COMS 4721: Machine Learning for Data Science
Lecture 20, 4/11/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute
Columbia University
So far, when thinking probabilistically we have focused on the i.i.d. setting.

- All data are independent given a model parameter.
- This is often a reasonable assumption, but was also done for convenience.

In some applications this assumption is bad:

- Modeling rainfall as a function of hour
- Daily value of currency exchange rate
- Acoustic features of speech audio

The distribution on the next value clearly depends on the previous values.

A basic way to model sequential information is with a discrete, first-order Markov chain.
MARKOV CHAINS
Imagine you see a zombie in an alley. Each time it moves forward it steps (left, straight, right) with probability \((p_l, p_s, p_r)\), unless it’s next to the wall, in which case it steps straight with probability \(p_w^s\) and toward the middle with probability \(p_m^w\).

The distribution on the next location only depends on the current location.

\(^1\)This problem is often introduced with a “drunk,” so our maturity is textbook-level.
We simplify the problem by assuming there are only a finite number of positions the zombie can be in, and we model it as a random walk.

The distribution on the next position only depends on the current position. For example, for a position $i$ away from the wall,

$$s_{t+1} \mid \{s_t = i\} = \begin{cases} i + 1 & \text{w.p. } p_r \\ i & \text{w.p. } p_s \\ i - 1 & \text{w.p. } p_l \end{cases}$$

This is called the first-order Markov property. It’s the simplest type. A second-order model would depend on the previous two positions.
A more compact notation uses a matrix.

For the random walk problem, imagine we have 6 different positions, called *states*. We can write the *transition matrix* as

$$
M = \begin{bmatrix}
p^w_s & p^w_m & 0 & 0 & 0 & 0 \\
p_l & p_s & p_r & 0 & 0 & 0 \\
0 & p_l & p_s & p_r & 0 & 0 \\
0 & 0 & p_l & p_s & p_r & 0 \\
0 & 0 & 0 & p_l & p_s & p_r \\
0 & 0 & 0 & 0 & p^w_m & p^w_s
\end{bmatrix}
$$

$M_{ij}$ is the probability that the next position is $j$ given the current position is $i$.

Of course we can jumble this matrix by moving rows and columns around in a correct way, as long as we can map the rows and columns to a position.
**First-order Markov chain (General)**

Let $s \in \{1, \ldots, S\}$. A sequence $(s_1, \ldots, s_t)$ is a first-order Markov chain if

$$p(s_1, \ldots, s_t) \overset{(a)}{=} p(s_1) \prod_{u=2}^{t} p(s_u|s_1, \ldots, s_{u-1}) \overset{(b)}{=} p(s_1) \prod_{u=2}^{t} p(s_u|s_{u-1})$$

From the two equalities above:

(a) This equality is always true, regardless of the model (chain rule).

(b) This simplification results from the Markov property assumption.

Notice the difference from the i.i.d. assumption

$$p(s_1, \ldots, s_t) = \begin{cases} p(s_1) \prod_{u=2}^{t} p(s_u|s_{u-1}) & \text{Markov assumption} \\ \prod_{u=1}^{t} p(s_u) & \text{i.i.d. assumption} \end{cases}$$

From a modeling standpoint, this is a significant difference.
**First-order Markov chain (general)**

Again, we encode this more general probability distribution in a matrix:

\[ M_{ij} = p(s_t = j | s_{t-1} = i) \]

We will adopt the notation that rows are distributions.

- \( M \) is a *transition matrix*, or *Markov matrix*.
- \( M \) is \( S \times S \) and each row sums to one.
- \( M_{ij} \) is the probability of transitioning to state \( j \) given we are in state \( i \).

