Multi-view Support Vector Machines for Distributed Activity Recognition

Ehsan Adeli Mosabbeb
Iran Univ. of Science and Technology
Email: eadeli@iust.ac.ir

Kaamran Raahemifar
Ryerson University, Canada
Email: kraahemi@ee.ryerson.ca

Mahmood Fathy
Iran Univ. of Science and Technology
Email: mahfathy@iust.ac.ir

***Abstract*—In this paper, we propose a Multi-view Distributed SVM model. Most distributed classification models, distribute the instances among their processing nodes, while we assume that one instance is formed as a combination of information from different sources. This makes our model a great choice for multi-view activity recognition in camera sensor networks. We demonstrate the effectiveness of the algorithm, using the IXMAS dataset.**

I. Introduction

Distributed Camera Sensor Networks (CSN) are great platforms for different applications, including human activity recognition, due to their ease of deployment and robustness [1]. Proper design of algorithms on these networks can decrease the amount of processing and communication between the nodes [2], where the whole scene is analyzed collaboratively.

SVMs are great tools for machine learning and have been developed for large-scale and distributed purposes. Most of these methods distribute the whole training or testing instances over the processing nodes [4], [6], [7]. For multi-view activity recognition, different cameras are looking at the scene and hence each camera has only one part of the scene representation [2], [3]. In this scenario, the instances are not distributed as a whole, but they are split among the nodes. A few number of learning methods have been developed for such cases [5], but they are not designed for parallel/distributed systems.

In this paper, we propose a new SVM formulation to solve the multi-view activity recognition problem in distributed CSNs. A multi-view regularization is introduced to pull the classification results of all views together, and a consensus regularization to bring the classification result to an agreement.

II. Multi-view Distributed SVM

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ ($\mathbf{x_i} \in \mathbb{R}^d, y_i \in \{+1, -1\}$) denote a set of n training instances. Our goal is to learn a function $f(\mathbf{x})$, which best predicts the y_i labels. For the linear case, we have $f(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i + b$ with $\mathbf{w}$ and b as the weight vector and a bias, respectively. Let's assume that the network of the processing nodes is modeled with a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \ldots, m\}$ as the set of camera nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ as the nodes that can communicate with each other. Therefore, an instance $\mathbf{x}_i$ would be split to $\{(\mathbf{x}_i^j)\}_{j=1}^{m}$, with j as the index of its processing node for an m-view representation. Consequently, we can train a single hinge loss SVM for each view, j, as:

$$\min_{\mathbf{w}_j \in \mathbb{R}^{d_j}} \frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n} max(0, 1 - y_i f_j(\mathbf{x}_i^j)) \qquad (1)$$

where C is a positive hyper-parameter, $f_j(.)$ is the learned function from j^{th} view and is defined as $f_j(\mathbf{x}_i^j) = \mathbf{w}_j^\top \mathbf{x}_i^j + b_j$.

For a multi-view learner, we can assume that a good learner could be learned from each single view [5] and these learners should be consistent with each other and perform same predictions. In order to pull the learning results of different views together, a regularization term is incorporated. For m different views, the regularization term would be defined as:

$$\frac{1}{2n}\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{i=1}^{n}\big(f_j(\mathbf{x}_i^j) - f_k(\mathbf{x}_i^k)\big)^2, \qquad (2)$$

Let's call the concatenation of all view $\mathbf{w}_i$s, as $\mathbf{w} = [\mathbf{w}_1 \mathbf{w}_2 \ldots \mathbf{w}_m]$. As a result the problem would be formulated as optimizing the following objective for finding all the $\mathbf{w}_i$s:

$$\begin{aligned}\min_{\mathbf{v}\in\mathbb{R}^d} &\frac{1}{2}\|\mathbf{v}\|_2^2 + C\sum_{j=1}^{m}\sum_{i=1}^{n} max(0, 1 - y_i f_j(\mathbf{x}_i^j)) \\ &+ \frac{\gamma_1}{2n}\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{i=1}^{n}\big(f_j(\mathbf{x}_i^j) - f_k(\mathbf{x}_i^k)\big)^2 + \frac{\gamma_2}{2}\|\mathbf{v} - \mathbf{w}\|_2^2 \\ &s.t. \mathbf{v} = \mathbf{w},\end{aligned} \qquad (3)$$

