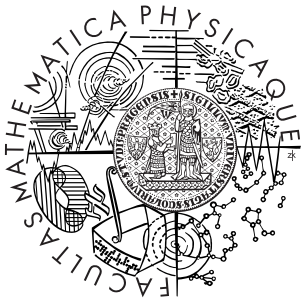
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Roman Zákutný

Použití Markovových rozhodovacích procesů pro modelování kolektivních her

Katedra softwarového inženýrství

Vedoucí diplomové práce: prof. RNDr. Jaromír Antoch, CSc.

Studijní program: Informatika - softwarové systémy

2010

Ďakujem pánovi profesorovi RNDr. Jaromírovi Antochovi, CSc. za pomoc, pripomienky, cenné rady a za odborné vedenie diplomovej práce.

Prehlasujem, že som svoju diplomovú prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa 6.8.2010 Roman Zákutný

Obsah

Název práce: Použití Markovových rozhodovacích procesů pro modelování kolektivních her
Autor: Roman Zákutný
Katedra (ústav): Katedra pravděpodobnosti a matematické statistiky
Vedoucí diplomové práce: prof. RNDr. Jaromír Antoch, CSc.
E-mail vedoucího: antoch@karlin.mff.cuni.cz

V tejto práci je navrhnutý a implementovaný model vychádzajúci z teórie Markovovho procesu so spojitým časom na jednu vybranú kolektívnu hru. Na vstupných dátach je prevedená rozsiahla analýza, na základe ktorej sú odvodené regresné modely pre odhady parametrov. Spustenou simuláciou je preukázaná použiteľnosť modelu a v porovnávacej analýze sú zhodnotené výhody nášho modelu oproti použitiu Markovovho reťazca s diskrétnym časom navrhnutom a implementovanom v mojej bakalárskej práci [1]. Na záver je prevedená diskusia možného rozšírenia na ostatné hry.

Klíčová slova: Markovov proces, simulácie, kolektívne hry, Java, Spring

Title: Use of Markov decision processes for modelling of collective games
Author: Roman Zákutný
Department: Department of Probability and Mathematical Statistics
Supervisor: prof. RNDr. Jaromír Antoch, CSc.
Supervisor's e-mail address: antoch@karlin.mff.cuni.cz

In this thesis, a model based on the continuous-time Markov process is built and implemented and later applied on an one chosen collective game. An extensive analysis of available data is carried out to build a regression model to estimate parameters of the game model. An usableness of the game model is shown by a simulation process. Pros and cons are evaluated in a comparison analysis against the application of the discrete-time Markov chains, how it was described in my bachelor thesis [1]. In conclusion are discussed possible extensions for other collective games.

Keywords: Markov process, simulations, team sports, Java, Spring

Úvod

Už veľmi dlhú dobu existuje mnoho dôvodov pre aplikovanie štatistických modelov v športe. Ich využitie spočíva v zlepšení strategických rozhodnutí, lepšieho rozpoznania taktiky súpera alebo vylepšenia pravidiel za účelom zvýšenia diváckeho záujmu. Niekedy sú modely aplikované z dôvodu testovania spravodlivosti pravidiel alebo štruktúry ligových či pohárových súťaží, inokedy športové udalosti poskytujú veľmi dobré dáta pre aplikovanie nových štatistických teórií. Jedna z najtypickejších požiadaviek na výstup je schopnosť predpovede výsledkov. Táto oblasť je zaujímavá ako z pohľadu fanúšika, tak z pohľadu výskumu a má značný dopad na trh stávok (tzv. spread betting market), pre ktorý sa stala jedným z hlavných výskumných zameraní. Pod pojmom športové udalosti sa skrýva rozsiahly počet hier jednotlivcov, dvojíc a kolektívov.

Motiváciou tejto práce je zamerať sa práve na kolektívne loptové hry a vylepšiť model zavedený v bakalárskej práci [1], ktorú som vypracoval v rámci bakalárskeho štúdia, a ktorá sa zameriava na simuláciu zápasov pomocou Markovových reťazcov s diskrétnym časom. Navrhnutý algoritmus bol aplikovaný na zápasy virtuálnej florbalovej ligy dostupnej v rámci webovej aplikácie, v ktorej sa zaregistrovaní hráči starajú o vlastné týmy a určujú stratégie. V reálnom nasadení sa však časom ukázalo, že dlhodobo výsledky dosahujú priemerné hodnoty, čím reálne rozdiely medzi týmami zanikajú. Príčinou je základný model s malým množstvom empirických dát a chýbajúca analýza o ich vlastnostiach, čo príliš zjednodušuje pohľad na hru ako takú. Súčasná aplikácia sa rozširuje o malý webový modul, ktorý je súčasťou tejto práce. Prehľadným spôsobom poskytuje získané dáta a predstavuje jednoduché rozhranie pre spustenie naimplementovaného simulačného algoritmu. Implementačnou platformou je Java 6.0 a technologický základ tvoria frameworky Hibernate 3.5 a Spring 3.0. Pôvodnú MVC architektúru frameworku Stripes (viď [1]) nahradzuje webový modul Spring MVC.

Práca je štruktúrovaná nasledovne. V 1. kapitole si predstavíme niekoľko známych prístupov, upozorníme na nedostatky a zamyslíme sa

nad možnosťami vylepšenia. Ťažiskom modelu sú reálne dáta, a preto si predstavíme niekoľko základných, ale aj sofistikovanejších prístupov, ako ich získať. V 2. kapitole sa oboznámime s Markovovým procesom so spojitým časom, jeho vlastnosťami, zavedieme si pojmy a tvrdenia nevyhnutné pre praktické použitie a ukážeme si vzťahy s ďalšími štatistickými modelmi. V 3. kapitole navrhneme vhodný model aplikovaním teoretických znalostí na jednu konkrétnu vybranú kolektívnu hru - futbal. Dôvodom je hlavne fakt, že tento šport sa stal fenoménom, a aj preto rôzne internetové portály poskytujú veľmi dobrý základ na získanie vhodných dát. Tie sú totiž v dnešnom svete veľmi cenné a so zvyšujúcimi sa požiadavkami je ich získavanie náročnejšie. Ich hlbšou analýzou sa budeme zaoberať vo 4. kapitole, kde zároveň overíme rôzne hypotézy vychádzajúce z iných štúdií, ale aj z očakávaní na základe skúsenosti. Po ich analýze získame vstupné parametre, ktorými inicializujeme vhodne modifikovaný Markovov proces, tj. simulačný model, ktorého implementáciu popisuje 5. kapitola. Predstavíme si navrhnutú viacvrstvovú architektúru aplikácie a najdôležitejšie vlastnosti použitých technológií. Pre podporu štatistických výpočtov použijeme dostupný štatistický software, ktorý spĺňa nároky kladené na naše výpočty, keďže je zložité toho dosiahnuť bežne dostupnými matematickými knižnicami. V poslednej 6. kapitole dosiahnuté výsledky vyhodnotíme a porovnáme s reálnymi dátami a so simuláciou postavenou na Markovových reťazcoch s diskrétnym časom. Na záver si zhrnieme výhody a nevýhody použitia nášho modelu oproti iným prístupom a prevedieme diskusiu o možnostiach aplikovania modelu na iné kolektívne hry.

Kapitola 1

Simulácie kolektívnych hier

Každý športový zápas sa riadi spravidla náhodným sledom udalostí, pričom náhodnosť určitým spôsobom ovplyvňujú strategické rozhodnutia hráčov v danom momente. Sú situácie, keď hráč nemá dostatok času alebo priestoru na racionálne zváženie ďalšieho kroku. To znamená, že je podstatne výhodnejšie, ak sa pokúsime stratégiu vyjadriť v obecnejšom princípe a nebudeme sa zaoberať detailmi, pri ktorých je veľmi zložité preukázať deterministické správanie. Zmyslom bude vyjadrenie obrannej alebo útočnej stratégie, výhody domáceho prostredia, či konfrontácie súperov. Samozrejme by sme dokázali nájsť veľké množstvo faktorov, ktoré by mohli priebeh hry ovplyvňovať (napr. podpis nového hráča, zranenie kľúčového hráča, počasie, množstvo divákov a pod.). Ich vplyv má však skôr subjektívny charakter a štatisticky je veľmi ťažko preukázateľný, pretože tieto informácie nemáme takmer vôbec k dispozícii. Preto sa v práci budeme zameriavať iba na objektívne merateľné faktory, ktoré dokážeme získať z dostupných dát.

1.1 Aplikované metódy

Ako sme už naznačili v úvode, schopnosť predikcie výsledkov športových udalostí, resp. simulovať ich priebeh, vedie k celkom rozsiahlemu výskumu podporovaného najmä trhom v oblasti stávok. Nie je ťažké dohľadať niekoľko ďalších štúdií zaoberajúcich sa formálnym popisom športových hier. Jedná sa o americký futbal [4], baseball [5], kriket [6] či ľadový hokej [7]. Popri kolektívnych hrách sú veľmi atraktívne analýzy tenisu, squashu či bedmintonu [8]. Jednoduchý Markovov proces s regresnými odhadmi parametrov je naznačený v Hirotsu, Wright [3].

Všetky metódy sú aj napriek svojej rôznorodosti založené na reálnom

pozorovaní, pretože vďaka nemu dokážeme skúmané javy popísať a správne sa rozhodnúť. Navyše, bez skutočných dát nedokážeme správnosť modelu otestovať. Drvivá väčšina štúdií skúma vzťahy medzi matematikou a športom, založené najmä na teórii pravdepodobnosti a matematickej štatistiky. Objavuje sa napríklad Simpsonov paradox [9], ktorý spochybňuje pravidlo, že čím väčšie množstvo dát máme k dispozícii, tým spoľahlivejšie výsledky dosahujeme. Aplikovanie dynamického programovania zasa popisuje Sackrowitz [10].

1.2 Nedostatky existujúcich prístupov

Od tejto chvíle sa budeme aj z dôvodov uvedených v úvode zaoberať konkrétnym športom - futbalom. Často aj kvôli nedostatku reálnych pozorovaní dochádza k predpokladu, že parametre zápasu sú počas celého priebehu konštantné, resp. prvotný predpoklad o rozdelení napozorovaných veličín nie je zamietnutý. Zhromažďovanie dát má niekoľko foriem. Vyberajú sa najpravdepodobnejšie výsledky zápasov na základe ich počtov v sezóne, v bežne dostupných tabuľkách určujúcich poradie tímov v prebiehajúcej sezóne sú k dispozícii počty výhier, remíz, prehier, strelených a obdržaných gólov. Ako si neskôr ukážeme, tieto charakteristiky nemajú dostatočnú informatívnu hodnotu.

Už *Maher* [2] vo svojej analýze prezentoval hru ako homogénny proces, v ktorom sa počet dosiahnutých gólov riadi Poissonovým rozdelením a pre odhad veličín v ďalších zápasoch využil Poissonov regresný model. Táto úvaha však naráža na fakt, že parametre zápasu sa v jeho priebehu s veľkou pravdepodobnosťou menia a konštantný odhad strednej hodnoty počtu gólov v rámci množstva zápasov nemusí byť zďaleka postačujúci. Pokúsime sa overiť túto úvahu, na základe výsledku navrhnúť Markovov proces so spojitým časom spôsobom podobným v [3], kde však taktiež nie je zamietnutá hypotéza, že prihrávky a góly sa riadia rozdelením v súlade s *Maherom*. V našom prípade zanalyzujeme rozdelenie parametrov a získané odhady sa pokúsime dynamicky meniť v závislosti na prebiehajúcom čase, aktuálnom skóre a taktike.

Cieľom práce ja taktiež zdokonaliť model popísaný v bakalárkej práci [1], postavený na Markovových reťazcoch s diskrétnym časom. Model bol navrhnutý pre florbal, avšak má v sebe základné vlastnosti aplikovateľné aj na futbal. Najväčším nedostatkom je absencia spojitého času a málo napozorovaných reálnych dát, čo spôsobuje, že parametrizovanie modelu

niekedy nezodpovedá realite a výsledky sú príliš rovnomerne rozložené a nezohľadňujú silu jednotlivých tímov.

1.3 Dolovanie dát

Reálne dáta sú dôležitým stavebným prvkom pri štatistickom modelovaní. Našou úlohou bude vyextrahovanie hodnotných informácií z veľkých objemov dát, ktoré sa použijú pri tvorbe efektívnych rozhodnutí. Zvyčajne je ľahko dostupný výsledok zápasu, prípadne čas a strelec gólu, pozícia tímu v tabuľke a jeho dosiahnuté ligové skóre. Z každého zápasu však dokážeme získať ešte bohaté množstvo ďalších informácií. Cesta k nim je však často zložitá alebo časovo náročná. Aj preto už existuje niekoľko firiem, ktorých biznis je postavený práve na zhromažďovaní a predávaní dát (napr. Opta Index Ltd. alebo Panini Digital). Takéto dáta sú však dnes na trhu veľmi cenné a drahé. Preto si načrtneme alternatívy.

Na začiatku je dobré si premyslieť, aké dáta vôbec potrebujeme. Zápas je reťazec veľkého množstva akcií (gól, strela, prihrávka, roh, aut, faul, priamy kop, hlavička, striedanie, karta, atď.). Každá akcia je charakterizovaná časom (kedy), pozíciou (kde) a exekútorom (kto). V základných štatistikách zväčša nájdeme góly, karty a striedania spolu s časom a hráčom. Ostatné akcie sú vďaka svojej početnosti zložitejšie zaznamenávané a teda ťažko dohľadateľné. Mnoho akcií dokážeme zredukovať na pár základných, ale postačujúcich. Sú to:

- *gól* - úspešná strela
- *prihrávka* - vhadzovanie, roh, priamy kop
- *striedanie* - zmena taktiky
- *karta* - zmena taktiky; červená = vylúčenie hráča z hry, žltá = zvýšené riziko vylúčenia (dve žlté karty znamenajú automaticky červenú)

Pozičné parametre akcie nám dávajú informáciu o pohybe lopty na hracom poli. Aby sme boli na základe empirických dát schopní dynamicky modelovať rozhodnutia všetkých hráčov v určitej situácii, museli by sme získať pozície všetkých hráčov v tejto situácii. Tieto pozície však k dispozícii bežne nie sú a preto je typickejšie popisovať hru jedine z pozície lopty. K tomuto popisu hry sa prikloníme aj my.

Ukážeme si 3 základné postupy pre získanie dát:

1) Pozorovanie

Pozorovaním reálnych zápasov (opakovane) dokážeme získať dáta šité na mieru. Keď však uvážime, že jeden zápas trvá 90 až 100 minút, tak sa jedná o časovo najnáročnejší prístup. Týmto spôsobom boli dáta zhromažďované v práci [1], čo malo za následok analýzy veľmi malej vzorky oproti celkovému počtu odohraných zápasov.

2) Parsovanie internetových zdrojov

Parsovanie verejne dostupných HTML/XML štruktúr je najrozšírenejšou metódou. Existuje mnoho verejne dostupných zdrojov, avšak kvalita dát je častokrát nedostatočná. Pre vytvorenie parsovacieho skriptu na mieru je potrebné detailne zanalyzovať štruktúru prezentácie dát, čo je zväčša HTML štruktúra v zlom formáte, v lepších prípadoch XML.

3) Segmentácia a analýza videa

Parsovanie videosekvencií je tou najsofistikovanejšou metódou. Jedná sa o všeobecný problém analýzy štruktúry videa a pochopenie jeho obsahu. Skúmaním športového obsahu vo videosekvencii sa už zaoberalo niekoľko štúdií. Medzi inými sú zaujímavé videoanalýzy striel [13] a klasifikácia sémantických herných stavov vo futbale [12].

Hlavnou myšlienkou je automatizovať segmentáciu videosekvencie do dvoch základných rekurentných sémantických herných stavov: hra a prerušenie. Z nášho pohľadu je zaujímavá najmä hra. Ako popisuje Xie a ďalší [11] a Tovinkere [12], pozorovaním môžeme dôjsť k záveru, že veľmi efektívnym nástrojom pre detekciu hry a prerušenia, je pochopenie spôsobu, akým producent video prenosu sníma hru.

Existujú totiž tri základné typy záberov v rámci prenosov: *globálny* (veľká časť ihriska), *lokálny* (na situáciu) a *detailný* (napr. na hráča). Rozlišovanie týchto záberov nám môže pomôcť v ďalšej analýze. Po prvé, v rámci hry evidentne prevládajú globálne zábery, pretože dodávajú divákovi najlepšiu informáciu o dianí, zatiaľ čo počas prerušenia prevládajú naopak lokálne a detailné zábery. A po druhé, typy záberov môžu byť identifikované na základe vlastností, akými sú pomer farieb (napr. v pixeloch) a intenzita pohybu. V globálnych záberoch prevládajú odtiene zelenej farby, v stredných je pomer menší a v detailných zelená farba takmer zaniká. Podobne je to

s detekciou pohybu, keď v stave hry očakávame jeho vyššiu intenzitu ako počas prerušenia.

Intenzita pohybu a pomer farieb sú dobrými merateľnými vlastnosťami obrazu. Analógiu rozpoznávania obrazu môžeme vidieť v rozpoznávaní izolovaného slova v hovorenej reči, ako to popisuje Rabiner [15]. Každý jeho model zodpovedá nejakej triede - foném hovorenej reči tak, ako hra a prerušenie vo videosekvencii. V rámci každého modelu existujú štruktúry, ktoré v reči popisujú prechody a zmeny v rámci foném a medzi nimi. V prípade videa sú to zmeny intenzity pohybu a prechody medzi rôznymi typmi záberov. Táto analógia nás môže viesť k myšlienke aplikovania matematického modelu známeho ako skryté Markovove modely (HMM).

HMM sa radia k metódam štatistického rozpoznávania a sú vhodné pre klasifikáciu sekvenčných dát s variabilnou dĺžkou. Markovov model, ako si neskôr ukážeme, sa dá reprezentovať ako stavový automat, pre ktorý platí, že jeho nasledujúci stav závisí iba na aktuálnom stave. Skrytý Markovov model sa dá charakterizovať ako jeden optimálny prechod Markovovým modelom pre dané vstupné dáta. Pre jeho nájdenie je nutné vyhodnotiť všetky alternatívy prechodu modelom.

Vzhľadom na komplexnosť a zložitosť tohto prístupu sa ním nebudeme ďalej zaoberať, ale v prípade záujmu odkážeme čitateľa na zmienené citované zdroje. Cieľom týchto odstavcov bolo poukázanie na HMM ako na jednu z možností pre získanie vhodných dát z videosekvencií.

Získané dáta

Dáta analyzované v rámci tejto práce sú získané z verejne dostupného portálu spoločnosti Guardian[1] postupom číslo 2 a uložené v databáze podľa schémy na Obr. A.4 (v prílohe). Jedná sa o flash aplikáciu, ktorá obsahuje dáta anglickej ligy z uplynulých 4 sezón 2006/07 až 2009/10. V lige hrá 20 týmov, čo znamená 380 zápasov za sezónu, resp. 1520 zápasov za 4 sezóny. 2 zápasy nebolo možné pre zlý formát načítaných XML štruktúr spracovať. Preto máme k dispozícii štatistiky z 1518 zápasov, čo je veľmi dobrá vzorka pre prijímanie či zamietanie hypotéz. V každej situácii je dostupný čas a pozícia lopty na ihrisku, nie však rozostavenie ostatných hráčov. Podľa navrhnutého zredukovania akcií tak dostávame pre oba týmy všetky prihrávky, strely, striedania a karty. Prihrávky a strely majú príznak úspechu definovaný podľa Tab. 1.1. Neúspešnú strelu chápeme dvojakým spôsobom, pretože odrazom

[1]Spoločnosť Guardian súhlasí s využitím získaných dát pre akademické účely, ale podmieňuje si to zverejnením iba ich vzorky. [5.6.2010]

sa lopta môže dostať naspäť k hráčovi rovnakého tímu (udržaná lopta), k súperovi (stratená lopta) alebo jej zakopnutím je strata automatická. Gól znamená zvýšenie skóre a stratu lopty. Preto strely a prihrávky môžeme zovšeobecniť ako zmeny kontroly nad loptou.

	Strela	Prihrávka
Úspešná	gól	udržaná lopta
Neúspešná	stratená alebo udržaná lopta	stratená lopta

Tabuľka 1.1: Úspešná vs. neúspešná strela a prihrávka.

Kapitola 2

Markovove procesy

V tejto kapitole sa zoznámime zo základnými pojmami z teórie náhodných procesov na základe [14].

Definícia 1. Nech (Ω, A, P) je pravdepodobnostný priestor, nech $T \subset \mathbb{R}$. Rodina reálnych náhodných veličín $\{X_t, t \in T\}$ definovaných na (Ω, A, P) sa nazýva *náhodný proces*. V prípade, že $T = \mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ alebo $T = \mathbb{N}_0 = \{0, 1, 2, \ldots\}$, hovoríme o *procese s diskrétnym časom*. Pokiaľ $T = [a, b]$, kde $-\infty \leq a < b \leq \infty$, hovoríme, že $\{X_t, t \in T\}$ je *proces so spojitým časom*. Dvojica (S, ε), kde S je množina hodnôt náhodných veličín X_t a ε je $\sigma-$algebra podmnožín S, sa nazýva *stavový priestor* procesu $\{X_t, t \in T\}$. Pokiaľ náhodné veličiny X_t nadobúdajú iba diskrétne hodnoty, hovoríme, že ide o *proces s diskrétnymi stavmi*, ak nadobúdajú hodnoty z nejakého intervalu, hovoríme o *procese so spojitými stavmi*.

Niekoľkokrát sa stretneme s pojmom exponenciálne rozdelenie, ktoré sa používa pre vyjadrenie čakania na určitú udalosť.

Definícia 2. Exponenciálne rozdelenie s parametrom $\lambda > 0$, s označením napr. $Exp(\lambda)$, je spojité rozdelenie na množine kladných čísel s distribučnou funkciou

$$F(X) = \begin{cases} 1 - e^{-\lambda x}, & x > 0 \\ 0 & inak, \end{cases}$$

tj. s hustotou

$$f(X) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0 & inak. \end{cases}$$

Náhodná veličina s exponenciálnym rozdelelním, $X \sim Exp(\lambda)$, má strednú

hodnotu a rozptyl

$$E(X) = \frac{1}{\lambda}, Var(X) = \frac{1}{\lambda^2}.$$

Poznámka. Exponenciálne rozdelenie je tzv. rozdelenie bez pamäti. Platí, že $P(X > x_1 + x_2 | X > x_1) = P(X > x_2)$. Tzn., že má konštantnú intenzitu výskytu pozorovaného javu, $\lambda(x) \equiv \lambda, x > 0$.

Niekedy sa parameter exponenciálneho rozdelenia neudáva ako intenzita pozorovaného javu, ale jej prevrátená hodnota, tj. stredná hodnota rozdelenia $\frac{1}{\lambda}$.

V práci sa budeme ďalej zaoberať iba procesmi so spojitým časom a diskrétnymi stavmi.

2.1 Markovova vlastnosť

V teórii pravdepodobnosti a matematickej štatistiky má náhodný proces Markovovu vlastnosť, ak je pravdepodobnosť prechodu do budúceho stavu podmienene závislá iba na aktuálnom stave. To znamená, že pravdepodobnosť výsledku v budúcom čase, ak poznáme výsledok v prítomnom čase a výsledky z minulých časov, je rovnaká, ako keď poznáme iba výsledok v prítomnom čase.

Definícia 3. Náhodný proces $\{X_t, t \in T\}$ má Markovovu vlastnosť, ak pre všetky $i, j, i_1, \ldots, i_n$ a pre všetky $0 \leq t_1 < \ldots < t_n < s < t$, pre ktoré $P(X_s = i, X_{t_n} = i_n, \ldots, X_{t_1} = i_1) > 0$, platí

$$P(X_t = j | X_s = i, X_{t_n} = i_n, \ldots, X_{t_1} = i_1) = P(X_t = j | X_s = i). \qquad (2.1)$$

2.2 Markovov proces so spojitým časom

Definícia 4. Systém celočíselných náhodných veličín $\{X_t, t \in T\}$ definovaných na pravdepodobnostnom priestore (Ω, A, P) sa nazýva Markovov proces so spojitým časom a spočetnou množinou stavov, ak má Markovovu vlastnosť (2.1).

Markovov proces si môžeme predstaviť ako orientovaný graf, ktorého vrcholy reprezentujú množinu stavov a hrany prechody medzi stavmi. Rozdiel oproti procesu s diskrétnym časom spočíva v tom, že prechodu do nového stavu predchádza zotrvanie v aktuálnom stave nejakú náhodnú dobu. Táto doba má exponenciálne rozdelenie.

Označme $P(X_t = j|X_s = i)$ ako $p_{ij}(s,t)$, čo nazývame pravdepodobnosti prechodu zo stavu i v čase s do stavu j v čase t. Podobne budeme $p_j(t) = P(X_t = j), j \in S$ nazývať absolútne pravdepodobnosti v čase t a $p_j = p_j(0) = P(X_0 = j), j \in S$ nazývame počiatočné pravdepodobnosti. Je zrejmé, že $p_j(t) \geq 0$ pre $\forall j \in S$ a $\sum_{j \in S} p_j(t) = 1$, pre pevné $t \geq 0$.

