Can Diversity Amongst Learners Improve Online Object Tracking?

Georg Nebehay[1], Walter Chibamu[2], Peter R. Lewis[2],
Arjun Chandra[3], Roman Pflugfelder[1], and Xin Yao[2]

[1] Austrian Institute of Technology, Austria
[2] CERCIA, School of Computer Science, University of Birmingham, UK
[3] Department of Informatics, University of Oslo, Norway
georg.nebehay.fl,roman.pflugfelder@ait.ac.at
wcc081,p.r.lewis,x.yao@cs.bham.ac.uk
chandra@ifi.uio.no

Abstract. We present a novel analysis of the state of the art in object tracking with respect to diversity found in its main component, an ensemble classifier that is updated in an online manner. We employ established measures for diversity and performance from the rich literature on ensemble classification and online learning, and present a detailed evaluation of diversity and performance on benchmark sequences in order to gain an insight into how the tracking performance can be improved.

1 Introduction

We deal with the problem of single-target model-free object tracking in videos, meaning that a single object is to be tracked and no a priori information about the object is available. Many authors (e.g. [14, 17, 24, 25]) formulate the task of object tracking as a binary classification problem, and use ensembles of multiple learners as binary classifiers. One of the elements required for accurate prediction in ensembles is error diversity [6]. While measures for diversity have been considered explicitly in the context of object tracking before [26], in this work, we take a different path and analyse the diversity in the state of the art object tracking method TLD (Tracking-Learning-Detection [17]) in order to gain an insight into how its performance can be improved by manipulating diversity.

As TLD consists of multiple interleaved components, we focus our analysis on its most influential component, a random fern classifier [23]. While it is not clear yet whether our findings generalize to the original TLD method, or to other object tracking methods, we do establish a baseline with the analysis of the random fern classifier, against which more involved methods can be evaluated in future. The contributions of this paper are threefold: firstly, we show how diversity can be measured in TLD. Secondly, we provide a detailed analysis with respect to diversity and performance. Thirdly, we hint at ways how performance might be improved.

This work is structured as follows. In section 2 we discuss related work in object tracking and machine learning. In section 3 we describe the state of the art tracking method TLD. In section 4, we lay out our experimental setup. In section 5 we present our analysis of diversity and performance, and section 6 gives conclusions and final remarks.

2 Related Work

In this section, we first review related work in online learning for object tracking, and secondly describe existing techniques for the engineering of diversity in ensembles of learners.

2.1 Online Learning in Object Trackers

Collins et al. [8] were the first to employ binary classification in a tracking context, the two classes being the object and the immediate surrounding. They employ feature selection in order to switch to the most discriminative colour space from a set of candidates and use mean-shift for finding the mode of a likelihood surface, thereby locating the object. In a similar spirit, Grabner et al. [14] perform online boosting and Babenko et al. [1] use multiple instance learning in order to find the location of the object. All of these methods use a form of reinforcement learning, meaning that the prediction of the classifier is directly used to update the classifier. While this approach enables the use of unlabelled data for training, it typically amplifies errors made in the prediction phase, thus leading to a degradation of tracking performance. In [15], this problem is addressed by casting object tracking as a semi-supervised learning problem, where only the first appearance of the object is used for updating. Both Kalal et al. [17] and Santner et al. [25] employ an optic-flow-based mechanism for labelling the available data in order to reduce the errors made in the prediction phase and demonstrate superior results.

2.2 Diversity of Ensembles in Object Tracking

In machine learning generally, *diverse* ensembles of classifiers often provide better prediction accuracy than any of the individual members of the ensemble [6]. Visentini et al. [26] employ a combined measure of diversity and performance to select classifiers from a pool for adaptive object tracking. Additionally, diversity has been considered more generally in computer vision. Bertolami and Bunke [2] use diversity measures as indicators for the accuracy of ensemble classification for handwriting recognition. Frinken et al. [13] increase the diversity of a handwriting recognition system by combining Neural Networks, Maximum Margin Hidden Markov Models and Hidden Markov Models, and show that high diversity leads to better results. Levy et al. [19] force classifiers to learn different aspects of the data by minimizing correlation between ensemble members and show improved results on visual recognition problems.

