Relative Margin Support Tensor Machines for gait and action recognition

Irene Kotsia and Ioannis Patras
School of Electronic Engineering and Computer Science
Queen Mary University of London, UK, Mile End Road, London E1 4NS
{irene.kotsia,i.patras}@elec.qmul.ac.uk

ABSTRACT
In this paper, we formulate the Relative Margin Support Tensor Machines (RMSTMs) problem, as an extension of the Relative Margin Machines (RMMs). While the typical Support Tensor Machines (STMs) aim at finding a solution that is greatly influenced by the data spread, the proposed RMSTMs maximize the margin in a way relative to the spread of the data. The efficiency of the proposed method is illustrated on the problems of gait and action recognition, where the results acquired verify the superiority of the method in terms of classification performance.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

General Terms
Delphi theory

Keywords
Σ-Support Vector Machines, Relative Margin Support Vector Machines, Multilinear Support Tensor Machines, Relative Margin Support Tensor Machines,

1. INTRODUCTION
The constantly increasing need for data storage, a result of the continuously growing size of high dimensional data, has made the need for proper information representations a vital one, constituting tensors an ideal choice. Indeed, during the past few years, tensors have greatly attracted the interest of scientific community, causing research to make a turn towards their direction.

Tensors are efficient representations of multidimensional objects, whose elements can be accessed with two or more indices. The number of indices required to access a tensor defines its order, while each index specifies a mode. Thus, tensors are ideal when it comes to representing images (2nd order tensors), gait sequences (3rd order tensors), color videos (4th order tensors) etc. They can be widely used in many areas, such as 3D object recognition, 3D face reconstruction, medical image analysis, activity recognition, gait recognition etc.

Thus, during the past few years, various fundamental methods have been extended to handle tensors. Among them the Multilinear Principal Component Analysis (MPCA) [1] (extension of PCA), Support Tensor Machines (STMs) [2] (extension of Support Vector Machines (SVMs)) and Canonical Analysis Correlation of tensors [3].

More specifically, most of the methods developed for gait recognition during the past few years have used gait silhouette sequences [4], [5], [6], [7] due to their easy extraction. Several studies have shown that gait silhouettes are ideal for object or person recognition when they are detailed enough, as humans can recognize readily the identity of the object or person under investigation [8]. Lately however, gait silhouette sequences have been regarded as 3 order tensors, thus enabling their manipulation using multilinear techniques [1].

Most common techniques regarding gait recognition include Hidden Markov Models (HMMs) [7], Linear Time Normalization [4], Gait Energy Images [6], Multilinear PCA [1] and temporal correlation of extracted gait silhouette sequences [5] (the above mentioned methods constitute the ones we will compare our results with, in order to prove the superiority of the proposed method in terms of classification accuracy).

Regarding action recognition, the research conducted in the past years can be distinguished in three categories: articulated model-based approaches [9], feature-based approaches [10], and template-based approaches [11], [12]. The first category includes methods that use articulated models and perform recognition based on those acquired models. The second category is based on the tracking of landmark points, thus performing recognition based on their acquired trajectories. The third category contains silhouettes, hand shapes etc. The temporal transitions of the above mentioned templates is studied to achieve recognition. Recently, the advantages of tensors have been studied in [3] in order to achieve action recognition using video tensors.

In this paper we propose the extension of Relative Margin Support Vector Machines (RMVMs) to Relative Margin Support Tensor Machines (RMSTMs). The typical STMs provide a solution that is based on finding the maximum margin that separates the data, ignoring their spread. The way to find the maximum margin solution while taking under consideration the spread of the data in-
stead of regarding the absolute large margin solution, is presented in this paper. Thus, novel formulations that deal with the above mentioned case are introduced in the RMSTMs problem.

The rest of this paper is organized as follows. Some useful notations that will be used throughout the paper are presented in Section 2. In Section 3, the Relative Margin Machines are presented. More specifically, a short overview of the maximum margin SVMs, the Σ-SVMs and the reasoning behind RMMs is provided (Subsections 3.1, 3.2 and 3.3, respectively). In Section 4 the extensions of SVMs, Σ-SVMs and RMMs, to the novel STMs, Σ-STMs and RMSTMs are described in detail (Subsections 4.1, 4.2 and 4.3, respectively). The power of the proposed classifiers is demonstrated in the gait and action recognition problems in Section 5. Finally, conclusions are drawn in Section 6.

