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

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

Another type of unsupervised learning

1. Clustering
2. Matrix factorization

Consider sequential data.

SEQUENTIAL DATA

So far, when thinking probabilistically we have focused on the i.i.d. setting.
(every observation is independent of every other observation and identically distributed.

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

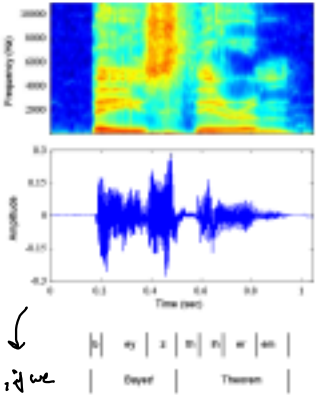
Examples:

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.



Clearly, if we look at the column slice there's a temporal dependence that exists in the data & we would want take that into consideration when modelling speech

MARKOV CHAINS

Discuss the simplest case of a Markov model, which is 1st order Markov model.

EXAMPLE: ZOMBIE WALKER[1]

Imagine you see a zombie in an alley. Each time it moves forward it steps

$$(\text{ left, straight, right }) \quad \text{with probability} \quad (p_l, p_s, p_r),$$

unless it's next to the wall, in which case it steps straight with probability p_s^w and toward the middle with probability p_m^w.

So in this case, if we model the random walk, we're also going to make the simplifying assumption:

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

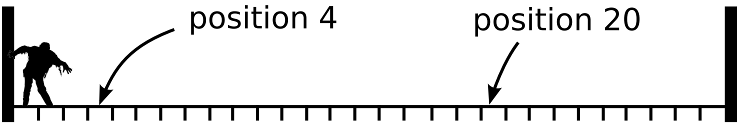
We want to model the location width-wise in relation to the wall. Assuming next location is dependent on current location.

[1] This problem is often introduced with a "drunk," so our maturity is textbook-level.

RANDOM WALK NOTATION

[If the position is dependent on previous 2 times, it would be an example of second order Markov chain. *]

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.
(States)



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

Add a latent variables, which gives the position as a function of time of the random walker

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

This is a simple Markov chain

Position at t+1, given at step t the position is i.

This is called the *first-order Markov property*. It's the simplest type. A second-order model would depend on the previous two positions.

Because the distribution of the state at time t+1 only depends on the position at the previous time.

* Most of the time when people are working with Markov chains,
they're working with first order Markov chains because the added complexity of working with higher orders is very difficult computationally.
So most of the time we'll make a first order Markov chain if we're going to use a Markov model.

MATRIX NOTATION (Transition these probabilities into a Matrix)

A more compact notation uses a matrix.

every position is now going to be called a state.

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

There's no prob. of skipping other steps.

$$M = \begin{bmatrix} p_s^w & p_m^w & 0 & 0 & 0 & 0 \\ p_l & p_s & p_r & 0 & 0 & 0 \\ 0 & p_l & p_s & p_r & 0 & 0 \\ 0 & 0 & p_l & p_s & p_r & 0 \\ 0 & 0 & 0 & p_l & p_s & p_r \\ 0 & 0 & 0 & 0 & p_m^w & p_s^w \end{bmatrix}.$$

Markov transition matrix or random walk matrix

In 3rd position, we have 0 probability of moving to the Ist one or Vth and VIth positions.

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.

We just have to have a coherent, correct way of permuting these probabilities so that they relate to reality.

First-order Markov chain (general)

Latent variable for a markov chain is a sequence of states $S_1 \ldots S_t$.

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

finite no. of states we can be in anytime.

addition assumption by 1st order Markov chain

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

joint distribution of a sequence of length t

*

From the two equalities above:

chain rule of probability

(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

conditionally dependent on the value of the sequence at previous timepoint

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

product of each one separately

From a modeling standpoint, this is a significant difference.

* The probability of state at time u given all of the states up until time u. This is always true. We can always condition on more than necessary, and say that this a way of representing this joint distribution.

** So the distribution of where I'm at a time u simplifies to being only conditioned on where I'm at on the location-- the state that I'm in in the previous timepoint.

First-order Markov chain (general)

· [How conditional probabilities map to the Markov matrix?]

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

j^{th} column in i^{th} row, probability of making a transition to state j given that I'm in state i.

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

prob. of transitioning from state i at t-1 to state j at time t.
↝ same for all time points. t.

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.

Now,

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

$$s_t \,|\, s_{t-1} \sim \text{Discrete}(M_{s_{t-1},:}).$$

← generate the state at time t using a discrete distribution where we pick out the row of transition matrix M indexed by the state the we're in a time t-1.

