Weather Prediction with Markov Chains

Math 42 Final Project
Department of Mathematics, UCLA

Junwon Choi
Brian Chau
Yifan Jiang
Mingyeong Kim
Nalin Chopra

Table of contents

01

Problem description

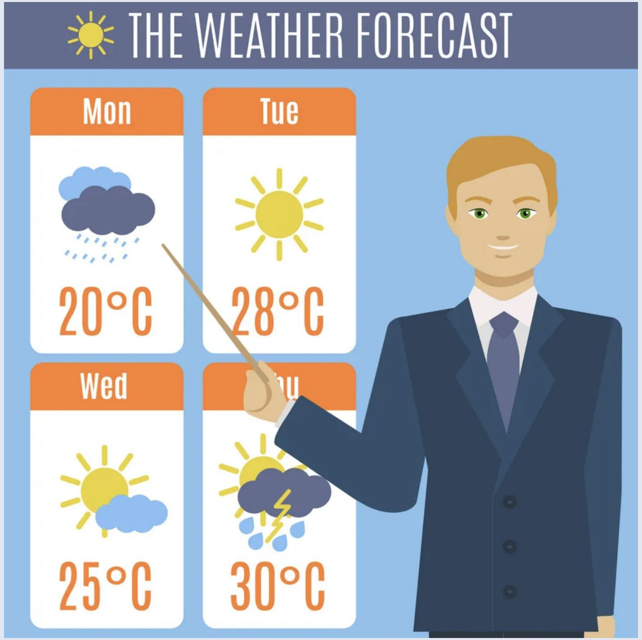
Present the problem we are attempting to solve with background information

How can we predict the weather?

Weather app? Guessing? News Forecasting?

Problem description

- The only method to predict the weather was to use the past data of local experience
- Mathematical model + the data = weather prediction



Three questions about weather prediction

- Which mathematical model is most appropriate for weather prediction?
- How can we fit the selected model to predict weather in a particular location?
- How accurate/significant is the model? How does it perform when fitted to different locations?