2. USEFUL NOTATIONS IN MULTILINEAR ALGEBRA

An n-th order tensor is a collection of measurements indexed by n indices, each index corresponding to a mode. Thus, vectors are first-order and matrices are second-order tensors [13]. We will use lower case letters (e.g. x) to denote scalars, boldface lowercase letters (e.g. X) to denote vectors and matrices, respectively. Tensors of order 3 or higher will be denoted by boldface Euler script calligraphic letters (e.g. X).

The i-th element of a vector x ∈ ℜ^n is denoted by x_i, i = 1, 2, . . . , n. In a similar way, the elements of an n-th order tensor X will be denoted by x_{i_1,i_2,...,i_n}, i_ℓ = 1, 2, . . . , n. To indicate the objects resulting by fixing one of the indices to a specific value, we introduce the generic subscript: that is, the i-th row of a matrix X is denoted as x_i. Unless otherwise stated, the j-th column of a matrix X will be denoted compactly as x_j = x_{j,i}.

The matricization (also unfolding or flattening of a tensor) is the reorderings of the tensor elements into a matrix. The n-mode matricization of a tensor A ∈ ℜ^{I_1 × I_2 × · · · × I_n}, denoted as A_{(n)}, arranges the n-mode fibers to become the columns of the final matrix. Each tensor element (i_1, i_2, . . . , i_n) maps to the matrix element (i_n, j) as

\[ j = 1 + \sum_{k=1, k \neq n}^{n} (i_k - 1)J_k, \text{ with } J_k = \sum_{k=1, k \neq n}^{n} I_k. \] (1)

In this paper, we will focus on n-th order tensors, i.e. X ∈ ℜ^{I_1 × I_2 × · · · × I_n} can represent a database consisting of L samples. Every database sample is a tensor of order (n−1) denoted as X_{i_n} ∈ ℜ^{I_1 × I_2 × · · · × I_{n−1}}.

For example, a database consisting of gait samples is a first-order tensor X ∈ ℜ^{I_1 × I_2 × · · · × I_2}, where I_1 and I_2 refer to the image dimensions (height and width), respectively, I_2 corresponds to the number of images in every tensor sample and I_1 is the number of gait samples in the database.

Let a ∈ ℜ^{I_1} and b ∈ ℜ^{I_2} be two non-negative real valued vectors. Their outer product yields a matrix C ∈ ℜ^{I_1 × I_2}

\[ C = a \otimes b \quad \text{with elements } c_{ij} = a_i b_j. \] (2)

Consequently, the outer product of n vectors \( a_1 ∈ ℜ^{I_{1}}, \ldots, a_n ∈ ℜ^{I_{n}} \) yields a tensor A ∈ ℜ^{I_1 × I_2 × · · · × I_n}.

An important operation between a tensor X ∈ ℜ^{I_1 × I_2 × · · · × I_n} and a matrix U ∈ ℜ^{J_1 × I_1} is the f-mode product denoted as X ⊗ f U which yields a tensor y of size I_1 × I_2 × · · · × I_{f−1} × J × I_{f+1} × · · · × I_n having elements [13]

\[ y_{i_1,\ldots,i_{f−1},j,i_{f+1},\ldots,i_n} = \sum_{i_f} x_{i_1,i_2,\ldots,i_{f−1},i_f} u_{j,i_f}, \quad j = 1,2,\ldots,J \] (3)

In the remaining of the paper, ⊗ and / operators will denote the elementwise multiplication and division between tensors, vectors and matrices.

3. RELATIVE MARGIN SUPPORT VECTOR MACHINES

3.1 Support Vector Machines (SVMs)

Let a dataset \((x_i, y_i)_{i=1}^n\), where \(x_i \in ℜ^n\) with \(y_i \in \{±1\}\). The maximum margin Support Vector Machines (SVMs) formulation is given as:

\[ \min_{w,b,ξ \geq 0} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} ξ_i \quad \text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - ξ_i, \forall 1 \leq i \leq n. \] (5)

The above formulation aims at maximizing the margin of the Support Vectors while minimizing the upper bound on the misclassification errors.

