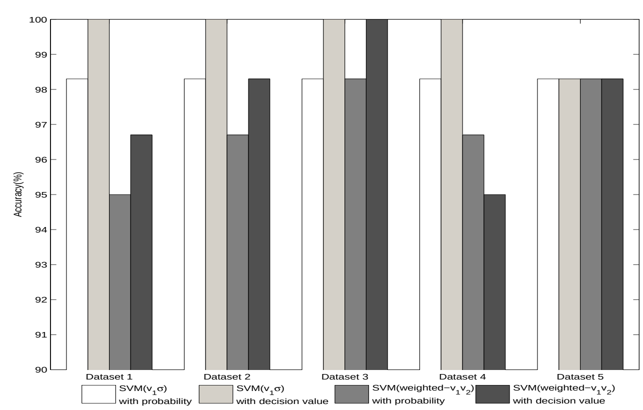


Fig. 7.13. Motion capture pattern classification accuracy.

Figure 7.14 shows that SVM outperforms all similarity measures except kWAS which includes more information than the feature vectors extracted for classification. Nevertheless, SVM classifiers still give more than 97% accuracy. In comparison, the two similar motions in each training class are used as patterns for computation of similarity with motions in the testing dataset using similarity measures. Motions in the testing datasets were recognized as the corresponding patterns with the highest similarities.

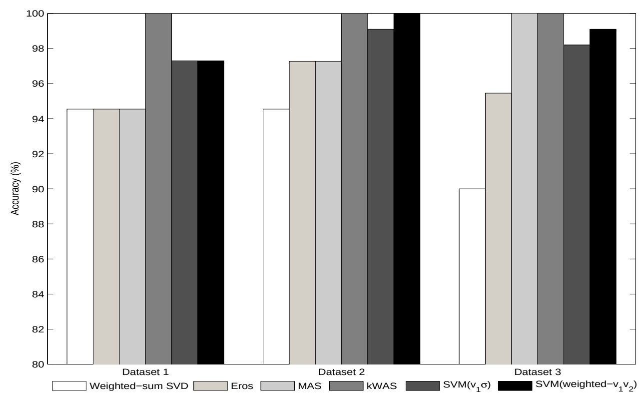


Fig. 7.14. Hand gesture classification accuracy of different approaches when two training patterns in each class are used for similarity testing. k is 5 for kWAS. The first singular vectors v_1 followed by normalized singular value vectors σ are used as the features for SVM ($v_1\sigma$), and the concatenated first two singular vectors v_1 and v_2 weighted by their corresponding singular value portions are used as the features for SVM (weighted-$v_1 v_2$).

Only one pattern for each different motion is usually assumed for similarity computation using similarity measures, while multiple similar motions are assumed for each class in the training of the SVMs. Figure 7.15 shows the recognition accuracies using similarity measures when only one pattern is used for each different motion. The proposed SVM classification approach obviously outperforms all the similarity measures in finding the most similar patterns or correctly classifying new motion patterns for all testing datasets.

Similar observations can be made for motion capture data as shown in Figures 7.16 and 7.17.

The CPU times taken by SVM classification and similarity measures are shown in Figure 7.18. For motion capture patterns, the proposed SVM classification takes CPU time comparable to that needed by MAS and kWAS, and takes less time than weighted-sum SVD and Eros. For CyberGlove patterns, classification by SVM takes a little more time than all the similarity measure approaches. We observed that after feature vectors have been extracted, clas-

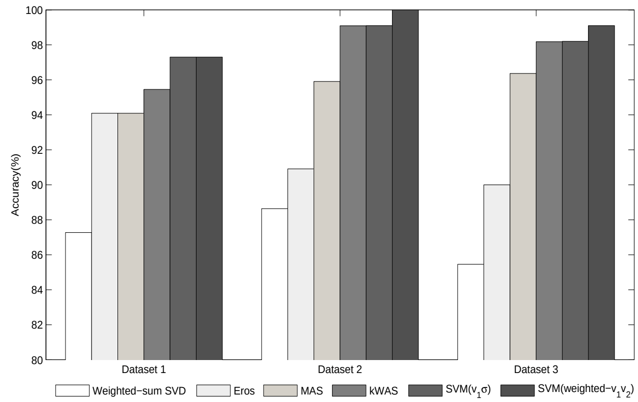


Fig. 7.15. Hand gesture classification accuracy of different approaches when only one training pattern in each class is used for similarity testing.

