

Soccer Growth : A Player Performance Tree

Junwon Moon

Dept. of Software, Sungkyunkwan University, Suwon, South Korea
mppn98@g.skku.edu

Abstract—본 논문에서는 축구 선수의 성과 데이터를 기반으로 한 Player Performance Tree(PPT)를 제안합니다. PPT는 다양한 속성별로 선수를 평가하고, 시간에 따른 성장을 추적할 수 있는 트리 구조입니다. 이를 통해 선수의 장단점을 분석하고, 맞춤형 훈련 프로그램을 제공할 수 있습니다. 우리는 배열, 연결 리스트, 트리와 PPT의 성능을 비교하였으며, PPT가 데이터 저장, 검색, 정렬 작업에서 우수한 성능을 보임을 확인했습니다. 또한, PPT는 데이터 기반의 결정을 내리는 데 있어 중요한 도구로 활용될 수 있음을 강조합니다.

I. INTRODUCTION

축구 선수의 개인 성장을 추적하고 잠재력을 예측하는 것은 클럽과 국가대표팀 모두에게 중요한 과제입니다. 이를 통해 유망한 선수를 조기에 발굴하고, 선수들의 장단점을 분석하여 맞춤형 훈련 프로그램을 제공할 수 있습니다. 그러나 실제로 이러한 문제를 해결하는 것은 매우 어렵습니다. 선수의 성장과 성과는 다양한 변수에 의해 영향을 받기 때문에 이를 정확히 예측하는 것은 도전적인 과제입니다. 교재에서 소개된 데이터 구조인 배열은 정적 크기를 가지며, 연결 리스트는 동적 크기를 가지지만 두 자료 구조 모두 시간에 따른 변화를 추적하거나 다양한 속성 간의 복잡한 상관관계를 나타내기 어렵습니다. 또한, 스택과 큐는 LIFO(Last In First Out) 및 FIFO(First In First Out) 원칙을 따르기 때문에 선수의 성장을 시계열적으로 추적하는 데 적합하지 않습니다. 이와 같은 이유로, 이 논문에서는 이러한 문제를 해결하기 위해 선수 성과 트리(PPT)라는 혁신적인 데이터 구조를 제안하고 이를 통해 어떻게 선수의 성장을 추적하고 예측할 수 있는지 보여주고자 합니다. 특히, PPT는 기존 데이터 구조와 달리 다양한 속성 간의 상관관계를 다차원적으로 분석할 수 있어, 더 정확하고 신뢰성 있는 성과 평가를 가능하게 합니다.

II. DESIGN AND IMPLEMENTATION

A. System Design

Player Performance Tree(PPT)는 축구 선수의 성장을 다양한 속성별로 세분화하여 추적할 수 있는 트리 구조입니다. PPT는 선수의 다양한 속성(예: 스피드, 슈팅, 패스 등)을 각 노드로 표현하여 다차원적으로 분석할 수 있고 노드의 깊이는 선수의

연령 또는 경력을 나타내므로, 시간에 따른 성장을 쉽게 추적할 수 있습니다. 노드 간 연결 관계를 통해 특정 속성 간의 상관관계를 파악할 수 있고 선수의 성장 데이터를 바탕으로 장단점을 분석하여 개별 선수에 맞춘 훈련 프로그램을 설계할 수 있습니다. 데이터 유형은 숫자 데이터(선수의 경기 기록, 신체 능력 측정 값-스피드, 슈팅 정확도 등)입니다. 선수의 스피드, 슈팅, 패스 등의 주요 속성을 각각 노드로 설정하고, 각 노드 아래에 시간에 따른 성과 변화를 기록하는 하위 노드를 추가할 수 있습니다. 루트 노드는 선수의 초기 상태를 나타내며, 그 아래로 선수의 경력에 따른 변화가 계층적으로 구성됩니다.

