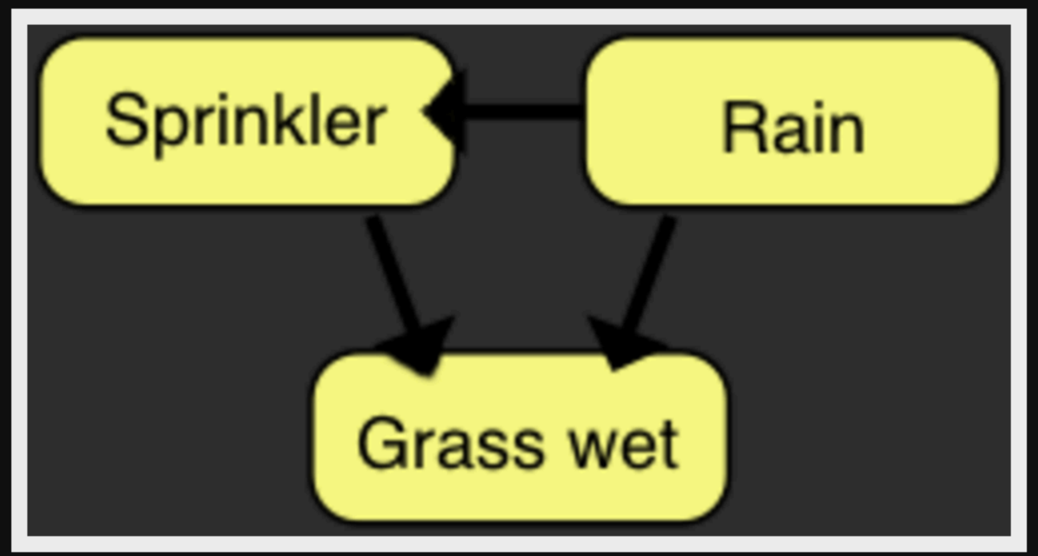
Bayesian Networks

David Duffrin

Bayesian Networks

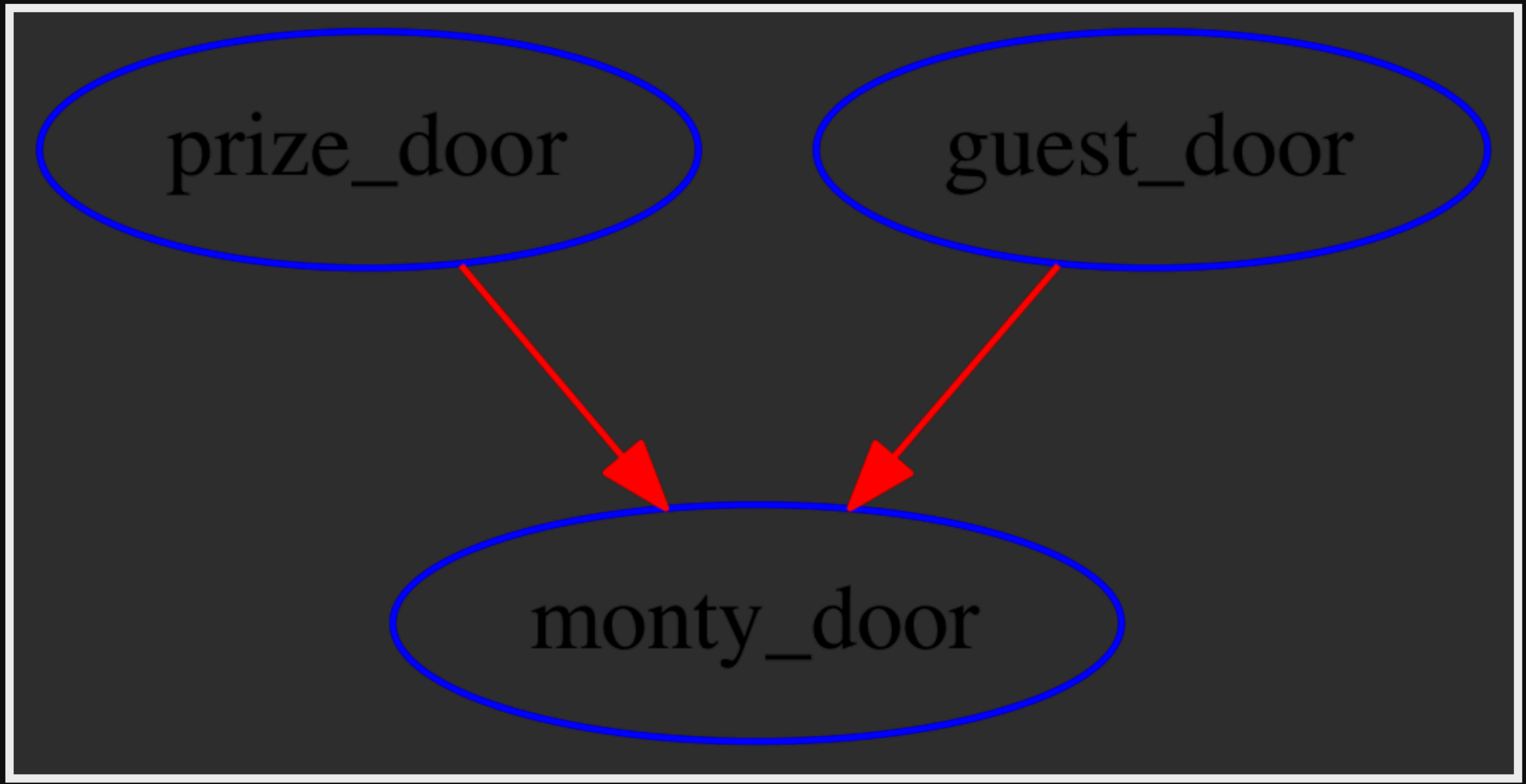
- Graphical Model
- Probabilistic (conditional)
- Directed
- Acyclic



Monty Hall

- There are three doors with one containing a prize
- Three phases:
  1. Choose a door
  2. Monty opens a door that you didn't choose AND that doesn't contain the prize (conditional)
  3. Choose to stay or switch

Monty's Bayes Net



Monty Hall Definitions

- Choose Door (C#)
- Monty opens Door (M#)
- Prize Door (P#)

Monty Hall Solution

1) There is a 1/3 chance that the prize is behind any door and 1/3 chance we choose any door

2) Probabilities of Monty choosing door n:

$$P(Mn \mid Cn) = 0$$

$$P(Mn \mid !Cn) = \frac{1}{2}$$

3) Probability that the prize is behind stay (x) or switch (y), with the opened door (z):

$$P(Px \mid Cx,\, Mz) = \frac{P(Mz \mid Cx,\, Px)\, P(Px \mid Cx)}{P(Mz \mid Cx)}$$

$$= \frac{\frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$$

$$P(Py \mid Cx,\, Mz) = \frac{P(Mz \mid Cx,\, Py)\, P(Py \mid Cx)}{P(Mz \mid Cx)}$$

$$= \frac{1 \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$$

Computer Science Aspect

- There are packages for graphical models and Bayesian Statistics (pomegranate)
- Allow you to input the states, transitions, and probabilities to generate probabilities for new examples
- Impute most likely outcome
- Fit based on Monte Carlo simulation (or real world data)

Khan Academy's Problem

- How to estimate Proficiency (Mastery) based on potentially incomplete exercise data
- Proficiency is the likelihood of getting the next problem correct (> threshold)
- Want to minimize the number of problems needed to complete to reach proficiency (so as to not waste time)