2.3 Engineering Diversity in Online Learning

The literature is abound with methods for encouraging diversity in ensembles. Attempts at consolidating these methods into taxonomies have also been made [6, 9], which can provide guidelines for encouraging diversity in different ways.

The taxonomy by Dietterich [9] consolidates ensemble creation methods into various categories with diversity encouragement being at the heart. For the discussion in this section, we assume a standard supervised learning problem: a learning algorithm is presented with a training set $\mathcal{S}$ $\{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_N, y_N)\}$ of size N for learning some unknown function $y = f(\mathbf{x})$. The learning algorithm outputs a classifier, which is a hypothesis $h_i \in \mathcal{H}$ about the true underlying function f. The various methods found in such taxonomies have been applied mostly in the offline learning mode. They can however be adapted to the online case (e.g. [21, 22]), where training instances continuously arrive one at a time as a stream of data. A brief overview of the taxonomy now follows:

Bayesian voting. In problems where it is possible to enumerate each hypothesis $h_i \in \mathcal{H}$, and calculate a prior $P(h)$, the problem of classifying a new example $\mathbf{x}$ amounts to computing $P(f(\mathbf{x}) = y|\mathcal{S}, \mathbf{x}) = \sum_{h \in \mathcal{H}} h(\mathbf{x}) P(h|\mathcal{S})$. This can be viewed as an ensemble consisting of all possible hypotheses in $\mathcal{H}$, where each hypothesis h is weighted by its posterior probability $P(h|S)$. However, Bayesian voting fails where it is not possible to enumerate all possible hypotheses and calculate the prior $P(h)$.

Manipulating training examples. L iterations of the learning algorithm are run. In each iteration a different subset of the training set $\mathcal{S}$ is used to train the classifier h_i, $i = 1 \ldots L$, thus generating multiple classifiers, each trained on a different training set. Example algorithms in this category are Bagging [3], Cross validated committees, and AdaBoost [12].

Manipulating input features. The input features are divided into feature subsets, and in each iteration i of the learning algorithm, a classifier is trained on a subset(s) of the input features. The random subspace method [16] falls into this category.

Injecting randomness. Some randomness can be induced into the learning setup, for example in a neural network ensemble by using different initial weights, or injecting noise into the input features following bootstrap sampling.

Manipulating output targets. The error-correcting output code technique [10] manipulates the y labels of the training examples in classification problems where the number of classes, k, is large. Instead of learning the problem on the original k classes, in each iteration $i = 1 \ldots L$, the k classes are divided into two subsets $\mathcal{A}$ and $\mathcal{B}$ (different in each iteration) and the input data re-labelled 0 and 1 respectively for classes in subsets $\mathcal{A}$ and $\mathcal{B}$. This results in L classifiers $h_1 \ldots h_L$. To classify a new data point $\mathbf{x}$, if $h_i(\mathbf{x}) = 0$, then each class in subset $\mathcal{A}$ receives a vote and if $h_i(\mathbf{x}) = 1$, then each class in subset $\mathcal{B}$ receives a vote. Once all L classifiers have voted, the class with the largest prediction is selected as the ensemble output.

Manipulating error functions. Diversity can be explicitly encouraged and maintained by defining and minimising a correlation term between ensemble members. Negative correlation encourages individual members to learn different parts of the training data (specialisation) allowing the ensemble to learn the entire training data better than any single or monolithic member [20]. Ensemble members are trained simultaneously allowing the members to interact and cooperate through a correlation penalty term that is introduced in the error function such that the individual error of each member is negatively correlated to the rest of ensemble errors [7].