Definícia 5. Markovov proces so spojitým časom je homogénny, ak pre jeho pravdepodobnosti prechodu platí

$$p_{ij}(s, s+t) = p_{ij}(t), s \geq 0, t \geq 0.$$

Poznámka. To znamená, že $p_{ij}(t)$ nezávisí na t. V opačnom prípade hovoríme o nehomogénnych Markovových procesoch so spojitým časom. To znamená, že $p_{ij}(t)$ závisí na t, resp. s časom sa mení.

Ďalej sa budeme zaoberať diferencovateľnosťou pravdepodobností prechodu. Nech pravdepodobnosti $p_{ij}(t)$ sú spojité pre $t > 0$ a nech $\lim_{t \to 0_+} p_{ii}(t) = 1$ a $\lim_{t \to 0_+} p_{ij}(t) = 0$ pre $i \neq j$. Definujeme $p_{ii}(0) = 1$ a $p_{ij}(0) = 0$ pre $i \neq j$.

Veta 1. *Pre $\forall i \in S$ existuje v 0 derivácia*

$$-p'_{ii}(0) = \lim_{h \to 0_+} \frac{1 - p_{ii}(h)}{h} = -q_{ii} = q_i \leq \infty$$

pre $\forall i, j \in S, i \neq j$ existuje v 0 derivácia

$$p'_{ij}(0) = \lim_{h \to 0_+} \frac{p_{ij}(h)}{h} = q_{ij} < \infty$$

a pre $\forall i \in S$ platí

$$\sum_{j \neq i} q_{ij} \leq q_i.$$

Dôkaz. Dôkaz vety sa dá nájsť v [14]. □

Definícia 6. Nech $p_{ij}(t)$, resp. $p_{ij}(t+h)$, je pravdepodobnosť prechodu zo stavu i do stavu j v čase t, resp. $t+h$. Potom definujeme deriváciu pravdepodobnosti prechodu

$$p'_{ij}(t) = \lim_{h \to 0_+} \frac{p_{ij}(t+h) - p_{ij}(t)}{h}.$$

Ďalej budeme uvažovať len také reťazce, pre ktoré

$$q_i = \sum_{j \neq i} q_{ij} \text{ pre } \forall i \in S. \tag{2.2}$$

Poznámka. Ak je množina stavov S konečná, vzťah (2.2) vždy platí, pretože

$$0 = 1 - \sum_{j \in S} p_{ij}(h), h \geq 0$$

a odtiaľ limitným prechodom

$$0 = \lim_{h \to 0_+} \frac{1 - \sum_j p_{ij}(h)}{h} = \lim_{h \to 0_+} \frac{1 - p_{ii}(h) - \sum_{j \neq i} p_{ij}(h)}{h} = q_i - \sum_{j \neq i} q_{ij},$$

pretože môžeme zameniť limitu a sumu.

Definícia 7. Nezáporné čísla q_{ij} definované vo Vete 1 sa nazývajú *intenzity prechodov* zo stavu i do stavu j, nezáporné číslo q_i sa nazýva *celková intenzita*. Matica $Q = \{q_{ij},\, i, j \in S\}$, kde $q_{ii} = -q_i$ sa nazýva *matica intenzít* prechodu.

Teraz ukážeme význam intenzít prechodu.

Veta 2. *Pre homogénny Markovov proces $\{X_t, t \in T\}$ so spočetnou množinou stavov platí pre $\forall s \geq 0$ a $\forall h > 0$*

$$P(X_t = i,\, s \leq t \leq s + h | X_s = i) = e^{-q_i h},$$

kde $e^{-q_i h} = 0$, ak $q_i = \infty$.

Dôkaz. Dôkaz vety sa dá nájsť v [14]. □

Veta 3. *Ak je $q_i = 0$, potom $p_{ii}(t) = 1$ pre $\forall t \geq 0$. Ak je $0 < q_i < \infty$, má doba, počas ktorej proces zotrvá v stave i, exponenciálne rozdelenie so strednou hodnotou $\frac{1}{q_i}$.*

Dôkaz. Dôkaz vety sa dá nájsť v [14]. □

Proces je teda charakterizovaný intenzitami prechodov q_{ij} medzi stavmi i a j za nejakú pevne stanovenú časovú jednotku. Nech $X(t)$ popisuje stav procesu v čase t a predpokladajme, že proces je v tomto čase v stave i. Hodnota q_{ij} určuje, ako často prechod medzi stavmi i a j za túto pevne stanovenú

časovú jednotku nastáva. Potom pravdepodobnosť, že sa v priebehu krátkeho časového kvanta dostane proces do stavu j, je daná ako

$$P(X_{t+h} = j | X_t = i) = q_{ij}h + o(h),\ i \neq j,$$

kde symbol $o(h)$ značí, že $\frac{o(h)}{h} \to 0$ pri $h \to 0_+$.

Pre lepšie vysvetlenie intenzity q_{ij} uvažujme hru, ktorá má dĺžku t minút, stav i predstavuje držanie lopty a j dosiahnutie gólu. Ak v celej hre prejde lopta zo stavu i do stavu j spolu n-krát, tak intenzita skórovania zo stavu i za jednu minútu je $q_{ij} = \frac{n}{t}$, tj. stredná doba zotrvania v stave i je $\frac{t}{n}$.

Poznámka. V literatúre sa niekedy namiesto Markovovho procesu vyskytuje pojem Markovov reťazec, chápaný ako proces s diskrétnym stavovým priestorom. Markovov reťazec je taktiež zvyčajne označovaný ako Markovov proces s diskrétnym časom a naopak, Markovov proces ako Markovov reťazec so spojitým časom.

2.3 Kolmogorovove diferenciálne rovnice a ich riešenie

Uvažujme Markovov proces so spojitým časom a množinou stavov $S = \{0, 1, \ldots\}$. V predchádzajúcej kapitole sme si zadefinovali intenzity prechodov pomocou derivácií pravdepodobností prechodov v bode 0. Teraz si ukážeme súvislosť medzi týmito intenzitami a deriváciami pravdepodobností v obecnom bode.

Veta 4. (Kolmogorove diferenciálne rovnice) *Predpokladajme, že $q_i < \infty$ pre $\forall i \in S$ a platí (2.2). Potom pravdepodobnosti prechodu $p_{ij}(t)$ sú diferencovateľné pre $\forall i, j \in S, t > 0$ a platí*

$$p'_{ij}(t) = -q_i p_{ij}(t) + \sum_{k \neq i} q_{ik} p_{kj}(t) = \sum_{k \in S} q_{ik} p_{kj}(t). \tag{2.3}$$

Maticovo je možné sústavu rovníc zapísať ako

$$P'(t) = QP(t).$$

Dôkaz. Dôkaz vety sa dá nájsť v [14]. □

Pravdepodobnosti prechodu Markovovho procesu so spojitým časom teda vyhovujú sústavám Kolmogorových rovníc. Naopak teraz predpokladajme,

že je daná sústava diferenciálnych rovníc typu 2.3. Nás zaujíma, či má takáto sústava riešenie, ktoré reprezentuje pravdepodobnosti prechodu nejakého Markovovho procesu.

Veta 5. *Nech* $Q = \{q_{ij},\, 0 \leq i, j \leq N\}$ *je matica, pre ktorej prvky platí*

$$\begin{aligned} q_{ij} &\geq 0,\, i \neq j, \\ q_{ii} &= -\sum_{i \neq j} q_{ij}. \end{aligned}$$

Potom existuje jediné riešenie sústavy (2.3)*, ktoré vyhovuje počiatočnej podmienke* $P(0) = I$ *a ktoré predstavuje sústavu pravdepodobností prechodu Markovovho procesu so spojitým časom a konečnou množinou stavov. Maticovo je možné toto riešenie zapísať v tvare* $P(t) = e^{Qt}$*, kde* e^{Qt} *je maticová exponenciálna funkcia definovaná predpisom*

$$e^{Qt} = \sum_{k=0}^{\infty} \frac{Q^k t^k}{k!}. \tag{2.4}$$

Poznámka. Zápis Q^k predstavuje k-tú mocninu matice Q a výsledkom je matica rovnakého rádu. Napr. pre $k = 2$ je $Q^2 = Q \times Q$. Zápis e^{Qt} predstavuje maticovú exponenciálnu funkciu, ktorej výstupom je taktiež matica rovnakého rádu.

Dôkaz. Dôkaz vety sa dá nájsť v [14]. □

2.4 Poissonov proces

Zaveďme si najprv pojem Poissonovho rozdelenia. Nech X nadobúda iba hodnoty $0, 1, \ldots$, a to s pravdepodobnosťou

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, \ldots,$$

kde $\lambda > 0$ je dané číslo. Potom hovoríme, že X má Poissonove rozdelenie s parametrom λ. Stredná hodnota aj rozptyl sú rovnaké a zároveň rovné tomuto parametru.

Uvažujme proces $\{X_t, t \geq 0\}$ celočíselných náhodných veličín, ktorý má nezávislé prírastky $X_{t+h} - X_t$ (počet udalostí v intervale $(t, t+h]$) a pre ktorý

platí

$$\begin{aligned} P(X_{t+h} - X_t = 1) &= \lambda h + o(h) \\ P(X_{t+h} - X_t = 0) &= 1 - \lambda h + o(h) \\ P(X_{t+h} - X_t \geq 2) &= o(h) \end{aligned}$$

rovnomerne pre všetky t.

Ide o Markovov proces so spojitým časom a množinou stavov $S = \{0, 1, \ldots\}$, s počiatočným rozdelením $p_0(0) = P(X_0 = 0) = 1$, $p_{j\neq 0}(0) = 0$ a intenzitami prechodov $q_{i,i+1} = \lambda$, $q_i = -q_{ii}$, $q_{ij} = 0$ inak. Pravdepodobnosti prechodov môžeme určiť zo sústavy Kolmogorových rovníc (2.3). Ak sa omedzíme iba na určenie absolútnych pravdepodobností $p_j(t)$, $j \in S$ s počiatočným rozdelením $p_i = p_i(0) = 1$, $p_j = p_{j\neq i}(0) = 0$, stačí riešiť sústavu diferenciálnych rovníc, ktoré odpovedajú $i-$temu riadku matice $P(t)$. Tzn. máme riešiť sústavu

$$p'_j(t) = -p_j(t)q_j + \sum_{k\neq j} p_k(t)q_{kj},\ j \in S$$

s počiatočnou podmienkou $p_i(0) = 1$, $p_{j\neq i}(0) = 0$, čo je v našom prípade sústava

$$\begin{aligned} p'_0(t) &= -\lambda p_0(t) \\ p'_j(t) &= -\lambda p_j(t) + \lambda p_{j-1}(t),\ 1 \leq j < \infty \end{aligned}$$

s počiatočnou podmienkou $p_0(0) = 1$, $p_{j>0}(0) = 0$. Táto sústava je sústavou obyčajných lineárnych diferenciálnych rovníc a namiesto riešenia jednej po druhej, využijeme metódu vytvárajúcej funkcie.

Uvažujme vytvárajúcu funkciu rozdelenia $\{p_j(t), j \in \mathbb{N}_0\}$,

$$\Pi(s,t) = \sum_{j=0}^{\infty} p_j(t)s^j$$

ako funkciu dvoch premenných s, t. Ak vynásobíme $j-$tú rovnicu sústavy výrazom s^j a potom formálne sčítame všetky takto vynásobené rovnice,

dostaneme vzťah

$$\begin{aligned}\sum_{j=0}^{\infty} p_j'(t)s^j &= -\lambda \sum_{j=0}^{\infty} p_j(t)s^j + \lambda \sum_{j=1}^{\infty} p_{j-1}(t)s^j \\ &\quad -\lambda \sum_{j=0}^{\infty} p_j(t)s^j + \lambda s \sum_{j=0}^{\infty} p_j(t)s^j.\end{aligned}$$

Tento výraz môžeme vyjadriť pomocou vytvárajúcej funkcie Π ako

$$\frac{\partial \Pi(s,t)}{\partial t} = -\lambda \Pi(s,t) + \lambda s \Pi(s,t) = -\lambda(1-s)\Pi(s,t) \tag{2.5}$$

a počiatočnú podmienku prepísať ako $\Pi(s,0) = 1$. Pre pevné s je možné (2.5) riešiť ako obyčajnú lineárnu diferenciálnu rovnicu v premennej t. Všeobecné riešenie tejto rovnice je

$$\Pi(s,t) = C(s)e^{-\lambda t(1-s)},$$

kde $C(s)$ je konštanta. Z počiatočnej podmienky plynie $C(s) = 1$ a teda

$$\Pi(s,t) = e^{-\lambda t + \lambda s t} = e^{-\lambda t} \sum_{j=0}^{\infty} \frac{(\lambda t)^j}{j!} s^j.$$

Tým dostávame hľadané absolútne pravdepodobnosti

$$p_j(t) = \frac{e^{-\lambda t}(\lambda t)^j}{j!},\ 0 \leq j < \infty,\ t > 0. \tag{2.6}$$

Ide o Poissonovo rozdelenie s parametrom λt. Odtiaľ tiež plynie, že stredný počet udalostí, ktoré nastanú v intervale $(0,t]$, je λt. Keďže počet udalostí v ľubovoľnom intervale $(s, s+t]$ závisí iba na dĺžke tohoto intervalu, odtiaľ plynie, že tiež prírastky $X_{s+t} - X_s$ majú Poissonovo rozdelenie s parametrom λt pre všetky $s, t > 0$.

Nech $X_1, \ldots X_n$ je náhodný výber (nezávislé rovnako rozdelené náhodné veličiny) z Poissonovho rozdelenia s parametrom λ a $k_1, \ldots, k_n$ je jeho realizácia. Tento parameter môžeme odhadnúť napr. pomocou metódy maximálnej vierohodnosti. Chceme zvoliť parameter λ tak, aby pravdepodobnosť $P(X_1 = k_1, \ldots, X_n = k_n|\lambda) = \prod_{i=1}^{n} P(X_i = k_i|\lambda)$ bola čo najväčšia. Ak vezmeme

logaritmickú funkciu

$$L(\lambda) = \ln \prod_{i=1}^{n} f(k_i|\lambda) = \sum_{i=1}^{n} \ln \left(\frac{e^{-\lambda} \lambda^{k_i}}{k_i!} \right),$$

položením $\frac{\partial}{\partial \lambda} L(\lambda) = 0$ dostaneme maximálny vierohodnostný odhad (MLE - *maximum likelihood estimation*)

$$\hat{\lambda}_{MLE} = \frac{1}{n} \sum_{i=1}^{n} k_i. \tag{2.7}$$

Rozlišujeme

- *homogénny Poissonov proces*, ak sa parameter λ v čase t nemení, tzn. je konštantný. Príkladom takéhoto procesu je napr. model zrodu a zániku bez zániku (v literatúre pure birth process).

- *nehomogénny Poissonov proces*, v ktorom je parameter λ už na čase t závislý, ale je to stále deterministická funkcia

- *zmiešaný Poissonov proces*, kde je parameter λ náhodná veličina z nejakého rozdelenia. Rozdelením tejto veličiny (napr. s distribučnou funkciou $G(\lambda)$) vlastne miešame rôzne homogénne Poissonove procesy. Pre každé $t > 0$ a celé $k \geq 0$ je

  $$P(N(t) = k) = \int_0^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} dG(\lambda). \tag{2.8}$$

  Toto odpovedá aj Bayesovskému pohľadu na situáciu s neznámym parametrom λ a jeho priorom $G(\lambda)$.

Z nášho pohľadu bude zaujímavejší práve Poissonov proces s meniacim sa parametrom λ ako varianta Markovovho procesu so spojitým časom.

2.4.1 Vzťah Poissonovho a negatívne binomického rozdelenia

Vykonávajme alternatívne pokusy končiace úspechom alebo neúspechom, pričom v každom z nich nastane úspech s pravdepodobnosťou p. Potom negatívne binomické rozdelenie $\mathrm{NBi}(r, p)$ má dve nasledujúce interpretácie:

1. X je náhodná veličina popisujúca počet neúspechov pred r-tým úspechom.

$$P(X = k) = \left(\begin{array}{c} r+k-1 \\ k \end{array} \right) p^r (1-p)^k,\ 0 < p < 1, r > 0$$

2. Negatívne binomické rozdelenie si tiež môžeme predstaviť ako súčet r nezávislých náhodných veličín, z ktorých každá má geometrické rozdelenie s rovnakým parametrom p.

Strednú hodnotu a rozptyl spočítame ako

$$\begin{aligned} E(X \sim \mathrm{NBi}(r,p)) &= r\frac{1-p}{p} \\ Var(X \sim \mathrm{NBi}(r,p)) &= r\frac{1-p}{p^2} = \frac{E(X)}{p}. \end{aligned}$$

Parametre p a r je možné odhadnúť pomocou momentovej metódy a budeme potrebovať odhad pre strednú hodnotu $\bar{x}$ a rozptyl s^2. Vychádzajúc z rovníc $\bar{x} = r\frac{1-\hat{p}}{\hat{p}}$ a $s^2 = \hat{r}\frac{1-\hat{p}}{\hat{p}^2}$ pre strednú hodnotu a rozptyl negatívne binomického rozdelenia, dostávame odhad:

- pre parameter $\hat{p}$

$$s^2 = \frac{\bar{x}}{\hat{p}} \Longrightarrow \hat{\mathbf{p}} = \frac{\bar{\mathbf{x}}}{\mathbf{s^2}} \tag{2.9}$$

- a pre parameter $\hat{r}$

$$\bar{x} = \hat{r}\frac{1-\frac{\bar{x}}{s^2}}{\frac{\bar{x}}{s^2}} = \hat{r}\frac{s^2-\bar{x}}{\bar{x}} \Longrightarrow \hat{\mathbf{r}} = \frac{(\bar{\mathbf{x}})^{\mathbf{2}}}{\mathbf{s^2}-\bar{\mathbf{x}}}. \tag{2.10}$$

Overdispersion v Poissonovom rozdelení

Pri Poissonovom rozdelení sme limitovaní rovnosťou strednej hodnoty a rozptylu. Ak však dáta preukážu oveľa väčší rozptyl ako strednú hodnotu, hovoríme o tzv. „overdispersion".

Nech X je náhodný výber so strednou hodnotou λ a rozptylom σ^2. Potom z rovnice (2.10) jednoduchou úpravou dostávame

$$\sigma^2 = \lambda + \frac{1}{r}\lambda^2, \tag{2.11}$$

čo znamená, že rozptyl je stále väčší ako stredná hodnota. Môžeme ihneď vidiet, že ak $r \to \infty$, potom stredná hodnota sa limitne rovná rozptylu,

čo predstavuje Poissonovo rozdelenie s parametrom λ. Na negatívne binomické rozdelenie tak môžeme nahliadať ako na rozšírenie Poissonovho rozdelenia, ktoré dovoľuje väčší rozptyl.

To nás vedie k záveru, že negatívne binomické rozdelenie je vhodnou alternatívou Poissonovho rozdelenia v prípade, že naše pozorovania preukazujú „overdispersion" efekt.

Poisson-Gamma rozdelenie

V predchádzajúcej sekcii sme videli, že negatívne binomické rozdelenie môže byť chápané ako rozšírenie Poissonovho rozdelenia. Teraz si ukážeme, ako je možné získať negatívne binomické rozdelenie zo zmiešaného Poissonovho rozdelenia s náhodným parametrom λ, ktorý pochádza z gamma rozdelenia. V literatúre môžeme nájsť taktiež termín Poisson-Gamma rozdelenie.

Rozoberme si voľbu gamma rozdelenia náhodného parametra λ. Toto rozdelenie je spojité nezáporné rozdelenie na $[0, +\infty)$, ktoré je zošikmené doprava. Preto je ideálny kandidát na modelovanie parametra λ. Parameter λ musí byť totiž nezáporný a nemal by nadobúdať extrémnych hodnôt. Gamma rozdelenie má vhodný tvar, lebo veľké hodnoty, ktoré nie sú žiadúce, nadobúda s veľmi malou pravdepodobnosťou.

Nech má náhodná veličina X pri pevnom λ Poissonovo rozdelenie, tj.

$$P(X = k|\lambda) = \frac{\lambda^k}{k!}e^{-\lambda},\ k = 0, 1, \ldots,\ \lambda > 0.$$

Gamma rozdelenie má hustotu

$$f(\lambda, \alpha, r) = \frac{\alpha^r}{\Gamma(r)}e^{-\alpha\lambda}\lambda^{r-1},\ \alpha, r, \lambda > 0,$$

kde α, r sú parametre gamma rozdelenia. Spočítame nepodmienenú pravdepodobnosť

$$P(X = k) = \int_0^\infty P(X = k|\lambda)f(\lambda)d\lambda = \int_0^\infty \frac{\lambda^k}{k!}e^{-\lambda}\frac{\alpha^r\lambda^{r-1}}{\Gamma(r)}e^{-\alpha\lambda}d\lambda,$$

odkiaľ po preznačení $p = \frac{\alpha}{\alpha+1}$ a zintegrovaní pomocou metódy per-partes, dostávame vzorec negatívne binomického rozdelenia (presné odvodenie viď [17])

$$P(X = k) = \binom{k + r - 1}{k} p^r(1 - p)^k.$$

2.4.2 Poissonov regresný model

Dôležitou štatistickou úlohou je hľadanie a skúmanie závislosti premenných, ktorých hodnoty získame pri realizácii experimentov. Vzhľadom k ich náhodnému charakteru reprezentuje nezávisle premenné náhodný vektor $\boldsymbol{X} = (X_1, \ldots, X_k)$ a závisle premennú náhodná veličina Y. Vektor $\boldsymbol{X}$ sa nazýva taktiež vektor vysvetľujúcich premenných a môže byť aj nenáhodný, ako býva v aplikáciách časté.

K popisu závislosti $\boldsymbol{Y}$ na $\boldsymbol{X}$ užívame regresnú analýzu, pričom túto závislosť vyjadruje regresná funkcia

$$y = E(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}),$$

kde $\boldsymbol{x} = (x_1, \ldots, x_k)$ je vektor hodnôt nezávisle premenných (hodnota vektoru $\boldsymbol{X}$), y je závisle premenná (hodnota vektoru $\boldsymbol{Y}$) a $E(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x})$ je podmienená stredná hodnota.

Pri vyšetrovaní závislosti $\boldsymbol{Y}$ na $\boldsymbol{X}$ získame realizáciou n experimentov súbor $(y_i, \boldsymbol{x_i})$, kde y_i sú pozorované hodnoty vektoru $\boldsymbol{Y}$ a $\boldsymbol{x_i}$ pozorované hodnoty vektoru nezávisle premenných $\boldsymbol{X}$ pre $i = 1, \ldots, n$.

Lineárny regresný model

Ako popisuje MacCullagh a Nelder [16], jednou zo základných metód regresnej analýzy je lineárny regresný model. Ten je možné zapísať pomocou matíc ako

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \varepsilon,$$

kde $\boldsymbol{Y}$ je vektor n hodnôt vysvetľovanej premennej, $\boldsymbol{X}$ je matica hodnôt vysvetľujúcich premenných, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ je vektor neznámych hľadaných regresných koeficientov a ε je vektor n hodnôt náhodnej zložky (spravidla s predpokladom normálneho rozdelenia) so strednou hodnotou 0. Koeficient β_0 nazývame intercept.

Ak je splnených niekoľko podmienok lineárneho modelu, môžeme neznámy vektor koeficientov $\boldsymbol{\beta}$ odhadnúť pomocou metódy najmenších štvorcov. Koeficienty môžeme taktiež odhadnúť pomocou metódy maximálnej vierohodnosti (ML - *maximum likelihood*). ML-odhady odhadujú hodnoty tak, aby tieto hodnoty maximalizovali vierohodnú funkciu, tj. odhady sú určené ako najpravdepodobnejšie hodnoty parametrov pre pozorované dáta. Pre podrobnejší výklad viď [16].

Zovšeobecnený lineárny regresný model

Zovšeobecnený lineárny model (GLM) poskytuje obecný rámec pre vytváranie jednotnej triedy modelov, ktoré pracujú so spojitými aj kategorizovanými závisle premennými a rozširuje klasický lineárny model o regresiu, ktorá má chybu modelu rozdelenú napr. podľa Poissonovho, gamma alebo negatívne binomického rozdelenia.

Zovšeobecnený lineárny model zahrňuje napr. lineárnu regresiu, modely analýzy rozptylu a log-lineárny model. Jednotlivé štruktúry a symboly označujeme rovnako ako v prípade lineárneho modelu. Náhodná zložka modelu má vektor stredných hodnôt $E(\boldsymbol{Y})$.

Lineárny prediktor η je systematická zložka v lineárnom modely a vyjadrujeme ju ako

$$\eta = \sum_{i=1}^{k} x_i \hat{\beta}_i,$$

kde $\boldsymbol{x} = (x_1, \ldots, x_k)$ je vektor vysvetľujúcich premenných a $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_k)$ je predikcia modelu $\sum_{i=1}^{k} X_i \beta_i$.