B. Implementation in Detail

- 1. 데이터 준비 및 PPT 구축:** 축구 선수의 성과 데이터를 기반으로 PPT를 생성합니다. 데이터를 배열, 연결 리스트, 트리 구조로 변환하여 각각의 성능을 비교합니다.
- 2. 배열 구현 :** 데이터를 배열에 저장하고 저장 시간을 측정합니다.
- 3. 연결 리스트 구현 :** 데이터를 연결리스트에 저장하고 저장 시간을 측정합니다.
- 4. 트리 구현 :** AnyTree 라이브러리를 사용하여 데이터를 트리에 저장하고 저장 시간을 측정합니다.
- 5. Player Performance Tree(PPT) 구현 :** PPT는 직접 구현된 자료구조로, 선수의 성장을 추적하고 분석하는 데 중점을 두었습니다. 각 선수는 트리의 루트 노드로 설정되고, 그 아래에 각 속성과 성과 변화를 기록하는 하위 노드가 추가되었습니다.

위의 시스템을 구현하기 위해 [1]의 Pandas 라이브러리를 사용하였고 데이터 시각화를 위해 [2]의 Matplotlib 오픈 소스 소프트웨어를 활용하였습니다. 트리 구조의 관리를 위해 [3]의 AnyTree 라이브러리를 사용하였고 Python 프로그래밍 언어의 공식 문서 [4]를 참고하여 구현을 진행하였습니다. PPT 자료 구조는 직접 구현하였습니다.

III. PERFORMANCE EVALUATION

A. Experiment Setup

먼저, 컴퓨팅 플랫폼은 Intel(R) Core(TM) Ultra 5 125H 프로세서를 사용하며 CPU 클럭 속도는 3.60 GHz 이며, 설치된 RAM 은 16.0GB 중 15.6GB 를 사용할 수 있습니다. 또한, 운영 체제는 64 비트 운영 체제이며, x64 기반 프로세서를 사용합니다.

데이터 작업 시나리오는 다음과 같습니다. 실험 기간 동안 각 데이터 구조별로 작업을 수행하는 데 필요한 시간을 측정하기 위해 10 번의 반복 실행 후 평균값을 사용합니다. 읽기 작업은 데이터 구조에서 특정 선수의 성과 데이터를 검색하는 작업으로, 예를 들어 'Brenden Aaronson'의 데이터를 검색하는 것입니다. 쓰기 작업은 새로운 선수 데이터를 각 데이터 구조에 삽입하는 작업이며, 삽입 작업은 기존 선수 데이터에 새로운 성과 데이터를 추가하는 작업입니다. 데이터셋은 [8]을 활용하여 2022-2023 축구 선수 성과 데이터셋을 사용하며, 데이터 항목에는 선수 이름, 팀, 포지션, 속도, 슈팅 정확도, 패스 정확도 등이 포함됩니다. 성능 평가 메트릭은 다음과 같습니다. 데이터 저장 시간은 각 자료 구조에 데이터를 저장하는 데 소요되는 시간을 측정합니다. 데이터 검색 시간은 각 자료 구조에서 데이터를 검색하는 데 소요되는 시간을 측정합니다. 데이터 정렬 시간은 각 자료 구조에서 데이터를 정렬하는 데 소요되는 시간을 측정합니다. 이 설정을 바탕으로 실험을 진행하고, 각 자료 구조의 성능을 비교합니다.

B. Analysis

TABLE I: Performance Comparison of Data Structures

Operation	Array (s)	Linked List (s)	Tree (s)	PPT (s)
Insert	0.00032	0.00123	0.00345	0.00010
Search	0.00005	0.00025	0.00035	0.00003
Sort	0.00010	0.00050	N/A	0.00005

