Online Detection of Concept Drift in Visual Tracking

Yichen Liu and Yue Zhou

Institute of Image Processing and Pattern Recognition,
Shanghai Jiao Tong University, Shanghai 200240
lycsjtu@163.com, zhouyue@sjtu.edu.cn

Abstract. In the field of data mining, detecting concept drift in a data stream is an important research area with many applications. However the effective methods for concept drift detection are seldom used in visual tracking in which drifting problems appear frequently. In this paper, we present a novel framework combining concept drift detection with an online semi-supervised boosting method to build a robust visual tracker. The main idea is converting updated templates to a data stream by similarity learning and detecting concept drift. The proposed tracker is both robust against drifting and adaptive to appearance changes. Numerous experiments on various challenging videos demonstrate that our technique achieves high accuracy in real-world scenarios.

Keywords: Concept Drift, Visual Tracking, Similarity Learning, Semi-Supervised Boosting.

1 Introduction

The distribution of a data stream is often not stable but changes with time, often these changes lead to performance degeneration in the old model. Thus detection of drifting concepts has extensive applications in data mining, such as spam filtering [1, 2]. To our knowledge, the first exploration of combining concept drift detection with visual tracking was introduced in [3]. They proposed a simple Bayesian approach to detect drift points. However, their method is applicable in limited situations when abrupt drift happens, such as light mutation. In this paper, we present an online learning method combined with concept drift detection to finish the tracking task in real-world scenarios.

There exists one key problem in online learning method for tracking: drifting. Slight inaccuracies in the tracker can lead to incorrectly labeled training examples. Each update to the tracker may introduce an error which may accumulate over time resulting tracking failure. To tackle the drifting problem, extensive techniques have been proposed. Grabner et al. [4] proposed a semi-supervised online boosting method which is based on the idea of [5]. In their approach, a fixed prior classifier which is trained from some labeled examples is used for supervising the update process. This method tackles the drifting problem by restricting the update in a certain range. But the tracker may fail when the target has a significant appearance change. In this case, one can employ concept drift detection techniques to revalidate the tracker.

C.K. Loo et al. (Eds.): ICONIP 2014, Part III, LNCS 8836, pp. 159–166, 2014.

In order to detect concept drift in visual tracking, the updated target templates need to be converted into a data stream. In this paper, we treat the similarity between templates and concept (i.e. prior classifier) as the data to be mined. Learning similarity functions is an area which has received considerable attentions in machine learning. Our learning approach is inspired by the work of Leistner et al. [6]. They proposed a similarity learning method based on semi-supervised boosting. Their technique enables us to measure the distance between newly labeled samples and the concept in feature space. As shown in Fig.1, concept drift manifests an apparent trend from the view of similarity.

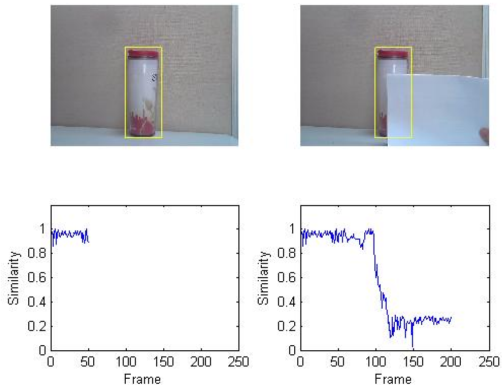

Fig. 1. An example of concept drift (occlusion) in visual tracking

Fig.1 demonstrates a special scenario (sudden occlusion) where only abrupt drift is considered. However in visual tracking, drifting types can be varied, thus a detection method which can accommodate for different situations has to be utilized. As data is generated constantly with the ongoing tracking process, the underlying distribution of data stream may change over time. In this paper, our change-detection algorithm is based on a two-window paradigm. Successive data points are maintained in two fixed-size windows: current window and reference window. We employ a statistical approach called L_1-distance-test [7] to verify whether the distributions of data points in the two windows are close or not. This test makes no assumption about the structure of the distributions and performs well in the application in visual tracking.

The remainder of this paper is organized as follow. After an introduction to the semi-supervised boosting method for similarity learning in Section 2, we introduce the drift detection algorithm L_1-distance-test and its application in visual tracking in Section 3. Section 4 presents the entire tracking framework of our concept drift detection based method. Section 5 demonstrates some experiments and results. Finally, our work concludes with Section 6.