Diversity Metrics Several measures for a quantitative assessment of diversity in ensembles have been proposed in the literature. Kuncheva et al. [18] have conducted a wide and detailed study of various diversity measures, and conclude that there is no unique way of measuring diversity, and in general, there is no direct or distinctive relationship between the diversity of an ensemble and its accuracy. One of the most commonly used diversity measures, the *Q-statistic* [18] is calculated in a pairwise manner for any two classifiers f_i and f_j:

$$Q_{i,j} = \frac{ad - bc}{ad + bc} \tag{1}$$

The symbols a, b, c, d refer to the number of times
a : f_i and f_j are correct,
b : f_i is correct, f_j is incorrect,
c : f_i is incorrect, f_j is correct,
d : f_i and f_j are incorrect.
$Q_{i,j}$ is closer to 1 if the output of the classifiers is not diverse, and is closer to -1 if their output is diverse. An overall measure for the diversity of an ensemble of size n is then obtained by averaging all of the pairwise measurements.

3 State of the Art in Object Tracking

3.1 Tracking-Learning-Detection

Kalal et al. [17] propose a solution to the tracking problem which they call *Tracking-Learning-Detection* (TLD). TLD consists of two separate components: A **frame-to-frame tracker** that predicts the location L_j of the object in frame I_j by calculating the optical flow between frames I_{j-1} and I_j and transforming L_{j-1} accordingly. Clearly, this approach is only feasible as long as the object is visible in the scene and fails otherwise. When the object is presumably tracked correctly (according to certain criteria) the location L_j is used in order to update a **Random Fern** classifier [23] with positive training data from patches close to L_j and negative data from patches that exceed a distance. This classifier is then applied in a sliding-window manner (see figure 1) in order to re-initialize the frame-to-frame-tracker after failure. Two additional stages not described here are used for classification.

3.2 Random Fern Classifier

The Random Fern classifier [23] operates on binary features $f_1 \dots f_n$ calculated on the raw image data. These features are randomly partitioned into groups of so-called *ferns* $F_1 \dots F_m$ of size s

$$\overbrace{f_1 \cdots f_s}^{F_1}, \overbrace{f_{s+1} \cdots f_{2s}}^{F_2} \cdots \overbrace{f_{(m-1)s+1} \cdots f_{ms}}^{F_m}. \tag{2}$$

Ferns essentially are non-hierarchical trees, meaning that the outcome of each fern is independent of the order in which features are evaluated. The main reason for favouring

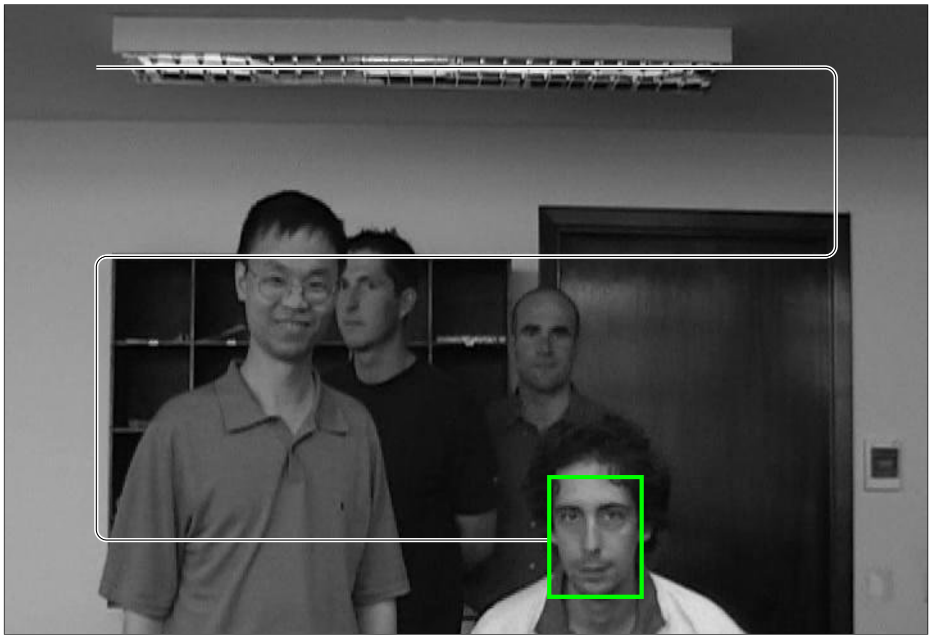

Fig. 1: In TLD, a binary ensemble classifier is used to locate the object of interest by applying it in a sliding-window manner. The ability for multi-scale detection is achieved by scaling the size of the detection window. Image is from the SPEVI[1] dataset.

ferns over trees is that they can be implemented extremely efficiently, an important property for real applications.