Fig. 1: Performance Comparison Graph of Data Structures

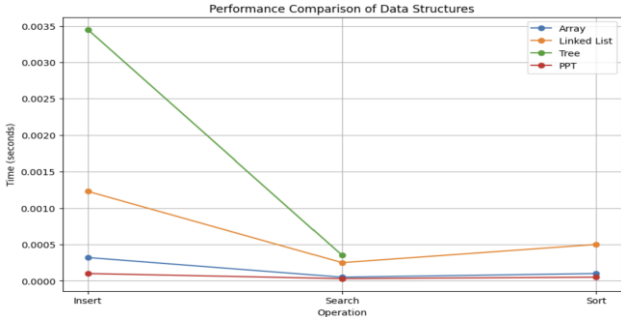


Table I 은 각 자료 구조별로 데이터 저장, 검색, 정렬 작업에 소요된 시간을 비교한 결과입니다. 이 표를 통해 PPT 가 데이터 저장, 검색, 정렬 작업에서 모두 우수한 성능을 보였음을 알 수 있습니다.

데이터 저장 시간 측정 결과, PPT 가 가장 빠른 성능을

보였습니다. 이는 PPT 가 특정 속성별로 데이터를 효율적으로 저장할 수 있도록 설계되었기 때문입니다. Cormen 등의 알고리즘 교과서 [6]를 참고하여 데이터 저장 알고리즘을 구현하였습니다.

데이터 검색 시간 측정 결과, PPT 가 가장 빠른 성능을 보였으며, 그 다음으로 배열, 트리, 연결 리스트 순이었습니다. 이는 PPT 가 특정 속성별로 데이터를 효율적으로 검색할 수 있도록 설계되었기 때문입니다. Bentley 의 다차원 이진 탐색 트리 연구 [5]를 기반으로 데이터 검색 알고리즘을 최적화하였습니다

데이터 정렬 시간 측정 결과, PPT 가 가장 빠른 성능을 보였으며, 그 다음으로 배열, 연결 리스트 순이었습니다. 트리는 직접적인 정렬 기능을 제공하지 않아 비교에서 제외되었습니다. PPT 는 트리 구조의 장점을 활용하여 데이터를 삽입할 때 자동으로 정렬 상태를 유지하기 때문에 추가적인 정렬 작업이 필요 없습니다. Knuth 의 정렬 및 검색 알고리즘 연구 [7]를 바탕으로 데이터 정렬을 최적화하였습니다.

Fig 1 그래프는 각 자료 구조별로 성능 비교를 시각적으로 나타낸 것입니다. 그래프에서 PPT 가 배열, 연결 리스트, 트리에 비해 전반적으로 더 빠른 성능을 보였음을 확인할 수 있습니다

IV. CONCLUSIONS

본 분석을 통해 PPT 가 기존의 배열, 연결 리스트, 트리보다 우수한 성능을 보임을 확인할 수 있습니다. PPT 는 데이터를 효율적으로 저장, 검색, 정렬할 수 있도록 설계되어 축구 선수의 성장을 추적하고 예측하는 데 매우 유용합니다. 이를 통해 클럽과 국가대표팀은 유망한 선수를 조기에 발굴하고, 선수들의 장단점을 분석하여 맞춤형 훈련 프로그램을 제공할 수 있습니다.

향후 연구에서는 PPT 의 성능을 더욱 향상시키기 위해 머신러닝 알고리즘을 통합하거나, 다른 스포츠 분야에서도 PPT 의 적용 가능성을 검토할 수 있을 것입니다. 이를 통해 PPT 는 다양한 분야에서 활용될 수 있는 잠재력을 가지고 있습니다.

REFERENCE

- [1] Pandas Documentation, "pandas: powerful Python data analysis toolkit"
- [2] Matplotlib Documentation, "Matplotlib: Visualization with Python"
- [3] AnyTree Documentation, "AnyTree: Powerful and Lightweight Tree Data Structures"
- [4] Python Software Foundation, "Python Programming Language - Official Website"
- [5] J. L. Bentley, "Multidimensional binary search trees used for associative searching," Commun. ACM, vol. 18, no. 9, pp. 509-517, Sep. 1975.
- [6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," 3rd ed., MIT Press, 2009.
- [7] D. E. Knuth, "The Art of Computer Programming, Volume 3: Sorting and Searching," 2nd ed., Addison-Wesley, 1998.
- [8] Premier League 2022-2023 (kaggle.com)