MAPACo-Training: A Novel Online Learning Algorithm of Behavior Models

Heping Li[1,2], Zhanyi Hu[1], Yihong Wu[1], and Fuchao Wu[1]

[1] National Laboratory of Pattern Recognition
[2] Digital Content Technology Research Center,
Institute of Automation,Chinese Academy of Sciences,
P.O. 2728, Beijing 100080, P.R. China
{hpli,huzy,yhwu,fcwu}@nlpr.ia.ac.cn

Abstract. The traditional co-training algorithm, which needs a great number of unlabeled examples in advance and then trains classifiers by iterative learning approach, is not suitable for online learning of classifiers. To overcome this barrier, we propose a novel semi-supervised learning algorithm, called MAPACo-Training, by combining the co-training with the principle of Maximum A Posteriori adaptation. This MAPACo-Training algorithm is an online multi-class learning algorithm, and has been successfully applied to online learning of behaviors modeled by Hidden Markov Model. The proposed algorithm is tested with the Li's database as well as Schuldt's dataset.

1 Introduction

Behavior modeling is driven by a wide range of applications, such as advanced user interface, visual surveillance, virtual reality and so on. The most existing works in this field focused on modeling the behaviors with manually labeling like [1,2,3,4]. For example, Li and Greenspan [1] built a multi-scale model from time-varying contours and Gong and Xiang [2] learned a Dynamically Multi-Linked Hidden Markov Model (DML-HMM). However, manual labeling of behavior patterns is laborious, impractical and error prone [5]. Recently, some behavior modeling methods based on semi-supervised/unsupervised learning [5,6,7,8] have been proposed. For instance, Xiang and Gong [5] discovered natural grouping of behavior patterns through unsupervised model selection and feature selection, and Zelnik-Manor and Irani [6] used the normalized-cut approach to automatically cluster the data and then build the statistical behavior model. Unfortunately, these methods need to get a great number of unlabeled examples beforehand, which are therefor unsuitable for online learning of behavior models and cannot automatically adjust the models' parameters according to the circumstantial changes.

The co-training approach proposed by Blum and Mitchell [9] is also a semi-supervised learning method. Levin et al. [10] used the co-training framework in the context of boosted binary classifiers to build the automobile detectors.

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 472–481, 2007.

And Yan and Naphade [11] proposed a multi-view semi-supervised learning algorithm which avoids the requirement of the co-training approach about that each view of examples is sufficient for learning the target concepts. However, these methods belong to the off-line learning category. Javed et al. [12] combined the co-training approach and boosting to propose an algorithm for online detection and classification of moving objects, where behavior modeling is not considered.

In this paper, we present a novel semi-supervised learning method called MAPACo-Training which combines the co-training approach and the principle of Maximum A Posteriori adaptation [8,13,16]. The proposed method can simultaneously train the parameters of multi-class models. We have successfully applied the method to online learning the parameters of behaviors modeled by Hidden Markov Model (HMM). Since it only needs a small labeled sample set beforehand, our method can alleviate the problem in the methods [1,2,3,4]. And unlike the approaches [5,6,7,8], the method can automatically adjust the parameters with the current example online.

The remainder of this paper is organized as follows: Motion signature representation is outlined in Section 2. Section 3 is a detailed description of MAPACo-Training. Experimental results are reported in section 4. And conclusions as well as future research directions are listed in section 5.

2 Motion Signature Representation

2.1 Feature Extraction

Background subtraction is used to detect foreground. In our approach, two types of features are considered: (1) shape feature; (2) optical flow feature [14].