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

MAXIMUM LIKELIHOOD

we have data and we want to infer the latent variables or model parameters.

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

Find matrix M that maximises the probability of an observed sequence.

log

index of states at 2 different time points

$$M_{\mathrm{ML}} = \arg\max_{M} p(s_1, \ldots, s_t | M) = \arg\max_{M} \sum_{u=1}^{t-1} \sum_{i,j}^{S} \mathbb{1}(s_u = i, s_{u+1} = j) \ln M_{ij}.$$

↳ every single transition

log joint likelihood of an individual sequence given M

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

every single time point.

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

transitioning from state i at time u to state j at time u+1

Empirically, count how many times we observe a transition from $i \to j$ and divide by the total number of transitions from i.

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

* Given the markov assumption, what that amounts to is maximizing over a sum of every single transition's likelihood.

** we have to sum over all of those possible indicators to pick out the correct event. And then that is going to pick out the correct log probability

*** If we just take the derivative and maximize subject to the constraints that m only has non-negative values and that each row has to sum to 1.

- Numerator → total no. of times we make the transition from state i to state j

Denominator → total no. of times we are in state i to begin with. [The total no. of times we transition from state i to another other state.]
[Normalizing]

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

Marginal of a joint probability of being in state u at time t. And being in state j at time t+1.

Represent this marginal probability as a marginalization of the joint probability.

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

marginal conditional prior

* (marginalization of joint prob.)

Sum over values for u to integrate or sum out this state variable at time t.

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

↳ gives a prob. distribution on S different states at time t.

* * * *

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

u^{th} row & j^{th} column

w_t evaluated at dimension 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.

$\underbrace{M \times M \cdots \times M}_{t \text{ times}}$

vector matrix way of writing it

*

So I can write this joint probability is the probability given
that I'm in state u at time t.
Of transitioning from state u to state j at time t plus 1.
Times a prior probability of being in state u at time t to begin with.
So I multiply these two things together, I sum over all of the states for time t.
And I get my marginal distribution of where I'm at at time t plus 1.

** let's now change notation,

w_t = row vector of lengths S, that gives a probability distribution on S different states at time t.
(Marginal probability of which of the S states I'm in at time t.)

***

$w_t \rightarrow$ marginal probability of which of the S states I'm at time t.

o where I am at time $t+1$ is equal to the same S dimensional row vector at time t times the matrix M

oo

We can let this vector w1 be a vector of all zeroes.
Except for a one placed in this dimension corresponding to the state we're in.
Which is simply another way of writing a probability on my state that
I'm in now that is deterministic.
If it's probability one of being where I'm at now then it's a guaranteed thing.

PROPERTY: STATIONARY DISTRIBUTION

Given current state distribution w_t, the distribution on the next state is

Prob. of any given state at time t+1 given my probability of where I'm at time t.

$$w_{t+1}(j) = \sum_{u=1}^{S} M_{uj} w_t(u) \quad \Longleftrightarrow \quad w_{t+1} = w_t M$$

What happens if we project an infinite number of steps out? $(t \to \infty)$

distribution of where I am at a infinite steps from now.

Definition: Let $w_\infty = \lim_{t\to\infty} w_t$. Then w_∞ is the *stationary distribution*. of markov chain M

- There are many technical results that can be proved about w_∞.
- Property: If the following are true, then w_∞ is the same vector for all w_0 *

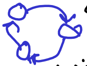
  True only if

  1. We can eventually reach any state starting from any other state, (where we are converging
  2. The sequence doesn't loop between states in a pre-defined pattern. between states in some deterministic way.) not encoded in M

  * * *

- Clearly $w_\infty = w_\infty M$ since w_t is converging and $w_{t+1} = w_t M$.

  w_∞ is a vector that this property



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

Property of eigenvalue and eigenvector:

1st and maximum value for λ_1

$$M^T q_1 = \lambda_1 q_1 \quad \Longrightarrow \quad \lambda_1 = 1, \quad w_\infty = \frac{q_1}{\sum_{u=1}^{S} q_1(u)}$$

for specific matrix M (≥0, all rows to sum)

q is same or different

how it changes with w? ?

We can recover a stationary distribution by taking the 1st eigenvector of M and normalizing it.

* w_∞ will be the same no matter what. So any given state I start out at, w_∞, my distribution on where I'm at a infinite steps from now is same in all cases.

→ So just telling me where I am starting is not going help me say anywhere I'm going in the infinite distance.