Khan Academy's Subject Structure

Topic

$$\%\ \text{Progress} = \frac{4\times(\#\ \text{skills mastered}) + 3\times(\#\ \text{skills at level two}) + 2\times(\#\ \text{skills at level one}) + 1\times(\#\ \text{skills practiced})}{4 \times \text{total}\ \#\ \text{skills in mission}}$$

Skill

- Mastered
- Level Two
- Level One
- Practiced
- Not Started (NA)

Exercise

- Correct
- Incorrect
- Not Started (NA)

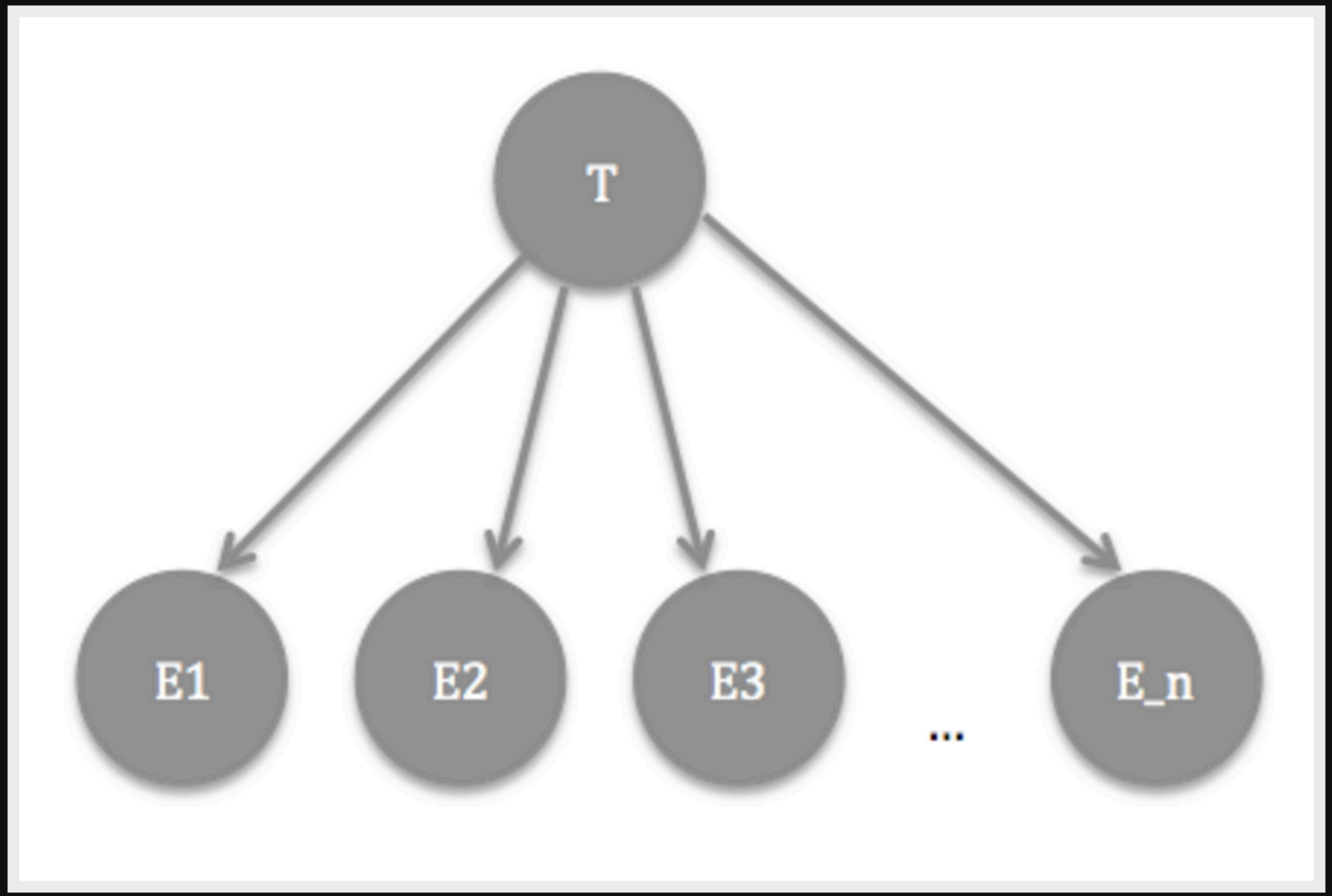
Previous Attempts to Estimate Skill

- Counting Not Started Exercises as Incorrect
  - Requires users to complete exercises even if Mastery is implied
- Streak-based model
  - A user that got 10 correct and then 1 incorrect would be seen the same as a user that never started