The size of the foreground region varies with the distance of object to camera, camera parameters and the size of object. We therefore need to normalize the foreground region. Firstly, we equidistantly divided the bounded rectangle of the foreground into $U \times V$ non-overlapping sub-blocks. Then, the normalized value of each sub-block is calculated as follows:

$$x_i^1 = s\_sub(i)/\max, \quad i = 1, 2, \ldots, num, \tag{1}$$

where $num = U \times V$ is the number of sub-blocks; *s_sub(i)* is the number of the foreground pixels in the i^{th} sub-block; *max* is the maximum value of $\{$*s_sub(i)*, $i = 1, 2, \ldots, num\}$. The optical flow value of each sub-block is calculated as follows:

$$x_i^j = f\_sub(i,j)/sum(i), i = 1, 2, \ldots, num, j = 2, 3, \tag{2}$$

where *f_sub(i,j)* with $j = 2, 3$ are respectively the sum of horizontal, vertical optical flow in the i^{th} sub-block; *sum(i)* is the pixel number in the i^{th} sub-block. Then, the feature vectors at frame t from shape and optical flows are as:

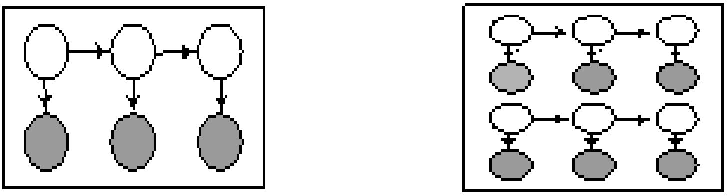
$$o_t^d = [x_1^d, x_2^d, \ldots, x_{num}^d], \quad d = 1, 2, 3.$$

2.2 Motion Signature Representation

Given the observation feature sequences $O_T^d = \{o_1^d, o_2^d, \cdots, o_t^d, \cdots, o_T^d\}$, two different Hidden Markov Models (HMMs) are adopted to build the behavior models. The one is a single continuous HMM from shape, and the other is like a Parallel Hidden Markov Model (PHMM) with two continuous HMMs from optical flow. These HMM topologies are shown in Figure 1, where shade circles are as observation nodes and clear circles as hidden nodes. For optical flow model, the two HMMs are learned independently. The output probability density function of learning is the following Gaussian Mixture Model (GMM):

$$p(o_t^d|\theta) = \sum_{k=1}^{K} \alpha_k p_k(o_t^d|\mu_k, \Sigma_k) \tag{3}$$

where $\theta = \{\alpha_k, \mu_k, \Sigma_k, k = 1, 2, \ldots, K\}$ represents the parameters of GMM including weight α_k, mean value μ_k and covariance matrix Σ_k of every mixture component; $\sum\limits_{k=1}^{K} \alpha_k = 1$.


(a)

(b)

Fig. 1. HMM topology: (a) shape model; (b) optical flow model

By the Forward procedure, we compute the observation probabilities $P(O_T^d|\lambda_c^d)$, $c = 1, 2, \ldots, C$ for the observation feature sequences O_T^d, where C is the class number of behaviors and λ_c^d is the HMM parameter set of the c^{th} class behavior. Since the output probability density function is GMM, the probabilities are normalized [15] as follows:

$$\bar{P}(O_T^d|\lambda_i^d) = P(O_T^d|\lambda_i^d) / \sum\nolimits_{c=1}^{C} P(O_T^d|\lambda_c^d). \tag{4}$$

And for optical flow model (Figure 1(b)), the following operation is further performed as:

$$\bar{P}(O_T^{2,3}|\lambda_i^{2,3}) = \frac{\bar{P}(O_T^2|\lambda_i^2)\bar{P}(O_T^3|\lambda_i^3)}{\sum_{c=1}^{C}\left[\bar{P}(O_T^2|\lambda_c^2)\bar{P}(O_T^3|\lambda_c^3)\right]}. \tag{5}$$

The Bayes classifier is used as our base classifier. According to the Bayes rule, the posterior

$$P(c|O_T^d) = \bar{P}(O_T^d|\lambda_c^d)P(c)/P(O_T^d),$$

where $P(c) = 1/C$, so $P(c|O_T^d) \propto \bar{P}(O_T^d|\lambda_c^d)$. Thus if $\bar{P}(O_T^d|\lambda_{c_0}^d) = \max_c \bar{P}(O_T^d|\lambda_c^d)$, O_T^d belongs to the c_0–th behavior.