3.2 Σ-Support Vector Machines (Σ-SVMs)

Below we will briefly present the Σ-SVMs, as proposed in [14]. Let's consider the whitening of the data with the covariance (total scatter) matrix:

\[ Σ = \sum_{i=1}^{n} x_i x_i^T - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j^T. \] (6)

Let us also consider \( μ = \frac{1}{n} \sum_{i=1}^{n} x_i \), the mean value of the data.

The formulation of Σ-SVMs is given by:

\[ \min_{w,b,ξ \geq 0} \frac{1}{2} ||w||^2 + D ||Σ^\frac{1}{2} w||^2 + C ξ^T 1 \]

\[ \text{s.t. } y_i \left( w^T (x_i - μ) + b \right) \geq 1 - ξ_i, \]

where \( 0 \leq D \leq 1 \) is the parameter that handles the two regularization terms, ||w||^2 and ||Σ^\frac{1}{2} w||^2.

The dual problem of (7) is then given by:

\[ \max_{0 \leq α_i \leq C} \sum_{i=1}^{n} α_i - \frac{1}{2} \sum_{i,j=1}^{n} α_i α_j y_i (x_i - μ)^T ((1 - D) I + D Σ)^{-1} x_j - μ \]

\[ \text{s.t. } y_i^T α_i = 1. \] (8)
3.3 Relative Margin Machines (RMMs)

Relative Margin Machines (RMMs) were introduced to deal with a possible bad scaling of the data [14]. In order to achieve that, the bounding of the projections of the training data was used. The trade off is now between the projections and the margin, resulting in finding a large relative margin.

The Relative Margin Machines (RMMs) are given by the following formulation:

\[
\min_{w, \xi, \alpha, \beta, \lambda} \frac{1}{2} ||w||^2 + C \xi^T 1
\]

s.t. \( y_i (w^T x_i + b) \geq 1 - \xi_i \),

\[
\frac{1}{2} (w^T x_i + b)^2 \leq \frac{\xi_i}{C}
\]

As can be seen, the above formulation has one extra parameter in addition to the SVMs parameters, \( B \) (where \( B \geq 1 \)). Let us denote as \( w_C \) and \( b_C \) the solutions obtained by solving the SVM (5) for a particular value of \( C \). If \( B > \max_i (w_C^T x_i + b_C) \), then the solution obtained is the same with the SVM estimate. If \( B \) is of a smaller value, the solutions obtained are different than that of the SVM estimate.

Let us assume that the value of \( B \) is smaller than the threshold. Then, the Lagrangian of (9) is the following:

\[
L_{RMMs}(w, \xi, \alpha, \beta, \lambda, B) = \frac{1}{2} ||w||^2 + C \xi^T 1
\]

- \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b - 1 + \xi_i) - \beta^T \xi +

\sum_{i=1}^{n} \lambda_i \left( \frac{1}{2} (w^T x_i + b)^2 - \frac{1}{B} B^2 \right)
\]

where \( \alpha, \beta, \lambda \geq 0 \) are the Lagrangian multipliers that correspond to the constraints.

Differentiating with respect to the primal variables and equating them to zero, it can be shown that:

\[
(1 + \sum_{i=1}^{n} \lambda_i x_i x_i^T) w - b \sum_{i=1}^{n} \lambda_i x_i = \sum_{i=1}^{n} \alpha_i y_i x_i,
\]

\[
b = \frac{1}{\lambda} \left( \sum_{i=1}^{n} \alpha_i y_i - \sum_{i=1}^{n} \lambda_i w^T x_i \right)
\]

C1 = \( \alpha + \beta \).

Denoting by:

\[
\Sigma_\lambda = \sum_{i=1}^{n} \lambda_i x_i x_i^T - \frac{1}{\lambda} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i x_i \lambda_j x_j
\]

and by:

\[
\mu_\lambda = \frac{1}{\lambda} \sum_{j=1}^{n} \lambda_j x_j,
\]

the dual of (9) can be shown to be:

\[
\max_{0 \leq \alpha \leq C, \lambda \geq 0} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i (x_i - \mu_\lambda)^T (1 + \Sigma_\lambda)^{-1} \sum_{j=1}^{n} \alpha_j y_j (x_j - \mu_\lambda) - \frac{1}{4} B^2 \lambda^2 1
\]

s.t. \( y_i \alpha = 1 \).