where γ_1 and γ_2 are positive parameters. The last term helps the algorithm converge more robust and faster. In this formulation, vector $\mathbf{v}$ is calculated with respect to the weight vectors from each view to preserve the global classification scheme, while each single view should also be able to classify the scene of its own. The last term and the constraint $\mathbf{v} = \mathbf{w}$ are to make the two vectors agree. In order to optimize the objective in parallel, we impose these conditions with respect to their corresponding parts relative to the set $\{\mathbf{x}^j\}_{j=1}^{m}$, as $\frac{\gamma_2}{2}\sum_{j=1}^{m}\|\mathbf{v}^j - \mathbf{w}^j\|_2^2$ and $\mathbf{v}^j = \mathbf{w}^j$, respectively.

III. Optimization

In order to solve (3), let's introduce Lagrangian multipliers $\lambda = \{\lambda_j \in \mathbb{R}^n\}_{j=1}^{m}$, and the Lagrangian function:

$$\begin{aligned}\mathcal{L}(\mathbf{v}, \mathbf{w}, \lambda) = &\frac{1}{2}\|\mathbf{v}\|_2^2 + C\sum_{j=1}^{m}\sum_{i=1}^{n} max(0, 1 - y_i f_j(\mathbf{x}_i^j)) \\ &+ \frac{\gamma_1}{2n}\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{i=1}^{n}\big(f_j(\mathbf{x}_i^j) - f_k(\mathbf{x}_i^k)\big)^2 \\ &+ \sum_{j=1}^{m}\big(\frac{\gamma_2}{2}\|\mathbf{v}^j - \mathbf{w}^j\|_2^2 + \lambda_j^\top(\mathbf{v}^j - \mathbf{w}^j)\big),\end{aligned} \qquad (4)$$

Using the Alternating Direction Method (ADM), we can write the update rules with their closed form solutions, as:

$$\mathbf{w}^{k+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \; L_1^k + L_2^k + \frac{\mu}{2}\sum_{j=1}^{m} \|\mathbf{v}^{j^k} - \mathbf{w}^{j^k} + \frac{\lambda_j}{\mu}\|_2^2, \quad (5)$$

$$\mathbf{v}^{k+1} = \frac{\gamma_2 \mu \mathbf{w}^{k+1} + n\lambda^k}{n\mu}, \quad (6)$$

$$\lambda_j^{k+1} = \lambda_j^k + \mu(\mathbf{v}^{j^{k+1}} - \mathbf{w}^{j^{k+1}}), \quad (7)$$

where μ is a step size, L_1 and L_2 are the second and third terms in (4). So, subproblem (5) is separable and could be optimized in parallel/distributed. Each machine j solves the problem in parallel (associated with data $\mathbf{x}^j$), and needs to communicate their so-far learned functions $f_j(\mathbf{x}^j)$ and their own share of weight vector, $\mathbf{w}^j$, with others. For any $\mu > 0$, as $k \to \infty$, it guarantees that $[\mathbf{w} - \mathbf{z}] \to \mathbf{0}$ (stop criterion).

IV. Activity Recognition in CSNs

Our task is to recognize activities present in a scene, which is captured with a networked set of cameras. Each scene is represented with a fix-length feature vector from each camera's view point, using histogram of densely sampled features from video space-time blocks, which are sampled in five dimensions, (x, y, t, σ, τ). σ and τ are the spatial and temporal scales, respectively. We use Histogram of Gradient (HoG), Histogram of optical Flow (HoF) [9] and Histogram of Motion Boundaries (HoMB). These histograms are computed on a regular grid at three different scales, with independent dictionaries. This is done by using K-means, and quantizing all descriptors to the closest ℓ_2 distance dictionary element. Each scene i is composed of a histogram feature vector $\mathbf{h}_i^j$ from the j^{th} view. So, scene $\mathbf{x}_i$ is described by $\{\mathbf{h}_i^j\}_{j=1}^m$.