In our proposed algorithm, we set $f_1^i = \bar{P}(O_T^1|\lambda_i^1)$ and $f_2^i = \bar{P}(O_T^{2,3}|\lambda_i^{2,3})$.

3 MAPACo-Training

In this section, we propose a new semi-supervised learning algorithm called Maximum A Posteriori Adaptation Co-Training (MAPACo-Training) which attempts to learn behavior models online. We first describe the principle of MAP adaptation, and then give the details of MAPACo-Training.

3.1 MAP Adaptation

MAP adaptation has widely been used in speaker and face verification [13]. Recently, Zhang et al. in [8,16] used it for unusual event detection and meeting event recognition. During the course of learning the parameters of GMM-based HMM in [8,16], the state-transition probabilities are kept fixed while the mean, variance and mixture weights are adapted as follows:

(1) According to the existing parameters, new statistical values are computed:

$$P(i|o_t^d) = \alpha_i p_i(o_t^d|\mu_i, \Sigma_i) \Big/ \sum\nolimits_{k=1}^{K} \alpha_k p_k(o_t^d|\mu_k, \Sigma_k) \tag{6}$$

$$\alpha_i^{new} = \sum\nolimits_{t=1}^{T} P(i|o_t^d) \Big/ T \tag{7}$$

$$\mu_i^{new} = \sum\nolimits_{t=1}^{T} o_t^d P(i|o_t^d) \Big/ \sum\nolimits_{t=1}^{T} P(i|o_t^d) \tag{8}$$

$$\Sigma_i^{new} = \frac{\sum_{t=1}^{T} P(i|o_t^d)(o_t^d - \mu_i^{new})(o_t^d - \mu_i^{new})^T}{\sum_{t=1}^{T} P(i|o_t^d)} \tag{9}$$

(2) New parameters are estimated as follows:

$$\hat{\alpha}_i = \rho \cdot \alpha_i^{new} + (1-\rho) \cdot \alpha_i^{old} \tag{10}$$

$$\hat{\mu}_i = \rho \cdot \mu_i^{new} + (1-\rho) \cdot \mu_i^{old} \tag{11}$$

$$\hat{\Sigma}_i = \rho \cdot \Sigma_i^{new} + (1-\rho) \cdot [\Sigma_i^{old} + (\hat{\mu}_i - \mu_i^{old})(\hat{\mu}_i - \mu_i^{old})^T] \tag{12}$$

where $\rho (0 \leq \rho \leq 1)$ is the scale factor.

We use the principle of MAP adaptation into our algorithm. More details about MAP adaptation can be found in [8,13,16].

3.2 MAPACo-Training Algorithm

The traditional co-training algorithm [9] needs to get a great number of unlabeled samples in advance and then train models by an approach of iterative learning. It is an off-line learning method. By combining the co-training and the MAP adaptation, we propose a novel online multi-class learning algorithm called MAPACo-Training as follows:

Input: Labeled data L including a small training sample set L_{tr} and a small validation sample set L_v with two views V_1 and V_2, threshold value $T_h > 1$ and $T_{num} \geq 1$.

Output: a classifier from the probabilities$f^1, f^2, ..., f^C$.