* * So that says that if I start in a particular state.
And then I transition according to the Markov transition matrix m.
At some point I can eventually reach any other state no matter where I start.

* * * And in those two cases if those are true.
Then no matter where we start we're gonna converge to the same stationary distribution.
We're gonna have the same uncertainty.
Or the same level of belief of where we're gonna be
an infinite number of steps from now.
No matter where we start.

A RANKING ALGORITHM

EXAMPLE: RANKING OBJECTS

We wanna construct this transition matrix, so
that the stationary distribution of that matrix first exists.
And secondly can be interpreted as telling us who the best teams are and
who the worst teams are and also give us a degree of everywhere in between.

we construct on our data

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
- Web pages based on popularity or relevance

(team/players)

Our goal is to rank objects from "best" to "worst." using markov chain

- 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.

every team is going to be 1 state in a Markov chain.

For 1000 team M: 1000x1000.

[Using markov chain as a artifical modelling construct
Markov chains we're constructing
doesn't actually correspond
to anything meaningful in reality.]

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.

Higher prob for team that win; lower prob. for teams that lose.

- 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:

Construct a transition matrix where:

- Transitions only occur between teams that play each other.

If 2 teams don't play each other then we have no information of how they relate to 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.

(not asymmetric prob.)

- The strength of the transition can be linked to the score of the game.

For ex: Close game: weaker skewing

One beats other by quite a bit: biased transition

EXAMPLE: TEAM RANKINGS

Then we iterate through every game 1 time to construct the transition matrix.

How about this?

Initialize $\widehat{M}$ 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

Self transition

If in j_1, stay in j_1 state.

(Diagonal has probability)

fraction of total pts. it had.

Blowout → ≈1

$$\widehat{M}_{j_1j_1} \leftarrow \widehat{M}_{j_1j_1} + \mathbb{1}\{\texttt{Team A wins}\} + \frac{\texttt{points}_{j_1}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}},$$

$$\widehat{M}_{j_2j_2} \leftarrow \widehat{M}_{j_2j_2} + \mathbb{1}\{\texttt{Team B wins}\} + \frac{\texttt{points}_{j_2}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}},$$

Some of these events is True & other is False

zero if they haven't already played each other

non-zero. Otherwise

j_1 to j_2

$$\widehat{M}_{j_1j_2} \leftarrow \widehat{M}_{j_1j_2} + \mathbb{1}\{\texttt{Team B wins}\} + \frac{\texttt{points}_{j_2}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}},$$

indicator that B wins.

Adding 1 + fraction pt j_2 scored

$$\widehat{M}_{j_2j_1} \leftarrow \widehat{M}_{j_2j_1} + \mathbb{1}\{\texttt{Team A wins}\} + \frac{\texttt{points}_{j_1}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}}.$$

After processing all games, let M be the matrix formed by normalizing the rows of $\widehat{M}$ so they sum to 1.

what's going to happen is the teams that win a lot and win by a lot are going to be transitioned to with higher probability and are going to stay to them.

Team wins a lot of games but against bad opponents, Then we are never going make in our random walk to those bad opponents. In order to get to a good team, we have to win a lot of games against other good times. *

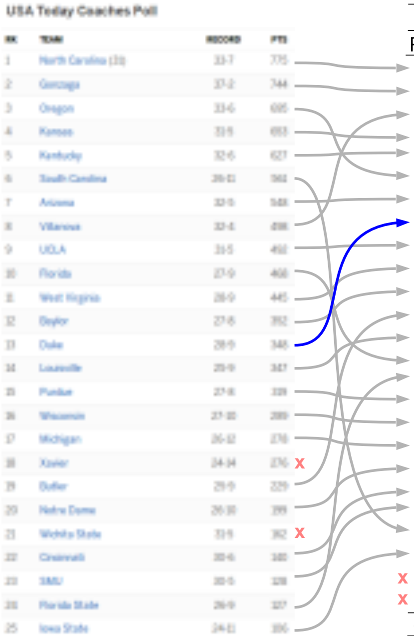
*

In that case, you're going to transition to those other teams more frequently,
which means because you beat those other teams,
they'll transition to you more frequently

*Stationary distribution:

So the stationary distribution can encode the probability
of which team I'm gonna be at an infinite distance from now.
And we can interpret the highest probability
team with the highest probability state is corresponding to the best one.

