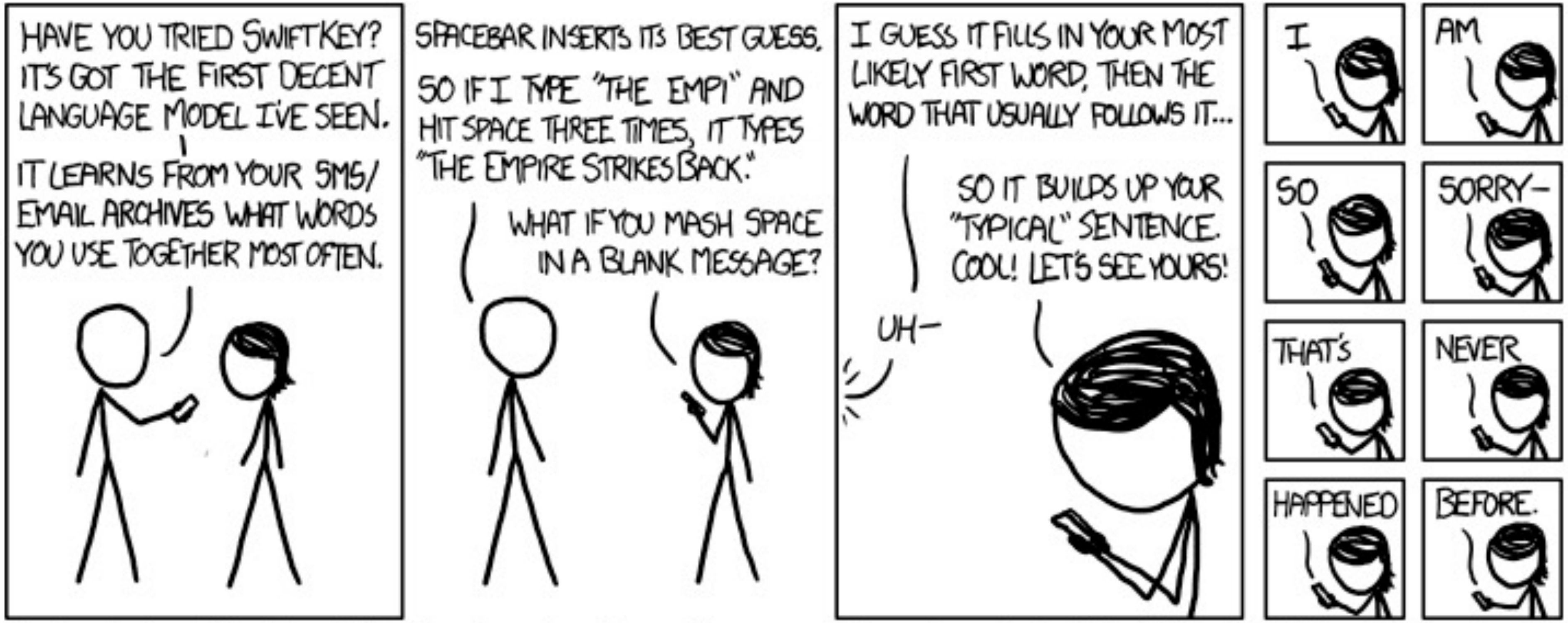


Title text: Although the Markov chain-style text model is still rudimentary; it recently gave me "Massachusetts Institute of America". Although I have to admit it sounds prestigious.

Markov Chains

CSE 312 23Su

Lecture 21

https://xkcd.com/1068/

So far…

We have spent a lot of time looking at probabilities for "single-shot" processes/experiments.

Random process → Outcome Distribution D

More generally…

Randomness can occur at different points in **time** and may depend on **previous outcomes**.

Random process 1 → Outcome Distribution D_1 → Random process 2 → Outcome Distribution D_2 → Random process 3 → Outcome Distribution D_3