MAPACo-Training

1. Create f_1^i and f_2^i ($i = 1, 2, \ldots, C$) using L_{tr} on V_1 and V_2. Set the new training sample set of each one of the C classes $L_b^i = \phi$ (b=1,2);
2. For $k = 1, 2, \ldots, C$
   - (a) For current sample S, assume $n = \max\limits_j \{f_1^j, j = 1, 2, \ldots, C,\ j \neq k\}$, $m = \max\limits_j \{f_2^j,\ j = 1, 2, \ldots, C,\ j \neq k\}$,
     - i. if $f_1^k / f_1^n \geq T_h$, the view V_2 of sample S is added into L_2^k as a new sample;
     - ii. if $f_2^k / f_2^m \geq T_h$, the view V_1 of sample S is added into L_1^k as a new sample;
     - iii. if $1 < f_1^k / f_1^n < T_h$and $1 < f_2^k / f_2^m < T_h$, the view V_1 of sample S is added into L_1^k as a new sample and the view V_2 of sample S is added into L_2^k as a new sample.
   - (b) if the sample number in L_b^k equals T_{num}, the parameters of model f_b^k are updated according to MAP equations (6)∼(12) with these samples in L_b^k and the scale factor ρ is decided by validation sample set L_v. And then let $L_b^k = \phi$.
3. Combine $f^i = \omega_1 f_1^i + \omega_2 f_2^i$ ($\omega_1 + \omega_2 = 1$) using L_v.
4. Create a new classifier using f^i according to the Bayes theory.

Similar to co-training, two base classifiers of every class model need to be trained on separate features of the same sample. How to select samples to train the models? In this algorithm, we use a threshold T_h to do it. The conditions (i)(ii) show if one base classifier can predict the label of the sample confidently, then we add this sample into the training set of the other base classifier of the corresponding class. The condition (iii) means that both base classifiers can get the same label according to the bayes rule, but neither of them is confident, which shows the sample is useful for improving the performance of the two classifiers.

During the course of updating parameters by MAP adaptation equations (6)∼(12), we use validation set L_v to decide the scale factor ρ. If the class prediction for a sample from the conditions (i)∼(iii) is not correct, which means the sample is a noise, the sample is no longer used for further learning by setting

$\rho = 0$ according to L_v. In our experiment, we assume the possible value of ρ is 0 or a constant$\bar{\rho}$ $(0 \leq \bar{\rho} \leq 0.5)$.

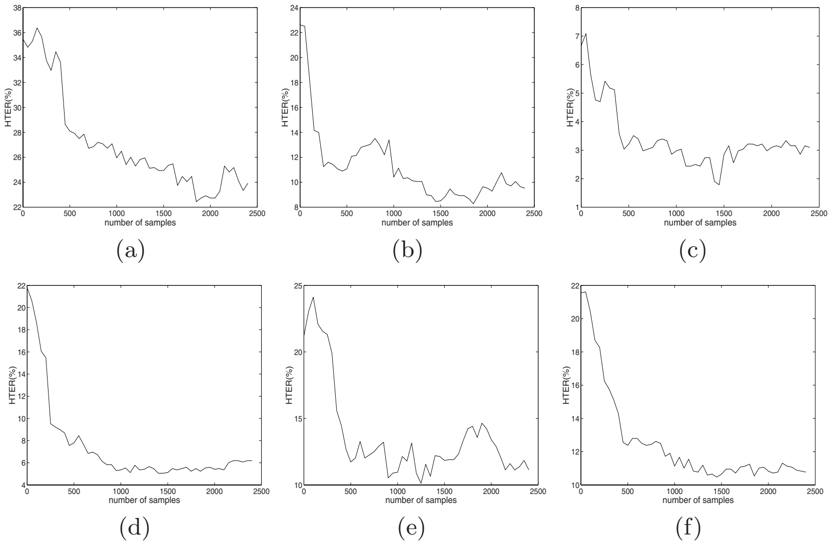
Remark: From the equations (6)∼(12), we can see that the MAP adaptation only uses the current samples to calculate the new statistical values and then gets the new parameters by simple weighted estimation. It avoids to directly train the HMM parameters from a great number of samples by EM algorithm and improves the computational efficiency. The MAPACo-Training algorithm starts from a small label sample set L_{tr} and then updates the parameters by the MAP adaptation. So the algorithm is suitable for online multi-class learning.

