Predicting Soccer Highlights from Spatio-temporal Match Event Streams (Supplementary material)

Anonymous

This supplementary material contains illustrations of the steps of highlight prediction with POGBA and detailed results of our auxiliary experiments.

Highlight prediction with POGBA

Figure 1 shows raw goal probability estimates by the generative model underlying the POGBA algorithm. Figure 2 shows the same sequence of probability estimates after exponential smoothing and the predicted highlights corresponding to the peaks in the sequence.

Auxiliary experiments

Goal attempt prediction

To illustrate the utility of considering full spatio-temporal data, we consider a simplified version of the shot prediction task, where we only predict whether a shot will occur in the next window, instead of predicting its attributes, as required by the full highlight prediction pipeline. We compare the performance of the nearest-neighbor classifier with two different distance measures. One is the DTW approach used in the previous experiments. The other is the Euclidean distance between the last events in each window, similar to the static baselines from the main experiment.

We label each window in the sequence collection as positive if there was a shot in the next window. On average, there are 1,125 windows per match, out of which 58 (5%) are positive. Due to the skewed class distribution, we use the area under the ROC curve (AUROC) as the performance measure. As in the main experiment, the ROC curves are obtained using the "leave-one-match-out" procedure, where a window in a particular match is classified based on its nearest neighbors from the other 68 matches.

Figure 3 illustrates the results. The DTW-based approach that considers full sequence data clearly outperforms the non-sequential baseline, obtaining AUROC values of 0.8 and 0.6, respectively.

Goal probability estimation

We evaluate various approaches to goal probability estimation. This is equivalent to a standard probabilistic classification task. We extract all preconditional events from the sequence data and assign a binary label to each event, where a positive label implies that a precondition (goal attempt) was followed by a critical event (goal). This yields 2,027 examples in total, out of which 188 (9%) are positive. We compare four state-of-the-art probabilistic classifiers: *Naive Bayes*, *Logistic Regression*, *Random Forests*, and *Extremely Randomized Trees*. We use the implementations in the *scikit-learn* library.[1] For both *Random Forests* and *Extremely Randomized Trees*, we set the number of trees in the forest to 1000. The ROC curves are obtained using 10-fold cross-validation.

Classifier	AUROC
Naive Bayes	0.727
Logistic Regression	0.733
Random Forests	0.771
Extremely Randomized Trees	0.794

Table 1: Goal probability estimation: AUROC values for various classifiers.

Table 1 and Figure 4 summarize the results. The ROC curves of the classifiers are close to each other, which indicates that the overall performance of the proposed pipeline is not sensitive to the choice of specific building blocks. We use *Extremely Randomized Trees*, which perform best in terms of AUROC.

Figures

See below.