Každá zložka vektoru Y má rozdelenie z exponenciálnej rodiny rozdelení s hustotou

$$f_Y(y, \theta, \phi, \omega) = \exp\left\{ \frac{y\theta - b(\theta)}{\phi} \omega + c(y, \phi, \omega) \right\}, \tag{2.12}$$

kde y je premenná hustoty, ϕ a θ sú parametre rozdelenia, $b(.), c(.)$ sú funkcie, ktorých tvar je daný konkrétnym rozdelením z exponenciálnej rodiny a ω je váha pozorovania, ktorá môže byť pre rôzne pozorovania rôzna. Ak je disperzný parameter ϕ známy, je rovnica (2.12) hustotou rozdelenia z exponenciálnej rodiny a má kanonický parameter θ. Ak parameter ϕ nepoznáme, potom to môže byť dvojparametrické rozdelenie (napr. negatívne binomické). Napr. pre Poissonovo rozdelenie s parametrom λ v rovnici (2.12) dostávame $\theta = \log \lambda, \phi = 1, b(\theta) = \exp(\theta)$.

Logaritmus vierohodnej funkcie (funkcia parametrov θ a ϕ pri známom y) je

$$L(\theta, \phi, y) = \ln f_Y(y, \theta, \phi).$$

Zo vzťahov známych pre vierohodnú funkciu (detailné odvodenie viď [16]) môžeme určiť strednú hodnotu μ a rozptyl σ^2 ako

$$\mu = (Y) = b'(\theta) \text{ a } \sigma^2 = Var(Y) = b''(\theta)\phi.$$

Vzťah medzi lineárnym prediktorom η a strednou hodnotou μ vysvetľovanej náhodnej veličiny Y vyjadruje spojovacia funkcia g (link function) ako

$$\eta = g(\mu).$$

Spojovacia funkcia $g(.)$ môže byť akákoľvek monotónna diferencovateľná funkcia. V klasickom lineárnom modely je spojovacou funkciou identita, tj. $\mu = \eta$. Keďže stredná hodnota $\mu = b'(\theta)$ je len funkciou kanonického parametra θ a spojovacia funkcia je monotónna, existuje inverzná funkcia, ktorou môžeme θ vyjadriť ako funkciu strednej hodnoty $\theta(\mu)$. Keď kanonický parameter rozdelenia je rovný lineárnemu prediktoru, tj. $\theta = \eta$, tak spojovacie funkcie takýchto rozdelení nazývame kanonické. Pre Poissonovo aj negatívne binomické rozdelenie ja kanonickou spojovacou funkciou logaritmus

$$\eta = \ln(\mu). \tag{2.13}$$

Výstižnosť modelu V klasickom modely sa jeho výstižnosť vyjadruje obvykle pomocou koeficientu determinancie (tesnosť preloženia) R^2, tj. ako podiel variability závislej veličiny vysvetlenej modelom na celkovej variabilite. Koeficientom determinancie rozumieme

$$R^2 = 1 - \frac{S_e}{S_t},$$

kde S_e je reziduálny a S_t celkový súčet štvorcov. V modely lineárnej regresie leží hodnota R^2 v intervale $[0, 1]$ a udáva, aký podiel rozptylu v pozorovaní závisle premennej sa podarilo regresiou vysvetliť. Ak model vysvetľuje závislú veličinu úplne, $R^2 = 1$.

Deviancia V zovšeobecnenom lineárnom modely sa dá tesnosť preloženia posudzovať analogicky. Uvažujme tzv. úplný model s k parametrami, ktorý by vysvetľoval hodnoty y presne. Keďže môžeme kanonický parameter vyjadriť ako funkciu strednej hodnoty, $\theta(\mu)$, môžeme vierohodnú funkciu zapísať ako $L(y, \phi, y, \omega)$. To je maximálne dosažiteľná hodnota vierohodnej funkcie. Pre model s jedným parametrom (nulový model, obsahuje iba intercept β_0) by sme dostali vierohodnú funkciu minimálnej hodnoty pre dané dáta. Dvojnásobok rozdielu medzi týmito vierohodnými funkciami je istou analógiou k celkovej variabilite v klasickom modely. Ak označíme odhad stredných hodnôt v modely s k parametrami ako $\hat{\mu}$ a odhad kanonického parametra pre tento model ako $\hat{\theta} = \theta(\hat{\mu})$ a $\tilde{\theta} = \theta(y)$, potom dvojnásobok

rozdielu vierohodných funkcií $L(y, \phi, y, \omega)$ a $L(\hat{\mu}, \phi, y, \omega)$ je

$$2\sum_{i=1}^{n} \omega_i \frac{y_i(\tilde{\theta}_i - \hat{\theta}_i) - b(\tilde{\theta}_i) + b(\hat{\theta}_i)}{\phi} = \frac{D(y, \hat{\mu})}{\phi},$$

kde funkciu $D(y, \hat{\mu})$ nazývame deviancia a je analógiou reziduálnej sumy štvorcov (RSS). Klasický lineárny model je zvláštnym prípadom zovšeobecneného modelu, keď spojovacia funkcia je identita, a potom pre normálne rozdelenú náhodnú zložku modelu je deviancia rovná reziduálnemu súčtu štvorcov

$$D(y, \hat{\mu}) = \sum_{i=1}^{n} (y_i - \hat{\mu}_i)^2 = RSS.$$

V zovšeobecnenom lineárnom modely je teda cieľom nájsť model, ktorý zmenšuje celkovú devianciu. Model by sme mali vyberať tak, aby sme čo najviac znížili devianciu úmernú rozdielu logaritmov vierohodných funkcií medzi úplným modelom a nulovým modelom s jedným parametrom. V prípade Poissonovej regresie počítame devianciu ako

$$D = 2\sum_{i=1}^{n} \left(y_i \log \frac{y_i}{\hat{y}_i} - (y_i - \hat{y}_i) \right),$$

kde $\hat{y}_i$ je predpoveď vysvetľovanej premennej y_i.

AIC V posudzovaní kvality modelu sa často zavádza termín „úspornosť". Vyjadruje totiž takú hľadanú vlastnosť modelu, ktorá váži jeho schopnosť predpovedať hodnoty vysvetľovanej premennej oproti jeho zložitosti. Pomer medzi presnosťou a zložitosťou modelu sa odráža aj v definícii štatistiky, ktorá sa pre vyjadrenie úspornosti modelu používa najčastejšie. Patrí sem aj tzv. AIC štatistika (Akaike Information Criterion). Presný postup výpočtu nie je podstatný, ale je dôležité vedieť, že hodnota AIC je výsledkom súčtu dvoch členov, z ktorých prvý je úmerný logaritmu RSS, zatiaľ čo druhý je penalizačný. Ak totiž do modelu pridáme ďalšiu premennú, môžeme síce zvýšiť jeho presnosť, ale tým taktiež rastie nebezpečenstvo nadhodnotenia modelu. Preto informačné kritériá penalizujú počet premenných a čím je model zložitejší (obsahuje väčší počet parametrov), tým je väčšia hodnota penalizačného člena. Výsledok je tak kompromisom medzi zložitosťou modelu a jeho presnosťou.

Z toho vyplýva skutočnosť, že najúspornejšie modely majú najnižšiu hodnotu AIC štatistiky.

Získanie modelu Výsledný model môže byť vytváraný postupne tak, že do modelu budú zaraďované tie regresory, ktoré znižujú devianciu vzhľadom k aktuálnemu modelu so zaradenými k parametrami. Regresory môžu mať kvalitatívny charakter alebo to môžu byť interakcie (súčiny) pôvodných regresorov. Metóda postupného výberu je veľmi citlivá k situácii, keď máme veľký počet regresorov, z ktorých vyberáme, a to hlavne, ak sú medzi niektorými veľké korelácie. Dobrým počiatočným krokom pri ich výbere je preto uvážlivosť na základe znalosti študovaných javov.

Negatívne binomická regresia ako rozšírenie Poissonovej regresie

V sekcii (2.4.1) sme si odvodili, resp. definovali, niekoľko dôležitých vzťahov medzi Poissonovým a negatívne binomickým rozdelením.

Uvažujme n nezávislých pozorovaní, pri ktorých očakávame, že sa riadia Poissonovým rozdelením, avšak sme zistili, že rozptyl výrazne prevyšuje strednú hodnotu. Zo vzťahu (2.11) vyplýva, že negatívne binomické rozdelenie $\mathrm{NBi}(r, p)$ so strednou hodnotou λ je robustnejšou alternatívou pre Poissonovo rozdelenie s parametrom λ, pretože pre parameter $r \to \infty$ sa rozptyl limitne blíži strednej hodnote. V našom príklade teda má zmysel uvažovať o negatívne binomickom rozdelení.

V sekcii 2.4.1 sme si ukázali, že negatívne binomické rozdelenie môžeme vyjadriť ako zmes Poissonových rozdelení s náhodným parametrom λ, ktorý pochádza z gamma rozdelenia. Ak dokážeme naše pozorovania zaradiť do k skupín (napr. na základe špecifických vlastností) a každá skupina $i = 0, \ldots, k-1$ sa riadi Poissonovým rozdelením s parametrom λ_i, ktoré je z gamma rozdelenia, tak na tieto skupiny môžeme nazerať ako na zmes Poissonových rozdelení. Potom pre odhady regresných koeficientov môžeme namiesto klasickej negatívne binomickej regresie použiť rozšírenú Poissonovú regresiu. Medzi regresory pridáme také premenné, ktoré popisujú rozdiely medzi skupinami.

Kapitola 3

Aplikácia Markovových procesov

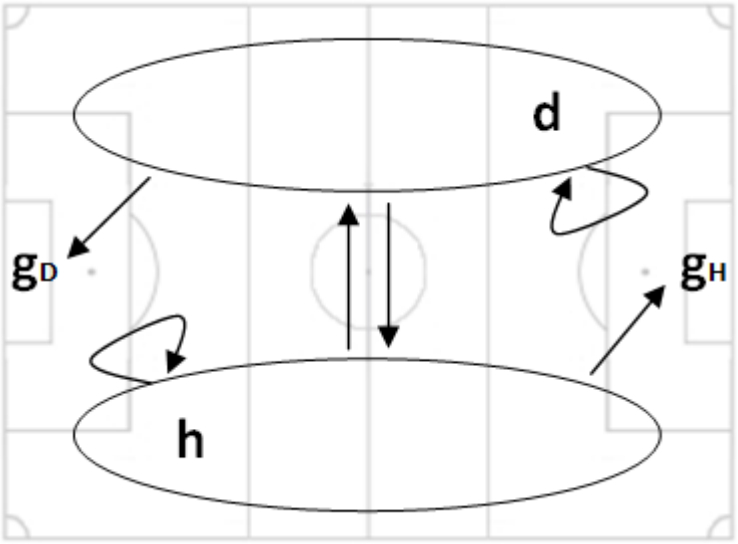
Ako sme už naznačovali, na futbalový zápas môžeme nazerať ako na množinu náhodných prechodov vznikajúcich pri výmene držania lopty alebo skórovania. Predpokladajme, že udalosti vznikajú nezávisle na minulých a časy medzi prechodmi sa chovajú exponenciálne. Preto modelovanie hry pomocou Markovovho procesu sa zdá byť ako rozumný prístup.

3.1 Hra ako Markovov proces

Zápas začína rozohrávkou jedného z týmov v strede hracieho poľa. V jeho priebehu sa kontrola nad loptou medzi týmami často mení a každý tým sa pokúša dosiahnuť gól, udržať si loptu alebo ju získať. V prípade skórovania sa automaticky rozohráva zo stredu hracieho poľa a loptu má tým, ktorý inkasoval.

Spôsobom podobným v [1] sa na hru môžeme v zásade pozerať ako na 4-stavový proces znázornený na Obr. 3.1. Jednotlivé stavy si podrobne popíšeme z pohľadu lopty. Uvažujme dva týmy D (domáci) a H (hostia) hrajúce proti sebe. Hra sa nachádza v stave d, ak má loptu pod kontrolou tým D a v stave h, ak má loptu pod kontrolou tým H. Domáci tým D dosiahne gól v prípade, že hra sa dostane do stavu g_D. Tým hostí H dosiahne gól v prípade, že sa hra dostane do stavu g_H. Šípky medzi stavmi označujú dovolené prechody medzi stavmi. Číselnú reprezentáciu šípok budeme chápať ako pravdepodobnosť prechodu medzi stavom, z ktorého vychádza a stavom, do ktorého vchádza. Ako naznačuje obrázok, domáci tým D môže skórovať, iba ak má loptu pod kontrolou. V prípade udržania lopty ostáva hra v stave d. V prípade straty sa hra presúva do stavu h. Pre tým hostí je to presne naopak.

Poznámka. V zápasoch padne niekedy aj vlastný gól. Keďže sa jedná o ojedinelú situáciu (necelé 1% všetkých gólov), nebudeme ju v našom modely brať do úvahy. Vlastné góly budeme do výpočtov zahrňovať ako klasické. Mnoho kolektívnych hier analógiu vlastného gólu ani nemá.



Obr. 3.1: 4-stavový herný Markovov model

Ako vidíme v Tab. 3.1, na model môžeme nahliadať ako na maticu, kde i-tý riadkový a i-tý stĺpcový index predstavuje stav i, resp. j-tý riadkový a j-tý stĺpcový index predstavuje stav j. Na pozícii $[i, j]$ takejto matice sa nachádza pravdepodobnosť prechodu zo stavu i do stavu j. Ak $i = j$, jedná sa o pravdepodobnosť, že hra v ďalšom kroku ostane v rovnakom stave. Ak je na pozícii $[i, j]$ hodnota 0, prechod zo stavu i do stavu j nie je možný. V obrázku potom takáto šípka neexistuje (napr. vlastné góly, tj. prechody medzi stavmi d a g_H, resp. g_D a h).

	d	h	g_D	g_H	$\sum$
d	p_{dd}	p_{dh}	p_{dg_D}	0	1
h	p_{hd}	p_{hh}	0	p_{hg_H}	1
g_D	0	1	0	0	1
g_H	1	0	0	0	1

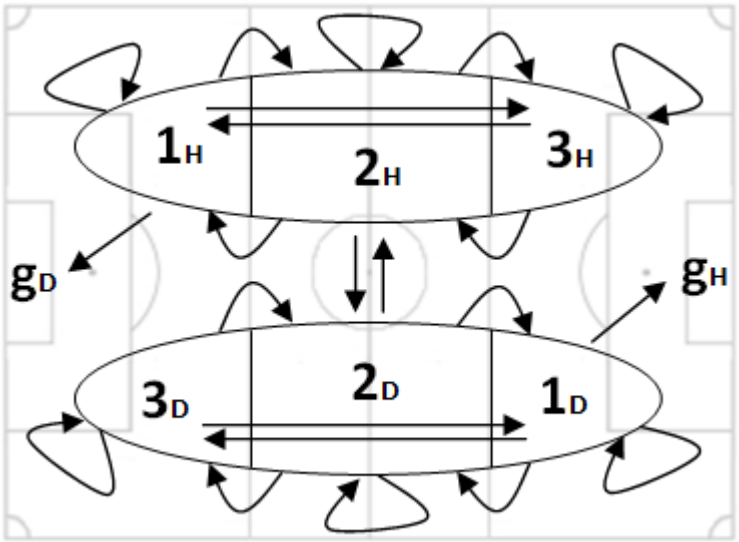
Tabuľka 3.1: Pravdepodobnostná matica prechodov k Obr. 3.1.

Takúto maticu nazývame pravdepodobnostná matica prechodov a je zrejmé,

že súčet hodnôt v každom riadku matice je jedna, pretože sa v ďalšom kroku musíme stále do nejakého stavu dostať (pripúšťa sa aj zotrvanie v pôvodnom).

Dáta, ktoré sme získali, nám umožňujú rozšíriť tento model o nové stavy, aby sme zjemnili prechodové fázy počas hry. Ako ukazuje Obr. 3.2, stavy d a h sme rozdelili na tri nové ($1_D, 2_D, 3_D$, resp. $1_H, 2_H, 3_H$), ktoré reprezentujú obranný, stredový a útočný sektor. Tak získavame 8-stavový Markovov model, ktorý sa stane základom pre ďalšie úvahy. Až na prechody medzi stavmi g_D a i_H, resp. g_H a i_D, kde $i \in \{1, 2, 3\}$, sú všetky ostatné dovolené.

Poznámka. V obrázkoch nie je zaznamenaný jeden typ prechodu, ktorý je však na základe pravidiel hry implicitný. Ako sme už naznačili v úvode kapitoly, po góle sa hra presúva do stredu hracieho poľa, tzn. pravdepodobnosť prechodu zo stavu g_D do stavu 2_H, resp. zo stavu g_H do stavu 2_D je rovná jednej.



Obr. 3.2: 8-stavový rozšírený herný Markovov model

Poznámka. Hraciu plochy by sme si teoreticky mohli rozdeliť na oveľa väčší počet sektorov, ale tým by sme znižovali kvantitatívnu hodnotu dát, pretože počet prechodov nám oproti stavom rastie kvadraticky.

V nasledujúcej podkapitole si odvodíme rovnice pre získanie pravdepodobností skórovania za aktuálnych podmienok, ako je držanie lopty konkrétnym týmom v konkrétnom sektore hracej plochy.

3.2 Proces skórovania

Hlavným cieľom vytvorenia modelu je schopnosť generovať čo najpresnejšie odhady výsledku zápasov, tzn. skóre. Skóre je definované počtom dosiahnutých gólov na oboch stranách. Zápas môžeme chápať ako systém, ktorý podlieha okamžitým zmenám spôsobeným náhodnými javmi - gólmi a prihrávkami. Proces ich pribúdania môžeme interpretovať ako model zrodu a zániku bez zániku, keďže góly nemôžu žiadnym spôsobom zaniknúť.

Model zrodu a zániku je špeciálnym prípadom Markovovho procesu so spojitým časom, kde stavy reprezentujú veľkosť populácie a prechody sú limitované na 1zrody a zániky. Za predpokladu exponenciálneho času medzi udalosťami sa na (homogénny) Poissonov proces môžeme pozerať ako na model zrodu a zániku bez zániku.

Popíšme si situáciu z pohľadu gólov. Ak sa Poissonov proces nachádza v čase t v nejakom zo stavov i_H alebo i_D, kde $i \in \{1, 2, 3\}$, v klasických odvodeniach rovníc pre pravdepodobnosti by sme uvažovali interval $[0, t]$, tj. doba t, za ktorú sa udialo práve n zmien (viď sekciu 2.4). V našom prípade nebudeme sledovať dobu, ktorá uplynula, ale dobu, ktorá ešte len nastane. Zápas má dĺžku 90 minút. Čas $t \in [0, 90]$ teda nepredstavuje dobu od začiatku zápasu, ale dobu do konca zápasu. Nás bude zaujímať, koľko zmien sa udeje v zostávajúcom čase t, resp. intervale $[t', 90]$, kde $t' = 90 - t$. V prípade $t = 90$ predstavuje táto doba celý zápas, tj. 90 minút do konca.

Pre jednoduchší výklad uvažujme opäť dva týmy D a H. Označme $P^M_{i_L}(n|t)$, pravdepodobnosť, že v zostávajúcom čase dĺžky t strelí tým $M \in \{D, H\}$ práve n ($n \geq 0$) gólov za počiatočných podmienok, že loptu má aktuálne pod kontrolou tým $L \in \{D, H\}$ a lopta sa nachádza v sektore $i \in \{1, 2, 3\}$ hracej plochy.

Odvodíme si pravdepodobnosti pre domáci tým D, ktorí drží loptu v sektore i (pre hosťujúci tým sú všetky odvodenia analogické). Všetky intenzity, z ktorých vychádzame, vzťahujeme k rovnakej pevne stanovenej časovej jednotke. Na základe tvrdení v kapitole 2, nech parameter $q_{i_D g_D}$ predstavuje intenzitu prechodu medzi stavmi i_D a g_D. Potom bez ohľadu na počet gólov, ktoré očakávame v zostávajúcom čase t, tj. v intervale $(t', 90)$, je pravdepodobnosť, že v časovom intervale $(t' - \Delta, t')$ strelí tým D práve jeden gól, rovná $q_{i_D g_D}\Delta + o(\Delta)$. Zmeny v intervaloch $(t' - \Delta, t')$ a $(t', 90)$ považujeme za nezávislé. Nesmieme ešte zabudnúť na alternatívny scenár, že situácia gólom neskončí. Ten má dva priebehy - tým D sa podarilo loptu udržať pod kontrolou alebo ju stratil.

Uvažujme dva susedné intervaly $(t' - \Delta, t'), (t', 90)$ pre malé Δ. Ak je

$n \geq 1$, môže v intervale $(t' - \Delta, 90)$ padnúť práve n gólov nasledujúcimi spôsobmi:

1. práve jeden gól za dobu $(t'-\Delta, t')$ s pravdepodobnosťou $q_{i_D g_D}\Delta + o(\Delta)$, $n-1$ gólov za dobu $(t', 90)$.

2. strata kontroly nad loptou za dobu $(t' - \Delta, t')$ s pravdepodobnosťou $q_{i_D j_H}\Delta$ (j je ľub. sektor hracej plochy) a n gólov za dobu $(t', 90)$.

3. udržanie kontroly nad loptou za dobu $(t' - \Delta, t')$ s pravdepodobnosťou $q_{i_D j_D}\Delta$ (j je ľub. sektor hracej plochy), n gólov za dobu $(t', 90)$.

Nech $S = \{1, 2, 3\}$ sú indexy jednotlivých sektorov hracieho poľa. Keďže súčet pravdepodobností je jedna, ponechanie kontroly nad loptou (tzn. mužstvo loptu nestratí) môžeme pre pevne daný sektor i vyjadriť ako

$$\sum_{j \in S} q_{i_D j_D}\Delta = 1 - (q_{i_D g_D} + \sum_{j \in S} q_{i_D j_H})\Delta.$$

Na základe tejto úvahy si odvodíme diferenciálne rovnice pre $P^D_{i_D}(n|t)$ a $P^D_{i_H}(n|t)$. Pre ľubovoľný sektor i určíme pravdepodobnosti počtu gólov domáceho týmu.

Pravdepodobnosť $P^D_{i_D}(n|t)$

Tzn. pravdepodobnosť, že domáci strelia n gólov v zostávajúcom čase t za predpokladu, že majú loptu pod kontrolou v sektore i. Popíšme si najprv slovne, ako chceme postupovať.

Musíme si uvedomiť, akými rôznymi spôsobmi môžeme n gólov dosiahnuť, ak sme v danej chvíli v danej situácii. V odvodení predpokladáme čas do konca zápasu $t + \Delta$, kde $\Delta \approx 0$. V prípade dosiahnutia gólu počas doby Δ sa hra presúva do stredu hracej plochy a loptu má v držaní súper. Od tohto momentu nás zaujíma pravdepodobnosť $P^D_{2_H}(n-1|t)$, tj. pravdepodobnosť, že tým D strelí $n-1$ gólov, ak do konca hry zostáva čas t a loptu má v držaní tým H v sektore 2. Podobne vyjadrujeme pravdepodobnosti pre ďalšie alternatívy - presun lopty do ďalšieho sektoru, strata lopty, atď. Pripomíname, že sektor i je pevne daný a $j \in S$.