EXAMPLE: 2016-2017 COLLEGE BASKETBALL SEASON



Markov chain ranking		
RK	SCORE	TEAM
1.	0.00826	North Carolina
2.	0.00825	Gonzaga
3.	0.00776	Villanova
4.	0.00748	Kansas
5.	0.00703	Kentucky
6.	0.00654	Oregon
7.	0.00650	Arizona
8.	0.00640	Duke
9.	0.00594	UCLA
10.	0.00550	West Virginia
11.	0.00546	Baylor
12.	0.00543	Butler
13.	0.00534	Louisville
14.	0.00533	Florida
15.	0.00518	Florida St
16.	0.00516	Purdue
17.	0.00512	Wisconsin
18.	0.00503	Michigan
19.	0.00496	Notre Dame
20.	0.00482	Cincinnati
21.	0.00474	SMU
22.	0.00470	South Carolina
23.	0.00467	Iowa St
24.	0.00455	Creighton
25.	0.00454	Virginia
422.	0.0006	Columbia NY

Finds stationary distribution by finding the 1st eigenvector of the transpose of that chain and normalizing. And then ranking based on stationary vector.

$8 < 13$: Proof of intelligence?

1,570 teams

22,426 games

SCORE = w_∞

A CLASSIFICATION ALGORITHM

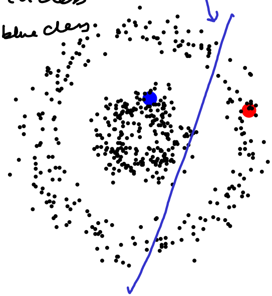
Not fully supervised learning. Something called semi-supervised learning.

SEMI-SUPERVISED LEARNING

↳ Idea: don't want to throw away unlabeled data
somehow gives structure of dataset

only 1 labelled of red class
only 1 " " blue class

linear classifier



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).

regression or classification problem.

Use all of data in X and whatever y is available to us.

start a random walk

We will define a classifier where, starting from any data point x_i, unlabelled want to label

- A "random walker" moves around from point to point (start x_i)
- A transition between nearby points has higher probability (& faraway prob. is low)
- A transition to a labeled point terminates the walk
- The label of a point x_i is the label of the terminal point

Construct a transition matrix b/w this point & all n points, including itself. N-dimensional prob. distr. strictly starting at this point.

One possible random walk matrix

1. Let the *unnormalized* transition matrix be

Gaussian kernel

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

Gaussian kernel takes proximity into consideration

prob. of transitioning from i^{th} to j^{th} point

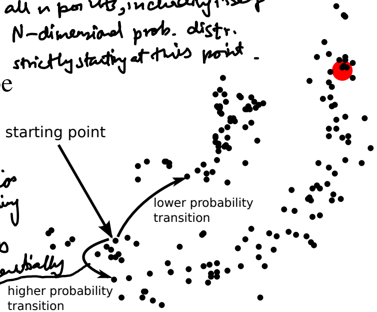
n x n kernel

prob. of transitioning outside range of b is essentially 0 ~

2. Normalize rows of $\widehat{M}$ to get M
3. If x_i has label y_i, re-define $M_{ii} = 1$ (self-transition) prob. = 1

now here markov transition matrix.

How we terminate.



n data points → n x n transition matrix

Property: Absorbing states

Imagine we have S states. If $p(s_t = i | s_{t-1} = i) = 1$, then the ith 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?

We're looking for k-dimensional probability distribution that says the probability of terminating at any one of these different absorbing states given that I start in stage j.

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

  [i.e., starting as point x_j that's unlabelled and terminating at one of k different labelled points.]

A: Start a random walk at j and keep track of the distribution on states. as a function of time.

- w_0 is a vector of 0's with a 1 in entry j because we know $s_0 = j$ (know where we are starting.)
- If M is the transition matrix, we know that $w_{t+1} = w_t M$.
- So we want $w_\infty = w_0 M^\infty$.

Superscript j:

$w^j_\infty = w^j_0 M^\infty$

Final distribution dependent on starting state

* Because this Markov chain does not satisfy the conditions that we had previously

Meaning:

Given that we start at any state we can reach any other state

That's not true because now given we started at a terminal state (an absorbing state), we cannot ever leave that state and reach any other state.

So the stationary distribution in this case is not going to be the same for all points.

PROPERTY: ABSORBING STATE DISTRIBUTION

Calculate: Probability distribution on terminating at any given absorbing state given that I start at a particular state.

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

Group:

Write M in convenient way:

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

Non-absorbing states at top.