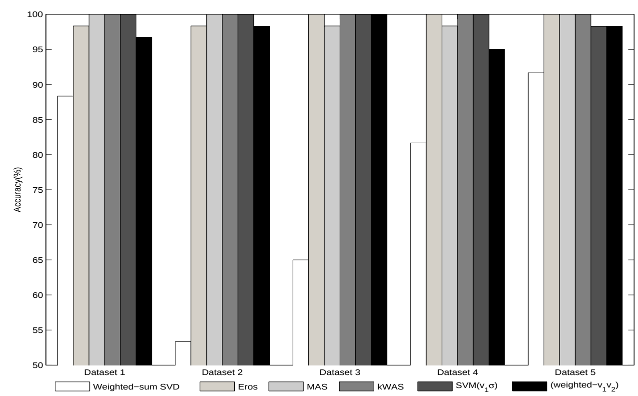


Fig. 7.16. Motion capture pattern classification accuracy of different approaches when four training patterns in each class are used for similarity testing.

sifying a CyberGlove pattern takes about 3.4 ms while classifying a motion capture takes only 1.1 ms. This is true although the feature vectors of CyberGlove patterns have only 44 attributes while the feature vectors of motion capture patterns have 108 attributes.

7.6.4 Discussion

After classification, DTW distance can be used to further determine motion directions if any class has motions following similar trajectories in different directions as discussed in [1]. The motions experimented with in this chapter, including hand gestures and captured human motions, were generated

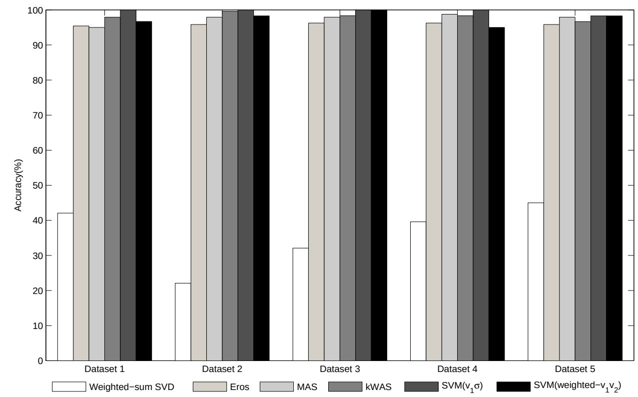


Fig. 7.17. Motion capture pattern classification accuracy of different approaches when only one training pattern in each class is used for similarity testing.

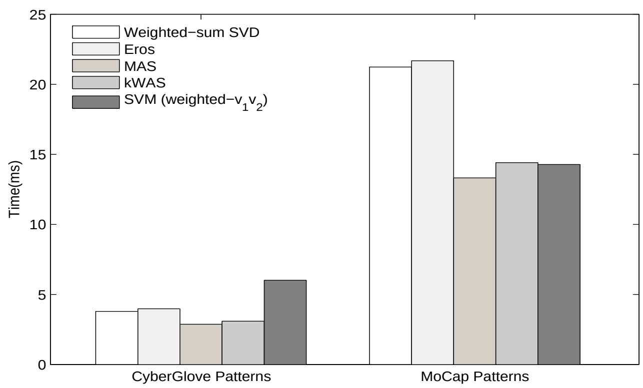


Fig. 7.18. Comparison of average CPU time taken by different approaches to classifying one motion pattern.

with different durations and different motion components can have different generation rates, showing that motions with different lengths and temporal dis-alignments can be effectively classified.

The weighted-sum SVD similarity measure computes the inner products of all the corresponding singular vectors weighted by their singular values, and takes the minimum of two sums of the inner products as the similarity measure of two matrices. Since the inner products of singular vectors can be both positive and negative, and the weights can be the singular values of either matrix, it's very likely that the weighted sum can drop or jump sharply even if a testing matrix approximately matches some training motion. In contrast, MAS does not consider all singular vectors for the similarity definition. It considers only the dominating first singular vectors and the singular values. By generating a new vector from the normalized singular values, MAS takes into consideration both the dominating direction and the shape of the motion data hyper-ellipsoid, while noises caused by variations of similar motions are reduced by not considering other singular vectors. Eros and kWAS both consider angular similarities of corresponding singular vector pairs, yet they differ in weighting the angular similarities. Eros uses the same weight vector for all patterns, while kWAS uses singular values of corresponding patterns as angular similarity weights, and the weights are different for different patterns. In contrast, the feature vectors we proposed in this chapter consider the dominating singular vectors, and similar to kWAS, the singular vectors are weighted by their associated singular values. The feature vectors do not consider as much information as kWAS, hence SVM classifiers cannot outperform kWAS if all motion patterns are considered for comparison. More singular vectors have been experimented for feature vector extracted, yet no obvious performance increases can be observed.

7.7 Conclusion

We have shown that by reducing multi-attribute motion data into feature vectors by different approaches, SVM classifiers can be used to efficiently classify multi-attribute motion data. Feature vectors are close to each other for similar motions, and are different for different motions as shown by the high accuracies of SVM classification we have achieved. RBF function has been used as the kernel function in this chapter, although other kernel functions can also be provided to the SVMs during the training process, which selects a small number of support vectors for the hyperplanes. The high accuracy and low CPU testing time make SVMs a feasible technique to classify and recognize multi-attribute data in real time.