Formálne si pravdepodobnosť vyjadríme ako

$$\begin{aligned} P^D_{i_D}(n|t+\Delta) &= P^D_{2_H}(n-1|t)q_{i_D g_D}\Delta + o(\Delta) \\ &\quad + \sum_S (P^D_{j_D}(n|t)q_{i_D j_D}\Delta + P^D_{j_H}(n|t)q_{i_D j_H}\Delta) \\ P^D_{i_D}(n|t+\Delta) &= P^D_{2_H}(n-1|t)q_{i_D g}\Delta + o(\Delta) \\ &\quad + P^D_{i_D}(n|t)q_{i_D i_D}\Delta + P^D_{i_H}(n|t)q_{i_D i_H}\Delta \\ &\quad + \sum_{S\setminus\{i\}} (P^D_{j_D}(n|t)q_{i_D j_D}\Delta + P^D_{j_H}(n|t)q_{i_D j_H}\Delta) \end{aligned}$$

Substitúciou

$$q_{i_D i_D}\Delta = 1 - (q_{i_D g_D} + q_{i_D i_H} + \sum_{S\setminus\{i\}} (q_{i_D j_D} + q_{i_D j_H}))\Delta$$

dostávame

$$\begin{aligned} P^D_{i_D}(n|t+\Delta) &= P^D_{2_H}(n-1|t)q_{i_D g_D}\Delta + P^D_{i_D}(n|t)(1 - (q_{i_D g_D} \\ &\quad + q_{i_D i_H} + \sum_{S\setminus(i)} (q_{i_D j_D} + q_{i_D j_H}))\Delta) \\ &\quad + P^D_{i_H}(n|t)q_{i_D i_H}\Delta + o(\Delta) \\ &\quad + \sum_{S\setminus\{i\}} (P^D_{j_D}(n|t)q_{i_D j_D}\Delta + P^D_{j_H}(n|t)q_{i_D j_H}\Delta) \end{aligned}$$

a po výslednej úprave

$$\begin{aligned} \frac{P^D_{i_D}(n|t+\Delta) - P^D_{i_D}(n|t)}{\Delta} &= P^D_{2_H}(n-1|t)q_{i_D g_D} + P^D_{i_D}(n|t)(-q_{i_D g_D} \\ &\quad - q_{i_D i_H} - \sum_{W\setminus\{i\}} (q_{i_D j_D} + q_{i_D j_H}))) \\ &\quad + P^D_{i_H}(n|t)q_{i_D i_H} + \frac{o(\Delta)}{\Delta} \\ &\quad + \sum_{S\setminus\{i\}} (P^D_{j_D}(n|t)q_{i_D j_D} + P^D_{j_H}(n|t)q_{i_D j_H}). \end{aligned}$$

Limitným prechodom $\Delta \to 0$ získavame diferenciálnu rovnicu pre výpočet

pravdepodobnosti počtu gólov domácich, ak sú pri lopte

$$\begin{aligned} P'^{D}_{i_D}(n|t) &= P^{D}_{2_H}(n-1|t)q_{i_D g_D} + P^{D}_{i_D}(n|t)(-q_{i_D g_D} \\ &\quad -q_{i_D i_H} - \sum_{S\setminus\{i\}}(q_{i_D j_D} + q_{i_D j_H}))) \\ &\quad +P^{D}_{i_H}(n|t)q_{i_D i_H} \\ &\quad + \sum_{S\setminus\{i\}}(P^{D}_{j_D}(n|t)q_{i_D j_D} + P^{D}_{j_H}(n|t)q_{i_D j_H}). \end{aligned} \tag{3.1}$$

Pravdepodobnosť $P^{D}_{i_H}(n|t)$

Tzn. pravdepodobnosť, že domáci strelia n gólov v zostávajúcom čase t za predpokladu, že loptu má pod kontrolou v sektore i tým hostí. Formálne si ju vyjadríme ako

$$\begin{aligned} P^{D}_{i_H}(n|t+\Delta) &= P^{D}_{2_D}(n|t)q_{i_H g_H}\Delta + \sum_{S}(P^{D}_{j_D}(n|t)q_{i_H j_D}\Delta \\ &\quad +P^{D}_{j_H}(n|t)q_{i_H j_H}\Delta) + o(\Delta). \end{aligned}$$

Analogickým postupom odvodíme

$$\begin{aligned} \frac{P^{D}_{i_H}(n|t+\Delta) - P^{D}_{i_H}(n|t)}{\Delta} &= P^{D}_{2_D}(n|t)q_{i_H g_H} + P^{D}_{i_H}(n|t)(-q_{i_H g_H} - q_{i_H i_D} \\ &\quad - \sum_{S\setminus\{i\}}(q_{i_H j_D} + q_{i_H j_H}))) + P^{D}_{i_D}(n|t)q_{i_H i_D} \\ &\quad + \sum_{S\setminus\{i\}}(P^{D}_{j_D}(n|t)q_{i_H j_D} + P^{D}_{j_H}(n|t)q_{i_H j_H}) + \frac{o(\Delta)}{\Delta} \end{aligned}$$

a limitným prechodom $\Delta \to 0$ získavame opäť diferenciálnu rovnicu pre výpočet pravdepodobnosti počtu gólov domácich, ak je tým hostí

pri lopte

$$\begin{aligned} P'^{D}_{i_H}(n|t) &= P^{D}_{2_D}(n|t)q_{i_H g_H}\Delta + P^{D}_{i_H}(n|t)(-q_{i_H g_H} - q_{i_H i_D} \\ &\quad - \sum_{S\setminus\{i\}}(q_{i_H j_D} + q_{i_H j_H}))) + P^{D}_{i_D}(n|t)q_{i_H i_D} \\ &\quad + \sum_{S\setminus\{i\}}(P^{D}_{j_D}(n|t)q_{i_H j_D} + P^{D}_{j_H}(n|t)q_{i_H j_H}). \end{aligned} \tag{3.2}$$

Keď sme si zadefinovali jednotlivé pravdepodobnosti, vyjadrime si sústavu diferenciálnych rovníc (3.1) a (3.2) jednotlivých sektorov hracej plochy ako

$$\begin{pmatrix} P'^{D}_{1_D}(n|t+dt) \\ P'^{D}_{2_D}(n|t+dt) \\ P'^{D}_{3_D}(n|t+dt) \\ P'^{D}_{1_H}(n|t+dt) \\ P'^{D}_{2_H}(n|t+dt) \\ P'^{D}_{3_H}(n|t+dt) \end{pmatrix} = A \times \begin{pmatrix} P^{D}_{1_D}(n|t) \\ P^{D}_{2_D}(n|t) \\ P^{D}_{3_D}(n|t) \\ P^{D}_{1_H}(n|t) \\ P^{D}_{2_H}(n|t) \\ P^{D}_{3_H}(n|t) \end{pmatrix} + B \times \begin{pmatrix} P^{D}_{1_D}(n-1|t) \\ P^{D}_{2_D}(n-1|t) \\ P^{D}_{3_D}(n-1|t) \\ P^{D}_{1_H}(n-1|t) \\ P^{D}_{2_H}(n-1|t) \\ P^{D}_{3_H}(n-1|t) \end{pmatrix}. \tag{3.3}$$

Symbol A reprezentuje maticu intenzít

$$\begin{pmatrix} -q'_{1_D 1_D} & q_{1_D 2_D} & q_{1_D 3_D} & q_{1_D 1_H} & q_{1_D 2_H} & q_{1_D 3_H} \\ q_{2_D 1_D} & -q'_{2_D 2_D} & q_{2_D 3_D} & q_{2_D 1_H} & q_{2_D 2_H} & q_{2_D 3_H} \\ q_{3_D 1_D} & q_{3_D 2_D} & -q'_{3_D 3_D} & q_{3_D 1_H} & q_{3_D 2_H} & q_{3_D 3_H} \\ q_{1_H 1_D} & q_{1_H g_H} + q_{1_H 2_D} & q_{1_H 3_D} & -q'_{1_H 1_H} & q_{1_H 2_H} & q_{1_H 3_H} \\ q_{2_H 1_D} & q_{2_H g_H} + q_{2_H 2_D} & q_{2_H 3_D} & q_{2_H 1_H} & -q'_{2_H 2_H} & q_{2_H 3_H} \\ q_{3_H 1_D} & q_{3_H g_H} + q_{3_H 2_D} & q_{3_H 3_D} & q_{3_H 1_H} & q_{3_H 2_H} & -q'_{3_H 3_H} \end{pmatrix},$$

kde $q'_{i_D i_D} = q_{i_D g_D} + \sum_{S\setminus\{i\}} q_{i_D j_D} + \sum_{S} q_{i_D j_H}$ a $q'_{i_H i_H} = q_{i_H g_H} + \sum_{S\setminus\{i\}} q_{i_H j_H} + \sum_{S} q_{i_H j_D}$, čo sú vlastne záporné súčty hodnôt v i-tom riadku matice A a matice B mimo hodnôt na pozíciách s indexom $[i,i]$ v matici A. Symbol B reprezentuje maticu intenzít

$$\begin{pmatrix} 0 & 0 & 0 & 0 & q_{1_D g_D} & 0 \\ 0 & 0 & 0 & 0 & q_{2_D g_D} & 0 \\ 0 & 0 & 0 & 0 & q_{3_D g_D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

kde posledné 3 riadky sú nulové, pretože neuvažujeme vlastné góly.

Poznámka. Matice A a B sme získali jednoduchou úpravou, kde $i-$tý riadok matice A a matice B dáva podľa (3.3) v súčte odvodenú rovnicu pre pravdepodobnosť počtu gólov pre daný sektor a držanie lopty. Prvá polovica riadkov označuje držanie lopty domáceho týmu, druhá polovica držanie lopty hosťujúceho týmu. Sektory sú pre každú z polovíc zoradené vzostupne podľa indexov.

Týmto zápisom sme získali predpis diferenciálnej rovnice tvaru

$$P'_n(t) = A \times P_n(t) + B \times P_{n-1}(t).$$

Riešenie:

1. Ak $n = 0$, pri počiatočných podmienkach $P_0(0) = 1$ a $P_{-1}(t) = 0$ (strelenie záporného počtu gólov nemá zmysel) riešime rovnicu

   $$P'_0(t) \quad = \quad A \times P_0(t),$$

   ktorej riešenie dostávame z Vety 5

   $$P_0(t) = e^{At}.$$

   Zápis e^{At} predstavuje exponenciálnu maticu popísanú v sekcii 2.3.

2. Pre $\forall n > 0$ aplikujeme vytvárajúcu funkciu popísanú v sekcii 2.4 a dostávame zápis absolútnych pravdepodobností

   $$P_n(t) = e^{At} \times \frac{(Bt)^n}{n!}, (n = 0, 1, 2, \ldots), \tag{3.4}$$

   ktorého parametre tvoria matice intenzít gólov a prihrávok A a B.
   Nech $b[i, j]$ je hodnota v matici B v i-tom riadku a j-tom stĺpci. Potom výsledkom $B' = (Bt)^n$ je matica rovnakého rádu ako B, v ktorej hodnota $b'[i, j] = (b[i, j] * t)^n$.
   Výsledkom e^{At} je matica rovnakého rádu ako A. Jej výpočet je realizovaný pomocou (2.4).

Poznámka. K rovnakému zápisu sa dopracujeme iteračným odvodením z riešenia jednoduchej diferenciálnej rovnice pre $n = 0$ a z riešenia pomocou metódy variácie konštánt pre každé $n > 1$.

Kapitola 4

Analýza dát a odhady

V tejto kapitole sa zameriame na štruktúru získaných dát. Štatistickými testami overíme hypotézy o ich rozdelení a odhadneme parametre modelu popísaného v predchádzajúcej kapitole. Pre tieto účely použijeme štatistický software Matlab[1] a R[2].

4.1 Odhad parametrov pre konkrétnu hru

Ak máme k dispozícii dáta z konkrétneho zápasu, môžeme odhadnúť intenzity prechodu z pohľadu domáceho týmu (pre hostí sú odvodenia analogické) zo stavu i do stavu j jednoducho ako pomer počtu akcií (prihrávok, gólov) a dĺžky stráveného času. Formálne

$$\begin{aligned} q_{i_D j_D} &= \frac{UP_{ij}^D}{T_i^D}, \\ q_{i_D j_H} &= \frac{NP_{ij}^D}{T_i^D}, \\ q_{i_D g_D} &= \frac{G_i^D}{T_i^D}, \end{aligned}$$

kde

- UP_{ij}^D je počet úspešných prihrávok domáceho týmu zo sektoru i do sektoru j (= počet prechodov zo stavu i_D do stavu j_D).

[1] `http://www.mathworks.com`
[2] `http://www.r-project.org`

- NP_{ij}^D je počet neúspešných prihrávok domáceho týmu zo sektoru i do sektoru j (= počet prechodov zo stavu i_D do stavu j_H).
- G_i^D je počet gólov domáceho týmu strelených zo sektoru i (= počet prechodov zo stavu i_D do stavu g).
- T_i^D je celkový čas domáceho týmu, počas ktorého mali loptu pod kontrolou v sektore i.

Ako určíme časy T^D a T^H, tj. časy celkového držania lopty v zápase? Poznáme dĺžku zápasu a počet akcií na oboch stranách. Počet prihrávok síce nemá priamy vplyv na výsledok zápasu, ale je vynikajúcou štatistickou metrikou. V zápase sú totiž najfrekventovanejším javom a dobre odzrkadľujú dianie. Preto najrozumnejším odhadom pre pomer času by mohol byť pomer celkového počtu prihrávok. To znamená, ak T^D je celkový čas držania lopty domáceho týmu a T^H hostí (platí $T^D + T^H = 90$ ako celkový čas zápasu), potom môžeme uvažovať aproximáciu

$$T^D : T^H \approx \left(\sum_{i,j \in S} UP_{ij}^D + \sum_{i,j \in S} NP_{ij}^D \right) : \left(\sum_{i,j \in S} UP_{ij}^H + \sum_{i,j \in S} NP_{ij}^H \right). \quad (4.1)$$

Podobnou aproximáciou určíme čas T_i^D pre daný sektor i, tj. čas držania lopty domácich (pre hostí určíme časy T_i^H analogicky)

$$T_i^D : T^D \approx \left(\sum_{j \in S} UP_{ij}^D + \sum_{j \in S} NP_{ij}^D \right) : \left(\sum_{i,j \in S} UP_{ij}^D + \sum_{i,j \in S} NP_{ij}^D \right). \quad (4.2)$$

Keďže štatistiku času držania lopty nemáme k dispozícii, overíme túto aproximáciu na základe získaných 10 zápasov (1. kolo anglickej ligy v sezóne 2010) s touto informáciou. Pomer prihrávok získame z našich dát.

Párovým t-testom overíme nulovú hypotézu, že sa rozdiely po dvojiciach medzi pomerom prihrávok a časom nelíšia. Párový test sa používa obvykle v situáciách, keď máme na každom z n objektov merané dve veličiny, pričom jednotlivé objekty môžeme považovať za nezávislé, ale merania na tom istom objekte už závislé sú.

V našom prípade, ak μ_1 je stredná hodnota pomerov prihrávok a μ_2 je stredná hodnota pomerov časov, tak chceme testovať hypotézu $H_0 : \mu_1 - \mu_2 = 0$, oproti alternatíve $H_1 : \mu_1 - \mu_2 \neq 0$ na hladine významnosti α. Za predpokladu, že rozdiely pomerov prihrávok a časov majú normálne

#zápas	pomer času	pomer prihrávok	#zápas	pomer času	pomer prihrávok
1	0,67	0,58	**6**	1	1,21
2	1,63	1,16	**7**	1,63	1,52
3	1,79	1,80	**8**	0,61	0,69
4	1,5	0,96	**9**	1,63	1,12
5	0,69	0,71	**10**	1,08	1,80

Tabuľka 4.1: Pomer počtu prihrávok a času držania lopty.

rozdelenie $N(\mu, \sigma^2)$, kde $\mu = \mu_1 - \mu_2$, úloha je prevedená na jednovýberový t-test hypotézy $H_0' : \mu = 0$ oproti alternatíve $H_1' : \mu \neq 0$.

Tab. 4.1 obsahuje namerané hodnoty. V stĺpci *pomer času* sa nachádza skutočná hodnota T^D/T^H. Hodnota 1 nám napríklad hovorí, že každý tým mal loptu v držaní 45 minút. V stĺpci *pomer prihrávok* sa nachádza hodnota $(\sum_{i,j\in S} UP_{ij}^D + \sum_{i,j\in S} NP_{ij}^D)/(\sum_{i,j\in S} UP_{ij}^H + \sum_{i,j\in S} NP_{ij}^H)$, tj. pomer vypočítaný z našich dát.

V Matlabe pomocou funkcie *ttest* získavame hodnoty *t-stat* $= 0,5624$ pri 9 stupňoch voľnosti a *p-hodnota* $= 0,588$. Na hladine významnosti $\alpha = 5\%$ teda nezamietame nulovú hypotézu a v odhadoch budeme používať aproximáciu (4.1). Pre jednotlivé sektory reálne dáta nemáme, budeme však vychádzať z predpokladu podobného chovania a teda používať aj aproximáciu (4.2).

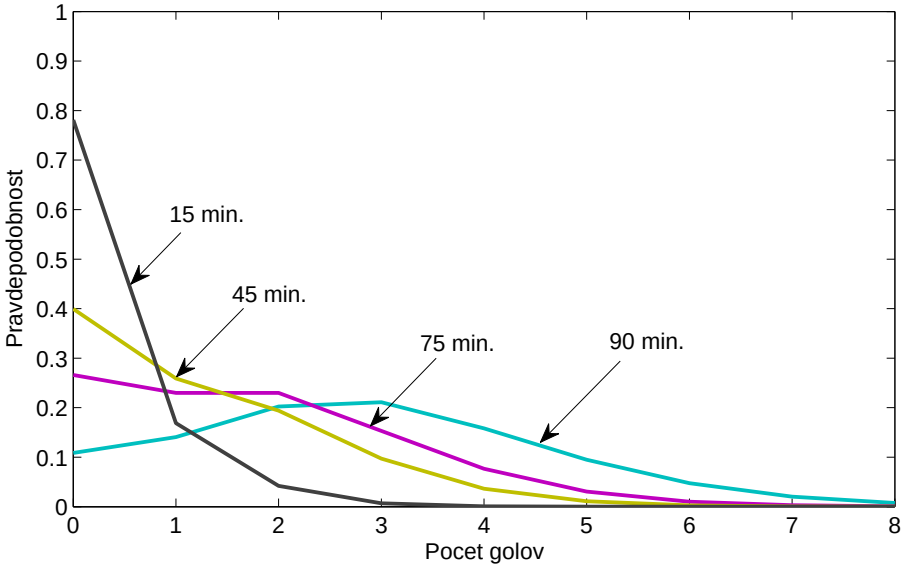
Príklad. *Pre znázornenie si na náhodne vybranom zápase s výsledkom, v ktorom domáci tým strelil* 3 *góly a loptu mal pod kontrolou* 44 *minút, môžeme pozrieť v Tab. 4.2 počet gólov (posledný stĺpec), úspešných a v zátvorke neúspešných prihrávok. Hodnoty boli určené na základe vzťahu* (3.4).

	Obrana	Stred	Útok	Gól
Obrana	45 (7)	42 (12)	3 (8)	**0**
Stred	14 (0)	130 (15)	33 (17)	**1**
Útok	0 (0)	11 (2)	60 (17)	**2**

Tabuľka 4.2: Počet prechodov medzi jednotlivými hernými stavmi v zápase z pohľadu domácich. V zátvorke je počet neúspešných prihrávok.

Na Obr. 4.1 vidíme graf s pravdepodobnosťami počtu gólov za predpokladu, že do konca zápasu zostáva 15, 45, 75 *a* 90 *minút. Čas* 90 *minút do konca vlastne predstavuje celý zápas a príslušná krivka ukazuje, že pri*

daných parametroch je pravdepodobnosť, že domáci tým strelí 3 *góly najväčšia, čo sa zhoduje so skutočným výsledkom.*



Obr. 4.1: Pravdepodobnosti počtu gólov v zostávajúcich 90, 75, 45 a 15 minútach.

Príklad slúžil iba pre ilustráciu toho, ako model funguje, a že dáva očakávané hodnoty. Je zrejmé, že takýmito vstupnými parametrami si v skutočných simuláciách nevystačíme, pretože štatistiky konkrétneho zápasu nebudeme mať k dispozícii. Ak chceme výsledky predpovedať, musíme sa pokúsiť tieto štatistické hodnoty získať inou cestou - odhadom. Pre odhady parametrov $UP^D, NP^D, G^D, UP^H, NP^H, G^H$ v budúcom zápase si preto zavedieme robustnejší model. Na základe nich potom určíme parametre T^D, T^H na základe aproximácie (4.1).

4.2 Regresný odhad herných parametrov

V tejto podkapitole bude naším cieľom odhadnúť parametre Markovovho procesu pomocou lineárnej regresie. Ako sme popisovali v sekcii 2.4.2, pre výber správnej regresnej funkcie je potrebné určiť pravdepodobnostné rozdelenie napozorovaných veličín. Ďalším krokom bude postupné odvodenie množiny regresorov, ktoré významne ovplyvňujú charakteristiky tímov.

4.2.1 Pravdepodobnostné rozdelenie parametrov

Už sme zmienili, že *Maher* [2] vo svojej štúdii nezamietol hypotézu, že počet gólov sa riadi Poissonovým rozdelením. Na našej vzorke dát overíme túto hypotézu pomocou χ^2 testu dobrej zhody. Ten nám umožňuje overiť, či má náhodná veličina vopred dané rozdelenie pravdepodobnosti. Je založený na tom, že náhodnú veličinu s multinomickým rozdelením je možné transformovať na veličinu majúcu rozdelenie χ^2.

Chí kvadrát test dobrej zhody pri neznámych parametroch

Nech H_0 je hypotéza, že náhodný výber z diskrétneho rozdelenia pochádza z Poissonovho rozdelenia. Pre použitie tzv. metódy χ^2 minima s neznámymi parametrami je postup nasledujúci:

1. Obor všetkých možných hodnôt veľkosti N rozdelíme do neprekrývajúcich sa tried. Zvolíme celé čísla $k \geq 3, r \geq 0, d \geq 1$. Do prvej triedy zaradíme hodnoty menšie rovné r. Ďalšie triedy sú tvorené postupne hodnotami $r < x \leq r + d, r + d < x \leq r + 2d, \ldots, r + (k - 3)d < x \leq r + (k - 2)d$. Posledná trieda obsahuje hodnoty väčšie rovné $r + (k - 2)d + 1$. Tým sme vytvorili k tried. Ich početnosti označme $X_r, X_{r+d,}X_{r+2d}, \ldots, X_{r+(k-2)d}, X_{r+(k-2)d+1}$.

2. Pre každú časť sa stanoví pravdepodobnosť p_i, že náhodná veličina nadobudne hodnotu z i-tej časti. V prípade Poissonovho rozdelenia

$$P(X = j) = q_j = \frac{\lambda^j}{j!} e^{-\lambda}, j = 0, 1, \ldots$$

   je táto pravdepodobnosť určená ako

$$\begin{aligned} p_1 &= \sum_{j=0}^{r} q_j, \\ p_i &= \sum_{j=r+(i-1)d}^{r+(i-1)d} q_j, 1 < i < k, \\ p_k &= \sum_{j=r+(k-2)d+1}^{\infty} q_j. \end{aligned}$$

3. Odhad parametra λ určíme ako

$$\frac{1}{n}\left[X_r\frac{\sum_{j=0}^{r} jq_j}{p_1}+\sum_{i=2}^{k-1}X_u\frac{\sum_{j=u-d+1}^{u} jq_j}{p_i}+X_{v+1}\frac{\sum_{j=v+1}^{\infty} jq_j}{p_k}\right], \quad (4.3)$$

kde $u = r + (i-1)d$ a $v = r + (k-2)d$. Ako počiatočnú aproximáciu λ_0 riešenia rovnice (4.3) určíme výberový priemer, na základe ktorého určíme všetky q_j. Takto ľahko dostaneme pravdepodobnosti p_1 až p_k a môžeme vypočítať ďalšiu aproximáciu λ_1. Tento krok sa iteračne opakuje.

4. Nakoniec sa porovnajú očakávané počty v jednotlivých častiach (Np_i) so skutočnými (X_i) pomocou vzorca

$$\chi^2 = \sum_{i=1}^{k}\frac{(X_i - Np_i)^2}{Np_i}.$$

Za platnosti H_0 má testová štatistika χ^2 rozdelenie chí kvadrát. Ak boli všetky parametre známe, je počet stupňov voľnosti $k-1$. Ak bol niektorý parameter neznámy, znižuje sa počet stupňov voľnosti o jeden za každý takýto parameter. V tomto prípade počítame s $k-2$ stupňami voľnosti, keďže sme odhadovali neznámy parameter λ. Hodnotu veličiny χ^2 porovnáme s kritickou hodnotou príslušného rozdelenia chí kvadrát na požadovanej hladine významnosti α (zvyčajne $0,05$). Za predpokladu, že $Np_i > 5$ pre $\forall i$, zamietame hypotézu H_0 na hladine významnosti α, ak

$$\chi^2 \geq \chi^2_{1-\alpha}(k-2),$$

kde $\chi^2_{1-\alpha}(k-2)$ je $(1-\alpha)$-kvantil rozdelenia χ^2_{k-2}. Ak nie je podmienka $Np_i > 5$ splnená, je možné najskôr niektoré výsledky zlúčiť.

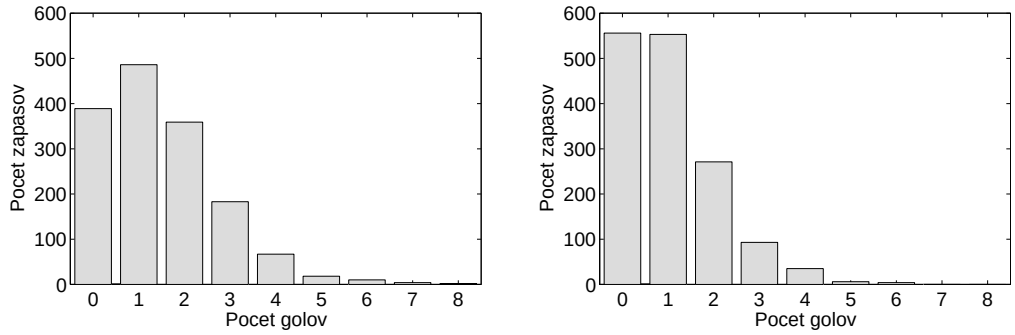
Poznámka. Obvyklým výstupom počítačových programov na testovanie hypotéz je tzv. *p-hodnota*. Je to pravdepodobnosť, že za predpokladu platnosti nulovej hypotézy H_0 nadobudne absolútna hodnota náhodnej veličiny hodnoty pozorovanej, alebo väčšej. Inak povedané je to najmenšia hladina významnosti, na ktorej by nulová hypotéza mala byť zamietnutá. Malá p-hodnota znamená, že pri platnosti nulovej hypotézy by výskyt nami pozorovaných dát nebol moc pravdepodobný.