3.3 Features

In [23], a feature vector of size s consists of s binary tests performed on gray-scaled image patches. Each test compares the brightness values of two random pixels (See figure 2). The locations of the tests are generated once at startup and remain constant throughout the rest of the processing. The same set of tests is used with appropriate scaling for all subwindows. Input images are smoothed with a Gaussian kernel to reduce the effect of noise.

3.4 Random Ferns in TLD

The posterior probability for each fern is

$$P(y=1|F_k) = \frac{P(y=1)P(F_k|y=1)}{\sum_{i=0}^{1} P(y=i)P(F_k|y=i)}. \tag{3}$$

[1] `http://www.eecs.qmul.ac.uk/~andrea/spevi.html`

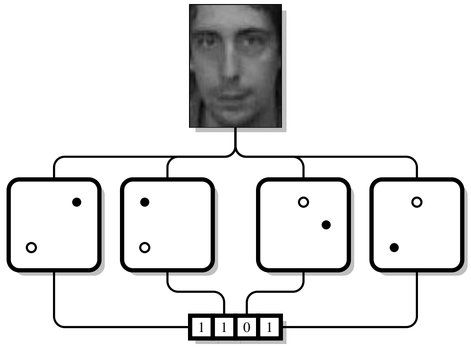

Fig. 2: Feature values depends on the brightness values of pairs of two random pixels. In this case, the outcome is the binary string 1101.

In TLD, the prior is assumed to be uniform, and the $P(F_k|y=i)$ are modelled as the absolute number of occurrences $\#p_{F_k}$ for positive training data and $\#n_{F_k}$ for negative training data. Therefore, the posterior probability becomes

$$P(y=1|F_k) = \frac{\#p_{F_k}}{\#p_{F_k} + \#n_{F_k}}. \tag{4}$$

When $\#p_{F_k} = \#n_{F_k} = 0$, then $P(y=1|F_k)$ is assumed to be 0 as well. Each training instance is used for training only if it was misclassified in the current frame. A decision is obtained by employing a threshold θ on the posterior probabilities combined using the mean rule

$$\frac{1}{m}\sum_{i=1}^{m} P(y=1|F_i) \geq \theta. \tag{5}$$

4 Experimental Setup

We conduct experiments according to the following novel pattern in order to assess the diversity and the performance of the Random Fern classifier in TLD. For each frame, we closely follow the predict-update cycle of classical online learning: first we let the classifier predict labels for all subwindows. We then measure performance and diversity using the ground truth values and update the classifier according to the misclassified examples. Each experiment is run 10 times with different seeds for the random number generator. Over these runs, the mean and standard deviation of the selected metrics for performance and diversity are reported. We apply the following modifications to the original algorithm [17].

- Majority voting is used instead of the mean rule. Crisp outputs are obtained by applying the threshold θ on the posterior probabilities of the individual classifiers.
- We replace the optic-flow based tracker with manually labeled ground truth.
- We disregard the two classification stages besides the random fern classifier.

The first modification enables the use of the Q statistic. We perform the last two modifications since we are interested only in the performance limits of the classifier. The analysis of this modified version gives us a baseline against which to evaluate more involved methods in the future.