Given a starting state, \( s_0 \), we generate a sequence \((s_1, \ldots, s_t)\) by sampling

\[ s_t | s_{t-1} \sim \text{Discrete}(M_{s_{t-1}, \cdot}) \]

We can model the starting state with its own separate distribution.
MAXIMUM LIKELIHOOD

Given a sequence, we can approximate the transition matrix using ML,

\[ M_{\text{ML}} = \arg \max_M p(s_1, \ldots, s_t|M) = \arg \max_M \sum_{u=1}^{t-1} \sum_{i,j} \mathbb{1}(s_u = i, s_{u+1} = j) \ln M_{ij}. \]

Since each row of \( M \) has to be a probability distribution, we can show that

\[ M_{\text{ML}}(i,j) = \frac{\sum_{u=1}^{t-1} \mathbb{1}(s_u = i, s_{u+1} = j)}{\sum_{u=1}^{t-1} \mathbb{1}(s_u = i)}. \]

Empirically count how many times we observe a transition from \( i \rightarrow j \) and divide by the total number of transitions from \( i \).

Example: Model probability it rains (\( r \)) tomorrow given it rained today with observed fraction \( \frac{\#\{r \rightarrow r\}}{\#\{r\}} \). Notice that \( \#\{r\} = \#\{r \rightarrow r\} + \#\{r \rightarrow no-r\} \).
PROPERTY: STATE DISTRIBUTION

Q: Can we say at the beginning what state we’ll be in at step $t + 1$?

A: Imagine at step $t$ that we have a probability distribution on which state we’re in, call it $p(s_t = u)$. Then the distribution on $s_{t+1}$ is

$$p(s_{t+1} = j) = \sum_{u=1}^{S} p(s_{t+1} = j | s_t = u) p(s_t = u).$$

Represent $p(s_t = u)$ with the row vector $w_t$ (the state distribution). Then

$$p(s_{t+1} = j) = \sum_{u=1}^{S} p(s_{t+1} = j | s_t = u) M_{uj} p(s_t = u).$$

We can calculate this for all $j$ with the matrix-vector product $w_{t+1} = w_t M$. Therefore, $w_{t+1} = w_1 M^t$ and $w_1$ can be indicator if starting state is known.
**Property: Stationary Distribution**

Given current state distribution \( w_t \), the distribution on the next state is

\[
  w_{t+1}(j) = \sum_{u=1}^{s} M_{uj} w_t(u) \iff w_{t+1} = w_t M
\]

What happens if we project an infinite number of steps out?

**Definition:** Let \( w_\infty = \lim_{t \to \infty} w_t \). Then \( w_\infty \) is the *stationary distribution*.

- There are many technical results that can be proved about \( w_\infty \).
- Property: If the following are true, then \( w_\infty \) is the same vector for all \( w_0 \):
  1. We can eventually reach any state starting from any other state,
  2. The sequence doesn’t loop between states in a pre-defined pattern.
- Clearly \( w_\infty = w_\infty M \) since \( w_t \) is converging and \( w_{t+1} = w_t M \).

This last property is related to the first eigenvector of \( M^T \):

\[
  M^T q_1 = \lambda_1 q_1 \implies \lambda_1 = 1, \quad w_\infty = \frac{q_1}{\sum_{u=1}^{s} q_1(u)}
\]
A RANKING ALGORITHM
We show an example of using the stationary distribution of a Markov chain to rank objects. The data are pairwise comparisons between objects.

For example, we might want to rank

- Sports teams or athletes competing against each other
- Objects being compared and selected by users

Our goal is to rank objects from “best” to “worst.”

- We will construct a random walk matrix on the objects.
- The stationary distribution will give us the ranking.
- Notice: We don’t consider the sequential information in the data itself. The Markov chain is an artificial modeling construct.
**Problem setup**

We want to construct a Markov chain where each team is a state.

- We encourage transitions from teams that lose to teams that win.

- Predicting the “state” (i.e., team) far in the future, we can interpret a more probable state as a better team.

One specific approach to this specific problem:

- Transitions only occur between teams that play each other.

- If Team A beats Team B, there should be a high probability of transitioning from B→A and small probability from A→B.

- The strength of the transition can be linked to the score of the game.
How about this?