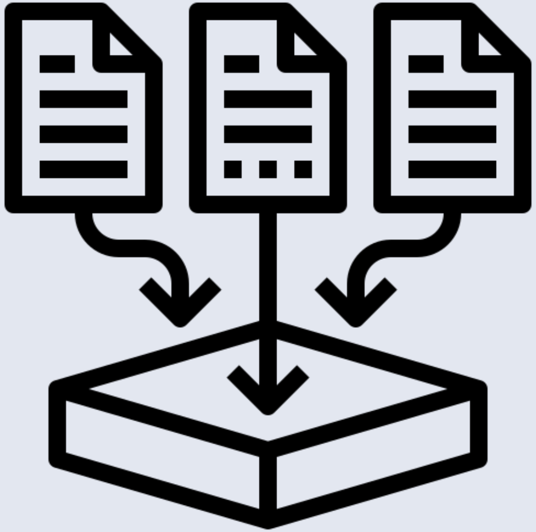
02

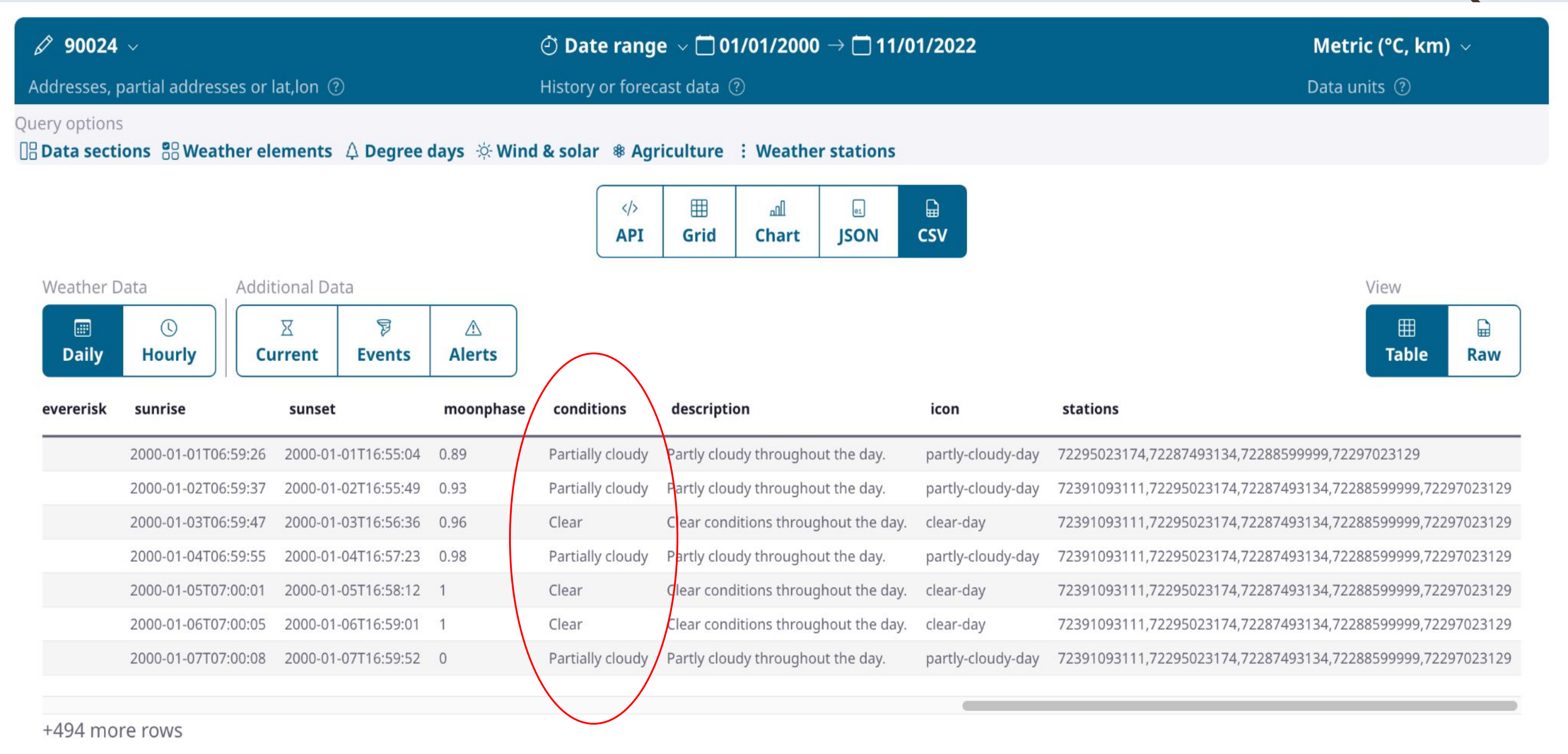
Simplifications

Explain the simplified way of the original problem. Justify the assumptions.

Data collection

- Weather data was collected from "https://www.visualcrossing.com"
  - UCLA (zip code - 90024)
  - From the year 2000 and 2022
- Task:
  - Find categorical variables for discrete Markov chain





evererisk	sunrise	sunset	moonphase	conditions	description	icon	stations
	2000-01-01T06:59:26	2000-01-01T16:55:04	0.89	Partially cloudy	Partly cloudy throughout the day.	partly-cloudy-day	72295023174,72287493134,72288599999,72297023129
	2000-01-02T06:59:37	2000-01-02T16:55:49	0.93	Partially cloudy	Partly cloudy throughout the day.	partly-cloudy-day	72391093111,72295023174,72287493134,72288599999,72297023129
	2000-01-03T06:59:47	2000-01-03T16:56:36	0.96	Clear	Clear conditions throughout the day.	clear-day	72391093111,72295023174,72287493134,72288599999,72297023129
	2000-01-04T06:59:55	2000-01-04T16:57:23	0.98	Partially cloudy	Partly cloudy throughout the day.	partly-cloudy-day	72391093111,72295023174,72287493134,72288599999,72297023129
	2000-01-05T07:00:01	2000-01-05T16:58:12	1	Clear	Clear conditions throughout the day.	clear-day	72391093111,72295023174,72287493134,72288599999,72297023129
	2000-01-06T07:00:05	2000-01-06T16:59:01	1	Clear	Clear conditions throughout the day.	clear-day	72391093111,72295023174,72287493134,72288599999,72297023129
	2000-01-07T07:00:08	2000-01-07T16:59:52	0	Partially cloudy	Partly cloudy throughout the day.	partly-cloudy-day	72391093111,72295023174,72287493134,72288599999,72297023129

+494 more rows

Analysis steps

- All seasons + Simplified (Long time frame)
- All seasons + Simplified (Short time frame)
- All seasons + 6 different weather conditions (Long time frame)
- All seasons + 6 different weather conditions (Short time frame)
- By season + 6 different weather conditions (Long time frame)
- All seasons + Simplified applied to UCSD/NYU (Long time frame)
    - For validation

Assumptions and Limitations

Assumptions:

- Weather condition for a day only depend on weather condition of the day before

Limitations:

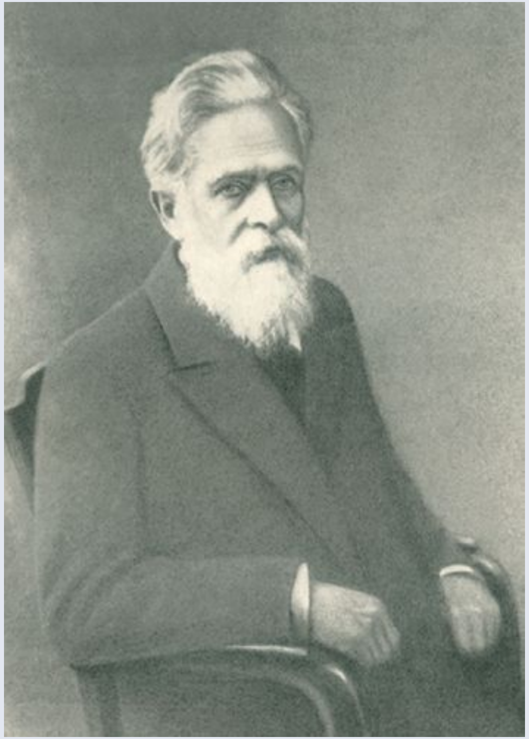
- Failure to account for other factors like temperature and humidity changes

03

Mathematical model

Markov chain explanation with the standard mathematical paradigm

What are Markov Chains?