Niekedy sa namiesto rovnice (4.3) za odhad berie iba aritmetický priemer $\bar{X}$. Dochádza však pritom k zvýšeniu pravdepodobnosti, že zamietneme hy-

potézu o Poissonovom rozdelení, aj keď je správna.

V nasledujúcich častiach sa pozrieme na pravdepodobnostné rozdelenie parametrov, ktoré potrebujeme pre inicializácii modelu - góly, úspešné a neúspešné prihrávky. S predpokladom exponenciálneho rozdelenia časov medzi týmito javmi overíme, či sa ich počet riadi Poissonovým rozdelením.

Pravdepodobnostné rozdelenie počtu gólov



(a) Góly domácich (b) Góly hostí

Obr. 4.2: Rozdelenie počtu gólov domácich a hostí v 1518 zápasoch za posledné 4 sezóny.

	Obr. 4.2(a)	Obr. 4.2(b)
E(X)	1,4611	1,0125
Var(X)	1,6951	1,1263
χ^2	26,901	11,110
počet stupňov voľnosti	5	4
p-hodnota	6×10^{-5}	0,025
zamietame H_0	áno	áno

Tabuľka 4.3: Výsledky χ^2 testu o Poissonovom rozdelení počtu gólov s hladinou významnosti 5%.

Grafy rozdelenia počtu gólov v zápasoch posledných 4 sezón vidíme na Obr. 4.2. Na prvých dvoch riadkoch Tab. 4.3 vidíme, že výberový rozptyl je v oboch prípadoch mierne väčší ako výberový priemer. Vzhľadom na typ problému a tvar grafov sa môžeme prirodzene domnievať, že sa počty riadia Poissonovým rozdelením. Neznámy parameter λ asymptoticky odhadneme

pomocou (4.3). Výsledky χ^2 testu dobrej zhody obsahuje taktiež Tab. 4.3. Z nich je zrejmé, že pri $\alpha = 5\%$ zamietame hypotézu H_0 o Poissonovom rozdelení.

V praxi to môžeme vysvetliť tak, že naše pozorovania obsahujú nadmerný počet extrémnejších výsledkov, ktoré jediným parametrom λ nepokryjeme. Dá sa očakávať, že väčší počet takých výsledkov prinášajú zápasy medzi týmami, ktoré sú kvalitatívne odlišné. V sekcii 2.4 sme podobnou úvahou dospeli k použitiu negatívne binomického rozdelenia s parametrami p a r, ktoré pokryje aj extrémnejšie hodnoty. Tab. 4.4 obsahuje výsledky χ^2 testu dobrej zhody s hypotézou H_0 o negatívne binomickom rozdelení počtu gólov. Parametre p a r sme odhadli na základe (2.9) a (2.10), takže stupeň voľnosti je znížený o 2. V tomto prípade sme nulovú hypotézu nezamietli.

	Obr. 4.2(a)	Obr. 4.2(b)
p	0,862	0,900
r	9,127	9,081
χ^2	7,229	2,987
počet stupňov voľnosti	4	3
p-hodnota	0,124	0,394
zamietame H_0	nie	nie

Tabuľka 4.4: Výsledky χ^2 testu o negatívne binomickom rozdelení počtu gólov s hladinou významnosti 5%.

Načrtnime si, ako by bolo možné zmes Poisonových rozdelení získať. V sezóne hrajú všetky týmy proti sebe systémom doma/vonku. Za posledné 4 sezóny hralo najvyššiu súťaž 28 týmov. Na základe rebríčku vytvoreného priemerne dosiahnutým počtom bodov za jednu sezónu rozdeľme týmy do troch výkonnostných kategórií v pomere $9 : 9 : 10$. Tým dostávame $3^2 = 9$ skupín, kde prvá pokrýva zápasy týmov prvej kategórie, druhá pokrýva zápasy, v ktorých domáci tým je z prvej a hosťujúci z druhej kategórie, atď. Zápasy, v ktorých domáce týmy patria do rovnakej kategórie a hosťujúce týmy patria do rovnakej kategórie, sa dajú očakávať ako kvalitatívne podobné, a teda rozdelenie počtu gólov by sa mohlo blížiť Poissonovmu rozdeleniu. Tab. 4.5 obsahuje výsledky χ^2 testu dobrej zhody s hypotézou H_0 o Poissonovom rozdelení v rámci výkonnostných skupín. Stĺpec D označuje výkonnostnú kategóriu domácich, H kategóriu hostí, s.v. je stupeň voľnosti, p-h. je p-hodnota a H_0 indikuje, či hypotézu zamietame.

(a) Góly domácich týmov

D	H	χ^2	s.v.	p-h.	H_0
1	**1**	0,97	3	0,81	nie
1	**2**	0,99	4	0,20	nie
1	**3**	2,83	4	0,59	nie
2	**1**	1,84	2	0,40	nie
2	**2**	2,17	3	0,54	nie
2	**3**	3,85	3	0,28	nie
3	**1**	4,97	2	0,08	nie
3	**2**	2,50	2	0,29	nie
3	**3**	0,24	2	0,89	nie

(b) Góly hosťujúcich týmov

D	H	χ^2	s.v.	p-h.	H_0
1	**1**	6,23	3	0,10	nie
1	**2**	3,01	2	0,22	nie
1	**3**	1,56	1	0,21	nie
2	**1**	2,38	3	0,50	nie
2	**2**	3,20	2	0,20	nie
2	**3**	1,91	1	0,17	nie
3	**1**	0,56	3	0,91	nie
3	**2**	0,73	2	0,69	nie
3	**3**	1,30	1	0,25	nie

Tabuľka 4.5: Výsledky χ^2 testu o Poissonovom rozdelení počtu gólov s hladinou významnosti 5% v rámci výkonnostných kategórií.

Keďže sme ani v jednom prípade nulovú hypotézu nezamietli, pre odhad strednej hodnoty počtu gólov λ_{gol} môžeme použiť Poissonov logaritmicko-lineárny model. Výhoda domáceho prostredia (*doma*) je bez ďalšej diskusie prirodzeným regresorom. Z Tab. 4.5 môžeme urobiť záver, že ďalším prirodzeným regresorom je príslušnosť vo výkonnostnej kategórii (označíme *ktgskore*, pretože tento regresor bude reprezentovať schopnosť skórovania v danej kategórii). Dostávame teda

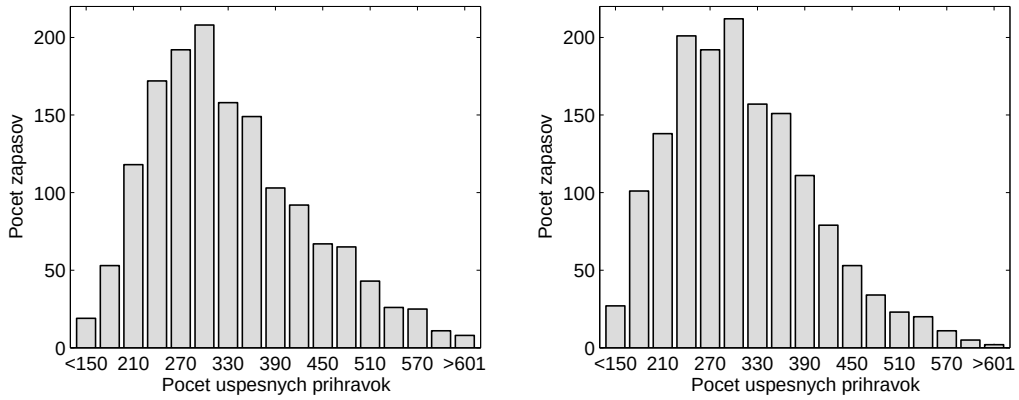
$$\ln(\lambda_{gol}) = \beta + \beta_{doma}X_{doma} + \beta_{ktgskore}X_{ktgskore} + \sum_{i=3}^{n} \beta_i X_i, \qquad (4.4)$$

kde X_i je množina vysvetľujúcich premenných, ktoré sa stanú neskôr predmetom nášho skúmania a β_i je množina príslušných koeficientov.

Pravdepodobnostné rozdelenie počtu prihrávok

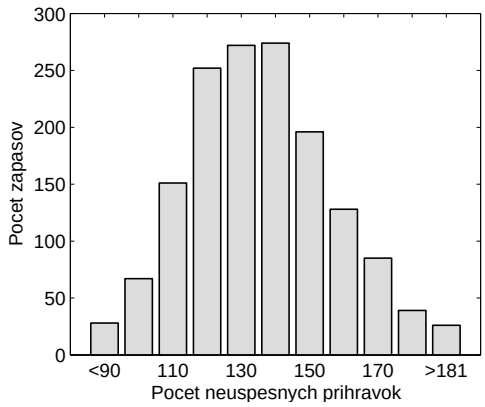
Prirodzene môžeme očakávať, že kvalitnejší tým dosiahne spravidla viac prihrávok a teda aj viac gólov. V našom modely počítame s dvoma typmi prihrávok - úspešné (udržanie lopty) a neúspešné (strata lopty, čo z pohľadu súpera znamená jej získanie).

Keďže počet úspešných prihrávok v zápase sa pohybuje medzi 100 až 800, agregujeme dáta v skupinách 30 prihrávok. V prípade neúspešných prihrávok je to v medzi 50 až 200, takže dáta agregujeme v skupinách 10 prihrávok. Na Obr. 4.3 vidíme rozdelenie počtu prihrávok domácich a hostí.

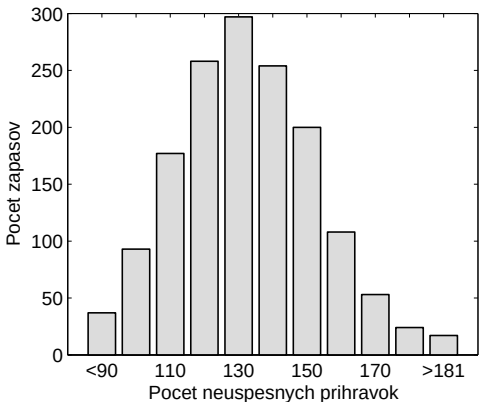


(a) Úspešné prihrávky domácich

(b) Úspešné prihrávky hostí



(c) Neúspešné prihrávky domácich



(d) Neúspešné prihrávky hostí

Obr. 4.3: Rozdelenie počtu úspešných a neúspešných prihrávok domácich a hostí.

Zápasy si opäť rozdelíme podľa výkonnostných kategórií, pretože sa dá domnievať, že počty úzko súvisia s kvalitou týmov. Ako vidíme v Tab. 4.6 (posledný riadok zodpovedá všetkým týmom bez zaradenia do kategórie), aj napriek rozdeleniu do kategórií, v každom riadku je evidentne vyšší rozptyl a oproti gólom sa nachádzame v úplne inej situácii. Poissonovo rozdelenie nepripadá do úvahy a núka sa tak použitie negatívne binomického rozdelenia.

(a) Domáci - úsp. prihr.

D	H	E(X)	Var(X)
1	**1**	351,4	9901,6
1	**2**	395,8	10238,6
1	**3**	393,3	12624,4
2	**1**	270,3	4972,4
2	**2**	285,1	6581,7
2	**3**	274,3	4672,1
3	**1**	260,1	3845,3
3	**2**	272,7	4962,6
3	**3**	261,2	4743,6
-	-	319,0	10479,1

(b) Hostia - úsp. prihr.

D	H	E(X)	Var(X)
1	**1**	329,8	7494,9
1	**2**	260,4	4865,0
1	**3**	248,8	3507,2
2	**1**	354,9	9201,6
2	**2**	264,8	5250,2
2	**3**	249,4	4204,6
3	**1**	345,7	9475,1
3	**2**	256,4	4963,4
3	**3**	240,6	4925,0
-	-	294,9	8284,4

(c) Domáci - neúsp. prihr.

D	H	E(X)	Var(X)
1	1	124,1	376,5
1	2	123,4	372,9
1	3	123,9	422,7
2	1	131,8	446,6
2	2	137,3	390,7
2	3	136,9	405,5
3	1	134,4	443,7
3	2	140,5	531,3
3	3	137,9	457,8
-	-	130,3	452,5

(d) Hostia - neúsp. prihr.

D	H	E(X)	Var(X)
1	1	121,2	321,4
1	2	123,7	411,4
1	3	122,9	460,1
2	1	125,6	353,1
2	2	131,3	355,5
2	3	134,9	505,9
3	1	140,0	466,2
3	2	133,7	534,0
3	3	137,2	322,7
-	-	127	423,5

Tabuľka 4.6: Priemer a rozptyl úspešných a neúspešných prihrávok v rámci výkonnostných kategórií.

Preto χ^2 testom dobrej zhody overíme hypotézu H_0, že sa počet prihrávok riadi negatívne binomickým rozdelením. Neznáme parametre p a r sme odhadli opäť na základe (2.9) a (2.10) a stupeň voľnosti je teda znížený o 2. Výsledky testu obsahuje Tab. 4.7. Tie ukazujú, že túto nulovú hypotézu

nezamietame.

(a) Domáci - úspešné prihrávky

D	H	χ^2	s.v.	p-h.	H_0
1	**1**	16,62	10	0,08	nie
1	**2**	10,75	11	0,46	nie
1	**3**	8,23	8	0,41	nie
2	**1**	9,90	8	0,27	nie
2	**2**	9,90	7	0,19	nie
2	**3**	3,11	6	0,79	nie
3	**1**	2,31	5	0,80	nie
3	**2**	5,01	6	0,54	nie
3	**3**	4,19	4	0,38	nie
-	-	39,01	14	0,00	áno

(b) Hostia - úspešné prihrávky

D	H	χ^2	s.v.	p-h.	H_0
1	**1**	8,02	10	0,62	nie
1	**2**	2,97	7	0,88	nie
1	**3**	9,02	4	0,06	nie
2	**1**	3,14	10	0,97	nie
2	**2**	16,03	7	0,02	*nie
2	**3**	12,78	5	0,02	*nie
3	**1**	14,31	8	0,07	nie
3	**2**	6,11	6	0,41	nie
3	**3**	1,68	3	0,64	nie
-	-	26,93	13	0,01	*nie

(c) Domáci - neúspešné prihrávky

D	H	χ^2	s.v.	p-h.	H_0
1	**1**	12,62	6	0,05	nie
1	**2**	2,60	6	0,85	nie
1	**3**	3,66	6	0,72	nie
2	**1**	5,29	6	0,50	nie
2	**2**	6,95	5	0,22	nie
2	**3**	9,57	5	0,88	nie
3	**1**	3,72	6	0,71	nie
3	**2**	9,13	5	0,10	nie
3	**3**	1,86	3	0,60	nie
-	-	6,89	7	0,44	nie

(d) Hostia - neúspešné prihrávky

D	H	χ^2	s.v.	p-h.	H_0
1	**1**	3,11	5	0,68	nie
1	**2**	9,86	6	0,13	nie
1	**3**	8,37	3	0,21	nie
2	**1**	13,21	6	0,04	*nie
2	**2**	1,33	5	0,93	nie
2	**3**	7,39	6	0,28	nie
3	**1**	2,79	6	0,83	nie
3	**2**	11,64	6	0,07	nie
3	**3**	2,59	3	0,45	nie
-	-	4,03	7	0,77	nie

Tabuľka 4.7: Výsledky χ^2 testu o negatívne binomickom rozdelení počtu úspešných a neúspešných prihrávok s hladinou významnosti 5% (*1%) v rámci výkonnostných kategórií.

Keďže kanonickou spojovacou funkciou negatívne binomického rozdelenia je logaritmus, pre odhad strednej hodnoty počtu úspešných prihrávok λ_{lopta_0} teda môžeme použiť logaritmicko-lineárny model pre každú výkonnostnú kategóriu (pre úspešné prihrávky označíme regresné koeficienty gréckym písmenom γ)

$$\ln(\lambda^{ktg}_{lopta_0}) = \gamma^{ktg} + \gamma^{ktg}_{doma} X_{doma} + \sum_{i=2}^{n} \gamma^{ktg}_i X_i \tag{4.5}$$

a analogicky pre odhad strednej hodnoty počtu neúspešných prihrávok $\lambda_{lopta\_}$ pre každú výkonnostnú kategóriu (pre neúspešné prihrávky označíme regresné koeficienty gréckym písmenom δ)

$$\ln(\lambda^{ktg}_{lopta\_}) = \delta^{ktg} + \delta^{ktg}_{doma} X_{doma} + \sum_{i=2}^{n} \delta^{ktg}_{i} X_i, \tag{4.6}$$

kde X_i je tak, ako v prípade gólov, množina ďalších možných vysvetľujúcich premenných. Tie sa pokúsime nájsť v nasledujúcej podkapitole.

4.2.2 Odvodenie regresného modelu

V predchádzajúcom odstavci sme si odvodili základné logaritmicko-lineárne modely pre góly (4.4), úspešné prihrávky (4.5) a neúspešné prihrávky (4.6). Pokúsime sa rozšíriť množinu vysvetľujúcich premenných o individuálne charakteristiky jednotlivých tímov, aby sme zvýšili presnosť odhadu. Tým máme na mysli vyjadrenie schopnosti skórovať alebo získať loptu, či naopak, inkasovať alebo stratiť loptu. Pre dva tímy D a H hrajúce proti sebe si rozšírime aktuálne modely. Počítanie regresných koeficientov bolo prevedené metódou maximálnej vierohodnosti v štatistickom software R pomocou funkcie *glm*.

Góly

Na základe dostupných dát zvolíme nasledujúce vyvetľujúce premenné, ktoré by mohli mať významný vplyv na odhad počtu gólov v konkrétnom zápase a popíšeme si ich všeobecne z pohľadu ľubovoľného tímu U:

1. $X_{skore_+}(U)$ ako celková schopnosť skórovania tímu U vo všetkých zápasoch.

2. $X_{skore_-}(U)$ ako celková schopnosť inkasovania tímu U vo všetkých zápasoch.

3. $X_{dvskore_+}(U)$ ako schopnosť skórovania tímu U v domácich zápasoch, ak hrá práve doma, resp. vonkajších zápasoch, ak hrá práve vonku.

4. $X_{dvskore_-}(U)$ ako schopnosť inkasovania tímu U v domácich zápasoch, ak hrá práve doma, resp. vonkajších zápasoch, ak hrá práve vonku.

5. $X_{taktika}(U)$ ako zvolené počiatočné rozloženie hráčov tímu U na ihrisku.

	Odhad (Chyba)	p-value
β	-1,470 (0,08)	$< 2 \times 10^{-16}$
β_{doma}	0,232 (0,04)	$2,3 \times 10^{-8}$
$\beta_{ktgskore}$	0,250 (0,05)	$< 2 \times 10^{-16}$
$\beta_{dvskore_+}$	0,341 (0,04)	$< 2 \times 10^{-16}$
β_{skore_-}	0,594 (0,06)	$1,29 \times 10^{-6}$

Tabuľka 4.8: Odhadnuté koeficienty pre góly na hladinou významnosti $\alpha < 0.001$.

Pre použitie Poissonovej regresie musíme splniť predpoklad rovnosti strednej hodnoty a rozptylu, čo znamená, že disperzný parameter ϕ je rovný 1. V prípade gólov sme vypočítali $\phi = 0,99$, čo nás k použitiu Poissonovej regresie oprávňuje. Pre tento účel použijeme implicitnú funkciu $glm(Y \sim X_1 + X_2 + \ldots + X_n, family = poisson)$.

Spôsobom popísaným v druhej časti sekcie 2.4.2 sme vybrali finálny model. Rozdiel deviancie nulového a parametrizovaného modelu dosahuje spomedzi všetkých testovaných modelov svojho maxima a hodnota AIC naopak svojho minima. Pre domáci tým D a hosťujúci tým H tak dostávame

$$\begin{aligned} \ln(\lambda_{gol}(D)) &= \beta + \beta_{doma}X_{doma} + \beta_{ktgskore}X_{ktgskore} \\ &\quad +\beta_{dvskore_+}X_{dvskore_+}(D) + \beta_{skore_-}X_{skore_-}(H) \\ \ln(\lambda_{gol}(H)) &= \beta + X_{doma}\beta_{doma} + \beta_{ktgskore}X_{ktgskore} \\ &\quad +\beta_{dvskore_+}X_{dvskore_+}(H) + \beta_{skore_-}X_{skore_-}(D), \end{aligned} \tag{4.7}$$

kde jednotlivé hodnoty odhadnutých koeficientov obsahuje Tab. 4.8. Ako vidíme, počiatočné rozloženie hráčov na hracej ploche reprezentované premennou $X_{taktika}$ nemá významný vplyv na počet strelených gólov.

Prihrávky

Pre odhad počtu prihrávok uvažujeme nasledujúce vyvetľujúce premenné, ktoré by mohli mať významný vplyv na odhad počtu prihrávok v konkrétnom zápase a popíšeme si ich opäť všeobecne z pohľadu ľubovoľného týmu U a jeho súpera V:

1. $X_{lopta_0}(U)$ ako celková schopnosť udržania lopty týmu U vo všetkých zápasoch.

2. $X_{lopta_+}(U)$ ako celková schopnosť získania lopty tímu U vo všetkých zápasoch.

3. $X_{lopta_-}(U)$ ako celková schopnosť straty lopty tímu U vo všetkých zápasoch.

4. $X_{vslopta_0}(U,V)$ ako schopnosť udržania lopty tímu U iba v zápasoch proti tímu V.

5. $X_{vslopta_+}(U,V)$ ako schopnosť získania lopty tímu U iba v zápasoch proti tímu V. Regresor $X_{vslopta_-}(U,V)$ nemusíme explicitne vyjadrovať, pretože je to rovnaká hodnota ako $X_{vslopta_+}(V,U)$.

V prípade prihrávok je v rámci každej kategórie evidentný „overdispersion" efekt (parameter ϕ sa pohybuje v rozmedzí 20 až 30 pre úspešné a viac ako 70 pre neúspešné), takže aplikujeme negatívne binomickú regresiu. Koeficienty budeme tak odhadovať pomocou dodatočne naimportovanej funkcie $glm.nb(Y \sim X_1 + X_2 + \ldots + X_n)$ z balíka *MASS*.

Rovnakým spôsobom ako pri góloch, získavame pre domáci tým D a hosťujúci tým H finálny model pre úspešné prihrávky v rámci každej výkonnostnej kategórie

$$\begin{aligned}
\ln(\lambda^{ktg}_{lopta_0}(D)) &= \gamma^{ktg} + \gamma^{ktg}_{doma}X_{doma} + \gamma^{ktg}_{lopta_0}X_{lopta_0}(D) \\
&\quad +\gamma^{ktg}_{lopta_+}X_{lopta_+}(H) + \gamma^{ktg}_{4}X_{vslopta_0}(D,H) \\
\ln(\lambda^{ktg}_{lopta_0}(H)) &= \gamma^{ktg} + \gamma^{ktg}_{doma}X_{doma} + \gamma^{ktg}_{lopta_0}X_{lopta_0}(H) \\
&\quad +\gamma^{ktg}_{lopta_+}X_{lopta_+}(D) + \gamma^{ktg}_{vslopta_0}X_{vslopta_0}(H,D),
\end{aligned}$$

kde hodnoty jednotlivých odhadnutých koeficientov a štandardných chýb v zátvorkách obsahuje Tab. 4.9 (na ďalšej strane).
Pre neúspešné prihrávky v rámci každej kategórie dostávame

$$\begin{aligned}
\ln(\lambda^{ktg}_{lopta_-}(D)) &= \delta^{ktg} + \delta^{ktg}_{doma}X_{doma} + \delta^{ktg}_{lopta_-}X_{lopta_-}(D) \\
&\quad +\delta^{ktg}_{lopta_+}X_{lopta_+}(H) + \delta^{ktg}_{vslopta_+}X_{vslopta_+}(H,D) \\
\ln(\lambda^{ktg}_{lopta_-}(H)) &= \delta^{ktg} + \delta^{ktg}_{doma}X_{doma} + \delta^{ktg}_{lopta_-}X_{lopta_-}(H) \\
&\quad +\delta^{ktg}_{lopta_+}X_{lopta_+}(D) + \delta^{ktg}_{vslopta_+}X_{vslopta_+}(D,H),
\end{aligned}$$

kde hodnoty jednotlivých odhadnutých koeficientov a štandardných chýb v zátvorkách obsahuje Tab. 4.10 (na ďalšej strane). Z toho je zrejmé,

Ktg	γ^{ktg}	γ^{ktg}_{doma}	$\gamma^{ktg}_{lopta_0}$	$\gamma^{ktg}_{lopta_+}$	$\gamma^{ktg}_{vslopta_0}$
11	4802 (45)	0	3 (0,1)	0	0
12	4283 (250)	66 (34)	3 (0,2)	3(2)	1 (0,3)
13	4597 (69)	113 (38)	2 (0,2)	0	1 (0,3)
21	4173 (214)	0	3 (0,1)	4(2)	0
22	4449 (79)	0	4 (0,2)	0	0
23	4267 (93)	0	3 (0,3)	0	2 (0,3)
31	4276 (242)	-63 (33)	3 (0,2)	2 (2)	1 (0,3)
32	4339 (355)	0	3 (0,4)	0	2 (0,3)
33	4218 (151)	0	3 (0,6)	0	2 (0,5)

Tabuľka 4.9: Odhadnuté koeficienty pre úspešné prihrávky v rámci výkonnostných kategórií na hladine významnosti $\alpha < 0,01$. Hodnoty koeficientov v 2. až 6. stĺpci sú kvôli úspore miesta prenásobené číslom 10^3.