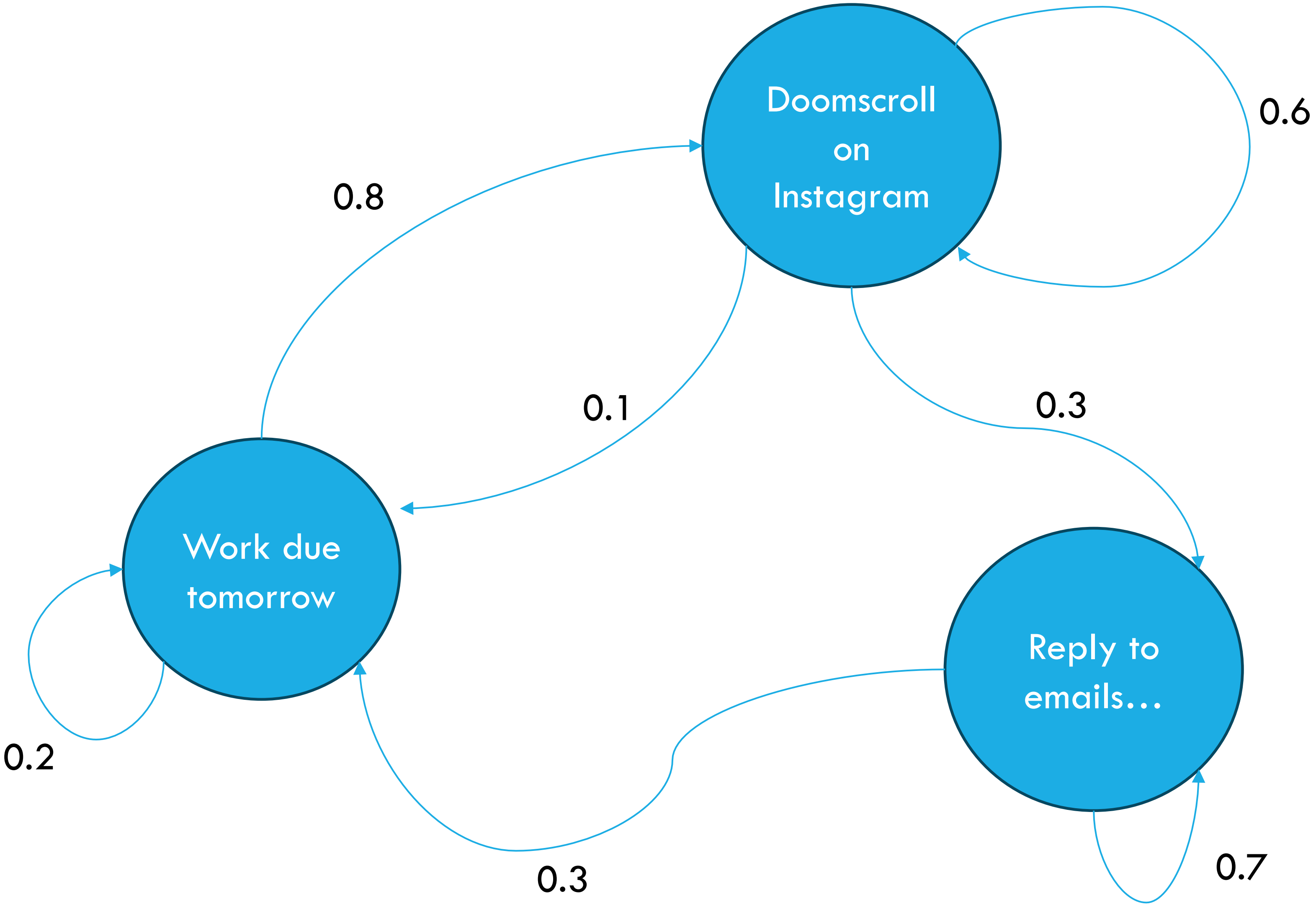
These kinds of processes are called **stochastic processes**.

Stochastic processes

Discrete-time stochastic process (DTSP)

A sequence of random variables $X_0, X_1, X_2, \ldots$ where X_t is the value at time t.

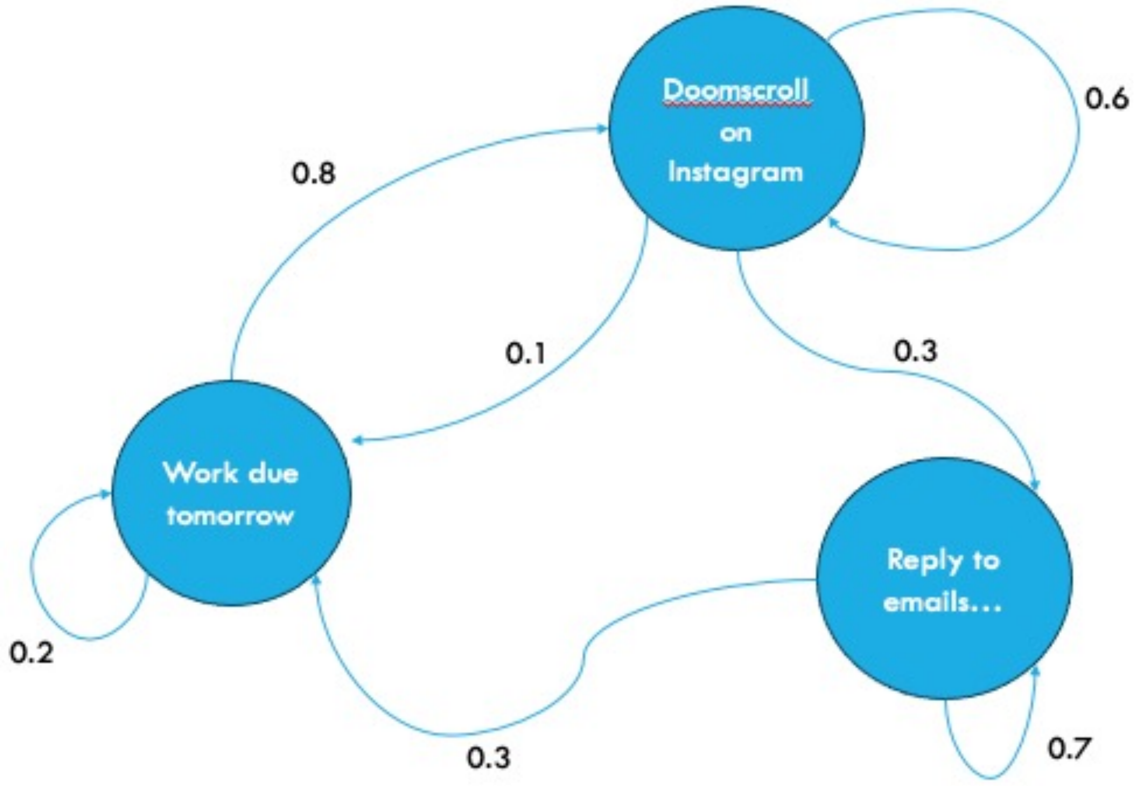
Day in the life



Our first Markov chain

At each time step *t*

- I can be in one of 3 **states** (work, Instagram, emails)