2 Semi-supervised Boosting for Similarity Learning

Similarity learning is a key step in our concept drift detection. It measures the similarity between samples and outputs a similarity score which is added to a data stream. The change of distribution in the data stream indicates the occurrence of concept drift in the tracking process. Usually in visual tracking, the target to track is manually selected in the first frame. Our prior classifier $H^P(\boldsymbol{x}) \in [-1, 1]$ is trained from the original labeled data using a boosting method. The prior classifier measures the similarity between updated templates and the target, generating a data stream which consists of similarity scores. As a confidence score is obtained from the prior classifier, according to [6], a distance measure is defined as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left| H^P(\boldsymbol{x}_i) - H^P(\boldsymbol{x}_j) \right| \tag{1}$$

This means samples that are close in distance share similar confidence scores. Then the distance measure is converted to a similarity measure by

$$S(x_i, x_j) = e^{\left(-\frac{d(x_i, x_j)^2}{\delta^2}\right)}, \tag{2}$$

Where δ is the scale parameter.

3 Online Detection of Concept Drift in a Data Stream

3.1 Testing Closeness of Distributions

If two unknown distributions over an n elements set are given, how to test whether they are statistically close is an interesting question. In this paper, we use the L_1-distance-test proposed in [7] for online distribution closeness testing. This method makes no assumption about the distributions and runs in time linear in the sample size. Our experiments show that the L_1-distance-test achieves high accuracy in the concept drift detection in visual tracking.

In the original L_1-distance-test algorithm, some elements appearing less than certain times are discarded before the test is performed. However, we omit this step in our test because the element set we use is small. Thus we give our simplified version of L_1-distance-test algorithm.

Table 1. The L_1-Distance-Test Algorithm

Algorithm 1 L_1-distance-test$\langle p, q, \varepsilon, \delta \rangle$
Sample $\vec{p}$ and $\vec{q}$ for
  $M = O(\max(\varepsilon^{-2}, 4) n^{2/3} \log n)$ times
Let S_p and S_q be the sample sets

4: Let n_i^p and n_i^q be the times element i appears in S_p and S_q
5: for m=1,2,..,M do
6: update n_i^p by checking m-th element in S_p
7: update n_i^q by checking m-th element in S_q
8: end for
9: Output 1 if $\sum_i |n_i^p - n_i^q| > \varepsilon M/8$
10: Otherwise output 0

In Algorithm 1, the parameters p and q are elements sets in two distributions, and parameters ε and δ can be tuned for adjusting the testing accuracy. Parameter n is the number of all possible elements. The presented algorithm runs in time complexity of O(M) if hashing technique is utilized. It has been proved in [7] that the L_1-distance-test generates a correct output with probability at least 1-δ.

3.2 Detecting Concept Drift in a Data Stream

Our change-detection algorithm is based on a two-window paradigm. Successive data points are maintained in two fixed-size windows: current window and reference window. The reference window focuses on the original data points that share high similarity with the target, however the current window focuses on the most recent data points, and slides forward whenever a new data point is added. Also the reference window is updated with each detected change. The L_1-distance-test is used to verify whether the distributions of data points in the two windows are close or not.

Table 2. The Concet Drift Detection Algorithm

Algorithm 2 Online Detection of Concept Drift
1: $c_0 \leftarrow 0$
2: for i=1…k do
3: $\text{Window}_{1,i} \leftarrow$ first $m_{1,i}$ points from time c_0
4: $\text{Window}_{2,i} \leftarrow$ next $m_{2,i}$ points in stream
5: end for
6: while new data is added to the stream do
7: Slide $\text{Window}_{2,i}$ by 1 point
8: if L_1-distance-test$\langle \text{Window}_{1,i}, \text{Window}_{2,i}, \varepsilon, \delta \rangle$=1
9: $c_0 \leftarrow$current time
10: report change at time c_0
11: clear all windows and GOTO step 2
12: end if
13: end while

Note that our detection algorithm processes the data stream in a discrete manner, so before adding the similarity score to the data stream, it needs to be discretized by mapping the similarity score to a corresponding integer in the element set.