4.1 Performance Measures

We use the following statistics to measure the performance, based on the occurrences of *True Positives* (TP), *False Negatives* (FN) and *False Positives* (FP) in each frame. TPs, FNs and FPs are found by comparing algorithmic output to manually annotated ground truth. Recall, given by

$$R_j = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FN}_j}, \tag{6}$$

measures the fraction of positive instances that were correctly classified as positive. Precision, given by

$$P_j = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FP}_j}, \tag{7}$$

measures the fraction of examples classified as positive that are truly positive. The F-measure, given by

$$F_j = \frac{2R_j P_j}{R_j + P_j}, \tag{8}$$

as the harmonic mean, combines precision and recall into a single measurement. We calculate R_j, P_j and F_j for each frame and report their average values R, P, F over the whole sequence.

As the employed set of subwindows is not exhaustive, there will typically be no single subwindow of the same location and the same dimension as the manual annotation. We therefore employ the measure used in the Pascal Visual Object Challenge [11] for overlap between two bounding boxes B_1 and B_2, namely,

$$overlap = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{I}{(B_1 + B_2 - I)}. \tag{9}$$

If the overlap between a manual annotation and a subwindow is larger than 0.5, the subwindow is labelled positive as well.

We employ the Q-statistic (section 2.3) as a measure for diversity in each frame and report averaged values over the whole sequence. While other diversity measures are available, we chose the Q-statistic as a starting point for our analysis primarily due to its widespread use. However, we plan to investigate different measures of diversity in future work.

4.2 Sequences

We employ the following six sequences for conducting our evaluation. These sequences were used in [17,27] for evaluating object tracking methods. **David** (761 frames) shows

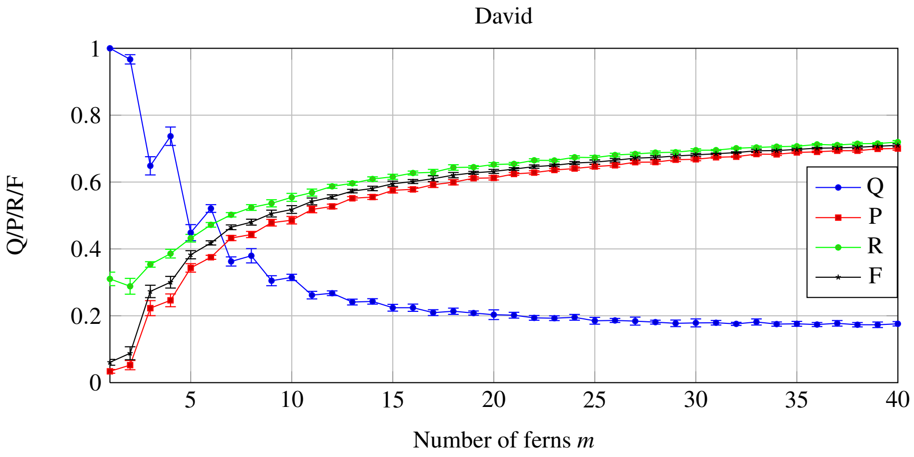

Fig. 3: Both diversity and performance exhibit a convergent behaviour when the number of ferns m is increased.

a person walking from an initially dark setting into a bright room and undergoing various changes in appearance. **Jumping** (313 frames) shows a person jumping rope causing motion blur. **Pedestrian 1** (140 frames), **Pedestrian 2** (338 frames) and **Pedestrian 3** (184 frames) show pedestrians being filmed by an unstable camera. **Car** (945 frames) shows a moving car, exposed to low contrast recording and undergoing multiple occlusions. The appearance of the car itself stays constant over the run of the sequence.

5 Diversity Analysis of TLD

In this section we present novel analyses of diversity within TLD based object tracking. Firstly, we explore the effect of varying the parameters of the system on the selected metrics. Secondly, we artificially increase diversity in the system and analyse the resulting effects. We use the parameters $m = 30, s = 14, \theta = 0.5$ unless noted otherwise.

5.1 Effect of Parameters

The parameter m steers the number of classifiers in the ensemble. Breiman [4] proved that an ensemble of randomized decision trees does not overfit as more trees are added, meaning that performance does not decrease. However it is not clear how m affects diversity. In figure 3 we plot Q and F against m for the sequence *David*. Increasing m leads to a convergent behaviour of Q, similar to the performance metric. Q converges more quickly than the performance metrics. These findings generalize to all sequences.