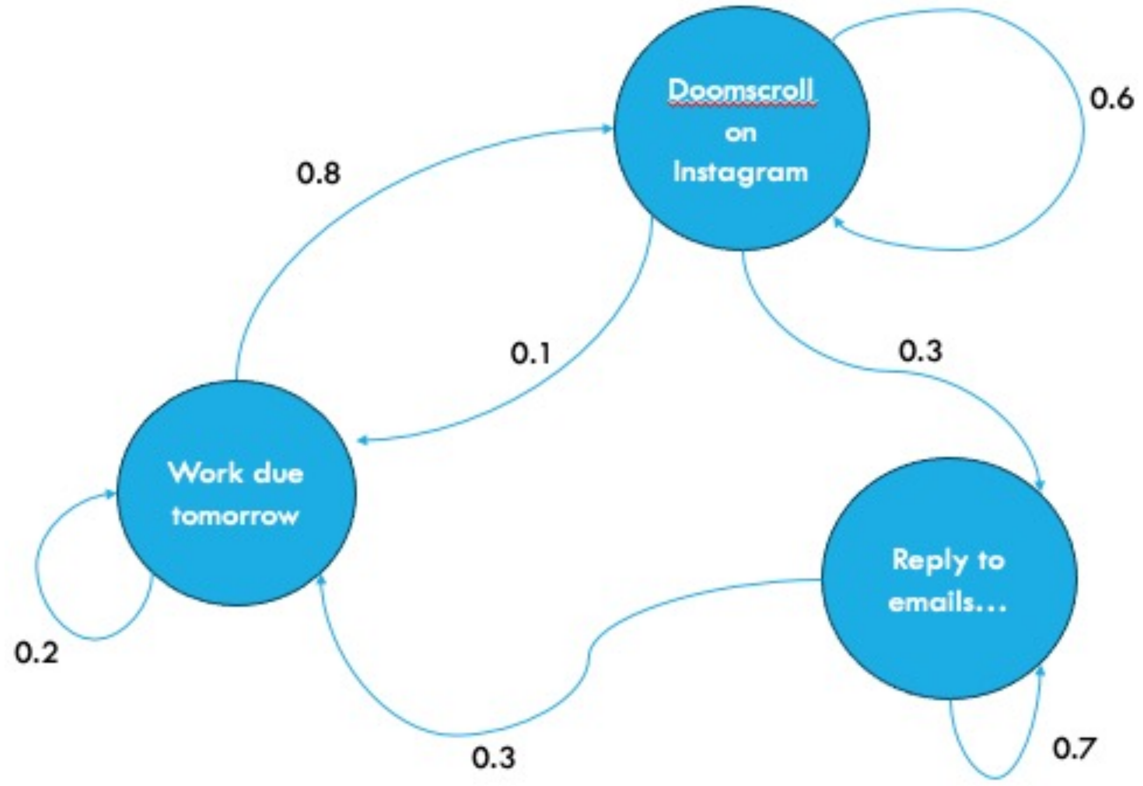


Our first Markov chain

At each time step *t*

- I can be in one of 3 **states** (work, Instagram, emails)

- If I am in some state *s* at time *t*, the labels of out-edges of s give the probabilities of moving to each of the states at time *t+1* (or staying at the same state)



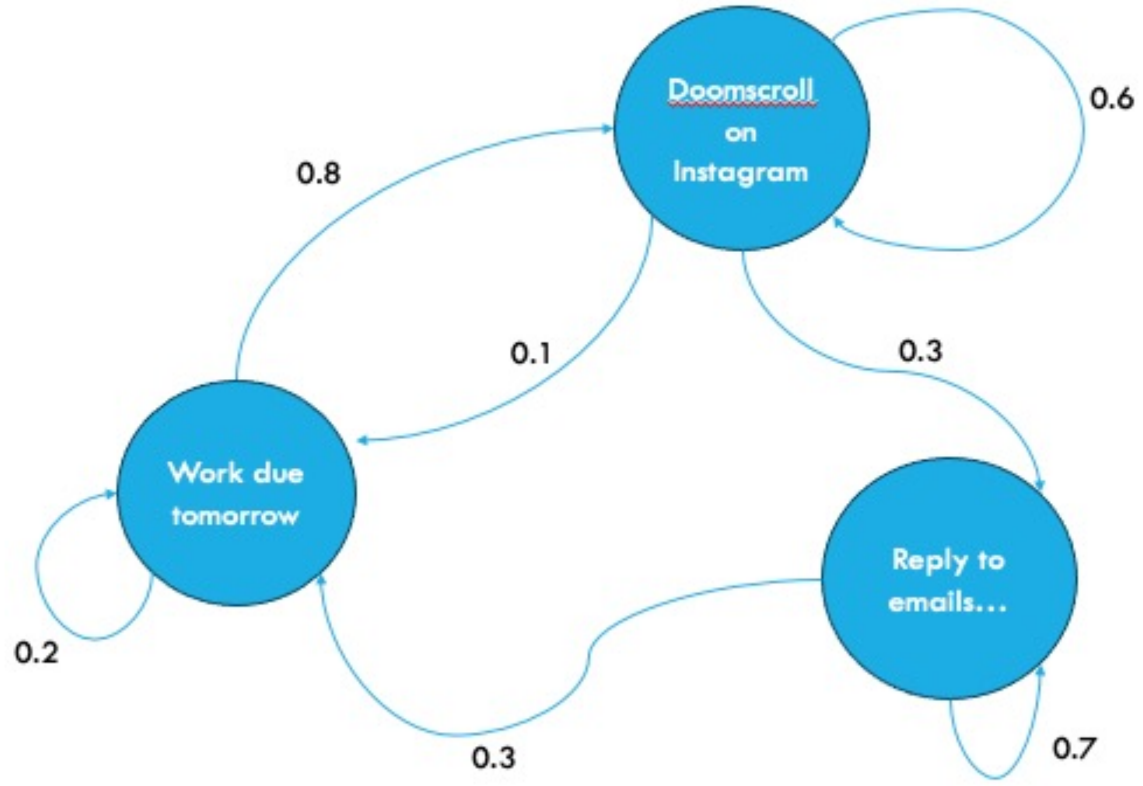
Hey! That looks familiar to me!

Yes, Markov chains are special kind of probabilistic (finite) automaton.