\( \Sigma_\lambda \) corresponds to a ”shape matrix” (potentially row rank) determined by the \( x_i \)s that have nonzero \( \lambda_i \). From the KKT conditions of (9) we have:

\[
\lambda_i \left( \frac{1}{2} (x_i w_k^T + b) + b \right)^2 - \frac{B^2}{2} = 0.
\]

Consequently \( \lambda_i > 0 \) implies that:

\[
\left( \frac{1}{2} (x_i w_k^T + b) + b \right)^2 - \frac{B^2}{2} = 0.
\]

Note that the constraint \( \frac{1}{2} (x_i w_k^T + b)^2 \leq \frac{B^2}{2} \) can be equivalently posed as two linear constraints: \( (x_i w_k^T + b) \leq B \) and \(- (x_i w_k^T + b)^2 \leq B \). Thus the problem to solve is a quadratic one.

4. RELATIVE MARGIN SUPPORT TENSOR MACHINES

The proposed method involves the extension of the RMMs proposed in [14] in order to deal with tensors. We will make a brief description of the typical Support Tensor Machines (STMs) [2] and continue with a detailed description of the novel \( \Sigma \)-STMs and Relative Margin Support Tensor Machines (RMSTMs).

4.1 Support Tensor Machines (STMs)

Let a dataset be represented by the tensor \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n} \), where \( I_n \) is the number of samples in the dataset. The dataset is separated into two classes with \( I_n^0 \) and \( I_n^1 \) denoting the number of samples of each class. The label \( y_i = 1 \) is assigned to the samples belonging to the first class, while the label \( y_i = -1 \) is assigned to the samples belonging to the second class.

For the maximum margin STMs we aim at finding a multilinear decision function:

\[
g(X) = \text{sign} \left[ X^{n-1} \prod_{i=1}^{n} x_i w_i + b \right].
\]

The projection vectors \( w_j \in \mathbb{R}^{I_j} \) for every dimension \( j = 1, \ldots, n \) and the bias term \( b \) are derived from solving the following soft STM problem:

\[
\min_{w_j \in \mathbb{R}^{I_j}, b, \xi_i} \left( \sum_{i=1}^{I_n} \xi_i \right)
\]

s.t. \( y_i \left[ X \prod_{i=1}^{n} x_i w_k + b \right] \geq 1 - \xi_i, \ 1 \leq i \leq I_n, \ \xi_i \geq 0. \)

If we keep every term but \( w_j \) fixed, then the problem becomes convex and quadratic. Otherwise, the above form of the optimization problem is not convex with respect to all projection vectors \( w_k \) with \( k = 1, \ldots, I_n \).

The \( j \)-th problem for solving with respect to \( w_j \) is given by:

\[
\min_{w_j \in \mathbb{R}^{I_j}, b, \xi_i} \left( \sum_{i=1}^{I_n} \xi_i \right)
\]

s.t. \( y_i \left[ w_j^T (X_i \times_j X_k) + b \right] \geq 1 - \xi_i, \ 1 \leq i \leq I_n, \ \xi_i \geq 0. \)

where \( g_j = \sum_{k=1, k \neq j}^{n} ||w_k||^2. \) The optimal vector \( w_j \) can be found by the saddle point of the Lagrangian:

\[
L^{(j)}_{STM}(w_j, b, \xi_i) = \frac{1}{2} ||w_j||^2 + C \sum_{i=1}^{I_n} \xi_i - \sum_{i=1}^{I_n} \alpha_i \left( \xi_i - \sum_{i=1}^{I_n} \alpha_i y_i \left[ w_j^T (X_i \times_j X_k) + b \right] - 1 - \xi_i \right) - \sum_{i=1}^{I_n} \alpha_i \xi_i
\]

as

\[
\nabla w_j L^{(j)}_{STM} = 0 \Rightarrow w_j = \frac{1}{\eta_j} \sum_{i=1}^{I_n} \alpha_i y_i (X_i \times_j X_k) w_k.
\]

The whole procedure is repeated iteratively for every mode, so as to find \( w_k, \ k = 1, \ldots, M \).
4.2 Σ-STM

In order to define the Σ-STM formulation, we follow the rationale behind Σ-SVMs. More specifically, let us define the n-mode scatter matrices as:

$$\Sigma_{(n)} = \sum_{k=1}^{m} (X_{k(n)} - \bar{X}_{(n)})(X_{k(n)} - \bar{X}_{(n)})^T$$

(20)

where $X_{k(n)}$ is the n-mode unfolding of the tensor sample $X_k$ and $\bar{X}_{(n)}$ is the mean value of $X_{k(n)}$.