Andrey Andreyevich Markov

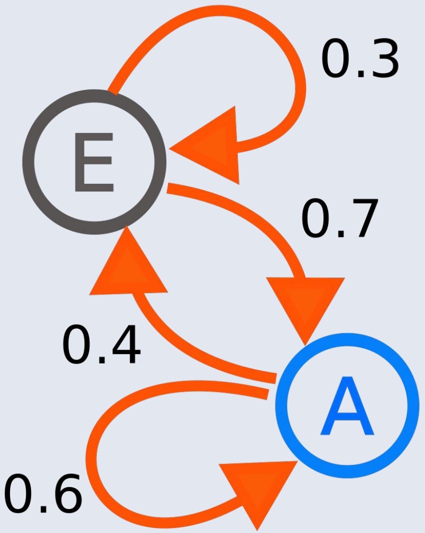
Markov chain

- A stochastic system that describes the sequence of possible events
- Properties:
  - The Markov assumption:

$$P(s_t|s_{t-1}, s_{t-2}, \ldots, s_0) = P(s_t|s_{t-1})$$

  - Conservation: the sum of the probabilities out of a state is equal to 1

Markov chain visual sample



Transition Matrix

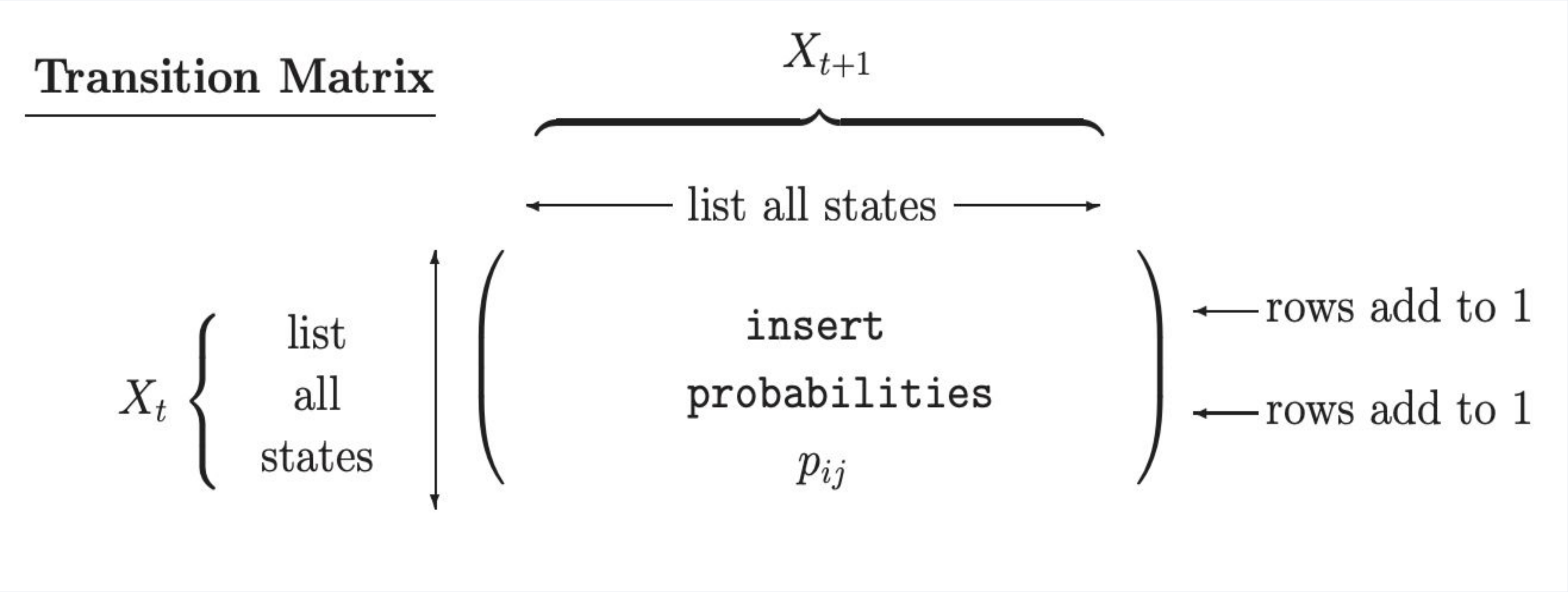


Image Source: https://www.stat.auckland.ac.nz/~fewster/325/notes/ch8.pdf

Application to Weather

- S = {Clear; Partially cloudy; Overcast; Rain; Rain, Partially cloudy; Rain, Overcast}
- A = {Rain, No Rain}

$$
\mathrm{T}_{\text{simplified}} = \begin{array}{cc} & \begin{array}{cc} \text{Rain (R)} & \text{No Rain (NR)} \end{array} \\ \begin{array}{c} \text{Rain (R)} \\ \text{No Rain (NR)} \end{array} & \begin{pmatrix} P(R|R) & P(NR|R) \\ P(R|NR) & P(NR|NR) \end{pmatrix} \end{array}
$$