They look a bit like Deterministic Finite Automata (DFAs) from 311…

> There are no input symbols on the edges. (The input symbol is always "tick of the clock")

> Out edges are marked with probabilities.



State Space

The state space $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ is finite (or countably infinite) so that each $X_t \in \mathcal{S}$.

Markov Property

The future is conditionally independent of the past given the present.

$$\mathbb{P}(X_{t+1} = x_{t+1} \mid X_0 = x_0, \ldots, X_t = x_t) = \mathbb{P}(X_{t+1} = x_{t+1} \mid X_t = x_t)$$

Stationary Transition Probabilities

We always transition from state s_i to s_j with probability independent of the current time.

Then, the number of transition probabilities is the number of ways to move from one state to another: n^2.

Transition Probability Matrix (TPM)

A TPM is a matrix that contains information about the probability of transitioning between states.

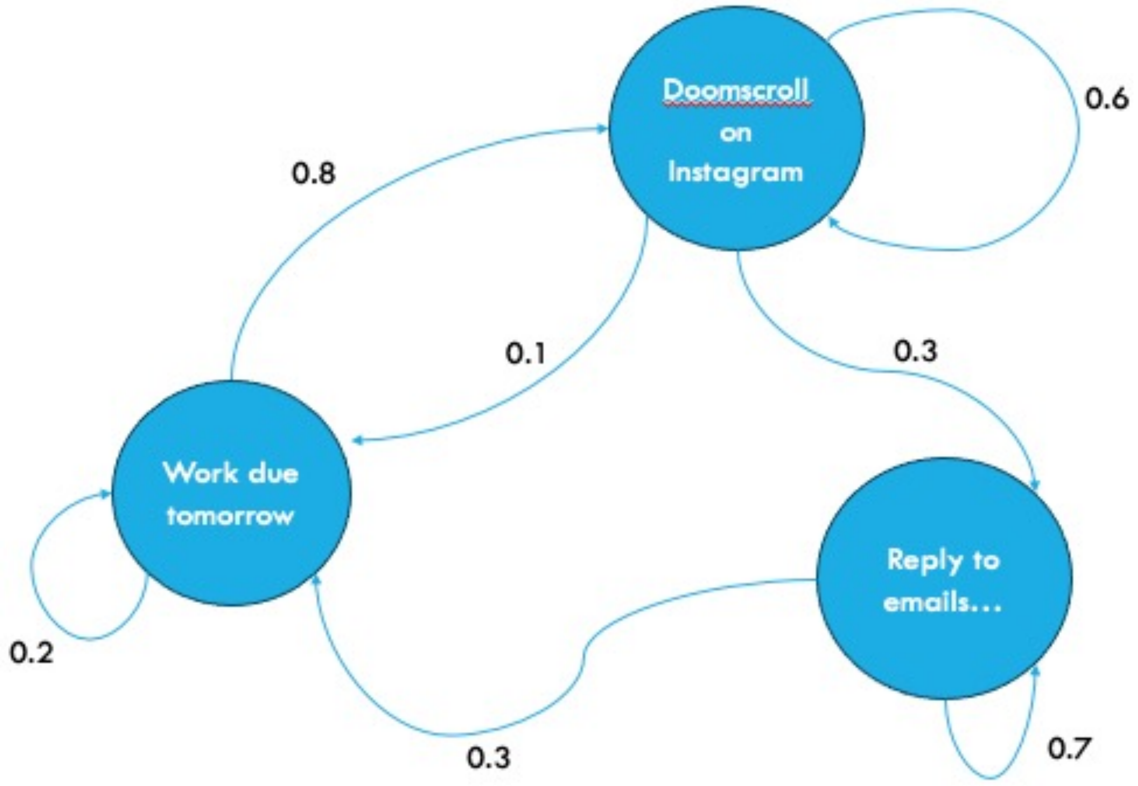
There are n rows and n columns.

The row represents our current state.

The column represents our next state.

So, the element at the ith row and jth col is $\mathbb{P}\left(X_{t+1} = s_j \mid X_t = s_i\right)$.

Transition Probability Matrix (TPM)