Initialize $\hat{M}$ to a matrix of zeros. For a particular game, let $j_1$ be the index of Team A and $j_2$ the index of Team B. Then update

$$\hat{M}_{j_1,j_1} \leftarrow \hat{M}_{j_1,j_1} + 1\{\text{Team A wins}\} + \frac{\text{points}_j}{\text{points}_{j_1} + \text{points}_{j_2}},$$

$$\hat{M}_{j_2,j_2} \leftarrow \hat{M}_{j_2,j_2} + 1\{\text{Team B wins}\} + \frac{\text{points}_{j_2}}{\text{points}_{j_1} + \text{points}_{j_2}},$$

$$\hat{M}_{j_1,j_2} \leftarrow \hat{M}_{j_1,j_2} + 1\{\text{Team B wins}\} + \frac{\text{points}_{j_2}}{\text{points}_{j_1} + \text{points}_{j_2}},$$

$$\hat{M}_{j_2,j_1} \leftarrow \hat{M}_{j_2,j_1} + 1\{\text{Team A wins}\} + \frac{\text{points}_{j_1}}{\text{points}_{j_1} + \text{points}_{j_2}}.$$

After processing all games, let $M$ be the matrix formed by normalizing the rows of $\hat{M}$ so they sum to 1.
### Example: 2014-2015 College Basketball Season

**MARKOV CHAIN RANKING**

<table>
<thead>
<tr>
<th>RK</th>
<th>Score</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0090112</td>
<td>Villanova</td>
</tr>
<tr>
<td>2</td>
<td>0.0079282</td>
<td>Kansas</td>
</tr>
<tr>
<td>3</td>
<td>0.0074781</td>
<td>North Carolina</td>
</tr>
<tr>
<td>4</td>
<td>0.0067752</td>
<td>Virginia</td>
</tr>
<tr>
<td>5</td>
<td>0.0065791</td>
<td>Oklahoma</td>
</tr>
<tr>
<td>6</td>
<td>0.0063760</td>
<td>Oregon</td>
</tr>
<tr>
<td>7</td>
<td>0.0058095</td>
<td>Michigan St</td>
</tr>
<tr>
<td>8</td>
<td>0.0056623</td>
<td>Xavier OH</td>
</tr>
<tr>
<td>9</td>
<td>0.0055031</td>
<td>Miami FL</td>
</tr>
<tr>
<td>10</td>
<td>0.0049979</td>
<td>West Virginia</td>
</tr>
<tr>
<td>11</td>
<td>0.0047690</td>
<td>Utah</td>
</tr>
<tr>
<td>12</td>
<td>0.0047131</td>
<td>Kentucky</td>
</tr>
<tr>
<td>13</td>
<td>0.0046578</td>
<td>Indiana</td>
</tr>
<tr>
<td>14</td>
<td>0.0046482</td>
<td>Seton Hall</td>
</tr>
<tr>
<td>15</td>
<td>0.0046097</td>
<td>Texas A&amp;M</td>
</tr>
<tr>
<td>16</td>
<td>0.0045635</td>
<td>Duke</td>
</tr>
<tr>
<td>17</td>
<td>0.0042596</td>
<td>Maryland</td>
</tr>
<tr>
<td>18</td>
<td>0.0042441</td>
<td>Purdue</td>
</tr>
<tr>
<td>19</td>
<td>0.0041866</td>
<td>Iowa St</td>
</tr>
<tr>
<td>20</td>
<td>0.0041599</td>
<td>St Joseph's PA</td>
</tr>
<tr>
<td>21</td>
<td>0.0040336</td>
<td>Notre Dame</td>
</tr>
<tr>
<td>22</td>
<td>0.0039794</td>
<td>Arizona</td>
</tr>
<tr>
<td>23</td>
<td>0.0039273</td>
<td>Providence RI</td>
</tr>
</tbody>
</table>