04

Solution

Techniques to solve the mathematical problem

Calculating Transition Matrix

Training Data:

Day 1: Rain

Day 2: Rain

Day 3: No Rain

Day 4: Rain

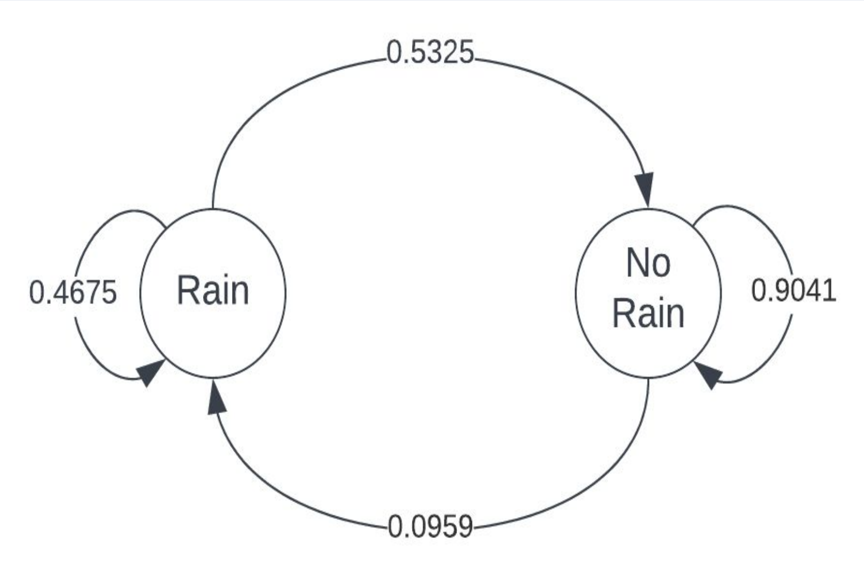
...

Day N: No Rain

- P(Rain | Rain)
- P(No Rain | Rain)
- P(Rain | No Rain)
- P(No Rain | No Rain)

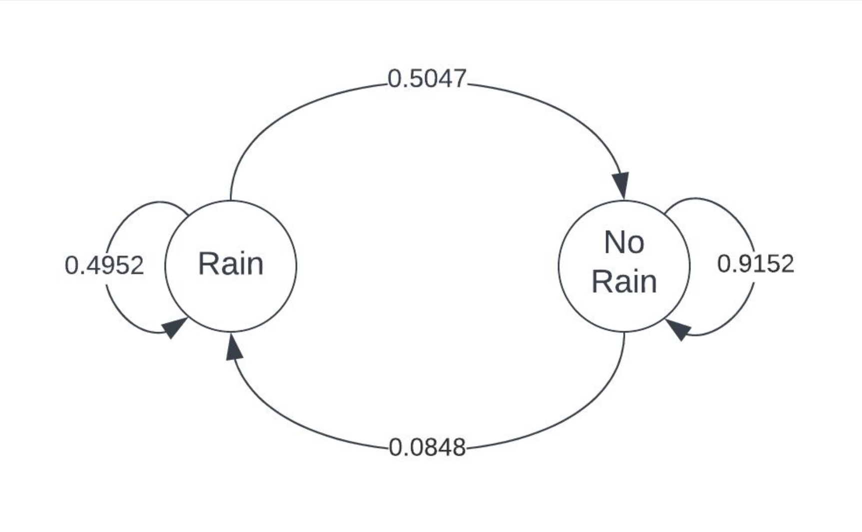
All Seasons, simplified analysis (long time frame)

$$T = \begin{array}{l} \\ \text{Rain (R)} \\ \text{No Rain (NR)} \end{array} \begin{array}{c} \begin{array}{cc} \text{Rain (R)} & \text{No Rain (NR)} \end{array} \\ \begin{pmatrix} 0.467 & 0.533 \\ 0.096 & 0.904 \end{pmatrix} \end{array}$$



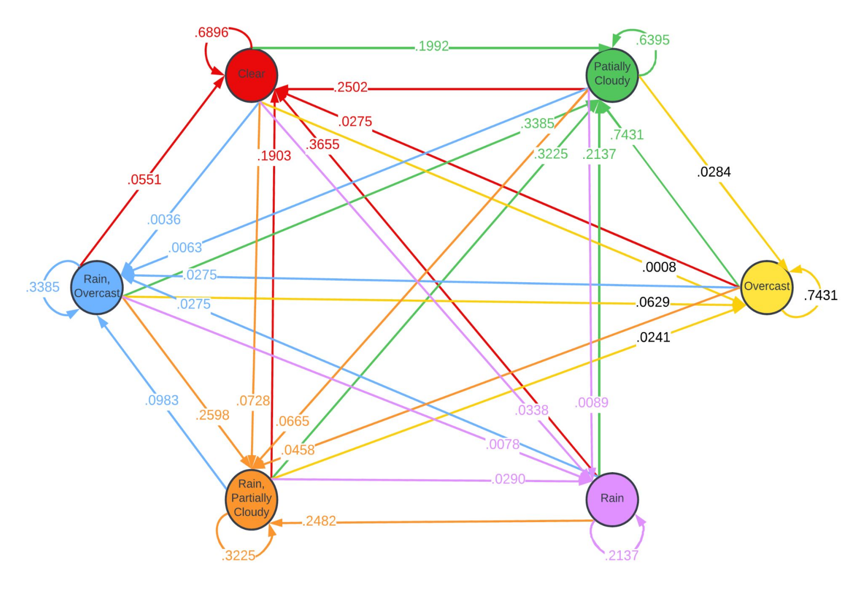
All Seasons, simplified analysis (short time frame)