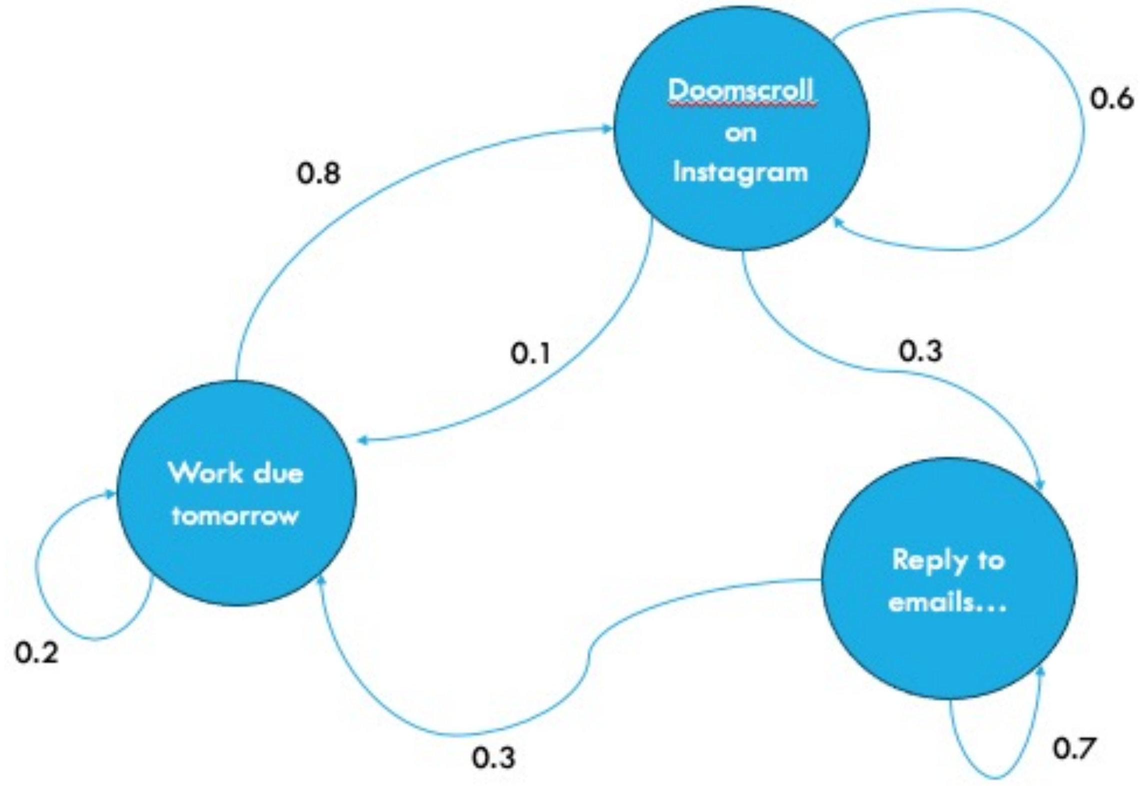


Transition Probability Matrix (TPM)

$$\begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

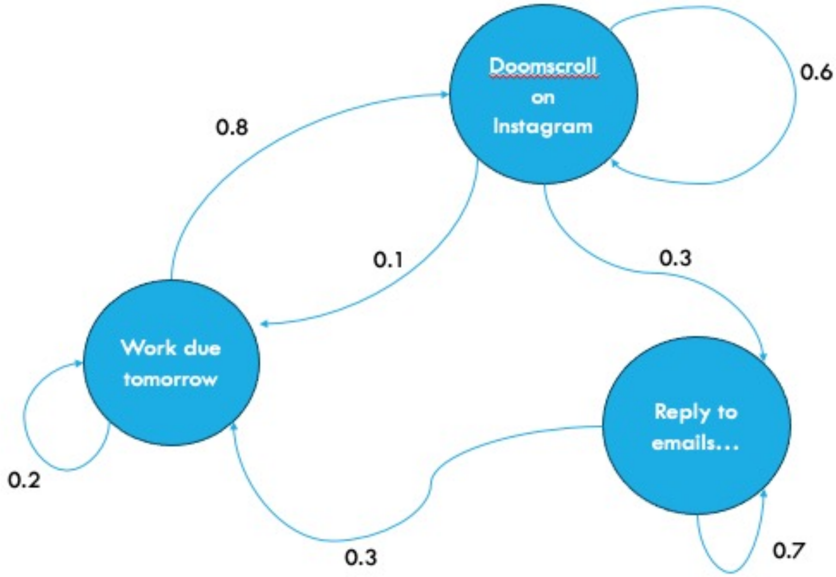
Note that each row sums to 1.

A TPM uniquely defines a Markov chain.



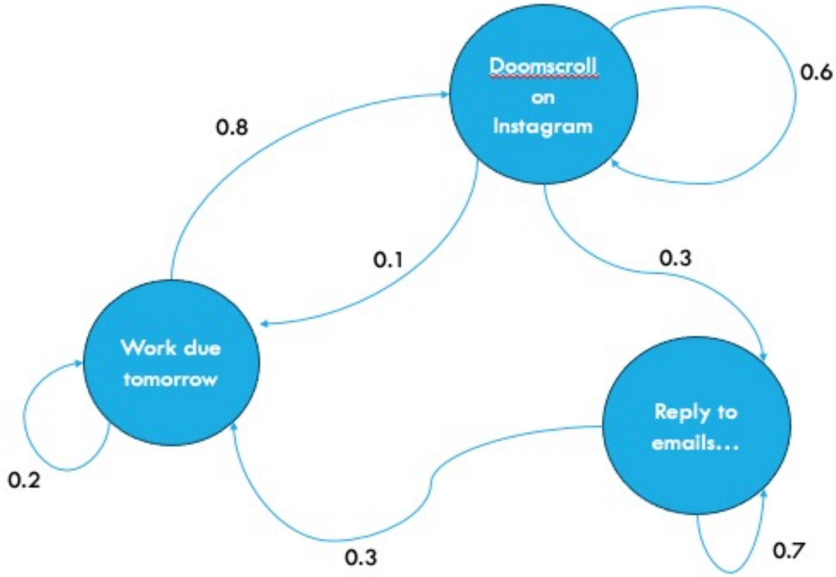
Extending the time...

Let's say we start at the state "Work". What is the probability of me being at state s at time t?



t	0	1	2
$\mathbb{P}(X_t = Work)$			
$\mathbb{P}(X_t = Insta)$			
$\mathbb{P}(X_t = Email)$			

Extending the time…



Let's say we start at the state "Work". What is the probability of me being at state *s* at time *t*?

t	0	1	2
$\mathbb{P}(X_t = Work)$	1	0.2	$= 0.2 \times 0.2 + 0.1 \times 0.8 + 0.3 \times 0 = 0.12$
$\mathbb{P}(X_t = Insta)$	0	0.8	$= 0.8 \times 0.2 + 0.6 \times 0.8 + 0 \times 0 = 0.64$
$\mathbb{P}(X_t = Email)$	0	0	$= 0 \times 0.2 + 0.3 \times 0.8 + 0.7 \times 0 = 0.24$

A Linear Algebra Perspective

$$\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} ad + bg & ae + bh & af + bi \end{bmatrix}$$

