Discrete random walks with memory: Models and applications

Ing. Tomáš Kouřim

Institute of Information Theory and Automation, AS CR Prague

16.9.2019

Outline

1. Prepare mathematical model
2. Describe its properties
3. Apply it on data

Outline

1. Prepare mathematical model
2. Describe its properties
3. Apply it on data

Outline

1. Prepare mathematical model
2. Describe its properties
3. Apply it on data

Random walk

Definition

A man starts from a point O and walks l yards in a straight line; he then turns through any angle whatever and walks another l yards in a second straight line. He repeats this process n times. I require the probability that after these n stretches he is at a distance between r and $r + \delta r$ from his starting point, O.

[Karl Pearson: The problem of the random walk. (1905)]

Where is the *"Drunken sailor"*?

Random walk

Definition

A man starts from a point O and walks l yards in a straight line; he then turns through any angle whatever and walks another l yards in a second straight line. He repeats this process n times. I require the probability that after these n stretches he is at a distance between r and $r + \delta r$ from his starting point, O.

[Karl Pearson: The problem of the random walk. (1905)]

Where is the *"Drunken sailor"*?

Random walk

Definition

Let$\{X_k\}_{k=1}^{\infty}$be a sequence of discrete random variables. For each positive integer n, let S_n denote the sum $X_1 + X_2 + \cdots + X_n$, with $S_0 = 0$. The sequence $\{S_n\}_{n=1}^{\infty}$ is called a random walk. If the common range of the X_k's is $\mathbb{R}_m$, then $\{S_n\}$ is a random walk in $\mathbb{R}_m$.

If for $\forall k$; $X_k \sim B(p = \frac{1}{2})$, the walk is called the standard random walk.

Random walk properties

- Discrete random process
- $n-$dimensional, on a matrix, graph, finite or infinite set
- Self avoiding, reinforced
- Brownian motion, polymer creation, games simulation, sports simulation

Random walk with memory

- Based on standard random walk (Bernoulli distribution with $p = 0.5$, discrete time).
- Constant total step size:

$$l_i^+ + l_i^- = 2 \ \forall i \in \mathbb{N}.$$

- At the beginning the step sizes are equal ($l_1^+ = l_1^- = 1$) and further for $t > 1$ evolve using a memory parameter $\lambda \in (0, 1)$:

$$X_{t-1} = 1 \rightarrow \begin{cases} l_t^+ = \lambda l_{t-1}^+ \\ l_t^- = 2 - \lambda l_{t-1}^+ \end{cases} \quad X_{t-1} = -1 \rightarrow \begin{cases} l_t^+ = 2 - \lambda l_{t-1}^- \\ l_t^- = \lambda l_{t-1}^- \end{cases}$$

- Loïc Turban. *On a random walk with memory and its relation with markovian processes.* Journal of Physics A: Mathematical and Theoretical (2010).

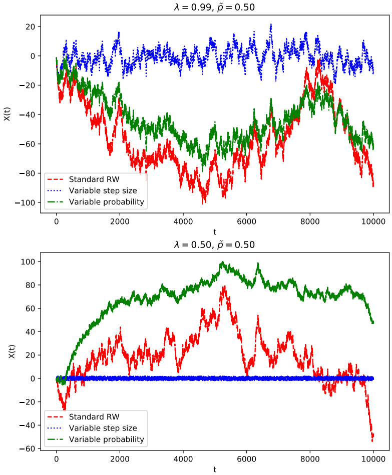
Random walk with varying transition probability

- Based on standard random walk (Bernoulli distribution with $p = 0.5$, discrete time).
- Step size remains constant, transition probability changes
- First step realized according to starting probability p_0 which then for $t > 1$ evolve using a memory parameter $\lambda \in (0, 1)$:

$$X_t = 1 \rightarrow p_t = \lambda p_{t-1}$$

$$X_t = -1 \rightarrow p_t = 1 - \lambda(1 - p_{t-1})$$

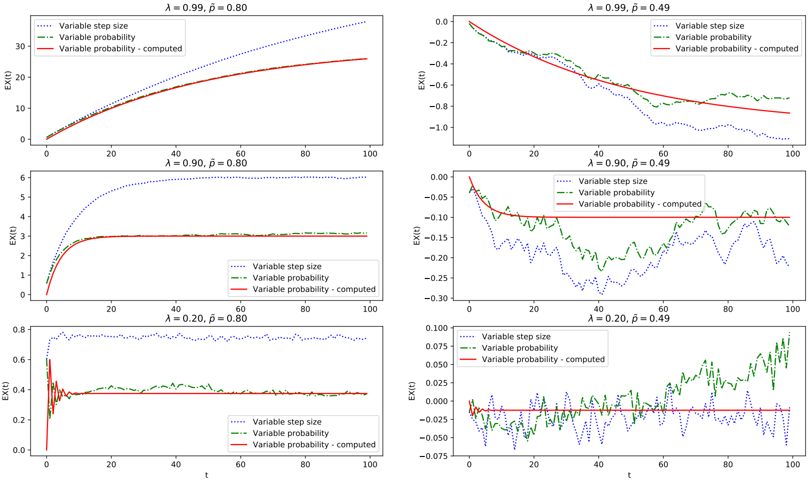
Example - RW evolution



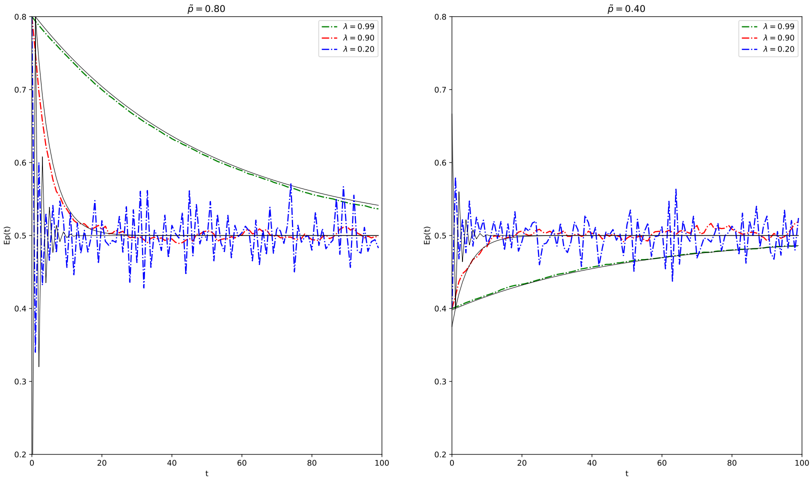
Random walk with varying transition probability - properties

- $P_t = \lambda P_{t-1} + \frac{1}{2}(1-\lambda)(1-X_t)$
- $P_t = p_0 \lambda^t + \frac{1}{2}(1-\lambda)\sum_{i=1}^{t} \lambda^{t-i}(1-X_i)$
- $EP_t = (2\lambda - 1)^t p_0 + \frac{1-(2\lambda-1)^t}{2}$
- $ES_t = S_0 + (2p_0 - 1)\frac{1-(2\lambda-1)^t}{2(1-\lambda)}$

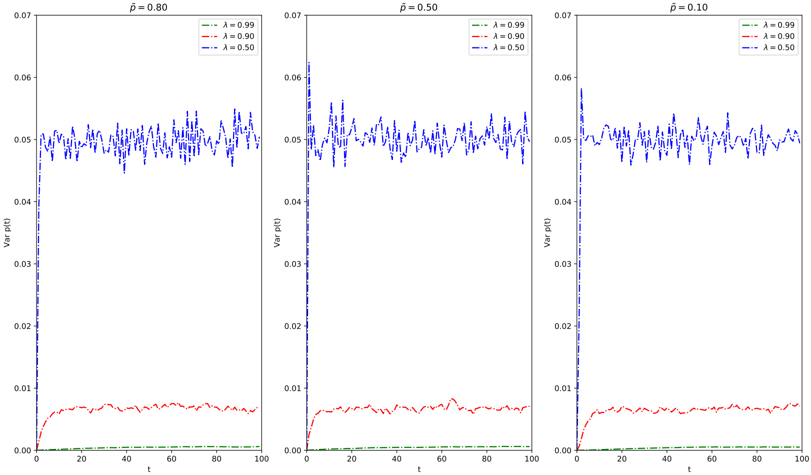
Example - Expected position of the walker