[1] http://scikit-learn.org

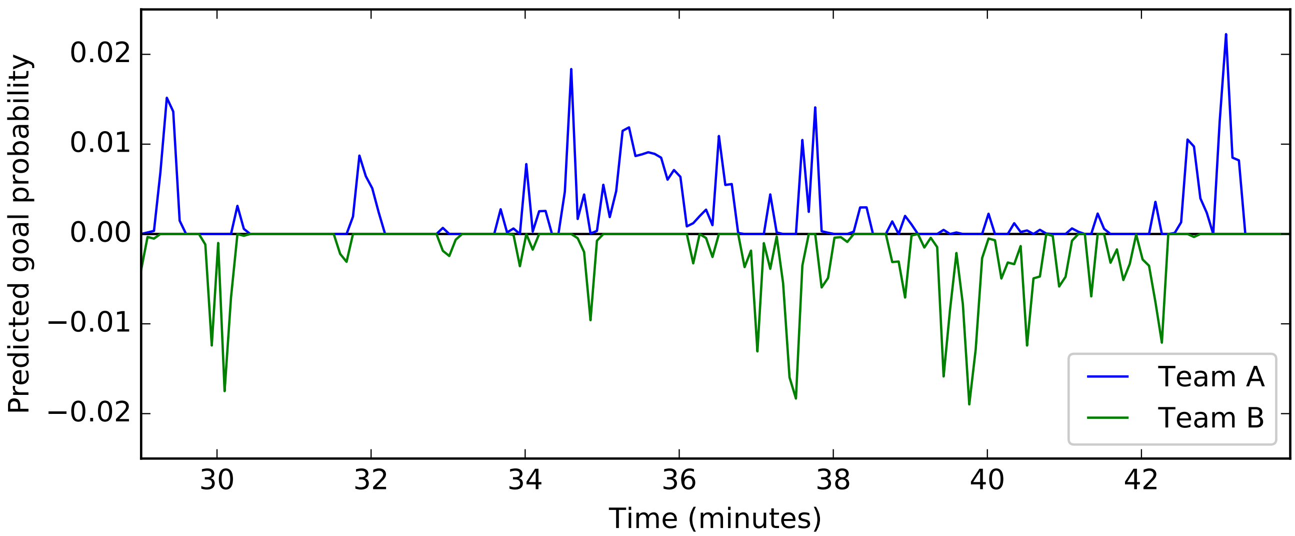

Figure 1: Raw goal probabilities for 15 minutes of gameplay. The goal probabilities of Team B have been reflected over the x-axis for clarity.

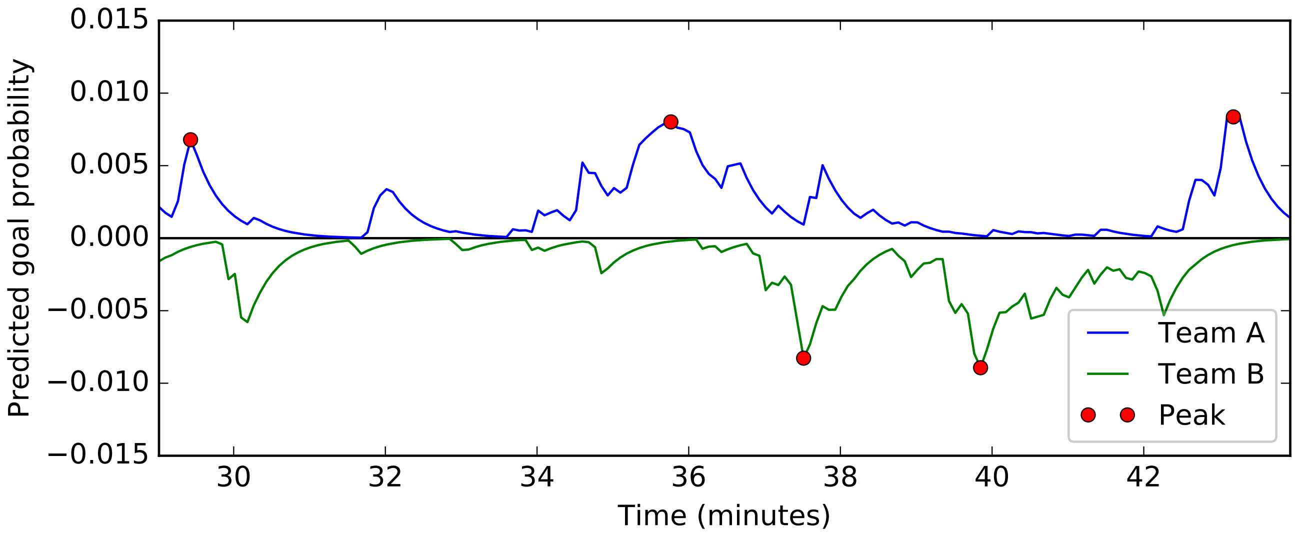

Figure 2: Smoothed probabilities for 15 minutes of gameplay. The goal probabilities of Team B have been reflected over the x-axis for clarity.