A Linear Algebra Perspective

$$[a \quad b]\begin{bmatrix} d & e & f \\ g & h & i \end{bmatrix} = [ad + bg \quad ae + bh \quad af + bi]$$

t	0	1	2
$\mathbb{P}(X_t = Work) = q_W^{(t)}$	1	0.2	0.12
$\mathbb{P}(X_t = Insta) = q_I^{(t)}$	0	0.8	0.64
$\mathbb{P}(X_t = Email) = q_E^{(t)}$	0	0	0.24

Let $[q_W^{(t)}, q_I^{(t)}, q_E^{(t)}]$ be a row vector.

$$M = \begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

A Linear Algebra Perspective

$$[a \quad b]\begin{bmatrix} d & e & f \\ g & h & i \end{bmatrix} = [ad+bg \quad ae+bh \quad af+bi]$$

t	0	1	2
$\mathbb{P}(X_t = Work) = q_W^{(t)}$	1	0.2	0.12
$\mathbb{P}(X_t = Insta) = q_I^{(t)}$	0	0.8	0.64
$\mathbb{P}(X_t = Email) = q_E^{(t)}$	0	0	0.24

$$M = \begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

Let $[q_W^{(t)}, q_I^{(t)}, q_E^{(t)}]$ be a row vector.

$$[q_W^{(0)}, q_I^{(0)}, q_E^{(0)}]M$$
$$= [1 \quad 0 \quad 0]\begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

$$= [0.2 \quad 0.8 \quad 0]$$
$$= [q_W^{(1)}, q_I^{(1)}, q_E^{(1)}]$$

A Linear Algebra Perspective

$$[a \quad b]\begin{bmatrix} d & e & f \\ g & h & i \end{bmatrix} = [ad + bg \quad ae + bh \quad af + bi]$$

t	0	1	2
$\mathbb{P}(X_t = Work) = q_W^{(t)}$	1	0.2	0.12
$\mathbb{P}(X_t = Insta) = q_I^{(t)}$	0	0.8	0.64
$\mathbb{P}(X_t = Email) = q_E^{(t)}$	0	0	0.24

$$M = \begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

Let $[q_W^{(t)}, q_I^{(t)}, q_E^{(t)}]$ be a row vector.

$$[q_W^{(1)}, q_I^{(1)}, q_E^{(1)}]M$$
$$= [0.2 \quad 0.8 \quad 0]\begin{bmatrix} 0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

$$= [0.12 \quad 0.64 \quad 0.24]$$
$$= [q_W^{(2)}, q_I^{(2)}, q_E^{(2)}]$$

In General…

$$\left[q_W^{(t)}, q_I^{(t)}, q_E^{(t)}\right] M = \left[q_W^{(t+1)}, q_I^{(t+1)}, q_E^{(t+1)}\right]$$

$$Let\ \boldsymbol{q}^{(t)} = \left[q_W^{(t)}, q_I^{(t)}, q_E^{(t)}\right]$$

$$Then,\ for\ all\ t \geq 0\ ,\ \boldsymbol{q}^{(t)}\boldsymbol{M} = \boldsymbol{q}^{(t+1)}$$

A little bit of induction

$Let\ \boldsymbol{q}^{(t)} = \left[q_W^{(t)}, q_I^{(t)}, q_E^{(t)}\right]$

$Then,\ for\ all\ t \geq 0\ ,\ \boldsymbol{q}^{(t)}\boldsymbol{M} = \boldsymbol{q}^{(t+1)}$

A little bit of induction

$Let\ \boldsymbol{q}^{(t)} = \left[q_W^{(t)},\ q_I^{(t)}, q_E^{(t)}\right]$

$Then, for\ all\ t \geq 0\ , \boldsymbol{q}^{(t)}\boldsymbol{M} = \boldsymbol{q}^{(t+1)}$

$q^{(1)} = q^{(0)}M$

$q^{(2)} = q^{(1)}M = q^{(0)}M^2$

$q^{(3)} = q^{(2)}M = q^{(0)}M^3$

$\ldots$

$q^{(t)} = q^{(0)}M^t$ for all $t \geq 0$

Stationary Distribution

From earlier, we know $q^{(t+1)} = q^{(t)}M$ for all $t \geq 0$.

So, if $q^{(t+1)} = q^{(t)}$, the distribution will never change again.

Stationary Distribution

From earlier, we know $q^{(t+1)} = q^{(t)}M$ for all $t \geq 0$.

So, if $q^{(t+1)} = q^{(t)}$, the distribution will never change again.

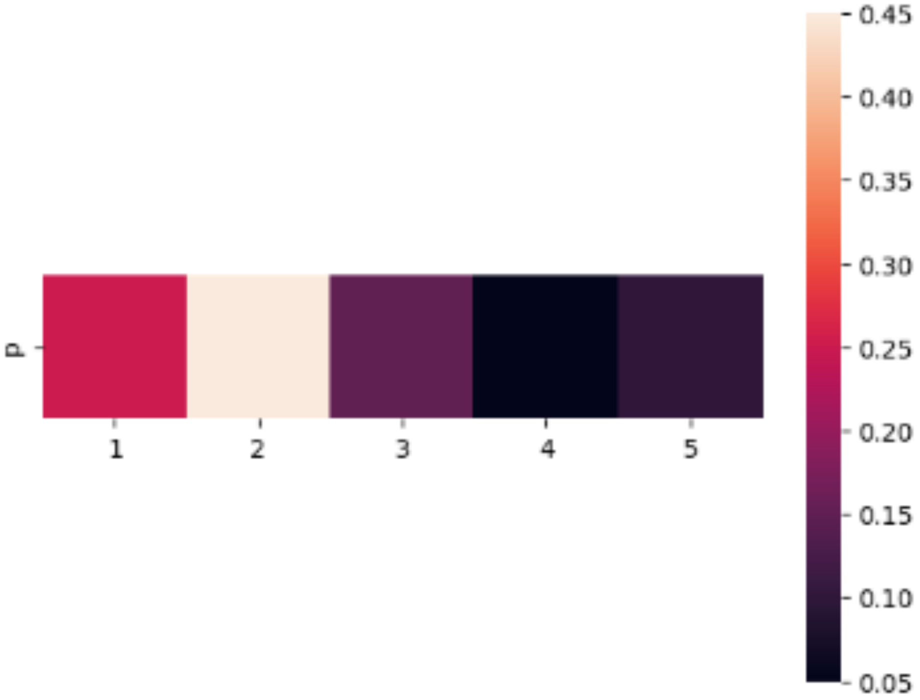
Known as the **stationary distribution**: (if it exists), for a Markov chain, is the n-dimensional row vector π (representing a probability distribution – non-negative and sums to 1) such that $\pi M = \pi$.