$$T = \begin{array}{r} \\ \text{Rain (R)} \\ \text{No Rain (NR)} \end{array} \begin{array}{c} \begin{array}{cc} \text{Rain (R)} & \text{No Rain (NR)} \end{array} \\ \begin{pmatrix} 0.495 & 0.505 \\ 0.085 & 0.915 \end{pmatrix} \end{array}$$



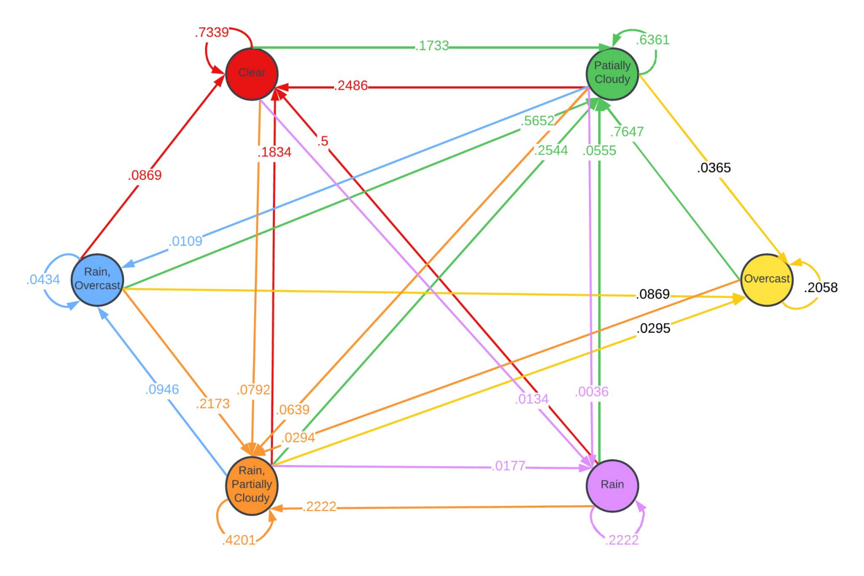
All Seasons, 6 weather conditions (long time frame)

$$T = \begin{array}{c} \\ C \\ PC \\ OV \\ R \\ RPC \\ ROV \end{array} \begin{array}{c} \begin{array}{cccccc} C & PC & OV & R & RPC & ROV \end{array} \\ \begin{pmatrix} 0.690 & 0.199 & 0.001 & 0.034 & 0.073 & 0.004 \\ 0.250 & 0.640 & 0.028 & 0.009 & 0.067 & 0.006 \\ 0.028 & 0.743 & 0.156 & 0.0 & 0.046 & 0.028 \\ 0.366 & 0.214 & 0.0 & 0.145 & 0.248 & 0.028 \\ 0.190 & 0.323 & 0.0242 & 0.029 & 0.335 & 0.098 \\ 0.055 & 0.339 & 0.0630 & 0.008 & 0.260 & 0.276 \end{pmatrix} \end{array}$$



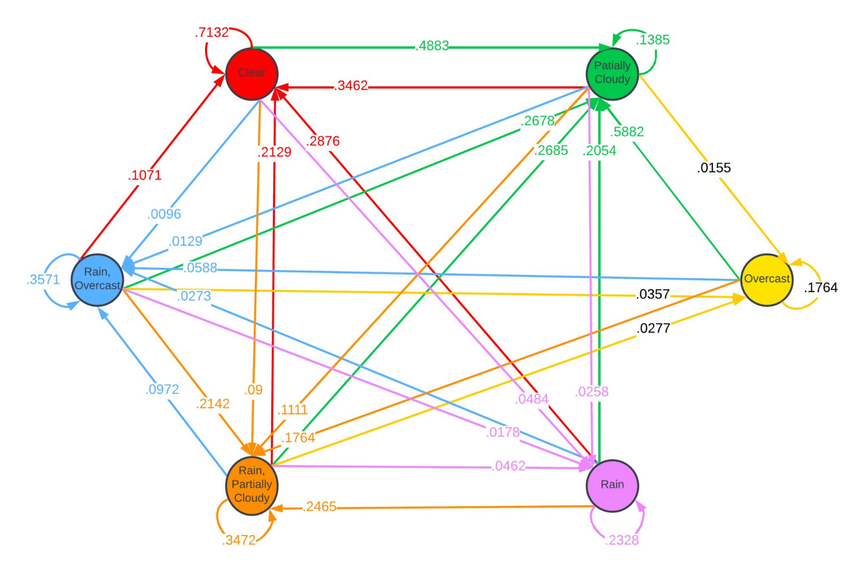
All Seasons, 6 weather conditions (short time frame)

$$
T = \begin{array}{c} \\ C \\ PC \\ OV \\ R \\ RPC \\ ROV \end{array}
\begin{array}{c}
\begin{array}{cccccc} C & PC & OV & R & RPC & ROV \end{array} \\
\begin{pmatrix}
0.734 & 0.173 & 0.0 & 0.013 & 0.079 & 0.0 \\
0.249 & 0.636 & 0.037 & 0.004 & 0.064 & 0.011 \\
0.0 & 0.765 & 0.206 & 0.0 & 0.029 & 0.0 \\
0.5 & 0.056 & 0.0 & 0.222 & 0.222 & 0.0 \\
0.183 & 0.254 & 0.0296 & 0.018 & 0.420 & 0.095 \\
0.087 & 0.565 & 0.087 & 0. & 0.217 & 0.043
\end{pmatrix}
\end{array}
$$