**USA Today Coaches Poll**

<table>
<thead>
<tr>
<th>RK</th>
<th>Team (Points)</th>
<th>Record</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Villanova (30)</td>
<td>35-5</td>
<td>750</td>
</tr>
<tr>
<td>2</td>
<td>North Carolina</td>
<td>33-7</td>
<td>720</td>
</tr>
<tr>
<td>3</td>
<td>Kansas</td>
<td>33-5</td>
<td>657</td>
</tr>
<tr>
<td>4</td>
<td>Oklahoma</td>
<td>29-8</td>
<td>643</td>
</tr>
<tr>
<td>5</td>
<td>Virginia</td>
<td>29-8</td>
<td>631</td>
</tr>
<tr>
<td>6</td>
<td>Oregon</td>
<td>31-7</td>
<td>596</td>
</tr>
<tr>
<td>7</td>
<td>Michigan State</td>
<td>29-6</td>
<td>488</td>
</tr>
<tr>
<td>8</td>
<td>Miami</td>
<td>27-8</td>
<td>480</td>
</tr>
<tr>
<td>9</td>
<td>Indiana</td>
<td>27-8</td>
<td>456</td>
</tr>
<tr>
<td>10</td>
<td>Syracuse</td>
<td>23-14</td>
<td>446</td>
</tr>
<tr>
<td>11</td>
<td>Xavier</td>
<td>28-6</td>
<td>361</td>
</tr>
<tr>
<td>12</td>
<td>Texas A&amp;M</td>
<td>28-9</td>
<td>358</td>
</tr>
<tr>
<td>13</td>
<td>Maryland</td>
<td>27-9</td>
<td>358</td>
</tr>
<tr>
<td>14</td>
<td>West Virginia</td>
<td>26-9</td>
<td>331</td>
</tr>
<tr>
<td>15</td>
<td>Iowa State</td>
<td>23-12</td>
<td>315</td>
</tr>
<tr>
<td>16</td>
<td>Kentucky</td>
<td>23-9</td>
<td>297</td>
</tr>
<tr>
<td>17</td>
<td>Notre Dame</td>
<td>24-12</td>
<td>285</td>
</tr>
<tr>
<td>18</td>
<td>Duke</td>
<td>25-11</td>
<td>263</td>
</tr>
<tr>
<td>19</td>
<td>Purdue</td>
<td>26-9</td>
<td>184</td>
</tr>
<tr>
<td>20</td>
<td>Utah</td>
<td>27-9</td>
<td>170</td>
</tr>
<tr>
<td>21</td>
<td>Gonzaga</td>
<td>28-8</td>
<td>157</td>
</tr>
<tr>
<td>22</td>
<td>Arizona</td>
<td>25-9</td>
<td>154</td>
</tr>
<tr>
<td>23</td>
<td>Wisconsin</td>
<td>22-13</td>
<td>149</td>
</tr>
<tr>
<td>24</td>
<td>Baylor</td>
<td>22-12</td>
<td>105</td>
</tr>
<tr>
<td>25</td>
<td>Iowa</td>
<td>22-11</td>
<td>82</td>
</tr>
</tbody>
</table>

1,549 total teams
22,033 total games
SCORE = stationary distribution
A CLASSIFICATION ALGORITHM
Imagine we have data with very few labels.

We want to use the structure in the dataset to help classify the unlabeled data.

We can do this with a Markov chain.

**Semi-supervised** learning uses partially labeled data to do classification.

- Many or most $y_i$ will be missing in the pair $(x_i, y_i)$.
- Still, there is structure in $x_1, \ldots, x_n$ that we don’t want to throw away.
- In the example above, we might want the inner ring to be one class (blue) and the outer ring another (red).
A random walk classifier

We will define a classifier where, starting from any data point $x_i$,

- A “random walker” moves around from point to point
- A transition between nearby points has higher probability
- A transition to a labeled point terminates the walk
- The label of a point $x_i$ is the label of the terminal point

One possible random walk matrix

1. Let the unnormalized transition matrix be

$$\hat{M}_{ij} = \exp \left\{ -\frac{\|x_i - x_j\|^2}{b} \right\}$$

2. Normalize rows of $\hat{M}$ to get $M$

3. If $x_i$ has label $y_i$, re-define $M_{ii} = 1$
**Property: Absorbing States**

Imagine we have $S$ states. If $p(s_t = i|s_{t-1} = i) = 1$, then the $i$th state is called an **absorbing state** since we can never leave it.