4 Experiments

We test our method from two datasets: Li's dataset [17] and Schuldt's dataset [18]. In the experiments, U=9 and V=5 are used for dividing the bounded rectangle of foreground. To each type of features such as shape, horizontal optical flow and vertical optical flow, the Principal Component Analysis (PCA) is used to reduce the 45-dimensional features to the 8-dimensional ones.

4.1 Results on Li's Dataset

We get a video consisting of five kinds of behaviors from Li's dataset [17], of which each one is performed by 18 subjects. Image size is of 160 × 120 pixels and frame rate is of 6 frames/sec. The video totals 38120 frames including "box"



(a) (b) (c)

(d) (e) (f)

Fig. 2. The learning curves: (a) box; (b) kick; (c) lookround; (d) standup (e) wave; (f) average HTER

Table 1. Initial confusion matrix

	box	**kick**	**lookround**	**standup**	**wave**
box	**37.14**	34.29	7.62	8.57	12.38
kick	12.86	**72.86**	1.90	12.38	0
lookround	1.43	1.90	**92.86**	3.33	0.48
standup	1.43	33.80	0.48	**63.81**	0.48
wave	16.67	2.38	14.76	5.24	**60.95**

Table 2. Final confusion matrix

	box	**kick**	**lookround**	**standup**	**wave**
box	**54.76**	18.10	3.33	7.14	16.67
kick	0	**86.67**	1.43	11.42	0.48
lookround	0	0	**95.72**	3.33	0.95
standup	0.95	3.80	0.48	**94.29**	0.48
wave	10.95	0.95	1.43	5.24	**81.43**

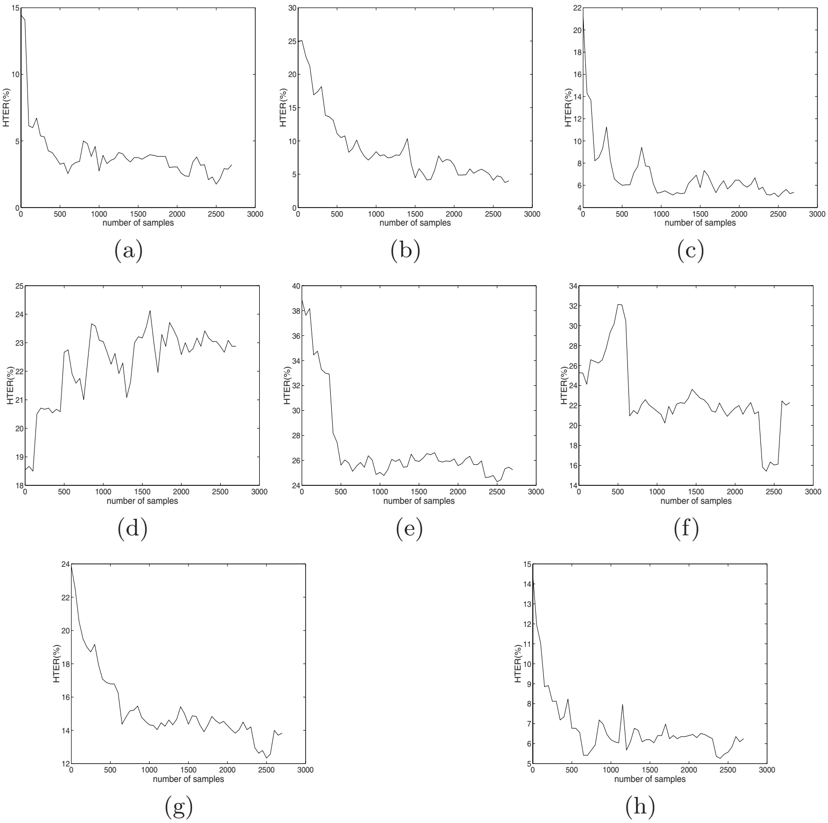
(8000 frames), “kick” (7600 frames), “lookround” (7820 frames), “standup” (7040 frames) and “wave” (7660 frames). We slice this video sequence into 3810 segments with the fixed time duration of 20 frames and the step length of 10 frames, where 25 segments in every class are selected for the small training sample setL_{tr}, 12 segments for the validation sample set L_v, 210 segments for the test sample set and the remaining segments for online learning. Parameters in our algorithm are preset as: T_h=1.5, T_{num}=5 and $\bar{\rho} = 0.2$. MAP adaptation is only used to update the means. The proposed algorithm is implemented in Matlab 6.0 and tested on a 2.0 GHz Pentium 4 PC with 256MB memory. The average time per frame is about 0.228s. As a result, our algorithm at the correct implement could be used for those applications with a frame rate of 6∼10 frames/sec.