The parameter θ directly influences recall and precision. High values of θ lead to an improvement of precision, as false positives are filtered out, and to a degradation of recall. Low values of θ lead to the inverse effect. Intuitively, both high and low values

		Sensitivity threshold θ				
Sequence	Metric	0.1	0.3	0.5	0.7	0.9
car	Q	0.31±0.01	0.29±0.01	0.28±0.01	0.27±0.01	0.26±0.01
	P	0.62±0.01	0.76±0.00	0.81±0.00	0.84±0.00	0.86±0.00
	R	0.95±0.00	0.92±0.00	0.90±0.00	0.85±0.00	0.73±0.00
david	Q	0.21±0.01	0.19±0.01	0.18±0.01	0.17±0.01	0.16±0.01
	P	0.28±0.02	0.56±0.01	0.67±0.00	0.74±0.00	0.74±0.01
	R	0.82±0.00	0.76±0.00	0.70±0.00	0.58±0.01	0.34±0.01
jumping	Q	0.24±0.01	0.22±0.01	0.21±0.01	0.21±0.01	0.20±0.01
	P	0.36±0.01	0.59±0.00	0.68±0.00	0.76±0.00	0.78±0.01
	R	0.85±0.00	0.77±0.00	0.70±0.00	0.58±0.00	0.35±0.01
pedestrian1	Q	0.30±0.01	0.27±0.01	0.26±0.01	0.26±0.01	0.25±0.01
	P	0.23±0.01	0.38±0.01	0.45±0.01	0.53±0.01	0.52±0.01
	R	0.53±0.01	0.44±0.01	0.38±0.00	0.26±0.01	0.14±0.01
pedestrian2	Q	0.31±0.01	0.29±0.01	0.28±0.01	0.27±0.01	0.26±0.01
	P	0.35±0.01	0.53±0.01	0.62±0.01	0.74±0.01	0.77±0.02
	R	0.71±0.01	0.68±0.01	0.65±0.01	0.58±0.01	0.45±0.01
pedestrian3	Q	0.47±0.01	0.45±0.01	0.44±0.01	0.44±0.01	0.42±0.01
	P	0.53±0.01	0.68±0.01	0.76±0.01	0.84±0.01	0.87±0.01
	R	0.92±0.01	0.87±0.01	0.83±0.01	0.75±0.01	0.57±0.01

Table 1: Increasing θ leads to an increase of diversity due to many positive instances being misclassified, thus increasing the size of the positive training set.

of θ should lead to a reduction of diversity, as the output of the individual classifiers become more similar. In table 1, θ is varied for all sequences. Surprisingly, Q decreases monotonically as θ is increased. The explanation for this effect is that high values of theta lead to many positive instances being misclassified, and therefore the set of positive training data becomes larger, causing a reduction of Q.

5.2 Increasing Diversity

In order to artificially increase diversity in the ensemble classifier, we restrict the location of the binary tests for individual classifiers to certain parts of the input image, thus decreasing the amount of information shared between them. For each classifier we randomly sample a value μ_j. We then generate the binary tests from the two-dimensional uniform distribution $U(max(0,\mu_j-\sigma),min(1,\mu_j+\sigma))$. Brown and Kuncheva [5] show that the majority vote error can be decomposed into the sum of individual errors (additive term), diversity measured on correctly classified instances called good diversity (subtractive term) and diversity measured on misclassified instances called bad diversity (additive term). While the decomposition of the F measure into analogous Q terms is unknown, the notions of good and bad diversity are still helpful in our context. For this experiment, we measure Q both on correctly classified instances (Q_{good}) and on misclassified instances (Q_{bad}).