4 Robust Tracking under the Framework of Concept Drift

4.1 Online Semi-supervised Boosting for Tracking

To finish the task of online semi-supervised boosting, Grabner et al. [8] introduced "selectors". Each selector $h^{sel}(\boldsymbol{x})$ contains a set of weak classifiers. At every training iteration t, a weak classifier $H_t(\boldsymbol{x})$ with lowest training error is picked. Thus in each selector, we can set the label and weight for unlabeled example by

$$y_t = sign\left(\tilde{z}_t(\boldsymbol{x})\right) \ and \ \lambda_t = |\tilde{z}_t(\boldsymbol{x})| \tag{3}$$

where y_t is the pseudo-label and λ_t is the corresponding weight. And $\tilde{z}_t(\boldsymbol{x})$ is the the pseudo-soft-label which is defined by

$$\tilde{z}_t(\boldsymbol{x}) = \frac{\sinh\left(H^P(x) - H_{t-1}\right)}{\cosh\left(H^P(x)\right)} = \tanh(H^P(\boldsymbol{x})) - \tanh(H_{t-1}(\boldsymbol{x})) \tag{4}$$

In the semi-supervised boosting based tracking approach [4], the tracking problem is formulated as binary classification between the target and background. Often in visual tracking, the target to track is manually selected in the first frame. A prior classifier is initialized by taking positive training samples and negative training samples from the target and background respectively. At every iteration, the classifier is evaluated in the local neighborhood to generate a confidence map which will be analyzed to find the target position. Note that the update process is restricted by the prior classifier.

However, the tracking approach described above has a major drawback. When the target has a significant appearance change, the tracker may fail. In this case, one can employ concept drift detection techniques to revalidate the tracker.

4.2 Tracking with Concept Drift Detection

The structure of our proposed tracker is depicted in Fig.2.

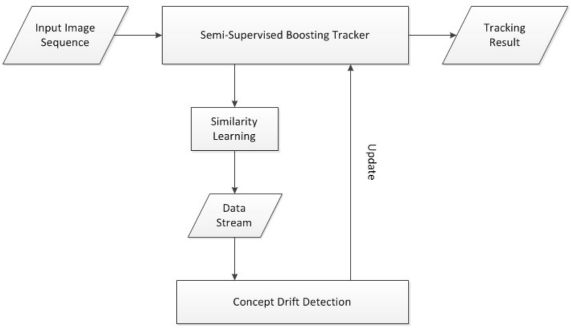

Fig. 2. The structure of our proposed tracker

In the semi-supervised boosting tracking, a template is evaluated by the prior classifier before updating. The updated templates restricted by the prior classifier are similar to the initial appearance, thus the tracker can't adapt to significant appearance changes. Using the drift detection algorithm discussed in Section 3, one can detect a major concept drift when significant appearance change happens. Our proposed tracker is both robust against drifting and adaptive to appearance changes.

5 Experiments

We compare our tracker with three other trackers, including SemiT[4], CT[9], MIL[10]. We include SemiT into our experiment because our tracker is implemented based on it. By adding drift detection module to SemiT, we can see a significant improvement in the tracking accuracy.

5.1 Quantitative Comparison

The four test sequences (*Dudek, football, girl, shaking*) for quantitative analysis exhibit extensive challenging properties: illumination changes, pose variations, occlusions, background clutters and so on. We use the center location error which is defined as the Euclidean distance of the center positions between the tracked target and the ground truth. The tracking results are shown in Fig.3.

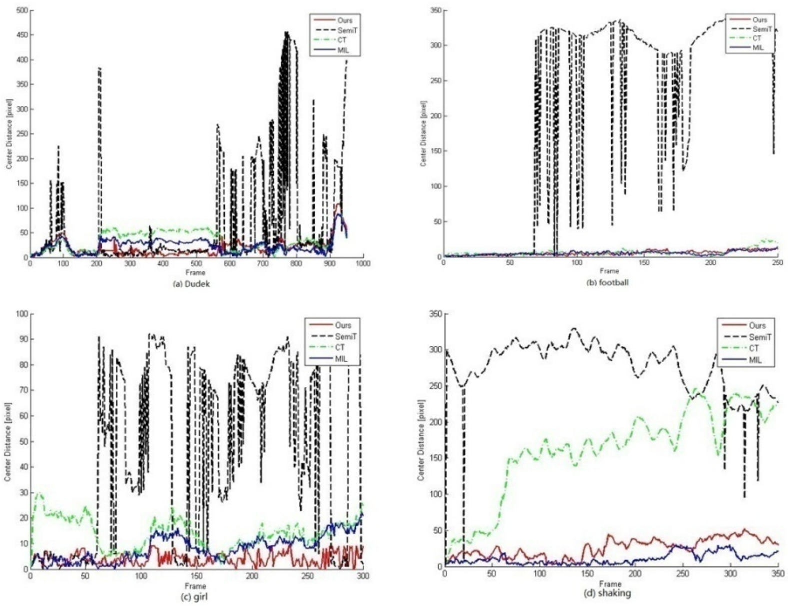

Fig. 3. The quantitative tracking results on the benchmark sequences

As we can see from the graphs, the online drift detection technique greatly improves the tracking accuracy of SemiT. The average center location errors of the tested trackers are shown in Table 1. The best and second best results are shown in red and blue respectively.