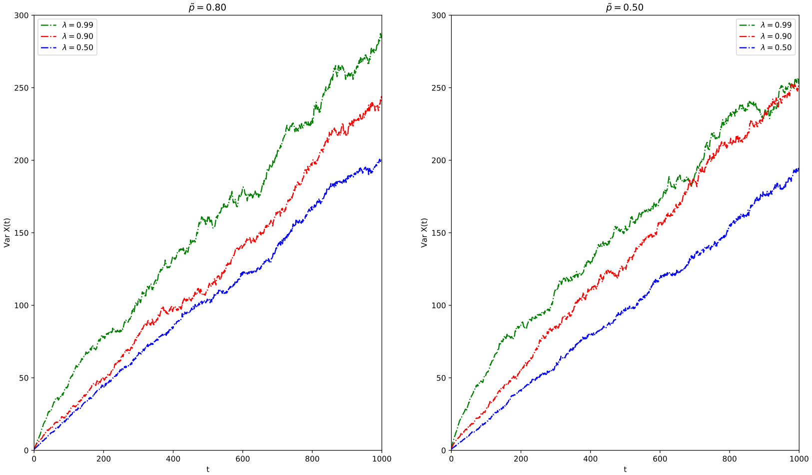
Example - Expected transition probability



Example - Transition probability variance



Example - Walker's position variance



Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Alternative definitions

- Formulas to obtain next transition probability P_i:
  - $\frac{1}{2}[(1+X_i)\lambda P_{i-1} + (1-X_i)(1-\lambda(1-P_{i-1}))]$
    - Success punished
  - $\frac{1}{2}[(1-X_i)\lambda P_{i-1} + (1+X_i)(1-\lambda(1-P_{i-1}))]$
    - Success rewarded
  - $\frac{1}{2}[(1+X_i)\lambda_0 P_{i-1} + (1-X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success punished with multiple λ
  - $\frac{1}{2}[(1-X_i)\lambda_0 P_{i-1} + (1+X_i)(1-\lambda_1(1-P_{i-1}))]$
    - Success rewarded with multiple λ

Data generation

- Using Python & Numpy package
- Different parameters:
  - $\lambda \in \{0.01, 0.1, 0.25, 0.5, 0.6, 0.8, 0.9, 0.99\}$
  - $\bar{\lambda} = \{[0.1, 0.2], [0.2, 0.5], [0.5, 0.8], [0.6, 0.9], [0.1, 0.9], [0.5, 0.99], [0.8, 0.01], [0.99, 0.9]\}$
  - $p_0 = \{0.01, 0.1, 0.25, 0.5, 0.8, 0.9, 0.99\}$
  - $n = \{2, 3, 4, 5, 10, 50, 100, 1000\}$
  - 100 walks for each combination

Fitting parameters on generated data

- Find $\bar{\lambda}$ with known p_0, model type
- Find p_0 with known $\bar{\lambda}$, model type
- Find p_0, $\bar{\lambda}$ with known model type
- Find p_0, $\bar{\lambda}$ and model type

Parameter fitting evaluation

- ?????

Real life applications

- Reliability analysis, medical data analysis
- Sport modelling
  - Tennis with model "Success rewarded"
  - Applied for live betting on US Open against Tipsport bookmaker
  - Source code and paper available at https://github.com/tomaskourim/mathsport2019

Results

- New models of Random walk described
- Properties derived
- Strenght of the models shown using simulated data
- Possible real life applications shown
- Source code and paper available at https://github.com/tomaskourim/amistat2019

Next steps????

- Model implementation
  - λ optimization
  - p_1 optimization
- Model improvement
  - Other versions of random walk with memory
  - Combination with other approaches
- Model testing
  - Model evaluation granularity
  - Performance on a larger dataset
  - Betting module for more bookmakers
  - Application of the model to *best-of-three* matches
- Application in other domains

Next steps????

- Model implementation
  - λ optimization
  - p_1 optimization
- Model improvement
  - Other versions of random walk with memory
  - Combination with other approaches
- Model testing
  - Model evaluation granularity
  - Performance on a larger dataset
  - Betting module for more bookmakers
  - Application of the model to *best-of-three* matches
- Application in other domains

Next steps????

- Model implementation
  - λ optimization
  - p_1 optimization
- Model improvement
  - Other versions of random walk with memory
  - Combination with other approaches
- Model testing
  - Model evaluation granularity
  - Performance on a larger dataset
  - Betting module for more bookmakers
  - Application of the model to *best-of-three* matches
- Application in other domains

Next steps????

- Model implementation
  - λ optimization
  - p_1 optimization
- Model improvement
  - Other versions of random walk with memory
  - Combination with other approaches
- Model testing
  - Model evaluation granularity
  - Performance on a larger dataset
  - Betting module for more bookmakers
  - Application of the model to *best-of-three* matches
- Application in other domains

Thank you.

tom@skourim.com