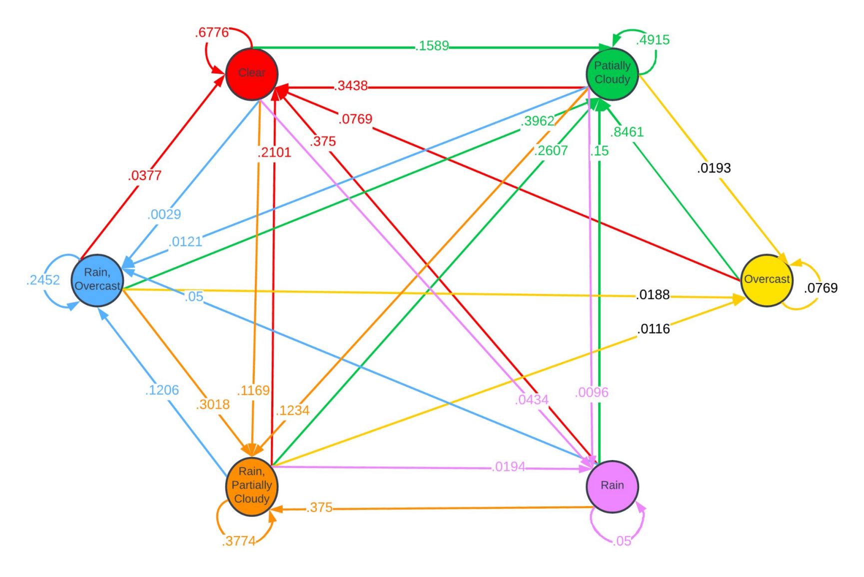
6 weather conditions, Fall

$$
T = \begin{array}{c c}
 & \begin{array}{cccccc} C & PC & OV & R & RPC & ROV \end{array} \\
\begin{array}{c} C \\ PC \\ OV \\ R \\ RPC \\ ROV \end{array} &
\begin{pmatrix}
0.713 & 0.139 & 0.0 & 0.048 & 0.090 & 0.010 \\
0.346 & 0.488 & 0.016 & 0.026 & 0.111 & 0.013 \\
0.0 & 0.588 & 0.176 & 0.0 & 0.176 & 0.059 \\
0.288 & 0.205 & 0.0 & 0.233 & 0.247 & 0.027 \\
0.213 & 0.269 & 0.028 & 0.046 & 0.347 & 0.097 \\
0.107 & 0.268 & 0.036 & 0.018 & 0.214 & 0.357
\end{pmatrix}
\end{array}
$$



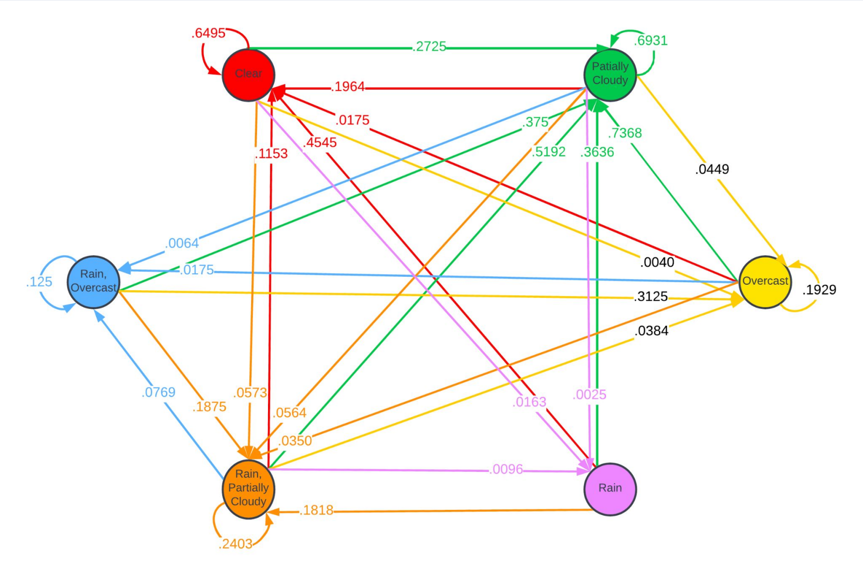
6 weather conditions, Winter

$$
T = \begin{array}{c c}
 & \begin{array}{cccccc} \text{C} & \text{PC} & \text{OV} & \text{R} & \text{RPC} & \text{ROV} \end{array} \\
\begin{array}{l} \text{C} \\ \text{PC} \\ \text{OV} \\ \text{R} \\ \text{RPC} \\ \text{ROV} \end{array} &
\begin{pmatrix}
0.677 & 0.158 & 0.0 & 0.043 & 0.116 & 0.002 \\
0.343 & 0.491 & 0.019 & 0.009 & 0.123 & 0.012 \\
0.076 & 0.846 & 0.076 & 0.0 & 0.0 & 0.0 \\
0.375 & 0.15 & 0.0 & 0.05 & 0.375 & 0.05 \\
0.210 & 0.261 & 0.012 & 0.019 & 0.377 & 0.121 \\
0.038 & 0.396 & 0.019 & 0.0 & 0.302 & 0.245
\end{pmatrix}
\end{array}
$$