Thus, the formulation of the Σ-STM is the following:

$$\min_{w_j, b, \xi \geq 0} \frac{1}{2}D \sum_{k=1}^{n-1} \left| \Sigma_{(1)}^{-1} w_k \right|^2 + \frac{C}{2} \sum_{i=1}^{m} \xi_i + C \sum_{i=1}^{m} \xi_i$$

s.t. $y_i \left( \langle \bar{X}_i, \Sigma_{(1)}^{-1} w \rangle + b \right) \geq 1 - \xi_i, \quad 1 \leq i \leq m, \quad \xi_i \geq 0.$

(21)

where $\bar{X}_i = \frac{1}{n} \sum_{k=1}^{n} X_i$ is the mean tensor for the mode (n) and $0 \leq D \leq 1$ is the parameter that handles the two regularization terms $\left| \Sigma_{(1)}^{-1} w \right|^2$ and $\left| \Sigma_{(1)}^{-1} \Sigma_{(2)} w \right|^2$.

For the j-th vector $w_j$, the above optimization problem is reformulated as:

$$\min_{w_j, b, \xi \geq 0} \frac{1}{2}D \eta_j w_j^T w_j + \frac{C}{2} \sum_{i=1}^{m} \xi_i + C \sum_{i=1}^{m} \xi_i$$

s.t. $y_j \left( \langle \bar{X}_j, \Sigma_{(1)}^{-1} w \rangle + b \right) \geq 1 - \xi_j, \quad 1 \leq j \leq m, \quad \xi_j \geq 0.$

(22)

where

$$\eta_j = \prod_{i=1, i \neq j}^{n-1} w_i^T \Sigma_{(j)} w_i.$$  

The Lagrangian is given by:

$$\mathcal{L}^{(j)}_{\Sigma-STM} = \left( (1-D) \eta_j + D \eta_j \Sigma_{(j)} \right) \theta_j + C \sum_{i=1}^{m} \xi_i - \frac{1}{2} a_i^T \left( \langle \bar{X}_i, \Sigma_{(1)}^{-1} w \rangle + b \right) - 1 + \xi_i - \frac{1}{2} \sum_{j=1}^{m} \alpha_j \theta_j.$$  

(23)

By letting $\Sigma_{(j)} = \left( (1-D) \eta_j + D \eta_j \Sigma_{(j)} \right)$, we have:

$$\nabla_{\theta_j} \mathcal{L}_{\Sigma-STM} = 0 \Rightarrow \nabla \theta_j = \sum_{i=1}^{m} a_i y_i (\theta_j, X_i - \bar{X}_j)^T$$

$$w_j = \sum_{i=1}^{m} \alpha_i y_i (\theta_j, X_i - \bar{X}_j)^T w_k.$$  

(24)

The whole procedure is repeated iteratively for every mode, so as to find $w_k, \quad k = 1 \ldots M$.

The dual problem of (22) is then defined as:

$$\max_{0 \leq \alpha_i \leq \lambda_i \geq 0} \frac{1}{2} \sum_{i=1}^{m} a_i^T \langle \bar{X}_i, \bar{X}_j \rangle \alpha_j \quad \theta_j = 0.$$  

(25)

subject to $\alpha_i^T \theta_j = 0.$

(26)

4.3 Relative Margin Support Tensor Machines

Following the same reasoning with the one used to define RMMs, we also use the projections of the training data to define Relative Margin Support Tensor Machines (RMSTMs). Their formulation is as follows:

$$\min_{w_j, b, \xi \geq 0} \frac{1}{2} \eta_j w_j^T w_j + C \sum_{i=1}^{m} \xi_i$$

s.t. $y_j \left( \langle \bar{X}_j, \Sigma_{(1)}^{-1} w \rangle + b \right) \geq 1 - \xi_j, \quad 1 \leq j \leq m, \quad \xi_j \geq 0.$

(27)

The partial optimization problem in terms of $w_j$ is given by:

$$\min_{w_j, b, \xi \geq 0} \frac{1}{2} \eta_j w_j^T w_j + C \sum_{i=1}^{m} \xi_i$$

s.t. $y_j \left( \langle \bar{X}_j, \Sigma_{(1)}^{-1} w \rangle + b \right) \geq 1 - \xi_j, \quad 1 \leq j \leq m, \quad \xi_j \geq 0.$

(28)

As can be seen, an extra parameter $B$ (where $B>1$) is introduced. If $w_C$ and $b_C$ the solutions acquired for the maximum margin STM problem for a specific value $C$ and

$$B = \max \{ \| w_j^T C X_i X_j w_r \| + b C \},$$

then the solution is the same for both STMs and RMSTMs. If however $B$ is of a smaller value, then the solution is different.