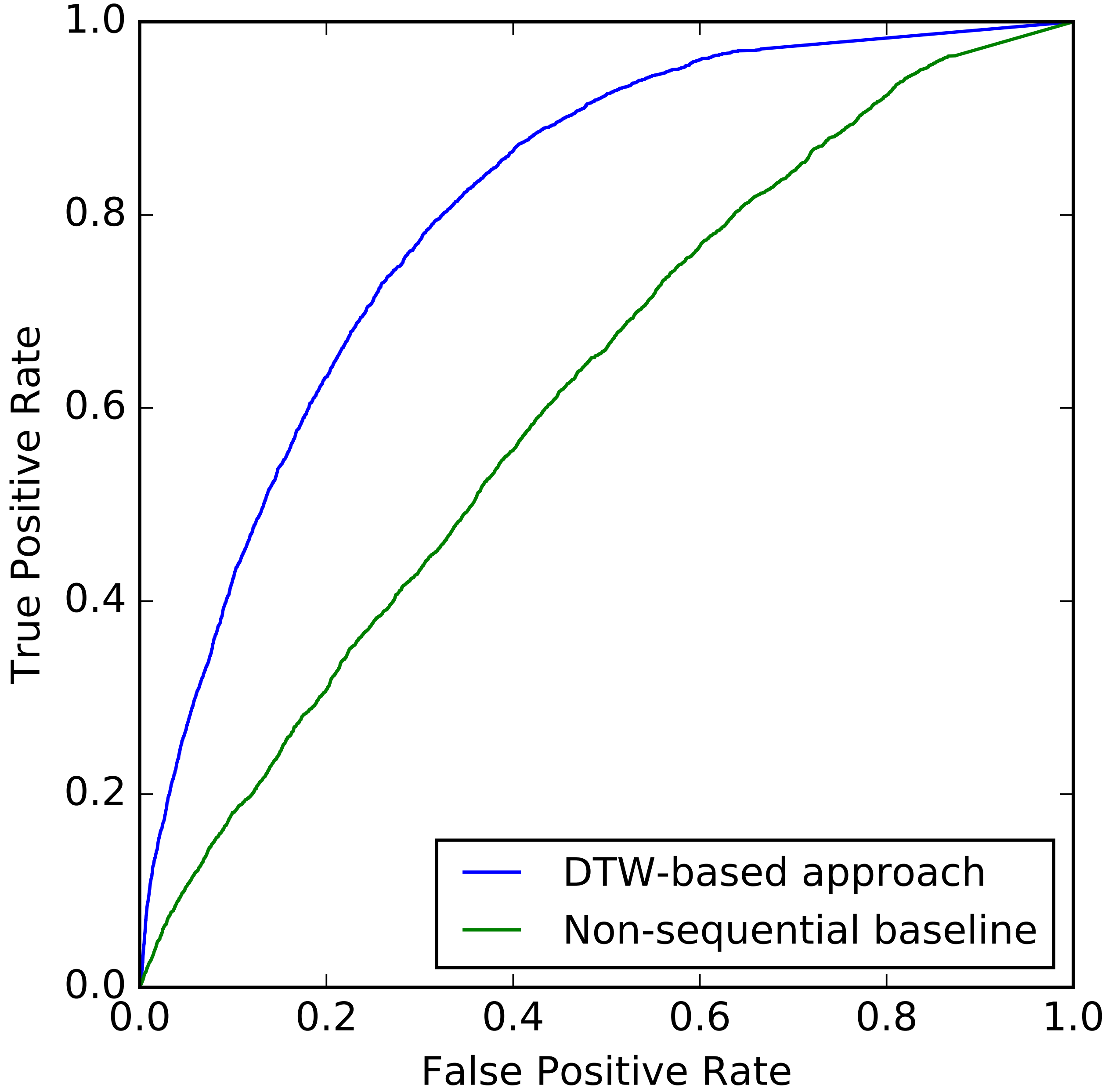

Figure 3: ROC curves for the shot prediction task. The DTW-based approach that considers the full sequence data dominates the baseline that only considers the last event in a sequence.