**Q:** Given initial state $s_0 = j$ and set of absorbing states $\{i_1, \ldots, i_k\}$, what is the probability a Markov chain terminates at a particular absorbing state?

- **Aside:** For the semi-supervised classifier, the answer gives the probability on the label of $x_j$.

**A:** Start a random walk at $j$ and keep track of the distribution on states.

- $w_0$ is a vector of 0’s with a 1 in entry $j$ because we know $s_0 = j$
- If $M$ is the transition matrix, we know that $w_{t+1} = w_tM$.
- So we want $w_\infty = w_0M^\infty$. 
PROPERTY: ABSORBING STATE DISTRIBUTION

Group the absorbing states and break up the transition matrix into quadrants:

\[
M = \begin{bmatrix}
A & B \\
0 & I
\end{bmatrix}
\]

The bottom half contains the self-transitions of the absorbing states.

Observation: \( w_{t+1} = w_t M = w_{t-1} M^2 = \cdots = w_0 M^{t+1} \)

So we need to understand what’s going on with \( M^t \). For the first two we have

\[
M^2 = \begin{bmatrix}
A & B \\
0 & I
\end{bmatrix} \begin{bmatrix}
A & B \\
0 & I
\end{bmatrix} = \begin{bmatrix}
A^2 & AB + B \\
0 & I
\end{bmatrix}
\]

\[
M^3 = \begin{bmatrix}
A & B \\
0 & I
\end{bmatrix} \begin{bmatrix}
A^2 & AB + B \\
0 & I
\end{bmatrix} = \begin{bmatrix}
A^3 & A^2 B + AB + B \\
0 & I
\end{bmatrix}
\]
**Geometric Series**

**Detour:** We will use the matrix version of the following scalar equality.

**Definition:** Let $0 < r < 1$. Then $\sum_{u=0}^{t-1} r^u = \frac{1-r^t}{1-r}$ and so $\sum_{u=0}^{\infty} r^u = \frac{1}{1-r}$.

**Proof:** First define the top equality and create the bottom equality

$$C_t = 1 + r + r^2 + \cdots + r^{t-1}$$

$$r C_t = r + r^2 + \cdots + r^{t-1} + r^t$$

and so

$$C_t - r C_t = 1 - r^t.$$

Therefore

$$C_t = \sum_{u=0}^{t-1} r^u = \frac{1-r^t}{1-r} \quad \text{and} \quad C_\infty = \frac{1}{1-r}.$$
A matrix version of the geometric series appears here. We see the pattern

\[ M^t = \begin{bmatrix} A^t & \left( \sum_{u=0}^{t-1} A^u \right) B \\ 0 & I \end{bmatrix}. \]

Two key things that can be shown are:

\[ A^\infty = 0, \quad \sum_{u=0}^{\infty} A^u = (I - A)^{-1} \]

Summary:

- After an infinite \# of steps, \( w_\infty = w_0 M^\infty = w_0 \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} (I - A)^{-1}B \\ I \end{bmatrix}. \)

- The non-zero dimension of \( w_0 \) picks out a row of \( (I - A)^{-1}B \).

- The probability that a random walk started at \( x_j \) terminates at the \( i \)th absorbing state is \( [(I - A)^{-1}B]_{ji} \).
Using a Gaussian kernel normalized on the rows. The color indicates the distribution on the terminal state for each starting point.

Kernel width was tuned to give this result.
Using a Gaussian kernel normalized on the rows. The color indicates the distribution on the terminal state for each starting point.

Kernel width is larger here. Therefore, purple points may leap to the center.