Multiplicative Update rules for Multilinear Support Tensor Machines

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 Multilinear Support Tensor Machines (MSTMs) problem in a similar to the Non-negative Matrix Factorization (NMF) algorithm way. A novel set of simple and robust multiplicative update rules are proposed in order to find the multilinear classifier. Updates rules are provided for both hard and soft margin MSTMs and the existence of a bias term is also investigated. We present results on standard gait and action datasets and report faster convergence of equivalent classification performance in comparison to standard MSTMs.

1 Introduction

Tensors have been an active research field for the past few years. They constitute efficient representations of multidimensional objects. Thus, they are often used to represent images (2nd order tensors), grayscale videos (3rd order tensors), color videos (4th order tensors) etc. Their applications span various areas, such as 3D face reconstruction, 3D object recognition, medical image analysis, activity recognition etc.

For this reason, various fundamental methods have been extended to handle tensors, such as the extension of Principal Component Analysis (PCA) to Multilinear PCA [4] and of Support Vector Machines (SVMs) to Support Tensor Machines (STMs) [7].

In [6], the objective function of SVMs has been reformulated taking under consideration the update rules used in the NMF algorithm, to result in a new set of update rules for soft and hard margin SVMs. In this paper we reformulate the dual MSTMs problem [7] as a matrix factorization problem inspired by the NMF update rules. To do so, we extend the method presented in [6] to handle higher order tensors as input. In this way, we extract a novel set of multiplicative updates in order to reach a solution for MSTMs, thus establishing the relationship between dual MSTMs and NMF algorithms.

The rest of this paper is organized as follows. Some useful notations are presented in Section 2. In Section 3, the problem of Multilinear Support Tensor Machines is formulated. The power of the proposed classifiers is demonstrated for the gait and action recognition problems in Section 4. Finally, conclusions are drawn in Section 6.

2 Useful Notations in Multilinear Algebra

The notations that we will use in this paper are consistent with the ones presented in [3],[4].

Moreover, for a matrix $\mathbf{G}$ we define the matrices $\mathbf{G}^+$ and $\mathbf{G}^-$ as

$$G_{ij}^{+} \triangleq \begin{cases} G_{ij} & \text{if } G_{ij} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

and

$$G_{ij}^{-} \triangleq \begin{cases} |G_{ij}| & \text{if } G_{ij} \leq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

3 Multilinear Support Tensor Machines

Let a dataset be represented by the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ where I_n is the number of samples in the dataset. The dataset is separated into two classes with I_n^A and I_n^B denoting the number of samples of each class. The samples of the two classes are stored in tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n^A}$ and $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n^B}$ with $I_n = I_n^A + I_n^B$. The label $y_i = 1$ is assigned to the samples $\mathcal{A}_{:i} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1}}$ with $i = 1, \ldots, I_n^A$ belonging to the first class , while label $y_i = -1$ is assigned to the samples $\mathcal{B}_{:i} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1}}$ with $i = 1, \ldots, I_n^B$ belonging to the second class.

We aim at finding a multilinear decision function $g(\mathcal{X}) = \text{sign}\left[\mathcal{X} \prod_{i=1}^{n-1} \times_i \mathbf{w}_i + b\right]$. The projection vectors $\mathbf{w}_j \in \mathbb{R}^{I_j}$ for every dimension $j = 1, \ldots, n$ and

the bias term b are derived from solving the following soft STM problem:

$$\begin{array}{l}\min_{\mathbf{w}_j|_{j=1}^M,b,\boldsymbol{\xi}} \frac{1}{2}\left|\left|\bigotimes_{k=1}^{n-1}\mathbf{w}_k\right|\right|^2 + C\sum_{i=1}^{I_n}\xi_i \\ \text{s.t. } y_i\left[\boldsymbol{\mathcal{X}}_{:i}\prod_{i=1}^{n-1}\times_k\mathbf{w}_k + b\right] \geq 1-\xi_i, \quad 1\leq i\leq I_n \\ \quad \xi_i \geq 0.\end{array} \tag{3}$$

