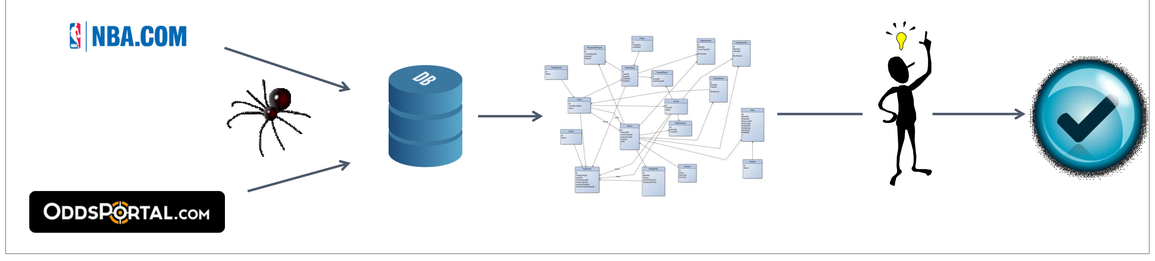
Sports Betting
Trying to Beat the Bookmakers in NBA

Kaido Lepik
University of Tartu
Faculty of Mathematics and Computer Science, Institute of Mathematical Statistics
Curriculum: Mathematical Statistics 2012/13
Public repository: *https://github.com/K-L/NBA*

DESCRIPTIVES

- Over 1 GB of data about more than 15000 NBA matches with odds on more than 5000:
  - box scores and play-by-play data,
  - 10 different bookmakers.
- Four different approaches, three of them not seen in literature.

WORK PROCESS



DESCRIPTIVES

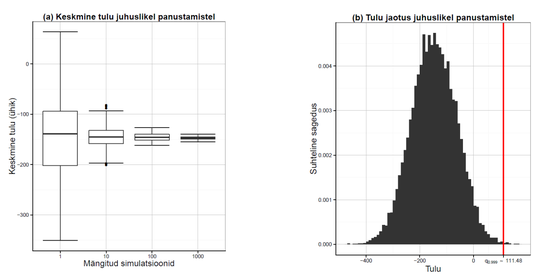
- Many machine learning algorithms and implementations.
- Simulating betting on chronologically ordered real matches with real odds:
  - close to 70% accuracy,
  - statistically significant profitability.

Data has been scraped from NBA.com and OddsPortal.com. It has been cleaned, validated and organized into a relational database. Play-by-play data has been extracted from textual information using regular expressions. At every time moment in an NBA match it is known which players were on the field, what events occurred, who had the possession etc. PS! All the data has only been used for educational purposes.

MOTIVATION

The objective of this project was to try and approach sports betting in professional manner and study how easy it is to find an edge and profit from sports betting.

Bookmakers do not offer fair odds on sports events. If a punter bets randomly, he loses in the long run. However, sports betting is a multi billion dollar business. Hence, it is worth to study whether it can be beaten.



Profit (loss) from random betting

BETTING EXPLAINED

We bet on a team to win a match only if we have found an edge. We have found an edge if according to our classifier, bookmakers have underestimated the probability of the team winning. We simulate betting on chronologically ordered matches which make up our test set. If home and away team's odds and conditional probabilities estimated by the classifier g are k_1, p_1 and k_0, p_0 respectively then we find the profit on a match as follows:

$$T(g,x,y) = \begin{cases} k_1 - 1 & p_1k_1 > 1, y = 1, p_1k_1 \geq p_0k_0 \\ k_0 - 1 & p_0k_0 > 1, y = 0, p_0k_0 > p_1k_1 \\ 0 & p_1k_1 \leq 1, p_0k_0 \leq 1 \\ -1 & otherwise \end{cases}.$$

Where's the profit?

It's very difficult to accurately estimate probabilities of any given team winning a match. Therefore, we wrongly find too much value in betting on outsiders. Hence, in addition to looking for better models, we might benefit from a more clever betting scheme.

RESULTS

Method	Accuracy	Return
Random	50%	-2.7%
Betting on home win	60.2%	-5.2%
Based on last result	**58.6%**	**+3.3%**
Betting on bookmakers' favourites	69.3%	-1.5%
Best logistic regression	68.9%	-3.3%
Best AdaBoost	67.6%	-4.5%
Best modified AdaBoost	**37.1%**	**+3.6%**
Homogeneous Poisson processes	65.1%	-3.9%
Non-homogeneous Poisson processes	64.7%	-3.5%
Profit-maximizing full tree (all features), max height 3	41.8%	-1.8%