Typically happens after a “long time” (mixing time).

Stationary Distribution E.g.

$$v = (0.25, 0.45, 0.15, 0.05, 0.10)$$

Figure 9.6.2: Belief Distribution v at $n = 0$



$$\begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Stationary Distribution E.g.

Figure 9.6.3: Belief Distribution vP at $n = 1$

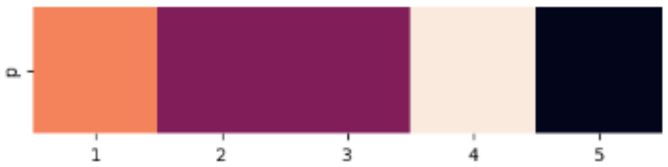


Figure 9.6.4: Belief Distribution vP^5 at $n = 5$

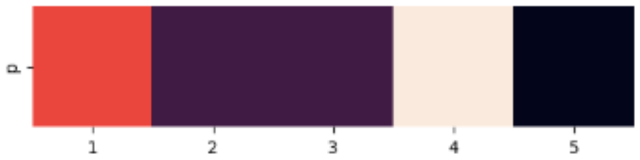


Figure 9.6.5: Belief Distribution vP^{10} at $n = 10$

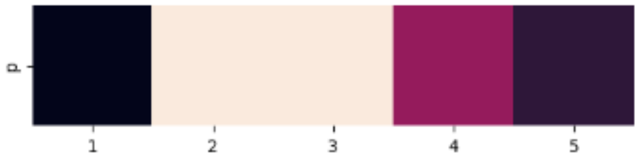
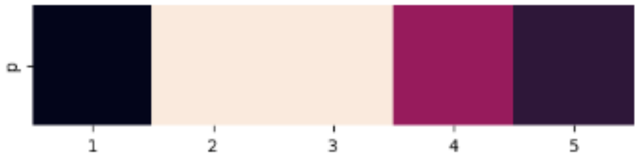


Figure 9.6.6: Belief Distribution vP^{100} at $n = 100$



Solving for the Stationary Distribution

You have a system of equations:

$$\begin{bmatrix}\pi_W & \pi_I & \pi_E\end{bmatrix}\begin{bmatrix}0.2 & 0.8 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0 & 0.7\end{bmatrix} = \begin{bmatrix}\pi_W & \pi_I & \pi_E\end{bmatrix}$$

$$\pi_W + \pi_I + \pi_E = 1$$

As $t \to \infty, q^{(t)} \to \pi$, no matter what $q^{(0)}$ is.

Markov Chains, Generalized

A set of n states $\mathcal{S} = \{s_1, \dots, s_n\}$

The state at time t is denoted by X_t

A $n \times n$ transition matrix M, with $M_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i)$

$q^{(t)} = \left(q_1^{(t)}, \dots, q_n^{(t)}\right)$ where $q_i^{(t)} = \mathbb{P}(X_t = i)$

Transitions: $q^{(t+1)} = q^{(t)} M$ so $q^{(t)} = q^{(0)} M^t$

Fundamental Theorem of Markov Chains

Recall $q^{(t)}$ is the distribution of being at each state at time t computed by $q^{(t)} = q^{(0)}M^{\mathrm{t}}$. As t gets large $q^{(t)} \approx q^{(t+1)}$.

For a Markov chain that is *aperiodic and irreducible** with transition probabilities M and for any starting distribution $q^{(0)}$ over the states

$$\lim_{t\to\infty} q^{(0)}M^t = \pi$$

Where π is the stationary distribution of M.

*Important concepts for practical use, but beyond the scope of 312.

Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo

A modelling technique.

Two common uses:

> Sampling from a (hard) distribution (Bayesian analysis!)

> Approximate solutions to complex optimization problems

Markov Chain Monte Carlo

A modelling technique.

Two common uses:

> Sampling from a (hard) distribution (Bayesian analysis!)

> Approximate solutions to complex optimization problems

Simulate a random walk that moves among possible configurations of a system and will converge to a useful distribution over these configurations.

Knapsack Problem

The 0-1 knapsack problem:

- Given n items with weights $w_1, \ldots, w_n > 0$ and values $v_1, \ldots, v_n > 0$
- Knapsack with weight limit W

Find the most valuable subset of items which satisfy the weight constraint of the knapsack.

The 0-1 knapsack problem

Given n items with weights $w_1, \ldots, w_n > 0$ and values $v_1, \ldots, v_n > 0$

- Knapsack with weight limit W

Find the most valuable subset of items which satisfy the weight constraint of the knapsack.

For e.g.,

Item 1 – 2 lb - $5, Item 2 – 4 lb - $9, Item 3 – 2 lb - $10

Knapsack limit: 4 lb

(Would prefer to choose Item 1 and 3)

MCMC formalization

Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \{0,1\}^n$ represent whether we took item i $(1 \leq i \leq n)$ or not.

We want to maximize the total value: $\sum_{i=1}^{n} v_i x_i$

With weight constraint: $\sum_{i=1}^{n} w_i x_i \leq W$.