The above optimization problem is not convex with respect to all projection vectors $\mathbf{w}_k$ with $k = 1,\ldots,I_n$, but is convex (and quadratic) if every term but $\mathbf{w}_j$ is kept fixed. If alternating optimization is used the j-th problem for solving with respect to $\mathbf{w}_j$ is given by:

$$\begin{array}{l}\min_{\mathbf{w}_j|_{j=1}^M,b,\boldsymbol{\xi}^j} \frac{\eta_j}{2}||\mathbf{w}_j||^2 + C\sum_{i=1}^{I_n}\xi_i^j \\ \text{s.t. } y_i\left[\mathbf{w}_j^T(\boldsymbol{\mathcal{X}}_{:i}\overline{\times_j}\mathbf{w}_k) + b\right] \geq 1-\xi_i^j, \quad 1\leq i\leq I_n \\ \quad \xi_i \geq 0\end{array} \tag{4}$$

where $\eta_j = \prod_{k=1,k\neq j}^{n}||\mathbf{w}_k||^2$. The optimal vector $\mathbf{w}_j$ can be found by the saddle point of the Lagrangian:

$$\begin{array}{l}L_j(\mathbf{w}_j,b,\boldsymbol{\xi}^j) = \frac{\eta_j}{2}||\mathbf{w}_j||^2 + C\sum_{i=1}^{I_n}\xi_i^j \\ -\sum_{i=1}^{I_n}a_i^j\left(y_i\left[\mathbf{w}_j^T(\boldsymbol{\mathcal{X}}_{:i}\overline{\times_j}\mathbf{w}_k) + b\right] - 1 + \xi_i^j\right) \\ -\sum_{i=1}^{I_n}\kappa_i\xi_i.\end{array} \tag{5}$$

as

$$\begin{array}{rl}\nabla_{\mathbf{w}_j}L_j & = 0 \Rightarrow \\ \mathbf{w}_j & = \frac{1}{\eta_j}\sum_{i=1}^{I_n}a_i^j y_i \boldsymbol{\mathcal{X}}_{:i}\overline{\times_j}\mathbf{w}_k.\end{array} \tag{6}$$

The whole procedure is repeated iteratively for every mode, so as to find $\mathbf{w}_k, \quad k = 1\ldots M$.

Let the basic functional that will be our main interest in the rest of the paper be defined as:

$$F(\mathbf{a}^j) = \frac{1}{2}\sum_{i,k}a_i^j a_k^j y_i y_k(\boldsymbol{\mathcal{X}}_{:i}\overline{\times_j}\mathbf{w}_r)^T(\boldsymbol{\mathcal{X}}_{:k}\overline{\times_j}\mathbf{w}_r) \tag{7}$$

where $\mathbf{a}^j$ are the Langrange multipliers. Using the above functional, the Wolf dual problem with respect to $\mathbf{a}^j$ can be reformulated as:

$$\begin{array}{l}\min_{\mathbf{a}^j} F(\mathbf{a}^j) - \sum_{i=1}^{I_n}a_i^j \\ \text{s.t. } \sum_{i=1}^{I_n}y_i a_i^j = 0,\ 0\leq a_i^j \leq C.\end{array} \tag{8}$$

The above problem will be used from now on to provide robust multiplicative update rules for finding the Lagrangian multipliers $\mathbf{a}^j$.

3.1 Multiplicative Update rules Using Semi-nonnegative Formulation

In this Section, sign-insensitive kernels based on NMF will be used to define two novel update rules. The semi-NMF algorithm will be used to achieve that. Semi-NMF requires only for the weight matrix to contain non-negative elements, while it imposes no constraints on the signs of the bases matrix elements.

We consider the following optimization problem:

$$\begin{array}{l}\min_{\mathbf{a}^j} F(\mathbf{a}^j) - {\mathbf{a}^j}^T\mathbf{1} \\ \text{s.t. } a_i^j \geq 0, i\in\{1\ldots n\},\end{array} \tag{9}$$

which is the optimization problem (8) when no slack variables or bias term b are considered. It can be proven that this problem (9) can be rewritten as:

$$\begin{array}{l}\min_{\mathbf{a}^j} \frac{1}{2}||\mathbf{X}_A^j\mathbf{a}_A^j - \mathbf{X}_B^j\mathbf{a}_B^j||_2^2 - {\mathbf{a}^j}^T\mathbf{1} \\ \text{s.t. } a_i^j \geq 0, i\in\{1\ldots n\},\end{array} \tag{10}$$

where $\mathbf{X}_A^j = [\boldsymbol{\mathcal{A}}_{:1}\overline{\times}_j\mathbf{w}_k|\ldots|\boldsymbol{\mathcal{A}}_{:I_n^A}\overline{\times}_j\mathbf{w}_k]$ and $\mathbf{X}_B^j = [\boldsymbol{\mathcal{B}}_{:1}\overline{\times}_j\mathbf{w}_k|\ldots|\boldsymbol{\mathcal{B}}_{:I_n^B}\overline{\times}_j\mathbf{w}_k]$. $\mathbf{a}_A^j \in \Re_+^{I_n^A}$ and $\mathbf{a}_B^j \in \Re_+^{I_n^B}$ are vectors of Lagrangian multipliers that correspond to the tensors $\boldsymbol{\mathcal{A}}_{:i}$ and $\boldsymbol{\mathcal{B}}_{:i}$, respectively. Using the theory provided in [6] and [1] we derive the multiplicative update rules for the vector $\mathbf{a}_A^j$ as:

