



Universidade Federal  
do Espírito Santo

CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA

Processos Estocásticos Aplicados à Computação  
Turmas 01 e 02 (EARTE) – 2021/2  
Prof. Rodolfo da Silva Villaça – [rodolfo.villaca@ufes.br](mailto:rodolfo.villaca@ufes.br)

Trabalho T1 – Aplicação de Cadeias de Markov na Simulação de um Jogo de Tênis

Objetivo:

Simular uma partida de tênis representando o jogo por meio de uma Cadeia de Markov. A partir desta representação, estudar o comportamento do jogo variando-se a diferença técnica, representada pela probabilidade de se vencer uma partida, entre os jogadores.

Definições Gerais:

1. As regras gerais sobre como funciona uma partida de tênis podem ser encontradas [aqui](#)<sup>1</sup>. No simulador a ser desenvolvido neste trabalho sempre considere que um jogador vence uma partida em um esquema “melhor de 3 sets”, ou seja: o vencedor da partida (*match*) é o jogador que vence 2 sets. Com isso os resultados possíveis para uma partida são 2-0, 2-1, e um jogo pode ter de 2 a 3 sets em disputa;
2. Considere uma partida disputada entre os jogadores A e B. Como não existe a possibilidade de empate em uma partida de tênis, sendo  $p$  a probabilidade de que o Jogador A marque um ponto em uma disputa, e  $q$  a probabilidade de que o Jogador B marque o ponto na mesma disputa, então faça  $p + q = 1$ , ou  $q = 1 - p$ ;
3. Por simplificação, considere que  $p$  e  $q$  são constantes em toda a partida. Ou seja, desconsidere efeitos externos à mesma, tais como temperatura, desgaste físico, emocional, apoio de torcida e histórico de partidas entre eles. Essa variação de  $p$  e  $q$  ao longo da partida não será explorada neste trabalho!
4. Uma possível representação de um *game* em uma partida de tênis por meio de uma Cadeia de Markov está representada na Figura 1, a seguir. Tal representação foi proposta em Sean Carver neste [post](#)<sup>2</sup>;

Instruções:

1. Neste trabalho você deverá simular dois tipos de partida, onde o valor de  $p$  é de livre escolha de cada grupo de acordo com as seguintes regras:

•

<sup>1</sup> <https://www.esportelandia.com.br/tenis/regras-do-tenis/>

<sup>2</sup> <https://towardsdatascience.com/markov-chain-models-in-sports-7cb907a6c52f>

CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA

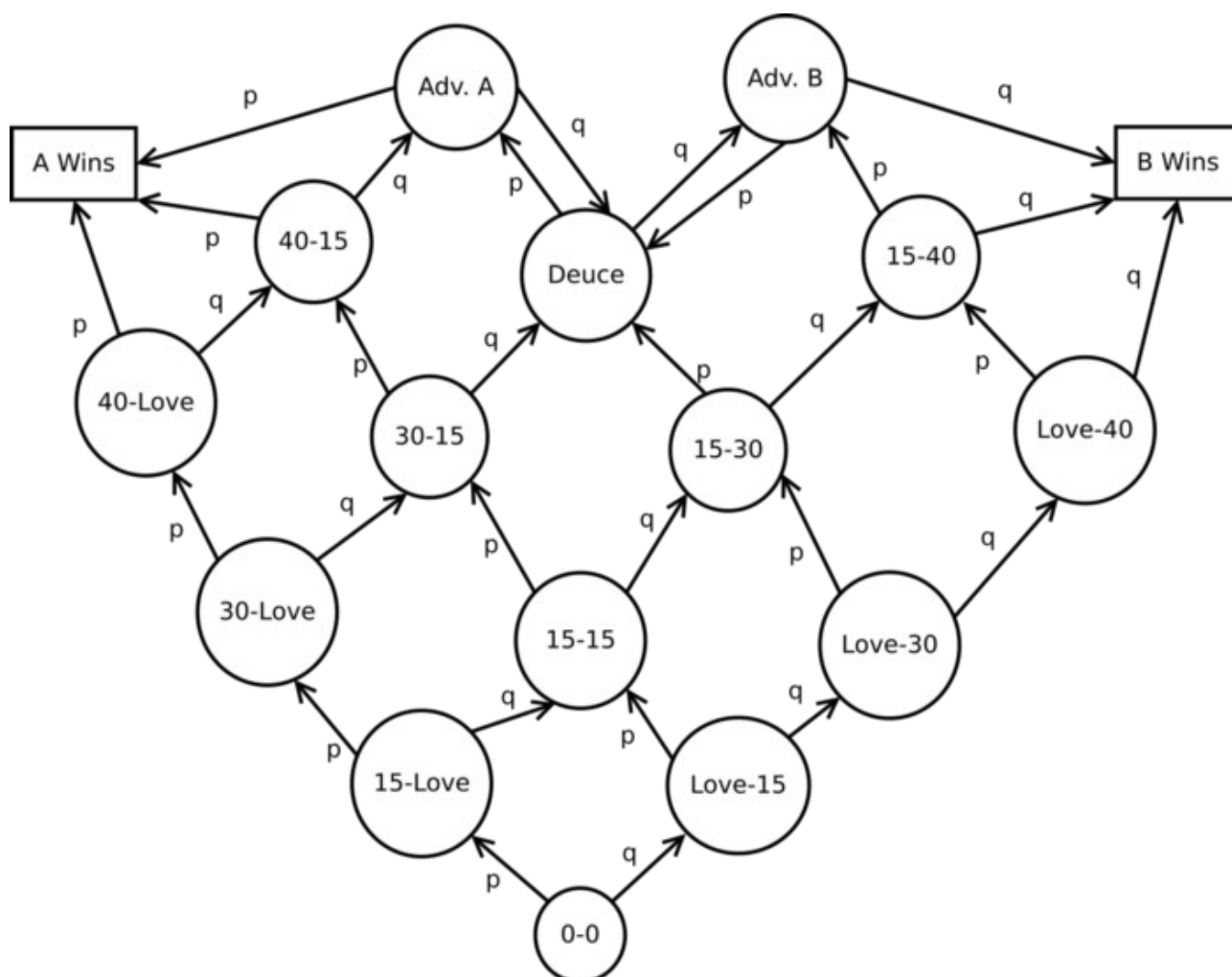


Figura 1: Representação de um jogo (game) em uma partida de tênis por meio de uma Cadeia de Markov. Na figura,  $p$  é a probabilidade de o Jogador A pontuar no game e  $q$  é a probabilidade do Jogador B pontuar no mesmo game. Sabe-se que  $p + q = 1$ . Adv (Advantage) representa uma vantagem no game, quando um jogador está a 1 ponto de fechar o game. Deuce representa um empate temporário, quando ambos estão próximos de fechar e vencer a partida. Love significa que o outro jogador ainda não pontuou no game.

- Partida 1: o Jogador A é muito melhor do que o Jogador B ( $0,70 \leq p \leq 0,80$ );
- Partida 2: os jogadores possuem nível técnico equivalente ( $0,45 \leq p \leq 0,55$ )

2. No simulador você sempre deverá iniciar um *game* pelo estado “0-0” (estado mais inferior na Figura 1). Neste estado inicial você deverá sortear aleatoriamente um vencedor do ponto em disputa no *game* (Jogador A ou B). **Importante: este sorteio aleatório deverá obedecer a distribuição de probabilidade entre  $p$  e  $q$ .** Na Figura 1 a transição representada por  $p$  representa que o Jogador A venceu o ponto, enquanto a transição representada por  $q$  representa que o Jogador B venceu o ponto em disputa.

CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA

Sendo assim, se no estado “0-0”, com probabilidade  $p$  o Jogador A fez o primeiro ponto do *game*, então haverá uma transição para o estado “15-Love” (segunda linha de baixo para cima à esquerda). Ao contrário, se no estado “0-0”, com probabilidade  $q$  o Jogador B fez o primeiro ponto do *game*, então haverá uma transição para o estado “Love-15” (segunda linha de baixo para cima à direita).

3. O *game* acaba quando um dos estados for alcançado primeiro: “A Wins” ou “B Wins” e um ponto será adicionado ao *set* do Jogador A ou B, respectivamente. A Cadeia de Markov é então reiniciada para simular o próximo *game*. Conforme as regras, vence o *set* o jogador que primeiro alcançar 6 ou 7 *games* (ver as regras do jogo). A partida acaba quando algum jogador vencer 2 *sets* no jogo (melhor de 3, ver as regras do jogo e as definições do trabalho).

4. Para cada partida (Partida 1 e Partida 2, ver a Instrução 1) você deverá executar ao menos  $n \geq 30$  simulações **com os mesmos valores de  $p$  e  $q$** . Ou seja, **não modifique os valores de  $p$  e  $q$  em cada uma das  $n$  simulações das Partidas 1 e 2**. Cada partida tem seus próprios valores, mas **eles deverão permanecer fixos em todas as  $n$  simulações de cada partida**.

5. Durante as simulações você deverá gerar um *log* em arquivo .txt ou .csv constando a evolução dos pontos em cada *game*, desde o início até o fim da partida, em cada simulação. **Importante: o formato do *log* pode ser definido pelo próprio grupo, mas é necessário que o formato esteja devidamente documentado ao fim do trabalho**. Por simplificação das análises posteriores, recomenda-se que para cada simulação também sejam feitos *logs* da evolução dos *games* em cada *set* e de cada *set* em cada partida (*match*).

6. Ao final das suas simulações, com base nos *logs* gerados (vamos denominá-los *datasets* a partir de agora) você deverá analisar os *datasets* e responder às seguintes perguntas:

- Qual a probabilidade do Jogador A/B vencerem as Partidas 1 e a Partidas 2? Para responder, considere  $X_i$  uma VA que representa o número de vitórias do jogador  $i$  em cada caso, e que nosso espaço amostral contem 3 partidas. Mostre uma análise estatística baseada em média e desvio padrão em cada caso.
- Qual a distribuição do número de *sets*, *games* e *pontos* nas Partidas 1 e 2? Mostre uma análise estatística baseada em média e desvio padrão em cada caso.

**CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA**

- Selecione aleatoriamente, com distribuição uniforme, 10 simulações dentre as  $n$  existentes em seus *datasets* originais. Refaça as 2 análises anteriores e explique as diferenças e semelhanças entre os resultados obtidos.

**Requisitos:**

1. O trabalho pode ser feito em grupos de 2 ou 3 alunos: não serão aceitos trabalhos individuais ou em grupos de mais de 3 alunos;
2. Os grupos poderão implementar os trabalhos usando qualquer uma dentre as linguagens de programação a seguir: C, Java, Python ou Go;
3. A submissão deverá ser feita por meio de um *link* com a disponibilização do código no Github, em modo de acesso público ou, minimamente, que meu e-mail [rodolfo.villaca@ufes.br](mailto:rodolfo.villaca@ufes.br) tenha direito de acesso ao código em um repositório privado;
4. A documentação para uso/teste do seu programa deverá ser feita integralmente na própria página do Github através do arquivo README<sup>3</sup>;
5. O grupo deverá gravar um vídeo de no máximo 5 min apresentando o funcionamento/teste do trabalho;
5. Avaliação: Adequação aos Requisitos (20%), Implementação (25%), Documentação – Instalação e Uso (25%), Documentação – Testes, Apresentação e Avaliação dos Resultados (30%),
6. Data de Entrega: 11/03/2022 pela Sala Virtual da disciplina no Google Sala de Aula;

Bom trabalho e boas férias!

---

3 <https://guides.github.com/features/wikis/>