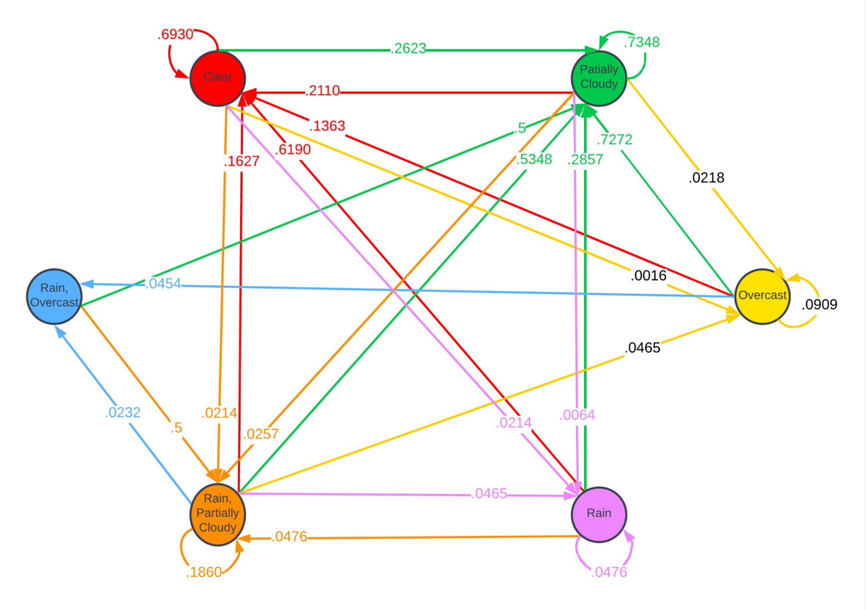
6 weather conditions, Spring

$$
T = \begin{array}{c} \\ C \\ PC \\ OV \\ R \\ RPC \\ ROV \end{array}
\begin{array}{c}
\begin{array}{cccccc} \mathrm{C} & \mathrm{PC} & \mathrm{OV} & \mathrm{R} & \mathrm{RPC} & \mathrm{ROV} \end{array} \\
\begin{pmatrix}
0.649 & 0.272 & 0.004 & 0.016 & 0.057 & 0.0 \\
0.196 & 0.693 & 0.044 & 0.002 & 0.056 & 0.006 \\
0.017 & 0.736 & 0.192 & 0.0 & 0.035 & 0.017 \\
0.454 & 0.363 & 0.0 & 0.0 & 0.181 & 0.0 \\
0.115 & 0.519 & 0.038 & 0.009 & 0.240 & 0.076 \\
0.0 & 0.375 & 0.312 & 0.0 & 0.187 & 0.125
\end{pmatrix}
\end{array}
$$



6 weather conditions, Summer

$$T = \begin{array}{cc} & \begin{array}{cccccc} \mathrm{C} & \mathrm{PC} & \mathrm{OV} & \mathrm{R} & \mathrm{RPC} & \mathrm{ROV} \end{array} \\ \begin{array}{c} \mathrm{C} \\ \mathrm{PC} \\ \mathrm{OV} \\ \mathrm{R} \\ \mathrm{RPC} \\ \mathrm{ROV} \end{array} & \begin{pmatrix} 0.693 & 0.262 & 0.001 & 0.021 & 0.021 & 0.0 \\ 0.211 & 0.734 & 0.021 & 0.006 & 0.025 & 0.0 \\ 0.136 & 0.727 & 0.090 & 0.0 & 0.0 & 0.045 \\ 0.619 & 0.285 & 0.0 & 0.047 & 0.047 & 0.0 \\ 0.162 & 0.534 & 0.046 & 0.046 & 0.186 & 0.023 \\ 0.0 & 0.5 & 0.0 & 0.0 & 0.5 & 0.0 \end{pmatrix} \end{array}$$



UCSD / NYU

- UCSD:

$$T = \begin{array}{c c} & \begin{array}{cc} \text{Rain (R)} & \text{No Rain (NR)} \end{array} \\ \begin{array}{c} \text{Rain (R)} \\ \text{No Rain (NR)} \end{array} & \begin{pmatrix} 0.467 & 0.532 \\ 0.095 & 0.904 \end{pmatrix} \end{array}$$

- NYU:

$$T = \begin{array}{c c} & \begin{array}{cc} \text{Rain (R)} & \text{No Rain (NR)} \end{array} \\ \begin{array}{c} \text{Rain (R)} \\ \text{No Rain (NR)} \end{array} & \begin{pmatrix} 0.467 & 0.532 \\ 0.095 & 0.904 \end{pmatrix} \end{array}$$

Same transition matrices as UCLA!