Bayesian Attempt to Estimate Skill

- Each exercise independently influences proficiency
- More difficult than Monty Hall because conditional probabilities are unknown
  - Difficult with incomplete data

Khan's Bayes Net



In [54]:

```python
import pandas as pd
bnet = pd.read_csv('bnet.csv')
```

In [110]:

```python
bnet.head()
```

Out[110]:

	addition_1	addition_2	addition_3	additi
0	0	0	0	0
1	0	-1	1	0
2	0	0	0	-1
3	1	1	1	1
4	1	-1	-1	-1

5 rows × 21 columns

Problem

- Proficiency is a hidden variable

Solution

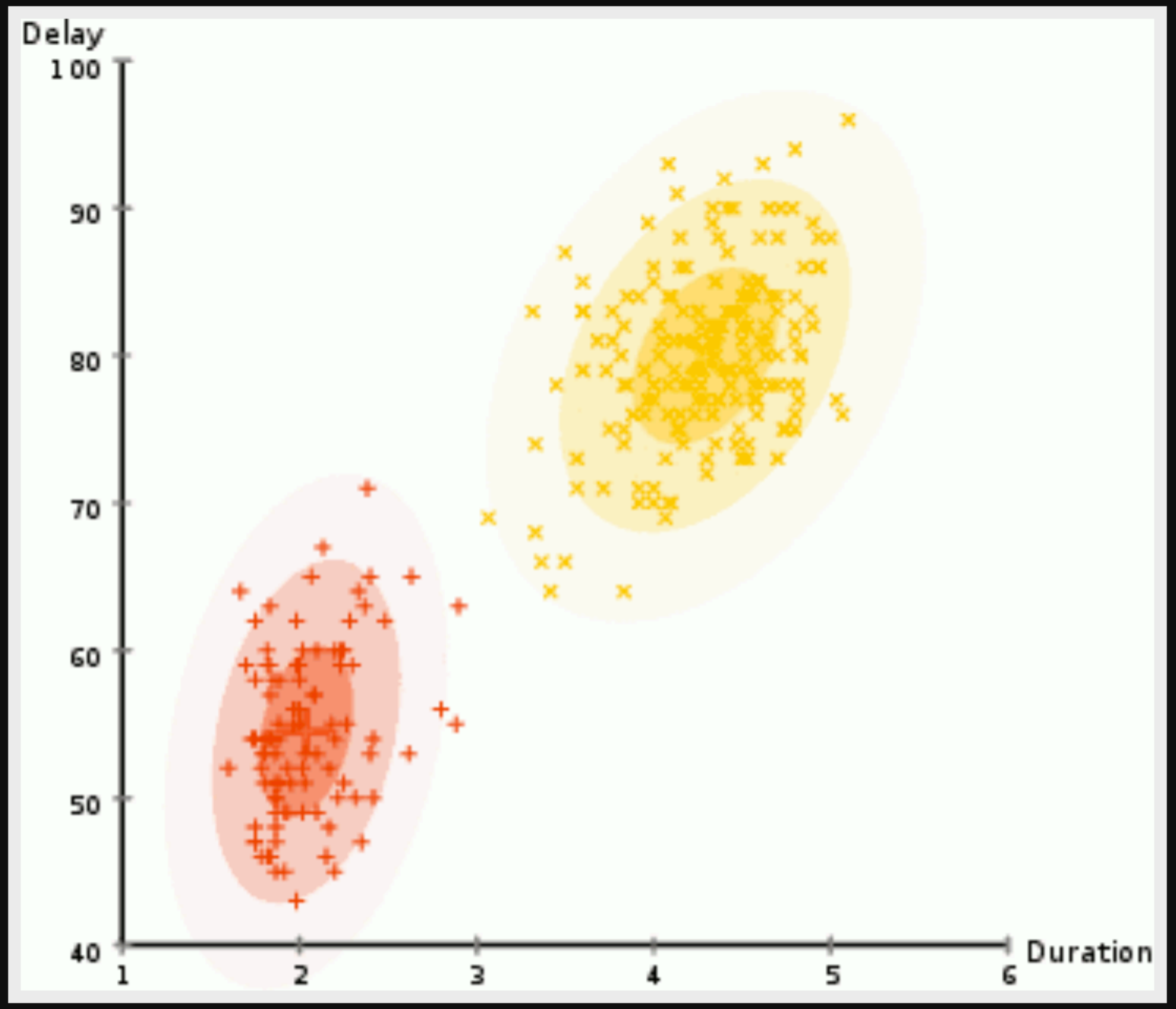
- Expectation Maximization algorithm to impute the topic proficiency based on the given exercise data