Table 1. The average center location errors of the four trackers

	Dudek	football	girl	shaking
SemiT[4]	124.3	232.4	63.5	275.1
CT[9]	38.2	4.1	13.9	147.8
MIL[10]	22.7	3.6	9.8	13.3
Ours	13.6	4.7	6.5	20.3

5.2 Qualitative Comparison

In this section, we perform qualitative evaluation on five testing sequences (*Dudek, girl, football, shaking, David*). Fig.4 shows the tracking results. Note that we won't draw the bounding box if the tracker loses the target.

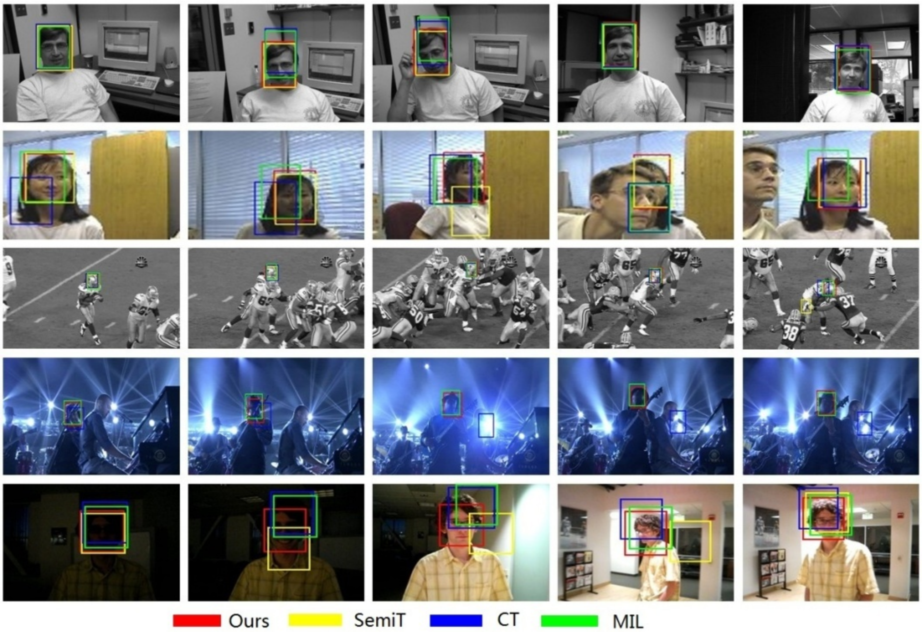

Fig. 4. The tracking results on *Dudek*, *girl*, *football*, *shaking*, *David* respectively

In general, our approach and MIL yield the best tracking results among the four. SemiT performs worst because it can't handle appearance changes properly and loses the target frequently. Both CT and MIL get a drift problem after the target is occluded in some frames. Besides, CT seems to have difficulty handling significant illumination changes.

6 Conclusion

In this paper, we propose a framework that combines concept drift detection with visual tracking to improve tracking performance. Experiments show that our tracker is of high accuracy in real-world scenarios. Further research could be done on exploring more powerful concept drift techniques to get a better tracking result.

References

1. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. Knowledge-Based Systems 18(4), 187–195 (2005)
2. Cunningham, P., Nowlan, N., Delany, S.J., Haahr, M.: A case-based approach to spam filtering that can track concept drift. In: The ICCBR, pp. 3–16 (2003)
3. Chen, L., Zhou, Y., Yang, J.: Object tracking within the framework of concept drift. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 152–162. Springer, Heidelberg (2013)
4. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
5. Kumar Mallapragada, P., Jin, R., Jain, A.K., Liu, Y.: Semiboost: Boosting for semi-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(11), 2000–2014 (2009)
6. Leistner, C., Grabner, H., Bischof, H.: Semi-supervised boosting using visual similarity learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
7. Batu, T., Fortnow, L., Rubinfeld, R., Smith, W.D., White, P.: Testing that distributions are close. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 259–269. IEEE (2000)
8. Grabner, H., Bischof, H.: On-line boosting and vision. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 260–267. IEEE (June 2006)
9. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
10. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 983–990. IEEE (June 2009)