Applying Transition Matrix

Transition Matrix → **Testing Data**

Applying Transition Matrix

Testing Data + Predictions

Date	Original Condition	Predicted Condition
Test Start Date ... Test End Date	Already downloaded	Appended at each iteration

Prediction process

At each step:

For example, with an all season simplified model (long time frame):

Let Day 1 of the model be a 'No Rain' day, which would give a vector $V = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. We take transition matrix $T = \begin{pmatrix} 0.467 & 0.533 \\ 0.096 & 0.904 \end{pmatrix}$.

- Multiply V and the transpose of T using a Python function.
- This outputs a vector with two elements containing probabilities of 'Rain' or 'No Rain' given the past day is 'No Rain'.
- Markov chain model's key feature is randomness: Python function to use the two probabilities of the transposed second row as inputs to make a random choice.
- This choice determines whether Day 2 will be 'Rain' or 'No Rain'.
- Repeat this step by a desired amount.

```python
def predict_weather_simplified(test_data):
    state = {0:'rain', 1:'no_rain'}
    n = len(test_data) #how many steps to test
    start_state = 1 #1 = No Rain
    test_result = test_data.copy()

    prev_state = start_state
    result = []
    result.append(state[start_state])
    while n-1:
        curr_state = np.random.choice([0,1], p=t_array[prev_state]) #taking the probability from the transition matrix
        result.append(state[curr_state])
        prev_state = curr_state
        n -= 1

    # curr_state = np.random.choice([0,1], p=t_array[prev_state]) #taking the probability from the transition matrix
    # result.append(state[curr_state])

    test_result['predicted_condition'] = result

    return test_result

def find_accuracy(predicted_result):
    correct_count = 0.0

    for i in range(len(predicted_result)):
        if predicted_result.loc[i, 'condition'] == predicted_result.loc[i, 'predicted_condition']:
            correct_count += 1

    correct_prop = correct_count / len(predicted_result)

    return correct_prop

def run_predictions_return_avg_accuracy(test_data, trial_count):
    accuracy_sum = 0.0
    for i in range(trial_count):
        predicted_result = predict_weather_simplified(test_data)
        accuracy = find_accuracy(predicted_result)
        accuracy_sum += accuracy
    avg_accuracy = accuracy_sum / trial_count

    return avg_accuracy
```

```python
# Sample prediction (for table graphic)

sample_prediction = predict_weather_simplified(all_seasons_test)
sample_accuracy = find_accuracy(sample_prediction)
print(sample_prediction.head())
print(sample_accuracy)
```

```
   index    datetime condition predicted_condition
0   6575  2018-01-01   no_rain             no_rain
1   6576  2018-01-02   no_rain             no_rain
2   6577  2018-01-03   no_rain             no_rain
3   6578  2018-01-04   no_rain             no_rain
4   6579  2018-01-05   no_rain             no_rain
0.7652292950034223
```

```python
run_predictions_return_avg_accuracy(all_seasons_test, 100)
```

```
0.7703011635865848
```

05

Result / Improvement

Analysis of results and discovery improvements

Result

The accuracy for all 10 of our models:

- All Seasons - Simplified(long time frame): **77%**
- All Seasons - Simplified(short time frame): **79%**
- All Seasons - 6 different weather conditions(long time frame): **37%**
- All Seasons - 6 different weather conditions(short time frame): **39%**
- By Season - 6 different weather conditions(long time frame)
  - Fall: **37% /** Winter: **35% /** Spring: **40% /** Summer: **45%**
- All Seasons - Simplified(long time frame) applied to UCSD/NYU
  - UCSD: **73%**
  - NYU: **43%**

Improvement

- Implementation of time-inhomogeneous Markov chains
    - Our Transition Matrix stays constant
    - Time-inhomogeneous Markov chains Transition Matrix is generated again at each time step
    - Could lead to increase in accuracy

Improvement

- Extending time frame of analysis with second-order Markov chains
  - What is it?(This model uses the past two states to predict the next state, instead of one past state)
  - Attempted to implement
  - Met problem that prevents us from doing it.
    - No observation where the two previous states were ‘clear’ --> ’overcast’
    - Need more data

06
Conclusion

Summarize what we have done and what we have learned

What we have done / learned

Project achievements:

- make a prediction model for weather that is independent of climate measurements

- expanded our model's scope by applying the transition matrices derived

Project takeaways:

- real world application of the Markov chain model

- common thinking process in a more accurate and systematic manner.

References

[1] How has weather forecasting changed over the past two hundred years? *The American Geosciences Institute*, 2022.

[2] Visual Crossing Corporation, 2022.

[3] Randall Swift Douglas D. Mooney. *A Course in Mathematical Modeling*, page 122. The Mathematical Association of America, 1999.

[4] Soumya Karlamangla. California is expected to enter a fourth straight year of drought. *The New York Times*, October.

[5] Henry Maltby, et al. Markov chains. *Brilliant.org*, December 2022.

Questions?