že na počet výmien držania lopty medzi týmami nemá významný vplyv výhoda domáceho prostredia.

V tejto chvíli máme odvodené rovnice, vďaka ktorým získame odhady stredných hodnôt parametrov, potrebných pre inicializáciu matíc A a B v odvodenej sústave (3.3). Mohli by sme prehlásiť model za plne použiteľný, avšak musíme si uvedomiť, že v takom prípade by sme zápas odhadovali ako celok. Nebolo by však správne zjednodušovať si situáciu na fakt, že hra sa dá prezentovať ako homogénny proces, ale naopak dá sa silne predpokladať, že parametre sa v priebehu zápasu menia. Napríklad v závislosti na čase. Špecifickejšie závislosti sa pokúsime zanalyzovať v nasledujúcej podkapitole.

4.3 Špecifická závislosť počtu gólov počas hry

Skúsenosť naznačuje, že má zmysel uvažovať nad dynamickými zmenami parametrov počas priebehu hry. Málokedy sa dá očakávať, že v prvých minútach za nerozhodného stavu a otvoreného priebehu padne viac gólov ako v posledných minútach, keď sa rozhoduje o konečnom skóre. Skúsime si zodpovedať nasledujúce otázky:

- Padá v postupujúcom priebehu v priemere stále viac gólov? Napr. z dôvodu únavy hráčov a väčšieho počtu chýb.
- Závisí počet gólov, resp. doba čakania na ďalší gól od aktuálneho skóre? Vyhrávajúci tým sa snaží navýšiť skóre, zatiaľ čo prehrávajúci vyrovnať. Ktorý efekt je evidentnejší?

Ktg	δ^{ktg}	δ^{ktg}_{doma}	$\delta^{ktg}_{lopta_-}$	$\delta^{ktg}_{lopta_+}$	$\delta^{ktg}_{vslopta_+}$
11	2936 (149)	0	6 (0,8)	6 (1)	3 (0,7)
12	2980 (159)	0	5 (1)	5 (1)	4 (0,7)
13	3157 (180)	0	5 (1)	4 (1)	4 (0,8)
21	3171 (148)	0	5 (0,8)	4 (1)	4 (0,7)
22	3142 (201)	0	5 (1)	4 (1,3)	4 (0,7)
23	3125 (199)	0	5 (1)	5 (1)	3 (0,7)
31	3232 (155)	0	4 (0,9)	4 (1)	4 (0,7)
32	3230 (159)	0	5 (0,7)	3 (1)	4 (0,7)
33	3517 (156)	0	4 (1)	4 (1)	2 (0,8)

Tabuľka 4.10: Odhadnuté koeficienty pre neúspešné prihrávky v rámci výkonnostných kategórií na hladine významnosti $\alpha < 0,01$. Hodnoty koeficientov v 2. až 6. stĺpci sú kvôli úspore miesta prenásobené číslom 10^3.

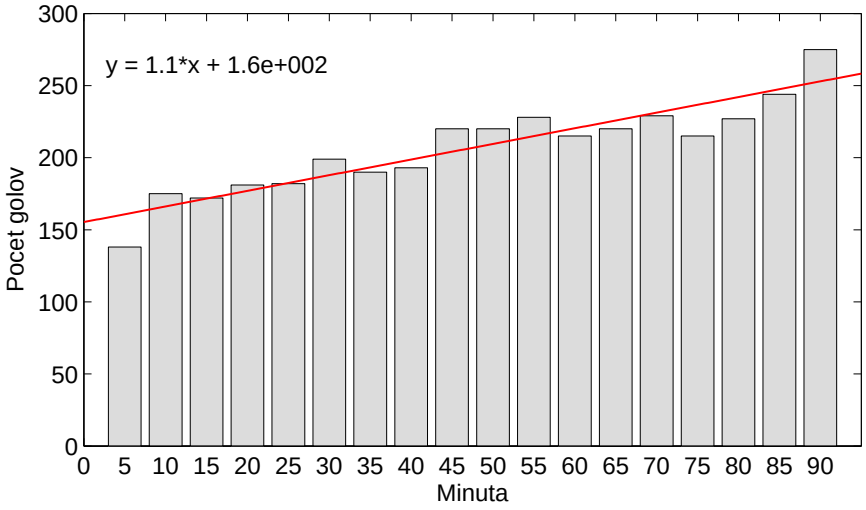
Postupne sa pokúsime potvrdiť či zamietnuť popísané efekty. V prípade potvrdenia budeme pribúdanie gólov v zápase interpretovať ako nehomogénny Poissonov proces, v ktorom sa parameter λ v závislosti na zistených faktoroch v čase mení. Takto budeme zápas chápať ako množinu po sebe idúcich časových intervalov, pričom v každom z nich už budeme uvažovať homogénny proces. Potom môžeme náš odvodený model (3.3) spustiť postupne pre každý časový interval zvlášť. Pritom predpokladáme, že vstupné parametre nového intervalu budú ovplyvnené výstupom z predchádzajúceho.

Závislosť počtu gólov na prebiehajúcom čase a skóre

Obr. 4.4 svojím trendom jasne naznačuje postupný nárast počtu gólov v závislosti na pribúdajúcom čase. Ako sme už naznačovali, núkajú sa prirodzene dva dôvody - úbytok síl hráčov a reakcia na aktuálne skóre.

Ako môžeme čo najjednoduchšie ukázať závislosť počtu gólov na čase, prípade ma aktuálnom skóre? Z dát môžeme zistiť intenzity gólov za daného skóre veľmi jednoducho, pretože vieme časy, ktoré medzi dvoma gólmi uplynuli. Zavedieme si model, aby sme sa vplyv času a skóre pokúsili preukázať na reálnych dátach.

Závislosť na skóre Nech T_{xy} je čas čakania na ďalší gól za aktuálneho skóre (x, y), $x, y = 0, 1, 2, \ldots$ a $I_{xy} \in \{0, 1\}$ indikuje, že gól bol ($I_{xy} = 1$) alebo nebol ($I_{xy} = 0$) dosiahnutý ešte pred koncom zápasu. Celý čas vychádzame z predpokladu, že sa časy medzi dvoma gólmi chovajú exponenciálne s intenz-



Obr. 4.4: Počty gólov v priebehu zápasu rozdelených v súčte do 5-minútových intervalov.

itou λ. Nech λ_{xy} predstavuje intenzitu skórovania za predpokladu, že aktuálne skóre zápasu je (x, y). Potom metódou maximálnej vierohodnosti môžeme λ_{xy} vypočítať ako podiel celkového počtu gólov a časov, ktoré do strelenia ďalšieho gólu uplynuli

$$\hat{\lambda}_{xy} = \frac{\sum_{i=1}^{N} I_{xy,i}}{\sum_{i=1}^{N} t_{xy,i}}, \tag{4.8}$$

kde $N = 1518$ je počet zápasov, $t_{xy,i}$ je čas čakania na ďalší gól a $I_{xy,i}$ je jeho príznak v i-tom zápase. Ak je v i-tom zápase aktuálne skóre (x, y) a je to zároveň aj konečný výsledok, potom $I_{xy,i} = 0$ a $t_{xy,i}$ je 90 mínus čas posledného gólu. Ak skóre (x, y) nikdy nenastane, $I_{xy,i}$ aj $t_{xy,i}$ je 0. Ak skóre (x, y) nenastalo ani v jednom zápase, nebudeme ho brať do úvahy. Tým predídeme deleniu nulou a navyše takéto skóre zjavne nie je bežné (k riešeniu tejto situácie sa neskôr vrátime). Tab. 4.11 zobrazuje intenzity skórovania za daného skóre (prepočítané na minútu zápasu). Aj keď počet gólov rastie v čase obecne, zdá sa byť evidentné, že skóre má na výsledok vplyv. Keď uvážime skóre $(2, 0)$, $(0, 2)$ a $(1, 1)$, dá sa očakávať, že v priemere bolo dosiahnuté v podobných časoch, no napriek tomu je intenzita značne odlišná.

Vytvorme si jednoduchý model, v ktorom sa pokúsime zohľadniť zmienené pozorovania. Intenzity skórovania, ktoré sú pre celý zápas konštantné, označme β_D a β_H pre domáci a hosťujúci tým. Tieto hodnoty získame pomo-

Skóre	Intenzita
λ_{00}	0,0250
λ_{10}	0,0289
λ_{01}	0,0293
λ_{11}	0,0302
λ_{20}	0,0353
λ_{02}	0,0315
λ_{21}	0,0327
λ_{12}	0,0369
λ_{22}	0,0372

Tabuľka 4.11: Odhadnuté intenzity (počet gólov za minútu) skórovania za aktuálne skóre (x, y).

cou odvodeného regresného modelu (4.7). Pre zavedenie časovej závislosti si zadefinujeme rozšírenie $\beta_D(t)$ a $\beta_H(t)$, kde $t \in [0, 1]$ predstavuje škálu celého zápasu. Označme $\beta_{D,xy}$ a $\beta_{H,xy}$ ako multiplikátory (podiely z intenzity), ktoré ovplyvňujú intenzity skórovania počas aktuálneho skóre (x, y). V bezgólovom zápase to znamená, že z hľadiska vplyvu skóre bude intenzita konštantná počas celého zápasu, tj. multiplikátor $\beta_{D,00} = \beta_{H,00} = 1$. Nakoniec si ešte zadefinujme pomocnú funkciu $\varphi(t)$, ktorá vráti aktuálne skóre v prebiehajúcom čase t, ako dvojicu $xy \in \{00, 01, 10, \ldots\}$. Potom si závislosť na aktuálnom skóre v čase t vyjadrime ako

$$\begin{aligned} \beta_D(t) &= \beta_{D,\varphi(t)}\beta_D \\ \beta_H(t) &= \beta_{H,\varphi(t)}\beta_H. \end{aligned}$$

V dobe medzi dvoma gólmi predpokladáme intenzity $\beta_D(t)$ a $\beta_H(t)$ konštantné, pretože multiplikátory $\beta_{D,\varphi(t)}$ a $\beta_{D,\varphi(t)}$ sa zmenia jedine pri vstrelení ďalšieho gólu. Napríklad, ak medzi 63. a 71. minútou zápasu sa aktuálne skóre $(1, 2)$ nezmenilo, tak $\varphi(63) = \varphi(64) = \ldots = \varphi(71) = 12$, čo odpovedá multiplikátorom $\beta_{D,12}$ a $\beta_{H,12}$.

Analýzou dát nebolo zistené, že by multiplikátory $\beta_{D,xy}$ a $\beta_{H,xy}$ boli v rámci rôznych výkonnostných kategórií evidentne odlišné a preto ich budeme počas aktuálneho skóre (x, y) pri simuláciách všetkých zápasov považovať za konštantné. Vráťme sa k riešeniu situácie, keď skóre (x, y) nie je bežné. Nebudeme totiž odhadovať všetky kombinácie. Keďže výsledkov s vyšším skóre je pomerne menej (alebo vôbec neexistujú), obmedzíme sa iba

na základnú množinu, v ktorej vyššie hodnoty zapíšeme pomocou rozdielov

$$\beta_{D,xy} = \begin{cases} \beta_{D,00} & x = 0, y = 0 \\ \beta_{D,10} & x = 1, y = 0 \\ \beta_{D,01} & x = 0, y = 1 \\ \beta_{D,11} & x = 1, y = 1 \\ \beta_{D,21} & x - y \geq 1, x \geq 2 \\ \beta_{D,12} & x - y \leq -1, y \geq 2 \\ \beta_{D,22} & x - y = 0, x, y \geq 2, \end{cases}$$

pričom multiplikátor $\beta_{D,xy} = \frac{\hat{\lambda}_{D,xy}}{\hat{\beta}_D}$, kde hodnotu $\hat{\lambda}_{D,xy}$ získame pomocou rovnice (4.8) obmedzenej iba pre domáce týmy. Pre hosťujúce týmy je celé odvodenie analogické. Tab. 4.12 obsahuje hodnoty multiplikátorov.

Multiplikátor	Domáci	Hostia
β_{10}	0,89	0,80
β_{10}	0,83	1,35
β_{01}	1,10	1,03
β_{11}	0,95	0,98
β_{21}	1,05	1,20
β_{12}	1,15	0,96
β_{22}	1,03	1,15

Tabuľka 4.12: Odhadnuté multiplikátory v závislosti na aktuálnom skóre.

Závislosť na čase Teraz si rozoberme situáciu z pohľadu času. Uvažujme spojitú závislosť intenzity skórovania na prebiehajúcom čase, ale nie na skóre. To môžeme interpretovať ako klasický nehomogénny Poissonov proces. Vzhľadom na trendovú krivku z Obr. 4.4 budeme túto závislosť modelovať lineárne. Preto definujme

$$\begin{aligned} \beta_D^*(t) &= \beta_D + \xi_D(t) \\ \beta_H^*(t) &= \beta_H + \xi_H(t) \end{aligned} \tag{4.9}$$

a tentokrát budeme konštantnú hodnotu intenzity predpokladať pre pevne stanovené časové (napr. 5 minútové) intervaly a nezávisle od aktuálneho stavu. V prvom polčase je intenzita skórovania oproti strednej hodnote nižšia a v druhom polčase naopak vyššia. Samozrejme by sme mohli uvažovať

aj akúkoľvek inú (kvadratickú, kubickú, ...) závislosť, ktorá má pozitívny charakter, ale z našich dát sme nenašli dôvody pre zamietnutie lineárnej závislosti. Vďaka zavedenému lineárnemu modelu (4.9) dostávame hodnoty koeficientov $\xi_D = 0,65$ a $\xi_H = 0,51$.

Zhrnutie Z odvodených multiplikátorov β_{xy} a koeficientovξ je vidieť, že predpoklad závislosti intenzity skórovania na čase a skóre bol správny. Nakoniec môžeme tieto odhady skombinovať do jediného modelu ako

$$\begin{aligned} \beta_D(t) &= \beta_{D,\varphi(t)}\beta_D + \xi_D(t) \\ \beta_H(t) &= \beta_{H,\varphi(t)}\beta_H + \xi_H(t), \end{aligned}$$

čím dostávame hodnoty prezentované v Tab. 4.13.

Multiplikátor	Domáci	Hostia
β_{10}	0,88	1,34
β_{01}	1,12	1,06
β_{11}	0,91	0,95
β_{21}	1,03	1,52
β_{12}	1,12	1,15
β_{22}	0,97	1,07
ξ	0,66	0,46

Tabuľka 4.13: Odhadnuté multiplikátory v závislosti na aktuálnom skóre a čase.

Teda úvaha, že intenzita skórovania v priebehu zápasu rastie, je správna. Okrem časového faktoru má svoj vplyv aj aktuálne skóre. Odvodené parametre intenzít využijeme pri simulácii priebehu zápasu.

Kapitola 5

Implementácia

V tejto kapitole sa zoznámime s frameworkami, ktoré sú technologickým základom našej aplikácie postavenej na platforme Java 6.0 a popíšeme si architektúru aplikácie spoločne so simulačným modelom.

5.1 Frameworky

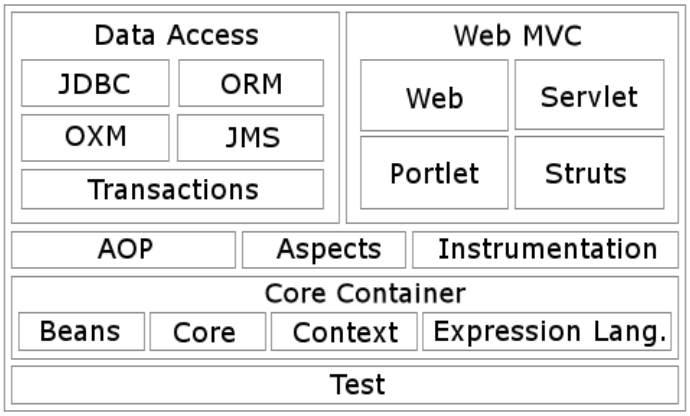
Technologický základ našej aplikácie tvoria Java frameworky *Spring 3.0* [18] a *Hibernate 3.5* [19]. Hibernate bol už popísaný v práci [1] a preto sa mu nebudeme detailnejšie venovať.

Nasledujúca sekcia je venovaná frameworku Spring, kde sa pokúsime zamerať iba na popis jeho hlavných prínosov a špecifických vlastností. Na konci si ukážeme, ako jednoducho do Springu zaintegrovať *Object-relational Mapping* (ORM) nástroj, akým je Hibernate a stručne porovnáme webový modul Springu s frameworkom *Stripes*, použitom v pôvodnej aplikácii.

Pre detailnejšie pochopenie frameworkov je vhodné siahnuť po referenčnej dokumentácii, ktorá je vo všetkých prípadoch veľmi kvalitná a zrozumiteľná.

5.1.1 Spring

Spring je v súčasnosti jeden z najkomplexnejších a najpoužívanejších viacvrstvových aplikačných frameworkov, ktorého hlavným cieľom je zjednodušiť ťažkopádnosť Java Enterprise Edition (JEE) [20]. Jeho hlavnými črtami sú prehľadnosť, neinvazívnosť, zameranie na architektúru a nie na technológiu, jednoduchá testovateľnosť a modulárnosť. Inými slovami nám Spring umožňuje sa sústrediť na architektúru a nie na technologické detaily tým, že ponúka množstvo abstrakcií nad JEE technológiami ako Java



Obr. 5.1: Hierarchia modulov v Springu.

Message Service (JMS), Java Database Connection (JDBC), Java Management Extensions (JMX), Java Naming and Directory Interface (JNDI) a iné. Na tomto mieste je nutné upozorniť, že Spring nie je náhradou JEE, ale iba odľahčeným JEE kontajnerom.

Spring sa drží niekoľkých zásad, ktorými sa v súlade s modernými princípmi a potrebami výrazne zjednodušuje vývoj. Typicky sa jedná o minimálnu závislosť na knižniciach tretích strán, *Plain Old Java Object* (POJO) prístup, tzv. *Open/Closed princíp*, čo v zmysle OOP znamená „otvorený pre rozšírenie, uzavretý pre modifikáciu" a v neposlednej rade tzv. *Convention over configuration*, čo predstavuje dizajnové paradigma v snahe oslobodiť vývojára od konfigurácie rutinných operácií v aplikácii.

Hlavnými stavebnými piliermi je návrhový vzor *Inversion of Control* a aspektovo orientovaný prístup. Tie si za chvíľu detailnejšie popíšeme. Keďže naša aplikácia vyžaduje prístup k databáze, zoznámime sa s transakčnou podporou Springu a so spôsobom zaintegrovania ORM frameworku, konkrétne Hibernate.

Celý framework je od verzie 3.0 rozdelený asi do 20 samostatných modulov v zmysle „použi, čo potrebuješ". Obr. 5.1 prehľadne popisuje ich hierarchiu. Fundamentálnu časť Springu tvorí Core kontajner, ktorý má na starosti manažment registrovaných tried (tzv. *Java beans*). Typicky používané sú moduly ORM a Test, v prípade webovej aplikácie modul MVC (Model-View-Controller).

Inversion of control a Dependency Injection

V klasickom modely programovania vytvárame nejakú triedu, ktorá potom využíva ďalšiu triedu, atď. Tým sú triedy pevne zviazané a zmena závislosti na inej triede (implementácii) vyžaduje zásah do kódu.

Inversion of Control (IoC) je všeobecný návrhový vzor, ktorý odstraňuje tesné väzby medzi objektami. Funguje na princípe presunutia zodpovednosti za vytvorenie a previazanie objektov z aplikácie na framework. IoC je však príliš abstraktný a často krát nie je jasný spôsob jeho implementácie. V princípe však chceme dosiahnuť, aby pri zmene implementácie nebolo nutné zasahovať do kódu, pretože konfigurácia je nastavená zvonku (napr. v XML).

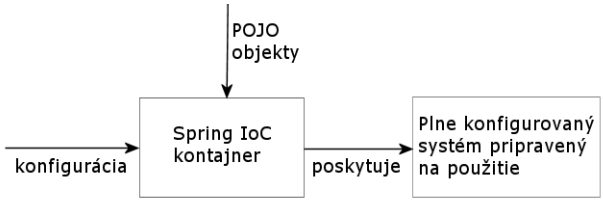
Dependency Injection (DI) je špeciálnym prípadom IoC, resp. predstavuje konkrétnu techniku jeho využitia. Nerieši teda len „čo", ale hlavne „ako". Cieľom DI je vlastný spôsob vloženia závislostí. Základné tri sú:

- *Setter Injection* - vkladanie prostredníctvom `set` metód (framework Spring)
- *Constructor Injection* - vkladanie prostredníctvom konštruktora (frameworky PicoContainer, Google Guice)
- *Interface Injection* - vkladanie prostredníctvom rozhrania (najmenej používaný spôsob; framework Avalon)

Objekty (beans) sú typicky vytvorené na základe XML alebo anotačnej konfigurácie, ktorá tieto objekty a závislosti medzi nimi definuje. Springom spravované objekty nazývame komponenty.

Spring je v podstate postavený na využití IoC a označovaný ako IoC kontajner, ktorý je implementovaný sadou modulou v rámci Core kontajnera a má pod správou vytváranie a poskytovanie objektov. Základom je sofistikovaná továrna trieda `BeanFactory`. Pre vkladanie závislostí využíva DI prostredníctvom konštruktoru a `set` metód. Typická je závislosť na rozhraní a nie na implementácii. Základný mechanizmus IoC znázorňuje Obr. 5.2.

V našom projekte uprednostníme konfiguráciu anotáciami, ktorá je oproti XML podstatne stručnejšia (vyžaduje Javu vo verzii 5+). Jej nevýhoda spočíva v tom, že do kódu v prípade zmeny zasiahnuť musíme. Nie však do implementácie, ale iba konfiguračnej anotácie. Minimálnu XML konfiguráciu však vytvoriť predsa len musíme, aby sme zadefinovali, ktoré triedy majú byť IoC kontajnerom načítané.



Obr. 5.2: Spring IoC kontajner

Pre znázornenie si uveďme jednoduchý príklad. Majme triedu `GameLauncher`, ktorá spúšťa simulácie nejakej hry. Ďalej majme implementáciu simulácie futbalu `SoccerSimulation`, ktorú chceme triede `GameLauncher` dodať. „Naivným" prístupom môžeme pri inštancovaní triedy `GameLauncher` v jej konštruktore vytvoriť objekt `SoccerSimulation`, čím vzniká tesná väzba.

Lepší prístup je naimplementovaný priamo v aplikácii, v balíčku `com.thesis.soccer.sim`. Spočíva vo vytvorení rozhrania `IGameSimulation`, ktoré bude v prípade futbalu implementovať trieda `SoccerSimulationImpl`. Túto závislosť rozhraním by sme mohli vložiť prostredníctvom `set` metódy alebo konštruktora do objektu `GameLauncher` manuálne, pretože my sme zodpovední za tieto objekty. Pomocou Springu však túto zodpovednosť delegujeme na IoC kontajner. Postup je veľmi jednoduchý. Triedy `GameLauncher` a `SoccerSimulationImpl` označíme anotáciou `@Service` alebo `@Component(name)`, kde atribút `name` je registrovaný názov komponenty. Na mieste, kde chceme závislosť „vstrieknuť", bude rozhranie `IGameSimulation` s anotáciou `@Autowired`. Keďže v tomto momente máme jedinú implementáciu rozhrania `IGameSimulation`, jeho implementácia `SoccerSimulationImpl` bude kontajnerom inštancovaná a vložená.

Ak je implementácii viac, k anotácii `@Autowired` je nutné pridať ešte anotáciu `@Qualifier` so zaregistrovaným názvom implementácie. Spring tak sám inštancuje objekty a vloží závislosti za nás. Každý vytvorený objekt má tzv. *scope*, ktorým určujeme, či je to *singleton* (jedináčik) alebo *prototype* (prototyp). Prednastavený scope je singleton.

V oblasti správy objektov je Spring naozaj mocným nástrojom a pre hlbšie pochopenie odporúčame referenčnú dokumentáciu.

Aspektovo orientovaný prístup

Aspektovo orientované programovanie (AOP) predstavuje evolúciu v modernom programovaní. Je potreba upozorniť, že jeho názov nemá nič spoločné s objektovo orietovaným programovaním. Motivácií pre jeho využitie je niekoľko. Niektoré rutinné operácie prítomné naprieč celou aplikáciou by sa mohli vyseparovať a aplikovať globálne. Bežné metodológie sa snažia pomocou metód, tried či balíčkov o oddelenie a zapúzdrenie operácií do samostatných entít. Niektoré oblasti sa však nedajú jednoducho oddeliť – tzv. prierezové koncerny (*cross-cutting concerns*). Patria sem napr. logovanie, autorizácia a transakcie. Refactoring kódu vykonávajúceho takéto operácie zvyčajne znamená rozsiahly zásah. Aspekty pomáhajú tento problém vyriešiť.

Vysvetlíme si základné pojmy, ktoré potrebujeme k pochopeniu AOP.