APPROACH
logistic regression and AdaBoost

Logistic regression solves

$$\max_{w,w_0} \prod_i p_1^{y_i}(x_i) p_0^{1-y_i}(x_i)$$

where conditional probabilities are derived from the model

$$\ln \frac{p_1(x)}{p_0(x)} = w^T x + w_0.$$

AdaBoost is a combination of weak classifiers

$$g(x) = sign \sum_{m=1}^{M} \alpha_m g^m(x)$$

where

$$g^m = \underset{g \in G}{\operatorname{argmin}} \sum_{i=1}^{n} \beta_i^m I_{g(x_i) \neq y_i}.$$

Mostly used normalized box scores as features with only a few derived features (30 in total). **Feature selection by greedy algorithms and simulated annealing heuristic.**

NOVEL APPROACH
classifying with Poisson processes

Let the players of team S for a given match be s_i. Let the previous number of matches by S which we look at be n. Let the number of seconds played by s_i in match j be $t_j^{s_i}$ during which S scored $\phi_j^{s_i}$ baskets and allowed the opponent to score $\varphi_j^{s_i}$ baskets. Then the intensities of team S scoring and allowing the opponent to score baskets in the match are

$$\mu_S = \frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\phi_j^{s_i}}{\sum_{j=1}^{n}\sum_{i=1}^{m} t_j^{s_i}} \text{ and } \nu_S = \frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\varphi_j^{s_i}}{\sum_{j=1}^{n}\sum_{i=1}^{m} t_j^{s_i}}.$$

Let the teams in a given match be denoted A and B. Then the intensity of team A scoring against team B in quarter k an l-point basket is $\lambda_{AB}^{k,l} = \kappa\mu_A^{k,l} + (1-\kappa)\nu_B^{k,l}$ where $\kappa \in [0,1]$. Assume that these are the intensities of Poisson processes with which to simulate baskets. Winner is the team with more wins or better average score over many simulations.

NOVEL APPROACH
modified AdaBoost algorithm

AdaBoost minimizes

$$\sum_{i=1}^{n} \exp(-y_i(\alpha_1 g^1(x_i) + \cdots + \alpha_M g^M(x_i))).$$

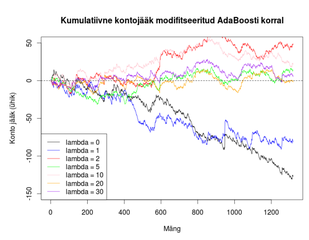
Instead minimize

$$\sum_{i=1}^{n} \exp(\lambda u_i(y_i) - y_i(\alpha_1 g^1(x_i) + \cdots + \alpha_M g^M(x_i)))$$

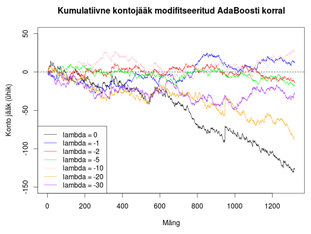
where

$$u_i(y_i) = \begin{cases} k_+(i), & y_i = +1 \\ k_-(i), & y_i = -1 \end{cases}.$$

If $\lambda > 0$ then we focus more on games where the underdog has won; if $\lambda < 0$ then we focus more on games where the favourite has won; if $\lambda = 0$ then we have the original AdaBoost algorithm. In the figures on the right, we can see that models with positive λ-s seem to be more profitable than the original AdaBoost.



a) Profitability of models with positive λ-s



b) Profitability of models with negative λ-s

NOVEL APPROACH
training trees by maximizing profit

Let the class and odds of home and away wins be +1, k_+ and -1, k_-, respectively. Then find a classifier from G such that

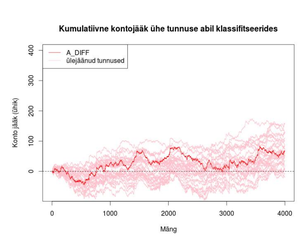
$$g_n = \underset{g \in G}{\operatorname{argmax}} \sum_{i=1}^{n} U_i(y_i, g(x_i))$$

where

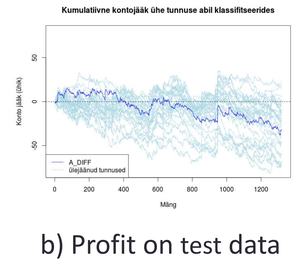
$$U_i(y_i, g(x_i)) = \begin{cases} k_+(i) & y_i = g(x_i) = +1 \\ k_-(i) & y_i = g(x_i) = -1. \\ -1 & y_i \neq g(x_i) \end{cases}$$

In the figures on the right, the profitability of this approach has been tested by training a decision tree with maximum height 2 using just 1 feature at a time. Profit on training data shows that we're doing the right thing but none of the models is profitable on test data.

We can deduce that bookmakers do not make systematic errors which our features can discover.



a) Profit on training data



b) Profit on test data