EM Algorithm

An iterative method for finding maximum likelihood estimates with two steps:

- Expectation - impute hidden node (categorize points - clustering)
- Maximization - computes parameters maximizing the expected log-likelihood
  - Adjusts the conditional probabilities of the exercise and proficiency

EM Example



In [105]:

```
from numpy import asmatrix, asarray, ones, zeros, mean, sum, arange, p
 dot, loadtxt
from numpy.random import random, randint
#import pickle

MISSING_VALUE = -1 # a constant I will use to denote missing integer
 values

# Impute hidden T - Expectation
def impute_hidden_node(E, I, theta, sample_hidden):

    theta_T, theta_E = theta

    # calculate the unnormalized probability associated with the hidd
en unit being a 0
    theta_E_wide = asarray( ones([E.shape[0],1]) * asmatrix(theta_E[:,
]) )
    p_vis_0 = I * (theta_E_wide * E + (1-theta_E_wide) * (1-E)) + (I=
=0)*1
    prob_0_unnorm = (1-theta_T) * prod(p_vis_0, 1)

    # calculate the unnormalized probability associated with the hidd
en unit being a 1
    theta_E_wide = asarray( ones([E.shape[0],1]) * asmatrix(theta_E[:,
]) )
    p_vis_1 = I * (theta_E_wide * E + (1-theta_E_wide) * (1-E)) + (I=
=0)*1
    prob_1_unnorm = theta_T * prod(p_vis_1, 1)

    hidden = prob_1_unnorm / (prob_0_unnorm + prob_1_unnorm)

    if sample_hidden:
        # set the hidden unit to a 0 or 1 instead of a probability of
 activation
        hidden = (hidden > random( hidden.shape ))*1

    return hidden
```

Code Overview

Originally starts with random conditional probabilities and proficiency probabilities and slowly converges toward the real values.

1) Reads in data (or simulates data with NA spaces)

2) Learn Phase - iterates X times through the EM cycle

- Expectation - infers proficiency for each student based on current conditional probabilities
- Maximization - recalculates the conditional probabilities based on the new proficiencies
  - New probabilities are used by the expectation phase on the next pass

