

Title: Review of actions speak louder than goals-valuing player actions in soccer

Rating:

Summary: n soccer matches, the traditional scoring method only considers key actions such as shots or goals, assigns constant weights to each action, and considers the situation only when the action occurs. This method fails to take into account the future impact of these actions on the game. To overcome these limitations, the paper proposes SPADL, a standardized language for individual player actions in soccer, and VAEP, a framework for predicting the outcomes and performances resulting from player actions. Furthermore, the paper demonstrates better predictions with an actual machine learning model based on this framework, showing improved performance over traditional methods.

Strengths:

1. The paper clearly identifies the limitations of traditional soccer scoring systems and proposes a structured data format and framework that can address these issues. This offers a new perspective that problems with model performance might not only arise during modeling but may also be due to unstructured data forms.
2. It covers the entire process of problem recognition and resolution in ML, including 'data preprocessing', 'modeling', 'inference', and 'application' provided through a Python package discussed in the paper, helping to understand the general processes involved in ML in practice.

Weaknesses:

1. While SPADL and VAEP can convey the context up to the point of a specific action, they still cannot judge broader contexts. For example, actions such as marking an opposing forward to prevent a pass, which are meaningful in actual games but not considered in the model, are not included. This limitation may lead to biases towards certain positions.
2. The model is also based on data without ground truth, making it difficult to measure its performance. Although the paper attempts to demonstrate performance using methods like ROC AUC, it lacks a detailed explanation of how false/true criteria were defined in this context.

Questions:

1. In addition to intuitively relevant information like ActionType, BodyType, Result, Time, Location, and Player, SPADL also includes seemingly unrelated attributes like Team. I wonder if such information actually influences the learning process.
2. Given the limited amount of data in soccer matches, I'm curious whether information like Time and Location can also show significant meaning.

Discussions:

1. The data structure SPADL proposed in the paper seems very flexible for adding different forms of action 'if collectable'. However, adding new actions would require retraining the model. Perhaps methods like Framing, discussed in class, could allow for inference without retraining when new actions are added.
-

Background:

1. The various features needed to predict travel time are broadly categorized into static or dynamic information over time. For instance, companies like Kakao and T-Map Mobility use the Customizable Contraction Hierarchies (CCH) algorithm to calculate expected travel times based on spatial information. While the preprocessing stage for learning all regional information nationwide using the CCH algorithm is quite lengthy, it takes relatively less time to reflect changes in the information that only affects the weights between certain edges (here, when the travel time on specific roads changes). In reality, factors that affect travel time include static information that barely changes over time, such as regional information and road structure, and dynamic information like weather and traffic accidents. Since navigation services must reflect real-time information due to the dynamic nature of such information, the CCH algorithm is used. As the importance of how dynamic the features are over time is crucial in this design proposal, we will categorize the features based on their dynamism.

Design:

1. Based on the discussion in the background section, the features to be considered for travel time prediction will be divided into four layers: static layer, cyclic layer, dynamic layer, and user layer.

The "static layer" includes information such as regional data, shortest path algorithms, and road structure information, which are static over time. Although GIS information is typically represented as continuous values based on latitude and longitude, shortest path algorithms operate on graph-based data structures. Thus, road structure information must be transformed into a graph format. In this format, each node of the graph must represent regional information in a geometrically repeating pattern without overlaps or omissions. Hexagons are the most uniform shapes to represent each node's regional and road structure information, used in shortest path algorithms like Dijkstra and CCH.

The "cyclic layer" contains information that changes periodically over time. For example, variations in traffic volume depending on weekdays or weekends and specific times like rush hours are periodic and highly predictable, significantly influencing expected travel times.

The "dynamic layer" includes real-time information that needs to be reflected, such as weather, traffic accidents, and current traffic conditions. Unlike the static and cyclic layers, it is difficult to predict when such events will occur, but it is crucial to predict how much they will impact travel time once they do occur.

Finally, the "user layer" includes personalized information such as driver and vehicle data, which are independent of time.

Challenges:

1. Now, the proposed feature structure in the design phase will explain how it can solve real-world problems. there are 5 problems to explain through our design: understanding distance, spatial division with hexagonal grids, influence of events and environments, database features, real-time computation.
2. "Understanding distance" involves defining the distance between starting and endpoint perspectives. In the proposed method, distance information is included in the static layer. Rather than merely training the model on the latitude and longitude of the start and end locations, the approach uses a shortest path algorithm. The model will keep the graph structure but learn to predict the weights (expected time) of the edges. This method resolves the issue where the L2 distance does not accurately reflect the actual travel distance.
3. "Spatial division with hexagonal grids" explains how geographic data presented in a hexagonal pattern affects features. As mentioned, if we assume each graph node represents an actual regional area in a hexagonal shape, then all nodes will have a hexagonal shape with six different neighboring nodes. If two neighboring hexagonal regions are not connected or cannot be traversed, the edge between these two nodes is removed to form the graph.
4. "Influence of events and environments" addresses how to quantify and train models on intermittent event and environment information. These are included in the dynamic and cyclic layers. For example, if a traffic accident on a road from point A to B delays travel time, the features representing this delay (accident scale, occurrence time, etc.) will be used in the model to learn the impact on the weight of the edge between nodes A and B. This method allows intermittent events to be reflected in travel time predictions.
5. "Database features" considers which features collected during service operation can best predict expected travel times. These features are broadly divided into two types. One type includes features related to time, such as real-time traffic information during commuting hours, which belongs to the cyclic layer. The other type includes features like driving styles and vehicle information, which are included in the user layer and managed there.
6. "Real-time Computation" addresses the technical solutions for handling real-time information concerning computational costs. As mentioned in the design phase, while transforming regional information into a graph format and learning the impact of features included in the dynamic layer on each edge's weight, this approach allows real-time information to be reflected more quickly than relearning the entire structure of the shortest path algorithm.

References

1. 카카오 테크 (2021).카카오맵과 CCH 알고리즘.
Retrieved from <https://tech.kakao.com/2021/05/10/kakaomap-cch/>
2. Customizable Contraction Hierarchies. Julian Dibbelt, Ben Strasser, and Dorothea Wagner. ACM Journal of Experimental Algorithmics, 2016.

Demo Video

Link to the demo video: <https://www.youtube.com/watch?v=yqBIeNaFrc4&abchannel=conauts>

Link to the demo website: <https://420-de-hw2-fe.vercel.app>