When σ is decreased, we make the following observations for all sequences in table 2: Q and Q_{good} decrease strongly. Q_{bad} starts out closely above the theoretical min-

		Feature locality $1-\sigma$				
Sequence	Metric	0.1	0.3	0.5	0.7	0.9
car	Q	0.28±0.01	0.23±0.01	0.16±0.01	0.10±0.00	0.08±0.00
	Q_{good}	0.27±0.01	0.22±0.01	0.15±0.01	0.10±0.00	0.07±0.00
	Q_{bad}	0.01±0.00	0.00±0.00	0.00±0.00	-0.00±0.00	-0.01±0.00
	F	0.86±0.00	0.86±0.00	0.86±0.00	0.85±0.00	0.77±0.01
david	Q	0.19±0.01	0.17±0.01	0.13±0.01	0.11±0.00	0.11±0.00
	Q_{good}	0.18±0.01	0.16±0.01	0.13±0.01	0.11±0.00	0.10±0.00
	Q_{bad}	0.00±0.00	0.00±0.00	-0.00±0.00	-0.00±0.00	0.00±0.00
	F	0.69±0.00	0.69±0.00	0.69±0.01	0.67±0.01	0.51±0.01
jumping	Q	0.22±0.01	0.20±0.02	0.16±0.02	0.09±0.01	0.07±0.00
	Q_{good}	0.21±0.01	0.19±0.02	0.15±0.02	0.09±0.01	0.07±0.00
	Q_{bad}	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.00±0.00
	F	0.70±0.00	0.70±0.01	0.69±0.01	0.66±0.01	0.44±0.02
pedestrian1	Q	0.26±0.01	0.25±0.01	0.21±0.01	0.15±0.01	0.08±0.00
	Q_{good}	0.23±0.01	0.22±0.01	0.18±0.01	0.13±0.01	0.07±0.00
	Q_{bad}	0.04±0.00	0.04±0.00	0.03±0.00	0.03±0.00	0.02±0.00
	F	0.41±0.01	0.41±0.01	0.41±0.02	0.40±0.02	0.34±0.02
pedestrian2	Q	0.27±0.01	0.26±0.01	0.21±0.01	0.14±0.01	0.08±0.00
	Q_{good}	0.26±0.01	0.25±0.01	0.20±0.01	0.13±0.01	0.07±0.00
	Q_{bad}	0.04±0.00	0.03±0.00	0.03±0.00	0.02±0.00	0.01±0.00
	F	0.66±0.01	0.67±0.01	0.66±0.01	0.62±0.03	0.49±0.04
pedestrian3	Q	0.45±0.01	0.44±0.02	0.38±0.02	0.22±0.01	0.09±0.01
	Q_{good}	0.44±0.01	0.42±0.02	0.37±0.02	0.21±0.01	0.08±0.00
	Q_{bad}	0.04±0.00	0.04±0.00	0.03±0.00	0.03±0.00	0.02±0.00
	F	0.81±0.01	0.81±0.01	0.80±0.01	0.77±0.01	0.71±0.02

Table 2: Diversity increases when the locations of the binary tests become more local. Q_{bad} indicates that diversity in the classification result of misclassified instances is already very high from the start.

imum $-\frac{1}{m}$, decreasing only slightly. Depending on the sequence, performance rapidly decreases at a certain value of σ. Increasing diversity the way we have seems to increase the error of the individual classifiers. For low values of σ, this increase is compensated for by a decreased Q_{good}, leading to a stable F. For high values of σ, the errors of the individual classifiers seem to outweigh the increased good diversity, leading to a reduction of F.

These observations suggest that we need to find a way to encourage diversity that keeps the individual classifiers from exhibiting an increased error. Since Q_{bad} is close to the theoretical minimum, increasing it can help us increase ensemble performance. Devising a training scheme that is informed by the wrongly classified instances may be one way of increasing Q_{bad}. Further analysis of the relationship between Q_{good}, Q_{bad}, and individual classifier performance, will shed more light on ways to encourage diversity that may lead to an increased overall performance.