Let us assume that the value of $B$ is smaller than the threshold.

Then, the Lagrangian of (28) is given by:

$$L_{RMSTM}(w_j, a^j, \beta, \alpha, \lambda^j) = \frac{1}{2} \eta_j w_j^T w_j + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i \left( y_i w_j^T \langle \bar{X}_i, \Sigma_{(1)}^{-1} \Sigma_{(2)} w \rangle + b \right) - 1 + \xi_i - \sum_{i=1}^{m} \lambda_i \left( \frac{1}{2} w_j^T \langle \bar{X}_i, \Sigma_{(1)}^{-1} \Sigma_{(2)} w \rangle + b \right)^2 - \frac{1}{2} B^2$$

(29)

where $\alpha, \beta, \lambda \geq 0$ are the Lagrangian multipliers corresponding to the constraints. Differentiating with respect to the primal variables and equating them to zero, it can be shown that:

$$\langle \eta_j I + \sum_{i=1}^{m} \lambda_i (X_{i}, \Sigma_{(1)}^{-1} \Sigma_{(2)} w) \rangle - \sum_{i=1}^{m} \alpha_i \eta_j \overline{X}_j = 0 \Rightarrow$$

$$b = \frac{1}{\lambda_i} \sum_{i=1}^{m} \alpha_i y_i - \frac{1}{\lambda_i} \sum_{i=1}^{m} \alpha_i w_j^T \langle \bar{X}_i, \Sigma_{(1)}^{-1} \Sigma_{(2)} w \rangle \overline{X}_j$$

(30)

Denoting by:

$$\Sigma_{\lambda} = \sum_{i=1}^{n} \lambda_i \overline{X}_i \overline{X}_j \overline{X}_j \overline{X}_i \overline{X}_i \overline{X}_j \overline{X}_i = \frac{1}{\lambda_i} \sum_{i=1}^{m} \alpha_i y_i - \frac{1}{\lambda_i} \sum_{i=1}^{m} \alpha_i w_j^T \langle \bar{X}_i, \Sigma_{(1)}^{-1} \Sigma_{(2)} w \rangle \overline{X}_j$$

by $\lambda_{\alpha}$ and $\lambda_{\beta}$, the dual of (29) can be shown to be:

$$\max_{0 \leq \alpha_i \leq \lambda_{\alpha} \geq 0, \quad \alpha_i \leq \lambda_{\beta} \geq 0} \sum_{i=1}^{m} \alpha_i \left( \frac{1}{2} \alpha_i^2 + \frac{1}{2} \lambda_{\alpha} \lambda_{\beta} \right) - \frac{1}{2} \lambda_{\beta} B^2$$

subject to $y_j \left( \langle \bar{X}_j, \Sigma_{(1)}^{-1} \Sigma_{(2)} w \rangle + b \right) \geq 1 - \xi_j, \quad 1 \leq j \leq m, \quad \xi_j \geq 0.$

(31)

From the KKT conditions of (29) we have:

$$\lambda_i \left( \frac{1}{2} \langle \bar{X}_i, \Sigma_{(1)}^{-1} \Sigma_{(2)} w \rangle + b \right)^2 \leq \frac{B^2}{2}.$$  

(32)
4.1 Gait recognition experiments

The database includes 452 sequences from 74 subjects (persons) walking in elliptical paths in front of the camera. Three variations (referred to as probe sets) are available, containing each one 71 sequences from each subject. The probe sets are increasing the complexity/difficulty with probe set A being the easiest one and probe set G being the most difficult. There are no common sequences between the gallery sets and any of the probe sets and each probe set is unique. The largest dimension of the gait samples contained in the database, i.e., tensors of dimension $128 \times 88 \times 40$ were used for the experiments. An example of a gait sequence can be seen in Figure 1.