3) Prints results

```
Ending State:
T          0: 0.455500      1:0.544500
E0  T=0    0: 0.321724      1:0.678276
E0  T=1    0: 0.000000      1:1.000000
E1  T=0    0: 0.294616      1:0.705384
E1  T=1    0: 0.000000      1:1.000000
E2  T=0    0: 0.717860      1:0.282140
E2  T=1    0: 0.529049      1:0.470951
E3  T=0    0: 0.723868      1:0.276132
E3  T=1    0: 0.168327      1:0.831673
E4  T=0    0: 0.356871      1:0.643129
E4  T=1    0: 0.354816      1:0.645184
Goal:
T          0: 0.250000      1:0.750000
E0  T=0    0: 0.450000      1:0.550000
E0  T=1    0: 0.050000      1:0.950000
E1  T=0    0: 0.400000      1:0.600000
E1  T=1    0: 0.050000      1:0.950000
E2  T=0    0: 0.760000      1:0.240000
E2  T=1    0: 0.580000      1:0.420000
E3  T=0    0: 0.870000      1:0.130000
E3  T=1    0: 0.280000      1:0.720000
E4  T=0    0: 0.380000      1:0.620000
E4  T=1    0: 0.340000      1:0.660000
```

```
E7 T=0  0: 0.411471     1:0.588529
E7 T=1  0: 0.105525     1:0.894475
E8 T=0  0: 0.058239     1:0.941761
E8 T=1  0: 0.002363     1:0.997637
E9 T=0  0: 0.140845     1:0.859155
E9 T=1  0: 0.000000     1:1.000000
E10 T=0 0: 0.104271     1:0.895729
E10 T=1 0: 0.000000     1:1.000000
E11 T=0 0: 0.427445     1:0.572555
E11 T=1 0: 0.104724     1:0.895276
E12 T=0 0: 0.455013     1:0.544987
E12 T=1 0: 0.115059     1:0.884941
E13 T=0 0: 0.550388     1:0.449612
E13 T=1 0: 0.184305     1:0.815695
E14 T=0 0: 0.151351     1:0.848649
E14 T=1 0: 0.027602     1:0.972398
E15 T=0 0: 0.170370     1:0.829630
E15 T=1 0: 0.069658     1:0.930342
E16 T=0 0: 0.696386     1:0.303614
E16 T=1 0: 0.366569     1:0.633431
E17 T=0 0: 0.740000     1:0.260000
E17 T=1 0: 0.381144     1:0.618856
E18 T=0 0: 0.746154     1:0.253846
E18 T=1 0: 0.326948     1:0.673052
E19 T=0 0: 0.697917     1:0.302083
E19 T=1 0: 0.280763     1:0.719237
E20 T=0 0: 0.782609     1:0.217391
E20 T=1 0: 0.273148     1:0.726852
```

Example User

- Let's look at an example of if a user answers E20 correctly and no other exercise
- Note: If we put in more exercises, the formula becomes very long