V. Experimental Results

To setup experiments, we have simulated the network environment, where each camera process is implemented in a single process on a processing core and the communication is done via IPC, with fully connected topology. The minimum size of a 3D patch is considered to be 18×18 pixels and 10 frames with $\sigma = 2$ and $\tau = 3$. Spatial and temporal samplings are done with 50% overlap. For each class one SVM is trained, with a leave-one-out cross-validation strategy. We carried out experiments using the IXMAS [8] dataset. This dataset is a challenging one, since the subjects freely choose their position and orientation. Therefore, each camera has captured different viewing angles, which makes the task harder. Figure 1 shows the results of the classification on each individual camera for IXMAS dataset, compared with the distributed algorithm (data from all views). Table I shows the overall recognition rate in comparisons with some state-of-the-art methods. The table shows recognition rates for methods both using 13 and 10 subjects for training and testing, as most of the previous methods use a subset of the dataset with 10 subjects, 11 actions and 4 out of the 5 views. Figure 2 also shows the execution time of each of these sets of data together with the communication and load overheads of the distributed algorithm. The centralized algorithm is run on the same machine, but on a single core.

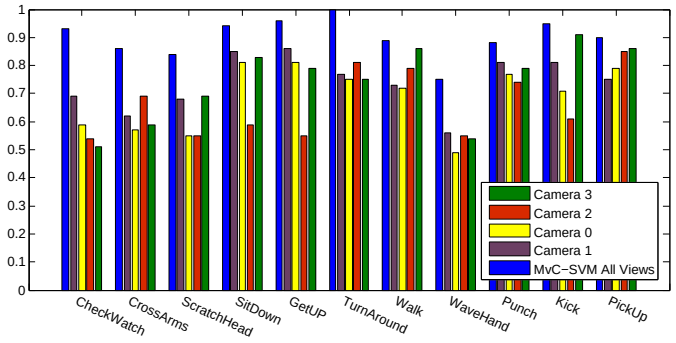

Fig. 1. Recognition results for single and all four views, on IXMAS.

TABLE I. Accuracy results on IXMAS dataset, # Sub. and #Act. indicate the number of subjects and actions used.

Approach	# Sub.	# Act.	Method	Accuracy
Srivastava et al. [2]	10	11	Distributed	81.4%
Weinland et al. [8]	10	11	Centralized	81.3%
Our Method	10	11	Distributed	90.0%
Wu and Jia [3]	13	12	View-invariant	91.67%
Our Method	13	12	Distributed	85.5%

VI. Conclusion

We have introduced a novel SVM model for use in distributed environments, where each training and testing instance is composed of data from different sources and have adapted it for multi-view activity recognition. Regularization terms are incorporated to bring all the views into agreement.

References

[1] Tron, R. and Vidal, R.: 'Distributed Computer Vision Algorithms', *IEEE Signal Process Mag.*, 2011, **28**, pp. 32-45.

[2] Srivastava, G., Iwaki, H., Park, J. and Kak, A.: 'Distributed and lightweight multi-camera human activity classification', *ICDSC*, 2009.

[3] Wu, X. and Jia, Y.: 'View-Invariant action recognition using latent kernelized structural SVM', *ECCV*, Springer-Verlag, 2012, pp. 411-424.

[4] Zhang C., Lee H. and Shin K.: 'Efficient distributed linear classification algorithms via the alternating direction method of multipliers', *J. Mach. Learn. Res.*, 2012, **22**, pp. 1398-1406.

[5] Shiliang S.: 'Multi-view laplacian support vector machines', *ADMA - Vol. Part II*, 2011, Springer-Verlag, pp. 209-222.

[6] Forero P., Cano A., and Giannakis G.: 'Consensus-Based Distributed Support Vector Machines'. *J. Mach. Learn. Res.* 2010, **11**, pp. 1663-1707.

[7] Chang E., Zhu K., Wang H., Bai H., Li J., Qiu Z., Cui H.: 'Parallelizing Support Vector Machines on Distributed Computers', *NIPS*, 2007.

[8] Weinland, D., Boyer, E. and Ronfard, R.: 'Action Recognition from Arbitrary Views using 3D Exemplars', *ICCV*, 2007 IEEE, pp. 1-7.

[9] Laptev, I., Marszaek, M., Schmid, C. and Rozenfeld, B.: 'Learning Realistic Human Actions from Movies', *CVPR*, 2008, IEEE.

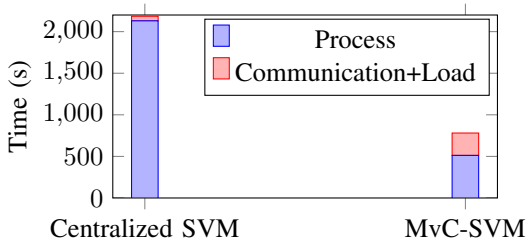

Fig. 2. Execution times on IXMAS datasets.