6 Conclusions and Future Work

In this work, we presented an analysis of the state of the art in object tracking with respect to diversity and showed how it is influenced by the intrinsic parameters of its ensemble classifier. We also showed how diversity can be increased artificially and conclude that performance is reduced due to an increased error of the individual classifiers. We plan to look into methods that increase good diversity while keeping the individual accuracy stable. We also acknowledge the fact that reducing bad diversity will help increase performance.

A better understanding of the relationship between performance of individual classifiers, as well as between good and bad diversity, will help show ways on how overall performance can be increased. We also plan to explicitly reduce correlation in the system by making use of algorithms similar to minimal correlation learning [19].

As only misclassified examples are used for training, the classifier highly overfits the training data. This does not to lead to a reduction in performance as long as sequences contain sufficient training examples. When short sequences with severe changes in appearance occur, performance is affected in a negative way. The results of Minku et al. [21] suggest that an increased level of diversity could help in exactly these cases.

References

1. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. Pattern Analysis and Machine Intelligence 33(8) (Aug 2011)
2. Bertolami, R., Bunke, H.: Diversity analysis for ensembles of word sequence recognisers. In: Structural, Syntactic, and Statistical Pattern Recognition (2006)
3. Breiman, L.: Bagging Predictors. Machine Learning 24, 123–140 (1996)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (Oct 2001)
5. Brown, G., Kuncheva, L.I.: “Good” and “Bad” Diversity in Majority Vote Ensembles. In: Multiple Classifier Systems. pp. 124–133 (2010)
6. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: A survey and categorisation. Journal of Information Fusion 6, 5–20 (2005)
7. Chen, H., Yao, X.: Multiobjective neural network ensembles based on regularized negative correlation learning. Knowledge and Data Engineering 22(12) (2010)
8. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
9. Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple Classifier Systens, LBCS-1857. pp. 1–15. Springer (2000)
10. Dietterich, T.G., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. Journal of Artificial Intelligence Research 2 (1995)
11. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88(2), 303–338 (Jun 2010)
12. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory 904, 23–37 (1995)
13. Frinken, V., Peter, T., Fischer, A., Bunke, H., Do, T.M.T., Artieres, T.: Improved handwriting recognition by combining two forms of hidden markov models and a recurrent neural network. In: Computer Analysis of Images and Patterns (2009)

14. Grabner, H., Bischof, H.: On-line boosting and vision. In: Computer Vision and Pattern Recognition. vol. 1 (2006)
15. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised On-Line boosting for robust tracking. In: European Conference on Computer Vision (2008)
16. Ho, T.K.: The random subspace method for constructing decision forests. Pattern Analysis and Machine Intelligence 20(8), 832–844 (Aug 1998)
17. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-detection. Pattern Analysis and Machine Intelligence 34(7), 1409–1422 (Jul 2012)
18. Kuncheva, L.I., Whitaker, C.J.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. Machine Learning 51(2), 181–207 (May 2003)
19. Levy, N., Wolf, L.: Minimal correlation classification. In: European Conference on Computer Vision (2012)
20. Liu, Y., Yao, X., Higuchi, T.: Evolutionary ensembles with negative correlation learning. Evolutionary Computation 4(4), 380 – 387 (Nov 2000)
21. Minku, L.L., White, A.P., Yao, X.: The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift. Knowledge and Data Engineering 22(5), 730–742 (May 2010)
22. Oza, N.C.: Online Bagging and Boosting. In: Systems, Man and Cybernetics (2005)
23. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast keypoint recognition using random ferns. Pattern Analysis and Machine Intelligence 32(3), 448–461 (2010)
24. Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H.: On-line random forests. In: International Conference on Computer Vision Workshops (2009)
25. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: PROST: Parallel robust online simple tracking. In: Computer Vision and Pattern Recognition (2010)
26. Visentini, I., Kittler, J., Foresti, G.: Diversity-Based classifier selection for adaptive object tracking. In: Multiple Classifier Systems (2009)
27. Yu, Q., Dinh, T.B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: European Conference on Computer Vision (2008)