Figure 2 gives the learning curves for behavior models of “box”, “kick”, “lookround”, “standup”, “wave” and average half-total error rate (HTER), where HTER=(FAR+FRR)/2 [8], FAR is false acceptance rate and FRR is false rejection rate. The horizontal axis shows the number of effective samples for estimating the parameters in the MAPACo-Training algorithm. The vertical axis shows the HTER. Figure 2(f) is the average HTER curve of all behaviors. From these curves, we can see the learning performance of behavior models can be markedly improved by MAPACo-Training, and after about 500 samples are used, the curves almost become stable. Table 1 gives the initial confusion matrix from the initial behavior models trained by the small training set L_{tr}, and Table 2 shows the final confusion matrix from the final behavior models by our algorithm. From these tables, we can see that when the initial recognition rate is low, those for “box”, “kick”, “standup” and “wave”, the final recognition rate is clearly improved. And when the initial recognition rate is high, that for “lookround”, the final recognition rate is still high.

4.2 Results on Schuldt's Dataset

We get a video sequence of 49813 frames from Schuldt's dataset [18] including "box" (8370 frames), "clap" (8476 frames), "wave" (8275 frames), "run" (7945 frames), "jog" (8170 frames) and "walk" (8577 frames). We slice this video sequence into 4978 segments with the fix time duration of 25 frames and the step length of 10 frames, where 30 segments in each class are selected for the small training sample set L_{tr}, 16 segments for the validation sample set L_v, 240 segments for the test sample set, and the remaining segments for online learning. Parameters in our algorithm are preset as: T_h=1.5, T_{num}=5 and $\bar{\rho} = 0.4$. MAP adaptation is only used to update the means and variances.

Figure 3 shows the learning curves. We can see that the learning results for all the behaviors except "run" are very good. For the behavior "run", the main reason of poor performance is that running of some people is very similar to the jogging of the others in this dataset [18], which is difficult to distinguish. From the initial confusion matrix (Table 3) and the final confusion matrix (Table 4),



(a) (b) (c)

(d) (e) (f)

(g) (h)

Fig. 3. The learning curves: (a) box; (b) clap; (c) wave; (d) run; (e) jog; (f) walk; (g) average HTER (h) run+jog

Table 3. Initial confusion matrix

	box	**clap**	**wave**	**run**	**jog**	**walk**
box	**79.17**	2.50	2.92	3.33	3.33	8.75
clap	24.17	**53.75**	9.17	3.33	2.08	7.50
wave	6.25	8.75	**60.00**	2.50	18.75	3.75
run	0.42	0	0	**80.00**	**14.16**	5.42
jog	2.50	4.58	0	**50.42**	**31.67**	10.83
walk	7.08	1.67	0	25.83	8.75	**56.67**

Table 4. Final confusion matrix

	box	**clap**	**wave**	**run**	**jog**	**walk**
box	**94.58**	2.50	2.08	0	0	0.84
clap	3.33	**94.17**	2.50	0	0	0
wave	0	7.50	**91.25**	1.25	0	0
run	0.83	0	0	**64.17**	**30.42**	4.58
jog	0.42	0.42	2.08	**30.42**	**59.58**	7.08
walk	0.41	0.42	3.33	17.92	20.00	**57.92**

we can see the confusion values between a pair of behaviors other than "run" and "jog" are not high. But for "run" and "jog", the HTER about "run" is only increased from 18.5% to 23% and nearly unchanged after about 2000 samples while the HTER about "jog" is declined from 38.5% to 24.5%. When we regard "run" and "jog" as one behavior "run+jog", the result becomes quite satisfactory as shown in Figure 3(h).