The acquired results for all probe sets when STMs and RMSTMs are shown in the last two columns of Tables 1 and 2 (rank 1 and 5 identification rates, respectively). The highest accuracy rate for every probe set is emphasized in bold. In the same Tables, a comparison of the recognition rates achieved for each probe set with the state of the art [1], [4], [5], [6], [7], is also provided.

As can be seen, the recognition rates of the proposed method are the highest for probe sets C to G, for rank 1 results. The acquired mean accuracy rate of RMSTMs is also higher than that of all other mentioned methods. For rank 5 results, the recognition rates (and the mean accuracy rates) of STMs and the proposed RMSTMs are the highest for all probe sets. It is also worth mentioning here that in state of the art methods, the recognition accuracy drops drastically as the difficulty of the probe set increases, while in STMs and the proposed RMSTMs the accuracy rates remain in satisfactory levels. This is due to the use of tensors for the description of the features used for classification.

More specifically, one can see that when all the frames are considered as a tensor from which features are to be extracted, instead of extracting features separately in a vector format and combining them afterwards, the spatial and temporal correlation of the initial information is preserved. For example, the geometry of spatial points is taken under consideration, something that does not happen when vectors are used. Even in the simplest case when an image is considered as a 2 mode tensor instead of a vector (as we will show below in the action recognition experiments 5.2), the results can be very promising.

5. EXPERIMENTAL RESULTS

In this Section, we will present the acquired experimental results in order to justify the superiority of RMSTMs over STMs. The gait and actions recognition problems will be studied. In our experiments, the leave-one-out cross-validation approach was used to test the generalization performance of the classifiers. The experiments were performed on an Intel Core 2 Quad PC (2.66 GHz) processor with 4GB RAM memory.

5.1 Gait recognition experiments

The database used for the gait recognition experiments was the USF HumanID Gait Challenge benchmark set version 1.7, as used in [1], for comparison reasons.

The database includes 452 sequences from 74 subjects (persons) walking in elliptical paths in front of the camera. Three variations are provided for each subject: viewpoint (left/right), shoe type (two different types) and surface type (grass/concrete). Seven experiments (referred to as probe sets) are available, containing each one 71 sequences from each subject. The probe sets are increasing the complexity/difficulty with probe set A being the easiest one and probe set G being the most difficult. There are no common sequences between the gallery sets and any of the probe sets and each probe set is unique. The largest dimension of the gait samples contained in the database, i.e., tensors of dimension $128 \times 88 \times 40$ were used for the experiments. An example of a gait sequence can be seen in Figure 1.
Figure 1: An example of a gait sequence.

Figure 2: An example of the action sequences (first column), spatiotemporal points detected (second column) and binary masks extracted for action classification (third column).
Table 1: Comparison of proposed method with state of the art (rank 1).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>79</td>
<td>99</td>
<td>94</td>
<td>100</td>
<td>92</td>
<td>99</td>
<td>78</td>
<td>81</td>
</tr>
<tr>
<td>B</td>
<td>66</td>
<td>89</td>
<td>83</td>
<td>85</td>
<td>85</td>
<td>88</td>
<td>75</td>
<td>77</td>
</tr>
<tr>
<td>C</td>
<td>56</td>
<td>78</td>
<td>78</td>
<td>80</td>
<td>76</td>
<td>83</td>
<td>83</td>
<td>86</td>
</tr>
<tr>
<td>D</td>
<td>29</td>
<td>35</td>
<td>35</td>
<td>30</td>
<td>39</td>
<td>56</td>
<td>56</td>
<td>78</td>
</tr>
<tr>
<td>E</td>
<td>24</td>
<td>29</td>
<td>24</td>
<td>33</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>81</td>
</tr>
<tr>
<td>F</td>
<td>30</td>
<td>15</td>
<td>17</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>75</td>
</tr>
<tr>
<td>G</td>
<td>10</td>
<td>24</td>
<td>21</td>
<td>29</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>82</td>
</tr>
<tr>
<td>Average</td>
<td>42</td>
<td>55</td>
<td>50</td>
<td>54</td>
<td>52</td>
<td>54</td>
<td>54</td>
<td>82</td>
</tr>
</tbody>
</table>