$$\mathbf{a}_A^j(t) = \mathbf{a}_A^j(t-1)\odot\sqrt{\frac{{\mathbf{G}_{AB}^j}^+\mathbf{a}_B^j(t-1) + {\mathbf{G}_A^j}^-\mathbf{a}_A^j(t-1) + \mathbf{1}}{{\mathbf{G}_A^j}^+\mathbf{a}_A^j(t-1) + {\mathbf{G}_{AB}^j}^-\mathbf{a}_B^j(t-1)}} \tag{11}$$

and the update rules for the vector $\mathbf{a}_B^j$ as:

$$\mathbf{a}_B^j(t) = \mathbf{a}_B^j(t-1)\odot\sqrt{\frac{{\mathbf{G}_{BA}^j}^+\mathbf{a}_A^j(t) + {\mathbf{G}_B^j}^-\mathbf{a}_B^j(t-1) + \mathbf{1}}{{\mathbf{G}_B^j}^+\mathbf{a}_B^j(t-1) + {\mathbf{G}_{BA}^j}^-\mathbf{a}_A^j(t-1)}} \tag{12}$$

where $\mathbf{G}_{AB}^j = [(\boldsymbol{\mathcal{A}}_{:i}\overline{\times}_j\mathbf{w}_k)^T\boldsymbol{\mathcal{B}}_{:k}\overline{\times}_j\mathbf{w}_k] = {\mathbf{X}_A^j}^T\mathbf{X}_B^j$,
$\mathbf{G}_A^j = [(\boldsymbol{\mathcal{A}}_{:i}\overline{\times}_j\mathbf{w}_k)^T\boldsymbol{\mathcal{A}}_{:k}\overline{\times}_j\mathbf{w}_k] = {\mathbf{X}_A^j}^T\mathbf{X}_A^j$,
$\mathbf{G}_B^j = [(\boldsymbol{\mathcal{B}}_{:i}\overline{\times}_j\mathbf{w}_k)^T\boldsymbol{\mathcal{B}}_{:k}\overline{\times}_j\mathbf{w}_k] = {\mathbf{X}_B^j}^T\mathbf{X}_B^j$ and
$\mathbf{G}_{AB}^j = {\mathbf{G}_{BA}^j}^T$.

In the above equations, $\mathbf{1}$ is an appropriately sized vector of ones and $\odot$ is the Hadamard product.

3.2 Multiplicative Update rules for Sign-Insensitive MSTMs

If NMF is used instead of semi-NMF, the update rules acquired are slightly different. More specifically[1]:

$$\mathbf{a}_A^j(t) = \mathbf{a}_A^j(t-1)\odot\frac{{\mathbf{G}_{AB}^j}^+\mathbf{a}_B^j(t-1) + {\mathbf{G}_A^j}^-\mathbf{a}_A^j(t-1) + \mathbf{1}}{{\mathbf{G}_A^j}^+\mathbf{a}_A^j(t-1) + {\mathbf{G}_{AB}^j}^-\mathbf{a}_B^j(t-1)} \tag{13}$$

and

$$\mathbf{a}_B^j(t) = \mathbf{a}_B^j(t-1)\odot\frac{{\mathbf{G}_{BA}^j}^+\mathbf{a}_A^j(t) + {\mathbf{G}_B^j}^-\mathbf{a}_B^j(t-1) + \mathbf{1}}{{\mathbf{G}_B^j}^+\mathbf{a}_B^j(t-1) + {\mathbf{G}_{BA}^j}^-\mathbf{a}_A^j(t-1)} \tag{14}$$

In these updates, the $\mathbf{G}$ matrix is split as follows:

$$G_{ij}^+ \triangleq \begin{cases} G_{ij} & \text{if } G_{ij} > 0, \\ G_{ij} + D_{ii} & \text{if } i = j \\ 0 & \text{otherwise.}\end{cases} \tag{15}$$

[1] We omit the derivations due to the lack of space.