5 Conclusion

In this paper, we proposed a semi-supervised learning algorithm called MAPACo-Training, which combines the traditional co-training algorithm and the principle of MAP adaptation. The algorithm is suitable for online learning of behaviors modeled by HMM. Experiments on two datasets also validate our method. In the future, we will explore a better way to train the models of similar behaviors like "run" and "jog".

Acknowledgment. This work was supported by the National Natural Science Foundation of China under grant Nos (60633070, 60475009) and by National Key Technology R&D Program under grant Nos (2006BAH02A03,2006BAH02A13).

References

1. Li, H., Greenspan, M.: Multi-scale gesture recognition from time-Varying contours. In: IEEE Int'l Conf. On Computer Vision, pp. 236–243. IEEE Computer Society Press, Los Alamitos (2005)

2. Gong, S.G., Xiang, T.: Recognition of group activities using dynamic probabilistic networks. In: IEEE Int'l Conf. On Computer Vision, pp. 742–749. IEEE Computer Society Press, Los Alamitos (2003)
3. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Trans. on Pattern Analysis and Machine Intelligence 23, 257–267 (2001)
4. Laptev, I., Linderberg, T.: Space-time interest points. In: IEEE Int'l Conf. On Computer Vision, pp. 432–439. IEEE Computer Society Press, Los Alamitos (2003)
5. Xiang, T., Gong, S.G.: Video behaviour profiling and abnormality detection without manual labeling. In: IEEE Int'l Conf. On Computer Vision, pp. 1238–1245. IEEE Computer Society Press, Los Alamitos (2005)
6. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 123–130. IEEE Computer Society Press, Los Alamitos (2001)
7. Zhong, H., Shi, J., Visontai, M.: Detecting unusual activity in video. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 819–826. IEEE Computer Society Press, Los Alamitos (2004)
8. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I.: Semi-supervised adapted HMMs for unusual event detection. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 611–618. IEEE Computer Society Press, Los Alamitos (2005)
9. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: 11^{th} Annual Conference on Computational Learning Theory (1998)
10. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual detectors using co-training. In: IEEE Int'l Conf. On Computer Vision, pp. 626–633. IEEE Computer Society Press, Los Alamitos (2003)
11. Yan, R., Naphade, M.: Semi-supervised cross feature learning for semantic concept detection in videos. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 657–663. IEEE Computer Society Press, Los Alamitos (2005)
12. Javed, O., Ali, S., Shah, M.: Online detection and classification of moving objects using progressively improving detectors. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 696–701. IEEE Computer Society Press, Los Alamitos (2005)
13. Reynolds, D.A., Quatieri, T.F., Dumn, R.B.: Speaker verification using adapted Gauusian mixture models. Digital Signal Processing 10, 19–41 (2000)
14. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: DARPA Image Understanding Workshop (April 1981)
15. Lv, F., Nevatia, R.: Recognition and segmentation of 3-D human action using HMM and multi-class adaboost. In: Proc. European Conference on Computer Vision, vol. IV, pp. 359–372 (2006)
16. Zhang, D., Gatica-Perez, D., Bengio, S.: Semi-supervised meeting event recognition with adapted HMMs. In: ICME. IEEE International Conference on Multimedia Expo (2005)
17. Li, H., Hu, Z., Wu, Y., Wu, F.: Behavior modeling and recognition based on space-time image features. In: International Conference on Pattern Recognition, vol. 1, pp. 243–246 (2006)
18. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: International Conference on Pattern Recognition, vol. 3, pp. 32–36 (2004)