- *JoinPoint* je jeden prípojný bod v programe (spustenie metódy, volanie metódy, atď.).
- *PointCut* je množina prípojných bodov získaná aplikovaním špeciálneho regulárneho jazyka *Pointcut Language* (PL).
- *Advice* je metóda, ktorá sa zachycuje na PointCut a implementuje dodatočné chovanie. Základné typy sú *before*, *after* a *around*.
- *Aspekt* zapúzdruje prierezové koncerny množinou prípojných bodov (PointCut) a aplikuje dodatočné chovania (Advices).

Vďaka týmto konštruktom dokážeme napríklad aplikovať logovanie na všetky `get` metódy. Každá metóda predstavuje záchytný bod (JointPoint). Množinu bodov (PointCut) popíšeme pomocou PL a na výsledok zachytíme metódu (Advice) implementujúcu skutočné logovanie. Tá sa môže vykonať pred vykonaním `get` metódy (before) alebo po nej (after). Samotná `get` metóda nič neimplementuje a „o ničom netuší". Najznámejšou implementáciou AOP je knižnica *AspectJ*, ktorá obsahuje vlastný prekladač a dokáže tak priamo meniť skompilovaný Javovský byte-code.

V Springu patria aspekty k základným stavebným kameňom a jeho najsilnejším vlastnostiam. Ich podpora je implementovaná modulom Spring AOP. Aspekty dokážeme aplikovať na akýkoľvek POJO objekt. Názvy anotácii v Springu sú pre zjednodušenie prebrané z knižnice AspectJ, ale je tu však jeden zásadný rozdiel. Spring žiadnym spôsobom nemení skompilovaný kód, ale aspekty implementuje programovo pomocou *proxy* objektov - zástupcov. To znamená, že každý Springom spravovaný objekt, na ktorý je aplikovaný aspekt, zastupuje proxy objekt, ktorý

- pred skutočným vyvolaním metódy vykoná príslušný *before advice* a
- po skutočnom vyvolaní metódy vykoná príslušný *after advice*.

Ak požiadame IoC kontajner o nejakú implementáciu rozhrania `A`, ktorú obaľuje proxy objekt, nevráti nám skutočnú implementáciu, ale práve tento obaľujúci proxy objekt, ktorý implementuje to isté rozhranie `A`.

Programovo riešený aspektovo orientovaný prístup samozrejme prináša niekoľko obmedzení. Tým prvým je, že existuje iba jediný prípojný bod v programe, a to spustenie metódy. Druhým zásadným obmedzením je, že aspekty môžeme aplikovať iba na `public` metódy. Metódy s inými modifikátormi viditeľnosti nie sú z proxy objektu dostupné.

Spring rozširuje množinu prípojných bodov o ďalšie proprietárne (napr. anotácie), ktoré je možné dohľadať v dokumentácii.

Transakčný manažment

Databázový prístup patrí k bežným požiadavkám aplikácií. Aby sme však dáta získali, je nutné vykonať niekoľko rutinných operácií: vytvorenie spojenia s databázou, spustenie transakcie pred položením dotazu, commit/rollback a ukončenie spojenia po získaní dát. Neustále opakovanie rutinných operácii je dôvodom k využitiu princípov, ktoré sme popísali v predchádzajúcej sekcii.

Transakčný manažment v Springu je implementovaný práve pomocou aspektov. Vďaka tomu dokážeme veľmi jednoducho nakonfigurovať transakčné chovanie pre jednotlivé metódy pomocu XML alebo anotácie `@Transactional`. Vyvolaniu transakčnej metódy vždy predchádza vytvorenie pripojenia do databázy a naštartovanie transakcie (before advice). Na konci dokážeme reagovať na výnimky podľa potreby. Ak bola transakčná metóda nakonfigurovaná iba na čítanie, Spring nedovolí uloženie zmenených dát (after advice).

Opäť však musíme počítať s niekoľkými obmedzeniami. Predstavme si, že máme dve metódy s anotáciou `@Transactional`. Tieto metódy samy o sebe netušia o transakčnom chovaní nič. Toto chovanie preberá ich obaľujúci proxy objekt. Ak však z jednej metódy v rámci našej implementácie vyvoláme druhú, tak tá sa nespustí v samostatnej transakcii, ale bude súčasťou existujúcej. Je to z toho dôvodu, že vnútorné volanie sa nevykonáva zvonku prostredníctvom proxy objektu. Tento problém je všeobecne známy a je potrebné na to myslieť už v dobe návrhu.

Spring a Hibernate

Hibernate patrí do rodiny ORM frameworkov. Ich myšlienka spočíva v mapovaní objektov na relačnú databázu. V objektovo orientovaných jazykoch tak každej tabuľke odpovedá tzv. entitná trieda a relačným vzťahom na základe cudzích kľúčov odpovedajú kolekcie v rámci entitných tried. Pri dotazovaní sa nepoužíva štandardný jazyk SQL, ale objektový jazyk, ktorý je mu podobný. V Hibernate je to tzv. Hibernate Query Language (HQL).

Hibernate má množstvo špecifických vlastností ako *cache prvej a druhej úrovne*, *oneskorená inicializácia (lazy loading)*, *dávkové operácie (batch fetching)*, *optimistické zamykanie (optimistic locking)* a *automatická kontrola zmien (dirty checking)*. V prípade hlbšieho záujmu odkazujeme čitateľa na knihu od samotného autora Hibernate [19].

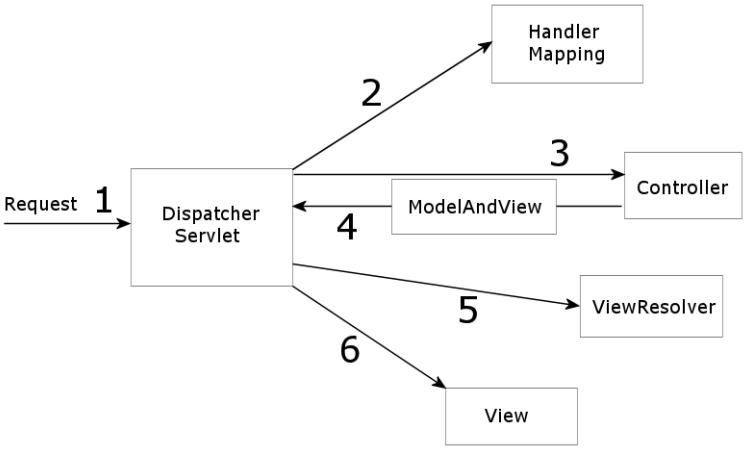
Spring ORM je modul poskytujúci integračnú vrstvu pre populárne API objektovo-relačného mapovania v Springu. Jeho cieľom je abstrahovať ORM technológie ako Hibernate, JDO či iBatis, čo teoreticky umožňuje neskoršiu zmenu frameworku bez zásahu do aplikačnej logiky. Spring to rieši napríklad tak, že poskytuje vlastnú hierarchiu výnimiek, ktoré na integračnej vrstve automaticky prekladá a teda konkrétna technológia nie je dôležitá. Takto integrovaný ORM framework môžeme používať so všetkými ostatnými funkciami, ktoré Spring ponúka, vrátane už popisovaného transakčného manažmentu, session manažmentu a testovania. Prístup k Hibernate API nie je nutný, pretože Spring pre tieto účely poskytuje DAO vrstvu s rodičovskou triedou `HibernateDaoSupport` a rozhraním `HibernateTemplate`, ktoré spoločne veškeré API zapúzdrujú.

Pre pripojenie do databázy a integráciu Hibernate je nutná XML konfigurácia triedy `AnnotationSessionFactoryBean`, ktorú poskytuje Springu ORM.

Spring MVC

Spring MVC aj Stripes patria do rodiny tzv. akčných frameworkov vychádzajúcich z návrhového vzoru MVC. Tým oddeľujú výkonnú časť od grafickej podoby. V tejto chvíli sa týmto vzorom nebudeme ďalej zaoberať. Bližšie popísaný ho nájdeme v práci [1].

Stripes síce patrí svojími minimálnymi nárokmi na pochopenie a svojím prínosom k najlepším webovým frameworkom a dokonca existuje možnosť jeho integrovania do Springu, ale tento krok je kontraproduktívny vzľadom



Obr. 5.3: Životný cyklus vlákna v Spring MVC.

na fakt, že Spring poskytuje vlastný MVC modul, ktorý obsahuje všetky prezentované vlastnosti Springu. Nechýba podpora lokalizácie, správy adries, validácia príchodzích dát a serializácia do *JavasScript Object Notation* (JSON) formátu .

Základom webovej aplikácie postavenej na platforme Java je servlet. Sám o sebe však poskytuje len minimálnu funkcionalitu a vedie k ťažkopádnemu a tzv. špagetovému kódu. Spring MVC poskytuje aplikačný rámec pre vývoj webových aplikácií a vytvára abstrakciu nad HTTP protokolom. Hlavným stavebným prvkom je objekt `DispatcherServlet`, ktorý je implementáciou návrhového vzoru *Front Controller* a predstavuje vstupnú bránu do aplikácie. Na Obr. 5.3 môžeme vidieť základný životný cyklus požiadavku od klienta na strane serveru. Každá URL adresa je obslúžená metódou v nejakom `Controller` objekte. O tom, ktorá metóda to bude, rozhoduje objekt `DispatcherServlet` prostredníctvom objektu `HandlerMapping` na základe predkonfigurovaných mapovaní. `Controller` si vyžiada od aplikačnej vrstvy požadovaný model, ktorý je príslušným `ViewResolverom` spracovaný a vrátený klientovi v požadovanom `View` formáte.

Každá trieda s anotáciou `@Controller` sa automaticky nainštancuje do kontrolnej vrstvy a je pod správou Springu. `Controller` objektov je v aplikácii typicky väčší počet a každý z nich zodpovedá za jednu operáciu, či sadu logických operácií. Každý `Controller` je z princípu singleton, takže je nutné dbať na *thread-safe* operácie. Každá metóda v rámci tohto objektu s anotáciou `@RequestMapping(value)` môže obslúžiť `GET/POST` požiadavku od klienta. Podmienkou je, aby sa vzor zdrojovej adresy (URI), relatívny

ku kontextu, zhodoval s atribútom `value` v anotácii, ktorý má formu regulárneho výrazu.

Spring MVC je v súčasnosti mocným nástrojom medzi akčnými frameworkami a ponúka širokú podporu pre jednoduchú implementáciu webových aplikácií. Jeho použitie prispieva k prehľadnosti aplikácie postavenej na Springu. Pridanou hodnotou je slobodná voľba pri výbere prezentačnej technológie.

5.2 Architektúra aplikácie

Aplikácia sa skladá z troch základných vrstiev - prezentačnej (view), kontrolnej (controller) a aplikačnej (model). Dáta medzi jednotlivými vrstvami sú prenášané prostredníctvom transfer objektov (TO). Definujeme dva typy transfer objektov - vstupný (input transfer objekt, označujeme ITO) a výstupný (output transfer objekt, označujeme OTO). Každý transfer objekt má v názve postfix na základe typu (napr. `MatchResultOTO`). Každý ITO objekt predstavuje štruktúru, ktorú očakáva aplikačná logika. Každý OTO objekt naopak predstavuje štruktúru, ktorú očakáva prezentačná vrstva. Jednotlivé vrstvy si teraz podrobnejšie popíšeme.

5.2.1 Vrstvy aplikácie

Prezentačná vrstva

Vzhľadom na fakt, že aplikácia je webová, je prezentačnou technológiou XHTML a JavaScript. Táto vrstva je taktiež plne objektová s pokročilými užívateľskými komponentami javascriptového frameworku *ExtJS*, ktoré sú typické v desktopových aplikáciach. To dovoľuje lepšiu interakciu a kontrolu na strane klienta, čo minimalizuje počet požiadavkov na server. Komunikácia so serverom prebieha v prevažnej väčšine asynchrónne v JSON formáte, čo je natívny (textový) zápis objektov v JavaScripte. Na strane servera sa JSON transformuje do ITO objektov.

Kontrolná vrstva

Túto vrstvu predstavuje sada `Controller` objektov, ktoré obsluhujú jednotlivé klientské požiadavky. Jej úlohou je deserializovať JSON formát do Javovských elementárnych dátových typov alebo ITO objektov, validovať vstupné parametre, vyvolať príslušnú aplikačnú logiku a odoslať odpoveď

na klienta. Podľa formátu odpovede rozlišujeme dva typy `Controller` objektov - `DataController` odpovedá asynchrónne vo formáte JSON a `ViewController` synchrónne vo formáte XHTML. Každý `Controller` má v názve postfix podľa svojho typu (napr. `SoccerDataController`). Ničmenej, táto vrstva by mala implementovať rozhodovaciu logiku aplikácie, tzn. má rozhodnúť, aký model sa pošle klientovi v odpovedi. Základnú hierarchiu `Controller` tried popisuje Obr. A.2 v prílohe.

Aplikačná (business) vrstva

Úlohou tejto vrstvy je poskytnúť požadovaný dátový model. Rozhranie medzi kontrolnou a aplikačnou vrstvou je postavené na návrhovom vzore *fasáda*. Každá metóda objektu `Facade` vyvolaná z objektu `Controller` dostane na vstupe množinu parametrov, na základe ktorých zostaví dotaz do databázy v jazyku HQL. Výsledok dotazu tvoria kolekcie doménových objektov, resp. entity objektov, ktoré sa následne pretransformujú do OTO objektov a posielajú naspäť na kontrolnú vrstvu. Každý fasádový objekt implementuje rozhranie a v názve má postfix `Facade` (napr. `SoccerFacade` implementuje `ISoccerFacade`). Základnú hierarchiu fasádových tried popisuje Obr. A.3 v prílohe.

5.2.2 Štruktúra aplikácie

Adresáre a balíčky v projekte

Keďže sa jedná o webovú aplikáciu, projekt je zabalený v adresári `war` (web archive). Ten má nasledujúcu štruktúru:

- `META-INF` - obsahuje základné informácie o archíve
- `static` - statické zdroje pre prehliadač (javascript, štýly a obrázky)
- `WEB-INF`
  - `classes` - skompilované Javovské triedy
  - `config` - konfiguračné súbory Springu (web, aplikačná vrstva, databáza)
  - `data` - dáta v CSV formáte pre inicializáciu databázy
  - `jsp` - prezentačné JSP súbory
  - `lib` - knižnice tretích strán potrebné pre beh aplikácie

  - `logs` - súbory pre logovanie aplikácie

Jednotlivé triedy sa nachádzajú v nasledujúcej balíčkovej štruktúre:

- `com.thesis.soccer`
  - `bl` - fasádové objekty
  - `bl.sim` - simulačný algoritmus
  - `bl.to` - input/output transfer objekty (ITO/OTO)
  - `pl.aspect` - aspekty zapúzdrujúce rutinné operácie
  - `pl.controller.*` - sada `Controller` objektov
  - `pl.json` - podporná vrstva pre konvertovanie JSON formátu do Javy a naopak
  - `dao.*` - pripojenie do databázy a dátový model aplikácie definovaný pomocou entity objektov v Hibernate
  - `test` - testovacie triedy

Uživateľské rozhranie

Samotná aplikácia je jednoduchá a poskytuje dva základné pohľady - tabuľku s výsledkami reálnych zápasov, ktoré sme použili v našom modely a možnosť spustenia simulácie zápasu pre vybrané dva týmy. Obe možnosti sú dostupné v menu na lište, ktorá sa nachádza v hornej časti obrazovky.

5.2.3 Simulačný modul

Trieda `SoccerSimulationImpl` s externým aplikačným rozhraním `IGameSimulation` implementuje navrhnutý štatistický model (viď Obr. A.1 v prílohe). Vstupné parametre predstavujú vstupné objekty `MatchTeamDataITO`, ktoré obsahujú regresory pre odhady parametrov oboch tímov. Výstupnou dátovou štruktúrou je objekt `MatchResultOTO` s výsledkom zápasu.

Simuláciu zápasu popisuje pseudo-algoritmus 5.1. Na začiatku je potrebné z dostupných dát získať vysvetľujúce premenné a vypočítať regresné odhady počtu gólov a prihrávok zo vzťahov (4.4), (4.5) a (4.6). Ďalším dôležitým krokom je vytvorenie matice intenzít prechodov medzi jednotlivými stavmi (sektormi). Tzn., ako často za nami stanovenú časovú jednotku prejde lopta z jedného stavu do ďalšieho. Samotná simulácia zápasu prebieha v niekoľkých

Algoritmus 5.1 Simulácia zápasu medzi týmami D a H.

získaj množinu regresorov pre oba týmy;
odhadni počet gólov pomocou (4.4);
odhadni počet úsp. prihrávok pomocou (4.5);
odhadni počet neúsp. prihrávok pomocou (4.6);
urči čas držania lopty pomocou (4.1);
inicializuj dĺžku iterácie v minútach;
vytvor maticu počtu prihrávok a gólov medzi jednotlivými stavmi;
inicializuj skóre na 0:0;
WHILE počet iterácií * dĺžka iterácie $<$ 90 **DO**
  urči stav, v ktorom prebieha hra;
  FOR `tým` **IN** {`D`,`H`} **DO**
    zostav matice intenzít `A` a `B` pre `tým` podľa (3.3);
    urči pravdepodobnosti počtu gólov v jednej iterácii (3.4);
    vyber náhodne počet gólov a uprav skóre;
    uprav odhad počtu gólov v závislosti na čase a skóre;
  END
  inkrementuj počet iterácií;
END
vypíš skóre;

iteráciách. Ich počet sa rovná podielu dĺžky zápasu (90 minút) a stanovenej časovej jednotky. V každej iterácii vychádzame zo stavu, v ktorom sa hra aktuálne nachádza a pre každý tým vypočítame pravdepodobnosti pre všetky možné počty gólov $(0, 1, 2, \ldots)$. Náhodne vyberieme počet strelených gólov, upravíme intenzitu skórovania v závislosti na aktuálnom čase a skóre a prejdeme do ďalšej iterácie.

Zaujímavá je metóda `calculateProbability(...)` v triede `ProbabilitySolver`, ktorá implementuje zložitejšie maticové operácie a vracia pravdepodobnosti na základe vzťahu (3.4).

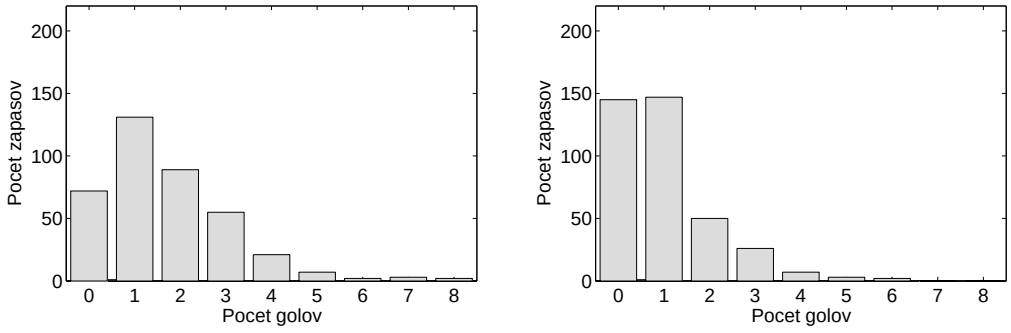
Kapitola 6

Výsledky

V tejto kapitole si zhrnieme dosiahnuté výsledky simulácií na základe nami navrhnutého modelu z kapitoly 3 a jej implementácie popísanej v kapitole 5. Keďže výstupom simulácie je predpoveď, nedokážeme jednoducho určiť jednoznačnú kvalitatívnu metriku, a preto pre zhodnotenie použijeme viacero štatistických ukazovateľov, ktoré simuláciami získame.

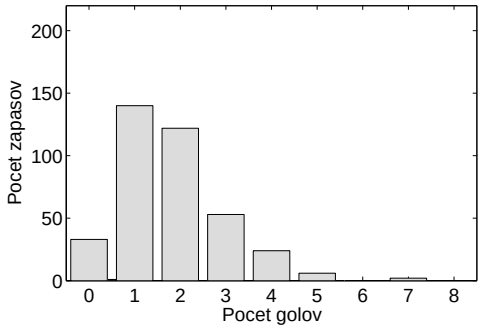
V rámci porovnávacej analýzy zvolíme nasledujúci postup. Na základe empirických dát z prvých troch sezón odhadneme vstupné parametre modelu a budeme simulovať priebeh štvrtej sezóny, ktorej skutočné výsledky máme taktiež k dispozícii. V jednej sezóne sa odohrá 380 zápasov, čo môžeme považovať za dostatočný rozsah pre objektívne tvrdenia o vlastnostiach modelu. Simulovať sezónu budeme dvakrát. Najskôr využijeme Markovov reťazec s diskrétnym časom (MRsD), navrhnutý a implementovaný v práci [1]. Vstupné parametre tohto modelu sú odhadnuté pomocou aritmetického priemeru a zohľadňujú iba počty gólov. Druhýkrát použijeme náš robustnejší Markovov model so spojitým časom (MMsS), ktorý inicializujeme regresnými odhadmi počtu gólov a prihrávok zohľadňujúce silu súpera. Rôzne štatistické ukazovatele získané z oboch simulácií porovnáme so skutočnými dátami.

Na Obr. 6.1 vidíme rozdelenie počtu gólov v reálnej sezóne a simulovanej sezóne pomocou implementácií MRsD a MMsS. Simulácia MRsD oproti realite evidentne predpovedá príliš veľa zápasov, v ktorých tým strelí jeden gól, čím znehodnocuje výhodu domáceho prostredia a kvality súperiacich tímov. Tým sa potieraju kvalitatívne rozdiely medzi týmami, čo sme už naznačovali v úvode práce.

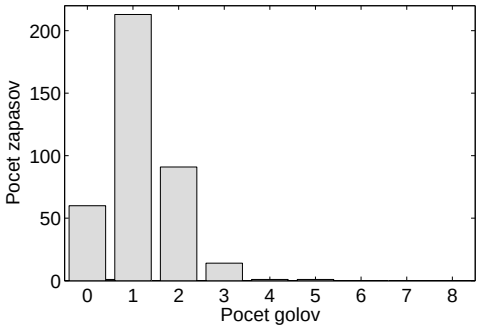


(a) Góly domácich v realite

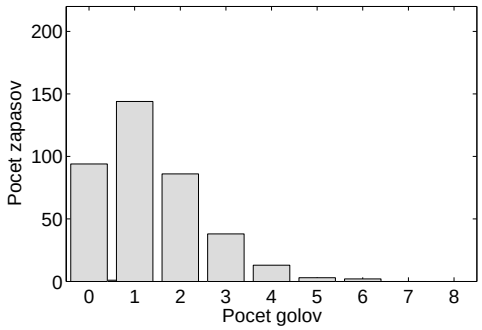
(b) Góly hostí v realite



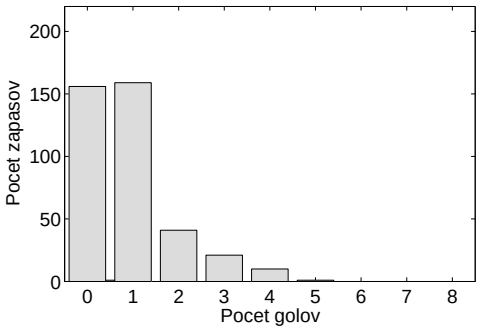
(c) Góly domácich v simulácii MRsD



(d) Góly hostí v simulácii MRsD



(e) Góly domácich v simulácii MMsS



(f) Góly hostí v simulácii MMsS

Obr. 6.1: Rozdelenie počtu gólov domácich a hostí v reálnych zápasoch, simulácii MRsD a simulácii MMsS.

V úvode sme písali, že modely pre predpovede zápasov sú cieľom výskumu hlavne na trhu stávok. V Tab. 6.1 môžeme vidieť zaujímavé ukazovatele, ktoré porovnávajú odhady simulácii MRsD a MMsS. Tieto ukazovatele neboli vybrané náhodne, ale práve na základe bežných stávkových príležitostí. Lepšia

Ukazovateľ	Počet záp.	Simulácia MRsD	Simulácia MMsS
tip na víťaza	380	123 (32,4%)	**191** (50,3%)
dvojtip	253	159 (62,8%)	**203** (80,2%)
výsledok	380	15 (3,94%)	**33** (8,68%)
počet gólov v zápase	380	57 (15,0%)	**70** (18,4%)
počet gólov domácich	380	61 (16,1%)	**84** (22,1%)
počet gólov hostí	380	102 (26,8%)	**135** (35,5%)

Tabuľka 6.1: Porovnanie správnych odhadov štatistických ukazovateľov na základe simulácií pomocou MRsD a MMsS.

hodnota z oboch simulácií je v každom riadku zvýraznená tučne a v zátvorke je percentuálne vyjadrenie voči počtu odohratých zápasov. Popíšme si, čo predstavujú jednotlivé stĺpce. Odhad *tipu na víťaza* znamená, že sme uhádli výhru domácich, remízu alebo výhru hostí bez ohľadu na presnosť výsledku. Odhad *dvojtipu* znamená, že ak náš model predpovedal výhru domácich alebo hostí, tak pre správny tip nám stačí aj remíza. Nezahrňujeme do neho predpovede remíz. Dvojtip teda neuhádneme jedine v prípade, ak vyhral opačný tým, ako sme predpovedali. Odhad *výsledku* je úplne presný výsledok. Pre *počet gólov v zápase*, *počet gólov domácich* a *počet gólov hostí* postačuje presný odhad opäť bez ohľadu na celkový výsledok.