Markov Chain

Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ represent whether we took item i $(1 \leq i \leq n)$ or not. We want to maximize the total value: $\sum_{i=1}^{n} v_i x_i$ with weight constraint: $\sum_{i=1}^{n} w_i x_i \leq W$.

Define a Markov chain.

The states are possible solutions.

How many?

Markov Chain

Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ represent whether we took item i $(1 \le i \le n)$ or not. We want to maximize the total value: $\sum_{i=1}^{n} v_i x_i$ with weight constraint: $\sum_{i=1}^{n} w_i x_i \le W$.

Define a Markov chain.

The states are possible solutions.

How many? 2^n!

Markov Chain

Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \{0,1\}^n$ represent whether we took item i $(1 \le i \le n)$ or not. We want to maximize the total value: $\sum_{i=1}^{n} v_i x_i$ with weight constraint: $\sum_{i=1}^{n} w_i x_i \le W$.

Define a Markov chain.

The states are possible solutions.

How many? 2^n!

Implicit TCM. Higher probability of “good” solution.

Markov Chain

Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ represent whether we took item i $(1 \leq i \leq n)$ or not. We want to maximize the total value: $\sum_{i=1}^{n} v_i x_i$ with weight constraint: $\sum_{i=1}^{n} w_i x_i \leq W$.

Define a Markov chain.

The states are possible solutions.

How many? 2^n!

Implicit TCM. Higher probability of "good" solution.

One "good" definition: solutions that satisfy the weight constraint.

Knapsack: Starter Algorithm

Algorithm 3 (First Attempt) MCMC for 0-1 Knapsack Problem

x ← vector of n zeros, where x_i is a binary vector in $\{0,1\}^n$ which represents whether or not we have item i. (Initially, start with an empty knapsack).
best_x ← x
for $t = 1, \ldots,$ NUM_ITER **do**
    $k \leftarrow$ a random integer in $\{1, 2, \ldots, n\}$.
    new_x ← x but with x[k] flipped ($0 \to 1$ or $1 \to 0$).
    if new_x satisfies weight constraint **then**
        x ← new_x
    if value(x) > value(best_x) **then**
        best_x ← x

MCMC Strategy (for optimization)

Define a Markov chain:

> States = possible solutions

> Implicitly defined TPM (higher probabilities on "good" solutions)

Run MCMC: Simulate Markov Chain for many iterations until we reach a "good" state (transitioning according to the TPM until we reach our stationary distribution).

A Better Algorithm

Algorithm 5 MCMC for 0-1 Knapsack Problem

subset $\leftarrow$ vector of n zeros, where subset is always a binary vector in $\{0,1\}^n$ which represents whether or not we have each item. (Initially, start with an empty knapsack).
best_subset $\leftarrow$ vector of n zeros
for $t = 1, \ldots,$ NUM_ITER **do**
    $k \leftarrow$ a random uniform integer in $\{1, 2, \ldots, n\}$.
    new_subset $\leftarrow$ subset but with subset[k] flipped ($0 \rightarrow 1$ or $1 \rightarrow 0$).
    $\Delta \leftarrow$ value(new_subset) $-$ value(subset)
    if new_subset satisfies weight constraint (total weight $\leq W$) **then**
        if $\Delta > 0$ OR ($T > 0$ AND Unif$(0,1) < e^{\Delta/T}$) **then**
            subset $\leftarrow$ new_subset
    if value(subset) $>$ value(best_subset) **then**
        best_subset $\leftarrow$ subset

A Better Algorithm

Check multiple "better" features:
- Does it satisfy the weight constraint?
- Is it a solution of higher value?

Algorithm 5 MCMC for 0-1 Knapsack Problem

```
subset ← vector of n zeros, where subset is always a b ... hether
    or not we have each item. (Initially, start with an emp
best_subset ← vector of n zeros
for t = 1, ..., NUM_ITER do
    k ← a random uniform integer in {1, 2, ..., n}.
    new_subset ← subset but with subset[k] flipped (0 → 1 or 1 → 0).
    Δ ← value(new_subset) − value(subset)
    if new_subset satisfies weight constraint (total weight ≤ W) then
        if Δ > 0 OR (T > 0 AND Unif(0, 1) < e^{Δ/T}) then
            subset ← new_subset
    if value(subset) > value(best_subset) then
        best_subset ← subset
```

A Better Algorithm

Exploitation: Transition to higher total values

Exploration: Random transitions occasionally, to avoid local optima

Algorithm 5 MCMC for 0-1 Knapsack Problem

subset $\leftarrow$ vector of n zeros, where subset is always a b hether or not we have each item. (Initially, start with an emp

best_subset $\leftarrow$ vector of n zeros

for $t = 1, \ldots,$ NUM_ITER **do**

  $k \leftarrow$ a random uniform integer in $\{1, 2, \ldots, n\}$.

  new_subset $\leftarrow$ subset but with subset$[k]$ flipped ($0 \rightarrow 1$ or $1 \rightarrow 0$).

  $\Delta \leftarrow$ value(new_subset) $-$ value(subset)

  if new_subset satisfies weight constraint (total weight $\leq W$) **then**

    if $\Delta > 0$ OR $\boxed{(T > 0 \text{ AND } \mathrm{Unif}(0, 1) < e^{\Delta/T})}$ **then**

      subset $\leftarrow$ new_subset

  if value(subset) $>$ value(best_subset) **then**

    best_subset $\leftarrow$ subset

Other MCMC Applications

Other hard optimization questions, like the travelling salesman problem.

PageRank

> Search engines rank results based on some 'quality' (determined by hyperlink analysis)

> Nodes are webpages, edges weighted by # of edges to node of total edges

> PageRank of a webpage is its stationary probability

Of course, the algorithm has greatly changed since then! And is more of a secret 😅