$$P(T_1 \mid E20_1) = \frac{P(E20_1 \mid T_1)\,P(T_1)}{P(E20_1)}$$

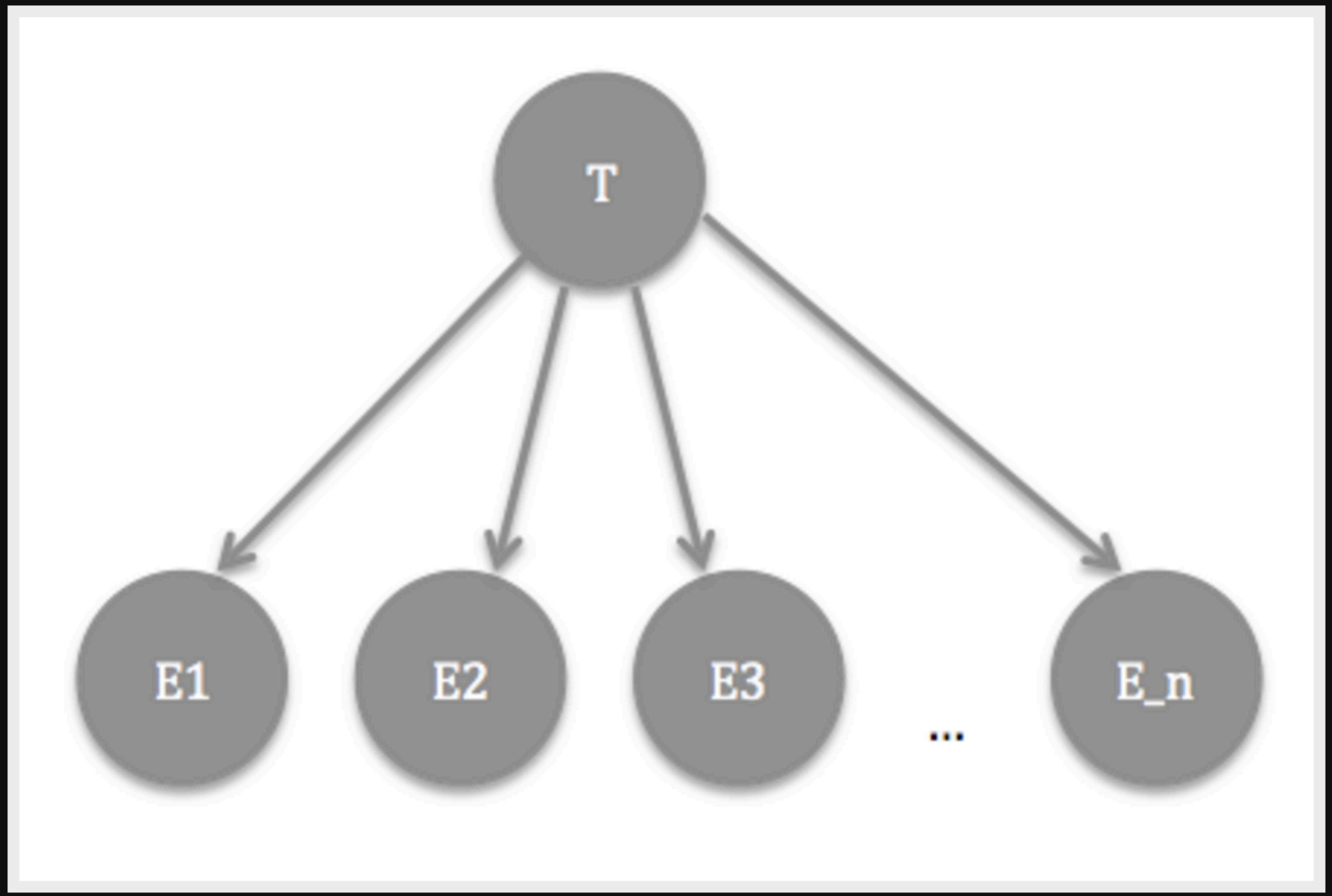
$$= \frac{0.726852 \cdot 0.764442}{0.764442 \cdot 0.726852 + 0.235558 \cdot 0.217391}$$

$$= 0.915616$$

Validation

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -4.51475    1.45531  -3.102  0.00192 **
ewma_3_capped     1.26465    0.74014   1.709  0.08751 .
ewma_10_capped    1.71547    2.04739   0.838  0.40210
current_streak    0.05308    0.05293   1.003  0.31591
log_num_done      0.23890    0.11430   2.090  0.03661 *
log_num_missed    0.01198    0.27270   0.044  0.96495
pct_correct       0.51366    0.20091   2.557  0.01057 *
E                 4.32014    0.09779  44.178  < 2e-16 ***
T                -0.33073    0.05696  -5.806 6.39e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
```

Khan's Bayes Net



Khan Academy's Other Problems

- Best ordering of topics to minimize time and maximize success
  - Originally based off of the school curriculum
- What software intervention would help the student
  - ie. Wrong answer on an exercise, so the website displays a video to rewatch that will help

Sources

http://derandomized.com/post/20009997725/bayes-net-example-wi
python-and-khanacademy

https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization

https://github.com/jmschrei/pomegranate