Using only a single motion pattern in the database to recognize similar motions allows for less variations in similar motion as shown in Figure 7.15 and Figure 7.17. By reducing multi-attribute motion data into feature vectors, and using a group of feature vectors for a class, a new motion has higher expected probability of being recognized by SVMs as optimal hyperplanes are obtained during the training phase.

Two different approaches are explored to extracting feature vectors. The first approach considers the first singular vectors and the normalized singular values, while the second approach takes into account the first two dominating singular vectors weighted by their associated singular values. Motions are classified irrespective of their directions at first, and taking into consideration the first left singular vectors by using DTW can further distinguish motions following similar trajectories in different directions [1].

Our experiments with hand gestures and various captured dances further show that SVM classifiers with decision values outperform SVM classifiers with probability estimates. We have addressed the problem of real time recognition of individual isolated motions accurately and efficiently. Further work needs to be done to explore the feasibilities of the two feature vector extraction approaches for segmenting and recognizing motion streams.

Acknowledgements

We would like to thank Gaurav Pradhan for generating the CyberGlove motion data and thank Punit Kulkarni for assisting with SVM. We also appreciate the reviewers for their insightful comments. The work of Dr. Khan was supported in part by the National Science Foundation grant, NGS-0103709. The work of Dr. B. Prabhakaran was supported in part by the National Science Foundation under Grant No. 0237954 for the project CAREER: Animation Databases.

References

1. Li C, Khan L, Prabhakaran B. Real-time Classification of Variable length Multi-attribute Motion Data. International Journal of Knowledge and Information Systems (KAIS) 2005;.
2. Cristianini N, Shawe-taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge, UK: Cambridge University Press; 2000.
3. Gordan M, Kotropoulos C, Pitas I. Application of support vector machines classifiers to visual speech recognition. In: Proceedings of the International Conference on Image Processing.. vol. 3; 2002. p. 24–28.
4. Vapnik VN. Statistical Learning theory. New York: Wiley; 1998.
5. Burian A, Takala J. On signal detection using support vector machines. In: International Symposium on Signals, Circuits and Systems. vol. 2; 2003. p. 609–612.
6. Shahabi C, Yan D. Real-time pattern isolation and recognition over immersive sensor data streams. In: Proceedings of The 9th Int'l Conference on Multi-Media Modeling; 2003. p. 93–113.
7. Yang K, Shahabi C. A PCA-based similarity measure for multivariate time series. In: Proceedings of The Second ACM International Workshop on Multimedia Databases 2004; 2004. p. 65–74.
8. Li C, Zhai P, Zheng SQ, Prabhakaran B. Segmentation and Recognition of Multi-attribute Motion Sequences. In: Proceedings of The ACM Multimedia Conference 2004; 2004. p. 836–843.
9. Li C, Prabhakaran B. A Similarity Measure for Motion Stream Segmentation and Recognition. In: Proceedings of the Sixth International Workshop on Multimedia Data Mining; 2005. p. 89–94.

10. Kahveci T, Singh A, Gurel Ar. Similarity Searching for Multi-attribute Sequences. In: Proceedings. of 14th Int'l Conference on Scientific and Statistical Database Management; 2002. p. 175 – 184.
11. Lee S, Chun S, Kim D, Lee J, Chung C. Similarity Search for Multidimensional Data Sequences. In: Proceedings. of 16th Int'l Conference on Data Engineering; 2000. p. 599 – 608.
12. Vlachos M, Hadjieleftheriou M, Gunopulos Ds, Keogh E. Indexing Multi-dimensional Time-series with Support for Multiple Distance Measures. In: SIGMOD; 2003. p. 216–225.
13. Huang XD, Ariki Y, Jack MA. Hidden Markov Models for Speech Recognition. Edinburgh University Press; 1990.
14. Kundu A, He Y, Bahl P. Recognition of Handwritten Words: First and Second Order Hidden Markov Model Based Approach. Pattern Recognition 1989;22(3):283–297.
15. Starner T, Weaver J, Pentland A. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998;20(12):1371–1375.
16. Shahabi C, Kaghazian L, Mehta S, Ghoting A, Shanbhag G, McLaughlin M. Analysis of Haptic Data for Sign Language Recognition. In: Proceedings of the 9th International Conference on Human Computer Interaction; 2001. p. 441 – 445.
17. Platt JC. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Smola AJ, Bartlett PL, Scholkopf B, Schuurmans D, editors. Advances in Large Margin Classifiers. Cambridge, MA: MIT Press; 2000. URL citeseer.nj.nec.com/platt99probabilistic.html.
18. Wu TF, Liu CJ, Weng RC. Probability Estimates for Multi-class Classification by Pairwise Coupling. Journal of Machine Learning Research 2004;(5):975–1005.
19. Chang CC, Lin CJ. LIBSVM: a library for support vector machines; 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
20. Golub GH, Loan CFVan. Matrix Computations. Baltimore,Maryland: The Johns Hopkins University Press; 1996.
21. Schutter BDe, Moor BDe. The singular value decomposition in the extended max algebra. Linear Algebra and Its Applications 1997;250:143–176.