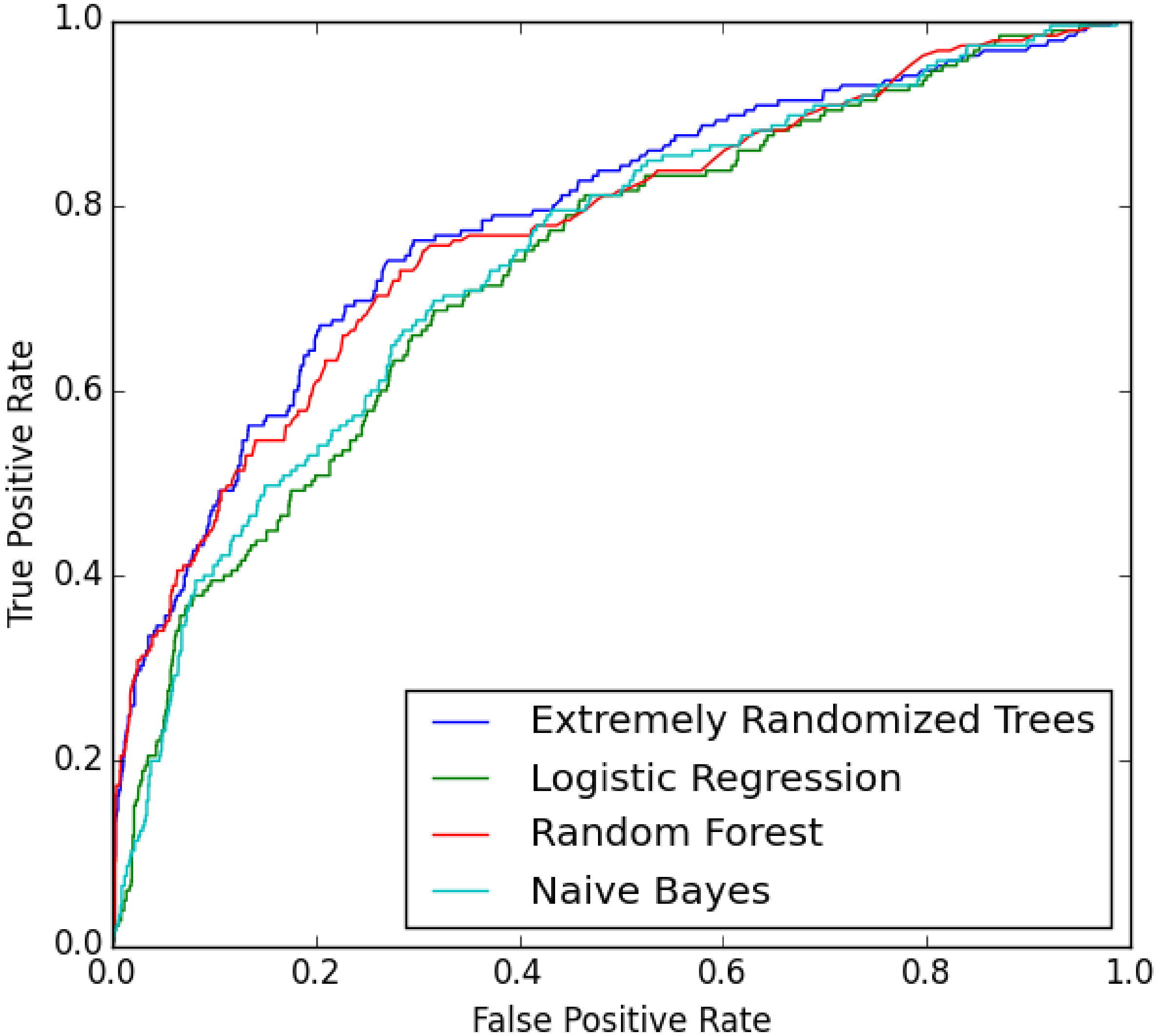

Figure 4: Critical event prediction performance. The ROC curves of various classifiers are close to each other. We use *Extremely Randomized Trees*, which is the best method in terms of AUROC, in the highlight prediction pipeline.