V každom z ukazovateľov náš model vracia lepšie hodnoty, v niektorých dokonca výrazne. Nedostatky implementácie MRsD si zhrnieme v závere práce. Teraz sa zameriame už iba na výsledky nášho modelu a pokúsime sa ich rozumne intepretovať.

Posledné štyri riadky v Tab. 6.1 obsahujú odhady presných počtov. Tieto štatistiky na presnosť sú však veľmi citlivé, pretože o ich správnosti rozhodujú detaily. Ak pripustíme chybu odhadu ± 1 gól, okamžite sa dostávame na 2 až 3 násobne vyššiu úspešnosť.

Prvý riadok (tip) a druhý riadok (dvojtip) Tab. 6.1 predstavujú najtypickejšie stávkové tipy. Kvalitu modelu tak môžeme porovnávať aj z hľadiska ziskovosti na základe stanovených kurzov. Dokázali sme odhadnúť 191 zápasov správne, z toho 120 výhier domácich, 42 remíz a 29 výhier hostí. To znamená, že ak na každý zápas vložíme rovnakú čiastku, dokážeme byť zisková pri priemernom výhernom kurze väčšom ako $380/191$, čo predstavuje zhruba číslo $1,99$. V prípade dvojtipu sme správne odhadli 203 z 253 zápasov, ak sme nepočítali zápasy, v ktorých sme predpovedali remízy. S priemerným kurzom dvojtipu aspoň $1,25$ začína byť náš model ziskový.

Presný kurz každého zápasu v sezóne nemáme k dispozícii, preto

Poradie	Realita	Body	Simulácia MMsS	Body
1	Chelsea	86	Chelsea	89
2	Manchester Utd.	85	Manchester Utd.	88
3	Arsenal	75	**Liverpool**	81
4	Tottenham	70	Arsenal	69
5	Manchester City	67	Tottenham	64
6	Aston Villa	64	Manchester City	64
7	**Liverpool**	63	Aston Villa	63
8	Everton	61	Everton	54
9	**Birmingham**	50	Blackburn	48
10	Blackburn	50	Fulham	48
11	Stoke City	47	**Portsmouth**	47
12	Fulham	46	Stoke City	41
13	**Sunderland**	44	West Ham Utd.	39
14	Bolton	39	Bolton	37
15	Wolverhampton	38	Wigan Athletic	33
16	Wigan Athletic	36	Wolverhampton	33
17	West Ham Utd.	35	Hull City	31
18	Burnley	30	Burnley	30
19	Hull City	30	**Sunderland**	28
20	**Portsmouth**	19	**Birmingham**	26

Tabuľka 6.2: Konečné poradie tímov v reálnej a simulovanej sezóne pomocou MMsS.

použijeme priemerné kurzy na základe dvoch po sebe nasledujúcich kôl. Na víťazstvo domácich je to hodnota okolo $1,7$, na remízu $3,2$ a na výhru hostí $2,5$. Pre naše predpovede tak dostávame priemerný výherný kurz $(1,7 * 120 + 3,2 * 42 + 2,5 * 29)/191 = 2,151$. Pri dvojtipoch sa pohybujú kurzy okolo hodnoty $1,25$. To by mohlo naznačovať podobnosť výsledkov, na základe ktorých stanovujú svoje kurzy spoločnosti na trhu stávok.

Poznámka. Keďže nemáme k dispozícii presný kurz na každý zápas v sezóne, musíme počítať s rizikom, že správne odhadujeme zväčša zápasy s nižším kurzom, kde je náš odhad vysoko očakávaný. To by znížilo náš vypočítaný priemerný výherný kurz.

Keďže sme v prípade dvojtipu správne odhadli 203 z 253 zápasov, rozdiel 50 ($13,2\%$) zápasov sme predpovedali presne opačne. Zamerajme sa teraz na možné dôvody, prečo táto situácia nastala. Odpoveď naznačuje Tab. 6.2, kde vidíme výsledné poradie tímov po všetkých odohraných zá-

pasoch pri ohodnotení troch bodov za víťazstvo a jedného bodu za remízu. Väčšina týmov sa nachádza na podobných priečkach. Výnimku však tvoria tučne označené týmy Liverpool a Portsmouth, ktoré zaostali za očakávaním a týmy Birmingham a Sunderland, ktoré naopak očakávania predčili. Súčet rozdielu reálneho a odhadnutého počtu bodov pre tieto štyri týmy je 86, čo predstavuje zhruba 29 nesprávnych odhadov. Problém je v tom, že tieto týmy hrali v predchádzajúcich sezónach výrazne lepšie, resp. horšie. V našej simulácii určujeme vstupné parametre iba na základe predchádzajúcich sezón. Lenže pred novou sezónou môže výkon týmu ovplyvniť mnoho faktorov, ktoré nedokážeme predvídať.

Riešením tohto efektu je, aby sme do regresných odhadov parametrov priebežne zahrňovali reálne dáta, ktoré pre simulovanú sezónu máme. V tých sa totiž odzrkadľuje aktuálna forma týmu. Vplyv nových dát môžeme zvýšiť zavedením váh, tzn. takéto dáta majú vyššiu váhu.

Záver

V tejto práci sme sa zaoberali aplikáciou Markovovho procesu so spojitým časom pre modelovanie kolektívnych hier. V úvode sme si predstavili niekoľko existujúcich prístupov so zameraním na ich nedostatky a popísali spôsoby, akými dokážeme získať reálne dáta. Zaviedli sme teóriu potrebnú pre pochopenie Markovovho procesu so spojitým časom, predstavili sme si model zrodu a zániku ako špeciálny prípad Markovovho procesu so spojitým časom, Poissonov proces, vzťah Poissonovho rozdelenia k negatívne binomickému rozdeleniu a Poissonovu regresiu. Ukázali sme, akým spôsobom sa dá hra popísať pomocou stavov a prechodov. Proces skórovania sme intepretovali ako model zrodu a zániku bez zániku, ktorý za predpokladu exponenciálneho času medzi udalosťami môže byť intepretovaný ako homogénny Poissonov proces. Nakoniec sme si odvodili sústavu diferenciálnych rovníc pre určenie pravdepodobností počtu gólov v zápase. Na základe χ^2 testu dobrej zhody sme overili hypotézy o predpokladanom rozdelení počtu parametrov a z výsledkov vytvorili vhodné regresné modely. Navrhnutý simulačný model sme naimplementovali v rámci webovej aplikácie postavenej na platforme Java s technologickou podporou frameworkov Spring a Hibernate.

Spojitý čas Jedným z hlavných prínosov tejto práce je zavedenie spojitého času hry. Predpoklad diskrétneho času by totiž znamenal, že nás stavy procesu zaujímajú jedine v okamihoch, ktoré tvoria (potenciálne nekonečnú) rastúcu postupnosť. Medzi týmito okamihmi by sa v Markovovom reťazci nič nedialo. Takýto model bol navrhnutý a implementovaný v práci [1] a práve absencia spojitého času bola považovaná za výrazný nedostatok.

Rozdelenie a odhady dát Lepšie výsledky sme dosiahli podstatne detailnejšou analýzou rozdelenia dát namiesto jednoduchého odhadu aritmetickým priemerom a do modelu sme okrem gólov zakomponovali ďalší dôležitý kvalitatívny parameter - prihrávky. Pri úvahách o rozdelení sme

vychádzali z citovanej literatúry, ktorá poskytuje veľmi dobrý základ. Častokrát sa však pre počty gólov a prihrávok vychádza z predpokladu Poissonovho rozdelenia. My sme však túto hypotézu zamietli. Jednoduchým zavedením Poissonovho rozdelenia by sme si až príliš zjednodušovali realitu. Táto nepresná úvaha zrejme vychádza z nedostatočného počtu pozorovaní. Pri tak vysokom počte pozorovaní, ktoré máme k dispozícii, sme preukázali, že rozptyl výrazne prevyšuje strednú hodnotu. Poissonovo rozdelenie nie je pre tieto prípady vhodné a preto sme použili robustnejšie negatívne binomické rozdelenie. Zistili sme však, že ak týmy vhodne zaradíme do výkonnostných kategórií, dokážeme negatívne binomické rozdelenie počtu gólov interpretovať ako zmes Poissonových rozdelení s parametrom λ, ktorý pochádza z gamma rozdelenia. Z toho sme vyvodili záver, že príslušnosť do kategórie patrí k prirodzeným vysvetľujúcim faktorom pri určovaní charakteristík týmov. V prípade prihrávok je aj po zaradení do kategórií prevýšenie rozptylu natoľko evidentné, že Poissonovo rozdelenie neprichádza do úvahy. Oproti ostatným citovaným metódam sme navyše preukázali oprávnenosť domnienky, že intenzita skórovania je dynamicky závislá na aktuálnom skóre a čase zápasu. Z výsledkov simulácií celej sezóny môžeme usúdiť, že analýza o rozdelení dát a robustnejšie regresné odhady vstupných parametrov pomohli výrazne znížiť mieru náhody v úspešnosti predpovedí.

Dolovanie dát a simulácie Jednou z najcennejších súčastí tejto práce je rozsiahly súbor dát, bez ktorého by sme nedokázali model správne navrhnúť a otestovať. Ako tieto dáta získať, nie je úlohou štatistiky, ale informatiky. V rámci práce boli vytvorené parsovacie skripty, ktoré prehľadávali internetové zdroje a zhromažďovali dôležité informácie. Vďaka tomu sa nám podarilo získať niekoľko 100MB cenných dát, nad ktorými sme mohli testovať správnosť bežne prijímaných hypotéz. Samotný simulačný modul vychádza priamo z navrhnutého štatistického modelu, je však zasadený v kontexte viacvrstvej architektúry. S využitím návrhových vzorov a popísaných moderných implementačných techník (aspekty, kontajnerom manažované objekty, ...) sa nám vo výsledku podarilo vytvoriť plnohodnotnú jednoducho rozšíriteľnú aplikáciu.

Nápady na rozšírenie Model, ktorý sme si odvodili, samozrejme nie je ideálny. Pre jeho rozšírenie by bolo vhodné uvažovať zmeny počtu hráčov v zápase z dôvodu taktických či vynútených striedaní alebo z dôvodu červenej karty. Pomer hráčov na ihrisku môže výrazne ovplyvňovať výkon týmov. Je potreba však myslieť aj na zložitosť modelu, ktorá tým veľmi narastá, pretože

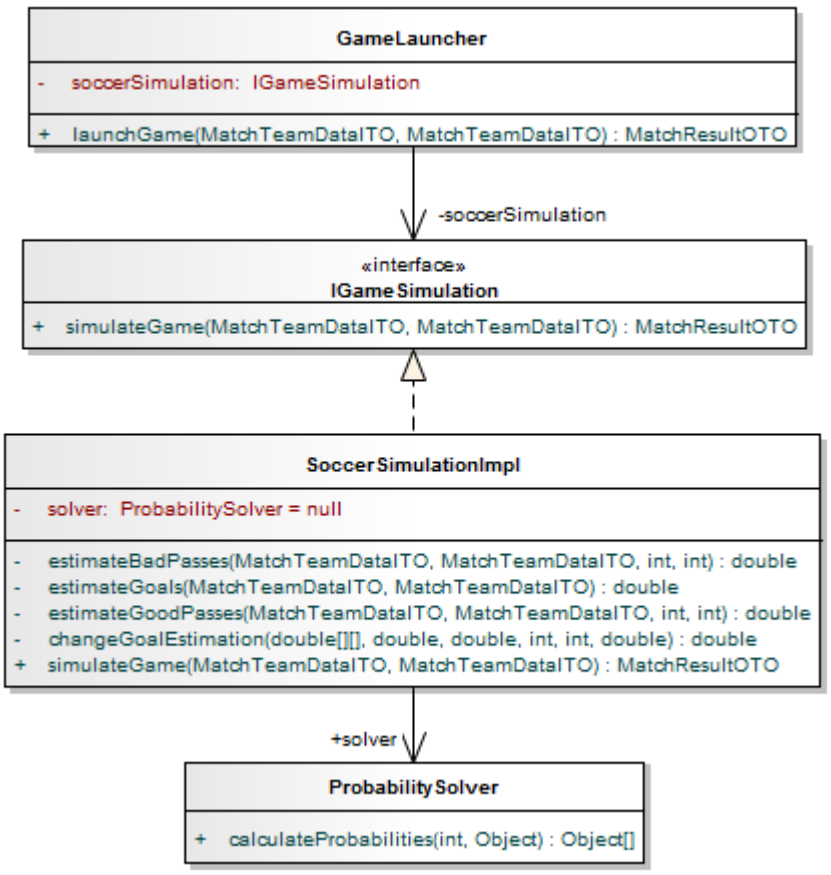
rozhodnutie, ktoré spravíme čisto zo štatistického hľadiska, môžeme o chvíľu posúdiť ako nesprávne. Počet striedaní je však častokrát obmedzený (vo futbale je to trikrát). Ďalším nedostatkom nášho modelu je fakt, že vychádzame jedine z pozície lopty a nie z pozícií hráčov. Je to z toho dôvodu, že takéto informácie o dianí na hracej ploche v každej situácii nemáme vôbec k dispozícii a ani rozsiahla analýza dostupných zdrojov nenasvedčovala, že by takéto dáta bolo možné získať.

Aplikácia Markovových procesov na kolektívne hry V tejto práci sme pre konkrétnu aplikáciu aj z dôvodu lepšej dostupnosti dát zvolili futbal. Každá hra má nejaké pravidlá a tie musíme v navrhovanom modely zohľadniť. Pravidlá sú pre rôzne hry častokrát odlišné. Napr. v basketbale sa po dosiahnutí bodu rozohráva z podkošového priestoru a nie zo stredu hracej plochy ako vo futbale. Vo volejbale tým loptu po získaní bodu nestratí. Navyše hry ako volejbal, resp. baseball, nie sú obmedzené časom, ale počtom dosiahnutých bodov, resp. počtom smien. Medzi typovo rovnaké hry ako je futbal zaraďujeme napríklad florbal, hokej a hádzanú a náš model by sa dal veľmi jednoducho na tieto hry aplikovať. Všetky športy však majú spoločnú tú vlastnosť, že sa dajú interpretovať pomocou stavov a prechodov medzi nimi. To nám umožňuje užiť Markovov proces ako vhodný štatistický model.

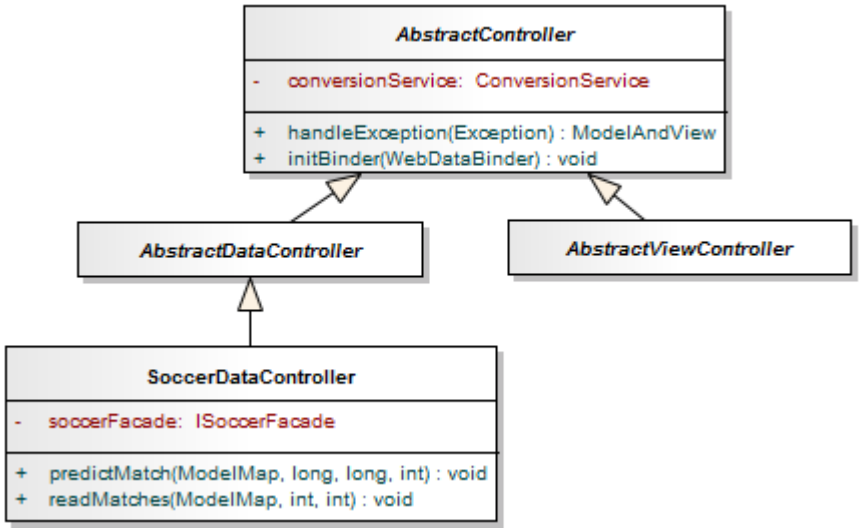
Na záver by bolo dobré poukázať na to, že v takmer každom informatickom probléme zohráva štatistika dôležitú úlohu. Ak vychádzame z empirických dát, informácie o ich vlastnostiach potrebujeme častokrát z dôvodu optimalizácií a správnych rozhodnutí. A práve cieľom štatistiky je nájsť „najlepšie" informácie z dostupných dát a preto v teórii rozhodovania zohráva dôležitú úlohu.

Dodatok A

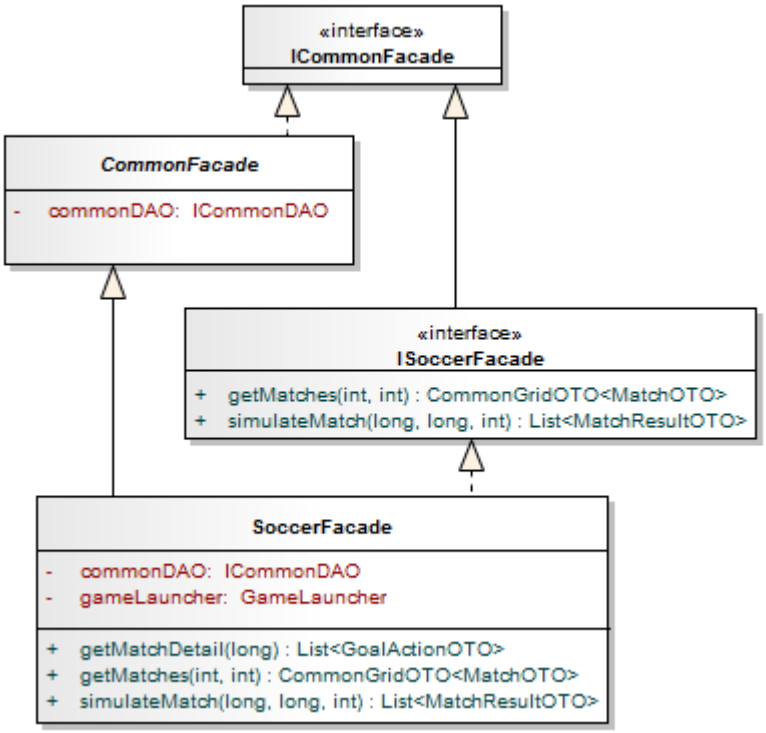
Diagramy



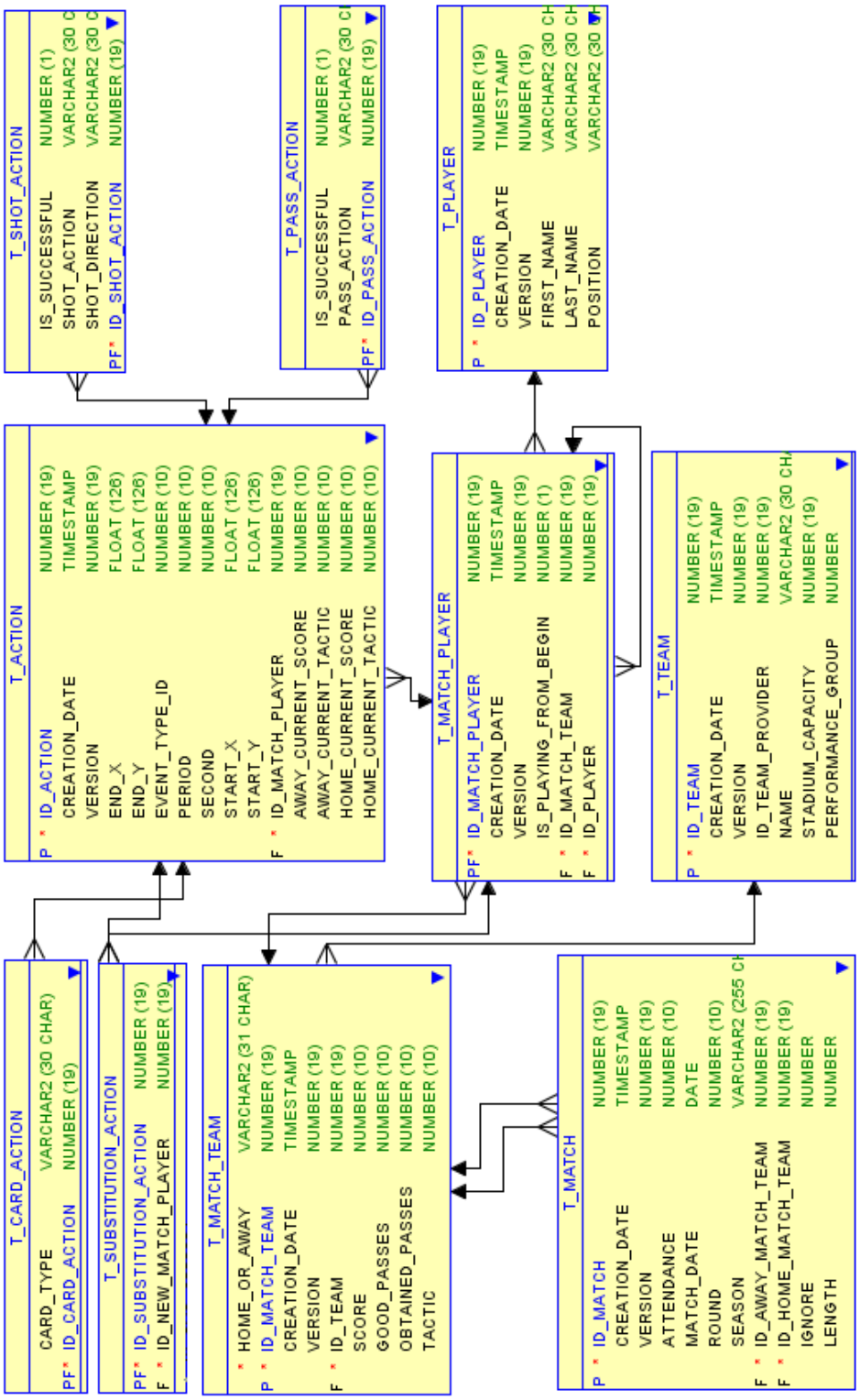
Obr. A.1: Štruktúra simulačných tried



Obr. A.2: Kontrolná vrstva - hierarchia `Controller` tried



Obr. A.3: Aplikačná vrstva - hierarchia `Facade` tried



Obr. A.4: Databázová schéma

Dodatok B

Príloha na CD

V zadnej časti práce je priložené CD, na ktorom sa nachádza

- elektronická verzia diplomovej práce
- výsledky štatistických meraní
- webová aplikácia
  - zdrojové kódy (*.java, *.js, *.css)
  - behové prostredia (JRE 6.0, Tomcat 6.0)
  - konfiguračné súbory (*,properties, *.xml)
  - dokumentácia (JavaDoc, užívateľská)
  - dáta (*.csv)

Literatúra

[1] Roman Zákutný (2007), *Simulace Sportovní manažer*, bakalárska práca.

[2] Maher (1982), *Modelling association football scores*, Stat Neerland **36**: 109-118

[3] Hirotsu, Wright (2002), *Using a Markov process model of an association football match to determine the optimal timing of substitution and tactical decisions*, J. Ops. Res. Soc., **53**, 88-96

[4] Boronico, Newbert (1999), *Play calling strategy in American football: a game-theoretic stochastic dynamic programming approach*, J. Sport Mngt. **13**, 13-21

[5] Bukiet, Harold, Palacios (1997), *A Markov chain approach to baseball*, Opns. Res. **45**, 14-23

[6] Clarke (1988), *Dynamic programming in one-day cricket - optimal scoring rates*, J. Opl. Res. Soc. **39**, 331-337

[7] Washburn (1991), *Still more on pulling the goalie*, Interfaces **21**, 59-64

[8] Perl, Miethling (1985), *Entwicklung optimaler Strategien am Beispiel von Badminton und Tennis*, Sportwissenschaft **15**, 170-182

[9] Wardrop (1995) *Simpson's paradox and the hot hand in basketball*, Am. stat. 49, 24-28

[10] Sackrowitz (2004), *Dynamic programming and time related strategies in sports*, Butenko, 31-41

[11] Xie, Chang, Divakaran, Sun (2002), *Structure analysis of Soccer Video with hidden Markov models*, Proceedings of International Conference on Acoustic, Speech and Signal Processing

[12] Qian, Tovinkere, August (2001), *Detecting semantic events in soccer games: Towards a complete solution*, IEEE International Conference on Multimedia and Expo

[13] Gong, Lim, Chua (1995), *Automatic parsing of TV soccer programs*, IEEE International Conference on Multimedia Computing and Systems, 167-174

[14] Zuzana Prášková, Petr Lachout (2001), *Základy náhodných procesů*

[15] Rabiner (1989), *A tutorial on hidden Markov models and selected applications in speech recognition*, IEEE 77, 257-285

[16] MacCullagh, Nelder (1983), *Generalized linear models - Monogrpahs on statistics and applied probability*

[17] Jack Weiss, *Statistical Analysis in Ecology and Evolution* (2006) `http://www.unc.edu/courses/2006spring/ecol/145/001/docs/lectures/lecture5.htm`

[18] Spring Framework 3.0, *Reference Documentation* `http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html`

[19] Christian Bauer, Gavin King (2007), *Java Persistence with Hibernate*

[20] Java 2 Platform, Enterprise Edition (J2EE) Overview `http://java.sun.com/j2ee/overview.html`