and

$$G_{ij}^{-} \triangleq \begin{cases} |G_{ij}| & \text{if } G_{ij} < 0 \\ |G_{ij}| + D_{ii} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

which is equivalent to :

$$\begin{array}{rl} \mathbf{G}_{new}^{+} & = \mathbf{G}^{+} + \mathbf{D}, \\ \mathbf{G}_{new}^{-} & = \mathbf{G}^{-} + \mathbf{D} \\ \mathbf{G} = \mathbf{G}_{new}^{+} & -\mathbf{G}_{new}^{-} = \mathbf{G}^{+} - \mathbf{G}^{-} \end{array} \quad (17)$$

where $\mathbf{D}$ is a non-negative diagonal matrix constructed as:

$$[\mathbf{D}_A^j]_{ii} = \max\left(0, \sum_{j \neq i} [\mathbf{G}_A^{j\,-}]_{ij} - [\frac{\mathbf{G}_{AB}^{j\,-}\mathbf{a}_B^j}{\mathbf{a}_A^j}]_i\right) \quad (18)$$

$$[\mathbf{D}_B^j]_{ii} = \max\left(0, \sum_{j \neq i} [\mathbf{G}_B^{j\,-}]_{ij} - [\frac{\mathbf{G}_{BA}^{j\,-}\mathbf{a}_A^j}{\mathbf{a}_B^j}]_i\right) \quad (19)$$

ensuring that $\mathbf{G}_{new}^{-}$ becomes positive semi-definite. Thus, the new updates are given by:

$$\mathbf{a}_A^j = \mathbf{a}_A^j \odot \frac{\mathbf{G}_{AB}^{j\,+}\mathbf{a}_B^j + \mathbf{G}_A^{j\,-}\mathbf{a}_A^j + \mathbf{1} + \mathbf{D}_A^j\mathbf{a}_A^j}{\mathbf{G}_A^{j\,+}\mathbf{a}_A^j + \mathbf{G}_{AB}^{j\,-}\mathbf{a}_B^j + \mathbf{D}_A^j\mathbf{a}_A^j} \quad (20)$$

$$\mathbf{a}_B^j = \mathbf{a}_B^j \odot \frac{\mathbf{G}_{BA}^{j\,+}\mathbf{a}_A^j + \mathbf{G}_B^{j\,-}\mathbf{a}_B^j + \mathbf{1} + \mathbf{D}_B^j\mathbf{a}_B^j}{\mathbf{G}_B^{j\,+}\mathbf{a}_B^j + \mathbf{G}_{BA}^{j\,-}\mathbf{a}_A^j + \mathbf{D}_B^j\mathbf{a}_B^j} \quad (21)$$

If the kernels used have non-negative values, the $\mathbf{G}^{-}$ set can be set equal to zero, while the $\mathbf{G}^{+}$ set can be set to the original matrix to achieve the split.

3.3 Soft Margin MSTMs

If we incorporate upper bound constraints of the form $\alpha_i \leq C$ where $\alpha_i = \min\{\alpha_i, C\}$ in (9), the dual problem is reformulated to the soft margin STMs. The term C is used to avoid overfitting.

3.4 Bias Term

MSTMs with a bias term are given by the following formulation:

$$\begin{array}{rl} & \min_{\mathbf{a}^j} F(\mathbf{a}^j) - \mathbf{a}^{j\,T}\mathbf{1} \\ \text{s.t. } a_i^j \geq & 0,\ \sum_i y_i a_i^j = 0,\ i \in \{1 \ldots n\}, \end{array} \quad (22)$$

We introduce a weight variable λ and rewrite the equality constraint $\sum_i y_i a_i^j = 0$ as the following two equality constraints :

$$\sum_{i \in A} \alpha_i = \lambda, \sum_{i \in B} \alpha_j = \lambda. \quad (23)$$

We also introduce the variables $c_k^j = \alpha_k^j / \lambda \ \forall \ k$. In order to reach a solution, we optimize λ keeping $\mathbf{c}^j$ fixed

Figure 1. An example of a gait sequence.

and then optimize $\mathbf{c}^j$ keeping λ fixed. Optimizing with respect to $\lambda = \frac{1}{F(\mathbf{c}^j)}$ we get:

$$\begin{array}{rl} & \min_{\mathbf{c}^j} \ F(\mathbf{c}^j) \\ \text{s.t. } c_i^j \geq & 0, \sum_{i \in A} c_i^j = 1, \\ \sum_{i \in B} c_i^j = 1, & i \in \{1 \ldots n\}, \end{array} \quad (24)$$