Absorbing states at bottom

↳ models the self-transition with prob. 1

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

distribution of where I'm at t+1 broken down into where I'm at t time M^{t+1}

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

vector of all zeros except 1 at starting state

So we need to understand what's going on with M^t. For the first two we have

Inductively figure this out:

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

$$M^3 = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^2 & AB + B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A^3 & A^2 B + AB + B \\ 0 & I \end{bmatrix}$$

Started out uncover a pattern which we're going to return to, in a second.

$$\begin{array}{cc} & s \\ & \begin{array}{cc} s-k & k \end{array} \\ \begin{array}{c} s-k \\ k \end{array} & \begin{bmatrix} A & B \\ O & I \end{bmatrix} \end{array}$$

s different states

k absorbing states

$A \rightarrow$ probability of transitioning from any non-absorbing state to any another non-absorbing state.

$B \rightarrow$ from non-absorb. to absorb.

If we make transition from any of top rows to B, then we make a transition to a absorbing state and our Markov Chain is essentially going to terminate there.

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

$$\begin{array}{rcccccccccccc} C_t & = & 1 & + & r & + & r^2 & + & \cdots & + & r^{t-1} & & \\ r\,C_t & = & & & r & + & r^2 & + & \cdots & + & r^{t-1} & + & r^t \end{array}$$

and so

$$C_t - r\,C_t \;=\; 1 - r^t.$$

Therefore

$$C_t \;=\; \sum_{u=0}^{t-1} r^u \;=\; \frac{1 - r^t}{1 - r} \qquad \text{and} \qquad C_\infty \;=\; \frac{1}{1 - r}.$$

PROPERTY: ABSORBING STATE DISTRIBUTION

A matrix version of the geometric series appears here. We see the pattern

limit case, matrix of all zeros.

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

Two key things that can be shown are:

why? why is $A = r$?

$$A^\infty = 0, \qquad \sum_{u=0}^{\infty} A^u = (I - A)^{-1}$$

Matrix version

So I am picking out the probability of which of observing states that I terminate at.

Summary:

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

vector of 0s except for 1 for state in which I start

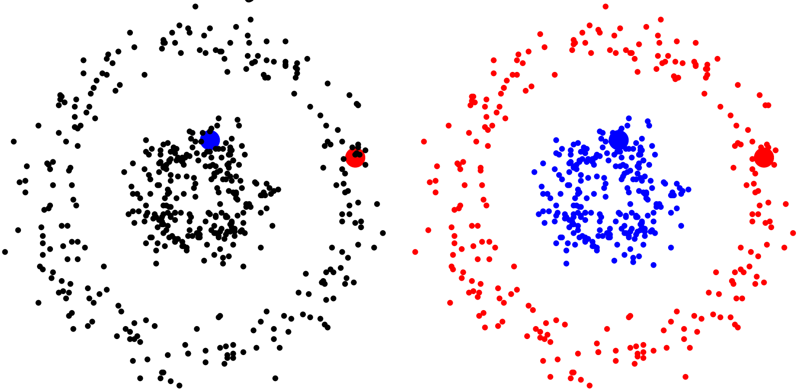
- The non-zero dimension of w_0 picks out a row of $(I - A)^{-1}B$.
- The probability that a random walk started at x_j terminates at the ith absorbing state is $[(I - A)^{-1}B]_{ji}$.

j^{th} element of the matrix

[A and B are sub-matrices of the random walk matrix that I construct using a Kernel.]

CLASSIFICATION EXAMPLE

color code based on the label we end at.

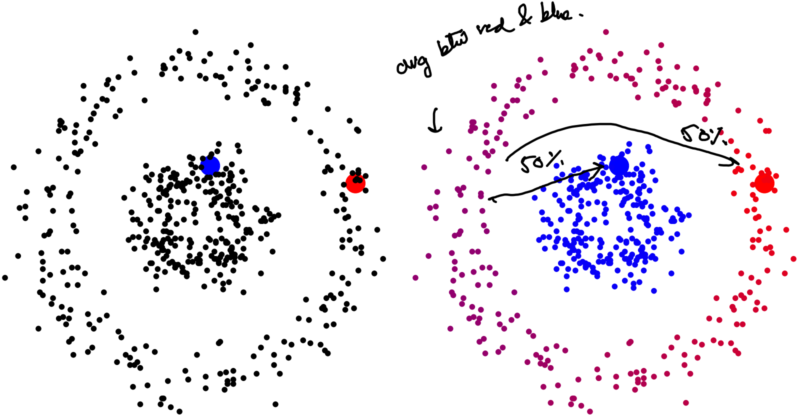


Using a Gaussian kernel normalized on the rows. The color indicates the distribution on the terminal state for each starting point.

Kernel width was tuned to give this result. (such that the probability of jumping over this chasm is 0.)

Classification example

If I ↑ the kernel width



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.

sensitive to parameter setting