Table 2: Comparison of proposed method with state of the art (rank 5).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>96</td>
<td>100</td>
<td>99</td>
<td>100</td>
<td>96</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>81</td>
<td>90</td>
<td>85</td>
<td>85</td>
<td>90</td>
<td>93</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C</td>
<td>76</td>
<td>90</td>
<td>83</td>
<td>88</td>
<td>81</td>
<td>88</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>D</td>
<td>61</td>
<td>65</td>
<td>65</td>
<td>55</td>
<td>55</td>
<td>71</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>E</td>
<td>55</td>
<td>65</td>
<td>67</td>
<td>55</td>
<td>52</td>
<td>60</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>F</td>
<td>46</td>
<td>60</td>
<td>58</td>
<td>41</td>
<td>58</td>
<td>59</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>G</td>
<td>38</td>
<td>50</td>
<td>38</td>
<td>47</td>
<td>48</td>
<td>50</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Average</td>
<td>64</td>
<td>74</td>
<td>72</td>
<td>67</td>
<td>69</td>
<td>66</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

The recognition accuracies achieved when STMs and RMSTMs were used, were equal to 84.46% and 87.76%, respectively. Thus, the use of RMSTMs introduces an increase of 3.3%. In order to better study the problem, the confusion matrices have been computed. The confusion matrix is a \( n \times n \) matrix containing information about the actual class label \( \text{Action}_{\text{ac}} \) (in its columns) and the label obtained through classification \( \text{Action}_{\text{cl}} \) (in its rows). The diagonal entries of the confusion matrix are the percentages that correspond to the cases when actions are correctly classified, while the off-diagonal entries correspond to misclassifications. The confusion matrix when STMs were used is given in Table 3, while the equivalent results when RMSTMs were used are provided in Table 4.

As can be seen from the confusion matrices, the use of RMSTMs improves the recognition accuracy results in the cases when most misclassifications were observed. More specifically, the misclassification of jump as bend and that one of pjump as wave2 are now omitted. Thus, even the use of even an image as a 2-mode can provide satisfactory results when tensors are used.

6. CONCLUSIONS

In this paper the novel Relative Margin Support Tensor Machines are proposed as an extension of the Relative Margin Support Vector Machines. The proposed RMSTMs exploit the spread of the data in order to find the final solution. The efficiency of the proposed method has been studied for the problems of gait and action recognition, where the results achieved illustrated the superiority of the method in terms of classification performance.

7. ACKNOWLEDGEMENT

This work was supported by the EPSRC grant ‘Recognition and Localization of Human Actions in Image Sequences’ (EP/G033935/1).

8. REFERENCES

Table 3: Confusion matrix of STMs.

<table>
<thead>
<tr>
<th>Action</th>
<th>Bend</th>
<th>Jack</th>
<th>Jump</th>
<th>Pjump</th>
<th>Run</th>
<th>Side</th>
<th>Skip</th>
<th>Walk</th>
<th>Wave1</th>
<th>Wave2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bend</td>
<td>100.0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jack</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jump</td>
<td>0</td>
<td>0</td>
<td>33.4</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pjump</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>66.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Run</td>
<td>0</td>
<td>0</td>
<td>22.2</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Side</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Skip</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>66.7</td>
<td>0</td>
</tr>
<tr>
<td>Walk</td>
<td>0</td>
<td>0</td>
<td>33.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>88.9</td>
</tr>
<tr>
<td>Wave1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>88.9</td>
</tr>
<tr>
<td>Wave2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
</tr>
</tbody>
</table>

Table 4: Confusion matrix of RMSTMs.

<table>
<thead>
<tr>
<th>Action</th>
<th>Bend</th>
<th>Jack</th>
<th>Jump</th>
<th>Pjump</th>
<th>Run</th>
<th>Side</th>
<th>Skip</th>
<th>Walk</th>
<th>Wave1</th>
<th>Wave2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bend</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jack</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jump</td>
<td>0</td>
<td>0</td>
<td>44.5</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pjump</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>77.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Run</td>
<td>0</td>
<td>0</td>
<td>22.2</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Side</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Skip</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>66.7</td>
<td>0</td>
</tr>
<tr>
<td>Walk</td>
<td>0</td>
<td>0</td>
<td>33.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>88.9</td>
</tr>
<tr>
<td>Wave1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>88.9</td>
</tr>
<tr>
<td>Wave2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1</td>
</tr>
</tbody>
</table>

2007.