The update rules are then derived as follows:

$$\mathbf{c}_A^j = \mathbf{c}_A^j \odot \frac{\mathbf{G}_{AB}^j\mathbf{c}_B^j + \mathbf{1}\mathbf{c}_A^{j\,T}\mathbf{G}_A^j\mathbf{c}_A^j}{\mathbf{G}_A^j\mathbf{c}_A^j + \mathbf{1}\mathbf{c}_A^{j\,T}\mathbf{G}_{AB}^j\mathbf{c}_B^j} \quad (25)$$

and

$$\mathbf{c}_B^j = \mathbf{c}_B^j \odot \frac{\mathbf{G}_{BA}^j\mathbf{c}_A + \mathbf{1}\mathbf{c}_B^{j\,T}\mathbf{G}_B^j\mathbf{c}_B^j}{\mathbf{G}_B\mathbf{c}_B^j + \mathbf{1}\mathbf{c}_B^{j\,T}\mathbf{G}_{BA}^j\mathbf{c}_A^j} \quad (26)$$

4 Experimental Results

The efficiency of the proposed classifier is shown in the gait and action recognition problems. The database used for the gait recognition experiments was the USF HumanID Gait Challenge data sets version 1.7, as used in [4], for comparison reasons. The viewpoint (left/right), the shoe type and the surface type (grass/concrete) change, thus providing 71 sequences organized in seven experiments (probe sets). 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.

For the action recognition experiments, the Weizmann database was used [2]. It contains nine activities (bend, jack, jump, pjump, run, side, skip, walk, wave1 and wave2) performed by nine subjects. We extract spatio-temporal salient points using the method presented in [5] and create a single image by projecting them on a single frame (i.e. by ignoring their temporal location). A set of dilations and erosions was applied in order to create creating the binary mask used for classification. An example of the binary mask for each one of the actions is provided in Figure 2.

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. The execution times were recorded to compare the

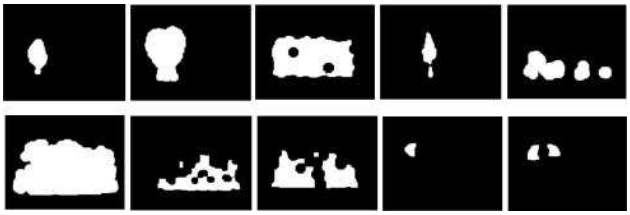

Figure 2. An example of the binary masks extracted for action classification.

Table 1. Recognition accuracy per probe set for the gait recognition problem.

Probe Set	A	B	C	D	E	F	G
Accuracy (%)	78.4	74.9	82.9	77.7	81.0	78.3	82.2

convergence times of the typical MSTMs and the proposed multiplicative MSTMs in order to prove the convergence superiority of the proposed update rules.

In Figure 3, we report the convergence value defined as $\prod_{k=1}^{M} \|w\|^2$ for both the proposed method and the classical STM updates rules. It is evident that the proposed method converges both faster and to a better optimum. More specifically, the typical MSTMs need approximately 10 iterations to reach the convergence value, while the equivalent number of iterations for the proposed set of update rules is close to 5. Thus, the proposed method causes the algorithm to converge faster when compared to the typical MSTMs (25% less time). Similar results in terms of convergence and speed were achieved for the action recognition dataset.

Finally, the recognition accuracy achieved for the gait recognition problems when both typical and the proposed MSTMs were used is shown in Table 1. The leave-one-out cross-validation approach was used to test the generalization performance of the classifiers. The accuracy achieved for the action recognition problem is equal to 84.4%.

5 Acknowledgment

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

6 Conclusions

A novel reformulation of the Multilinear STMs problem is proposed in this paper. The minimization problem is approached in a way similar to that of NMF algorithm, producing new sets for multiplicative update rules both for hard and soft MSTMs. The existence of bias is also investigated, while the relationship between MSTMs and NMF is also highlighted. The superiority of the proposed update rules in terms of convergence speed and power is verified by experiments performed for the gait and action recognition problems.

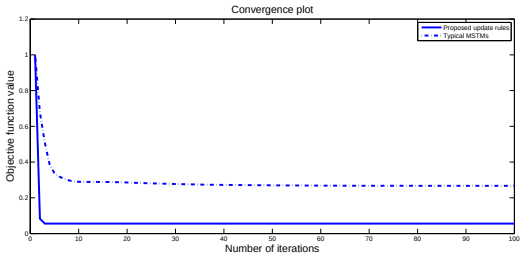

Figure 3. Convergence of objective function value of the proposed update rules.

References

[1] C. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, accepted for publication.

[2] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[3] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[4] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, 19(1):18–39, 2008.

[5] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics*, 36(3):710–719, June 2006.

[6] V. K. Potluru, S. M. Plis, M. Mrup, V. D. Calhoun, and T. Lane. Efficient multiplicative updates for support vector machines. *Proceedings of the 2009 SIAM Conference on Data Mining (SDM)*, 2009.

[7] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank. Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42, 2007.