CPSC 440: Advanced Machine Learning

Learning Markov Chains

Mark Schmidt

University of British Columbia

Winter 2022

Last Time: Markov Chains

- We discussed the chain rule of probability

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)p(x_4 \mid x_1, x_2, x_3)p(x_5 \mid x_1, x_2, x_3, x_4)$$

- In Markov chains we assume Markov property that $x_j \perp x_1, x_2, \ldots, x_{j-2} \mid x_{j-1}$.

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)p(x_4 \mid x_3)p(x_5 \mid x_4),$$

  which only models dependencies between consecutive features.

- 3 ingredients of Markov chains:
  - State space:
    - Set of possible states (indexed by c) we can be in at time j ("rain" or "not rain").
  - Initial probabilities:
    - $p(x_1 = c)$: probability that we start in state c at time $j = 1$ (p("rain") on day 1).
  - Transition probabilities:
    - $p(x_j = c \mid x_{j-1} = c')$: probability that we move from state c' to state c at time j.
    - Probability that it rains today, given what happened yesterday.

Homogenous Markov Chains

- For rain data it makes sense to use a homogeneous Markov chain:
  - Transition probabilities $p(x_j \mid x_{j-1})$ are the same for all times j.

- An example of parameter tieing:
  1. You have more data available to estimate each parameter.
     - Don't need to independently learn $p(x_j \mid x_{j-1})$ for days 3 and 24.
  2. You can have training examples of different sizes.
     - Same model can be used for any number of days.
     - We could even treat the rain data as one long Markov chain ($n = 1$).

Homogenous Markov Chains

- With discrete states, we could use tabular parameterization for transitions,

$$p(x_j = c \mid x_{j-1} = c') = \theta_{c,c'},$$

where $\theta_{c,c'} \geq 0$ and $\sum_{c=1}^{k} \theta_{c,c'} = 1$ (and we use the same $\theta_{c,c'}$ for all j).

  - So we have a categorical distribution over c values for each c' value.

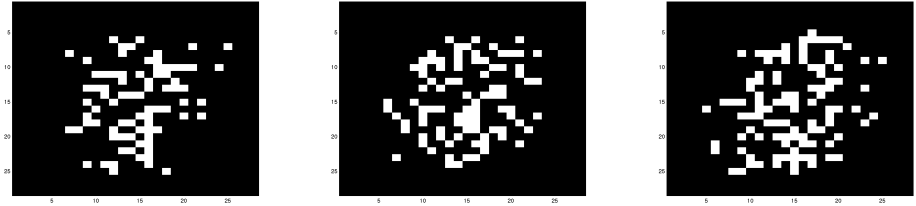
- MLE for homogeneous Markov chain with discrete x_j and tabular parameters:

$$\theta_{c,c'} = \frac{(\text{number of transitions from } c' \text{ to } c)}{(\text{number of times we went from } c' \text{ to anything})},$$

so learning is just counting.

Density Estimation for MNIST Digits

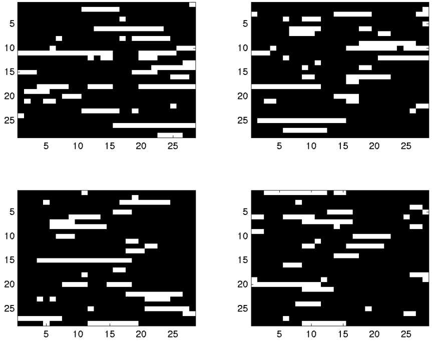
- We've previously considered density estimation for MNIST images of digits.
- We saw that product of Bernoullis does terrible



- This model misses correlation between adjacent pixels.
  - Could we capture this with a Markov chain?

Density Estimation for MNIST Digits

- Samples from a homogeneous Markov chain (putting rows into one long vector):



- Captures correlations between adjacent pixels in the same row.
  - But misses long-range dependencies in row and dependencies between rows.
  - Also, "position independence" of homogeneity means it loses position information.

Inhomogeneous Markov Chains

- Markov chains could allow a different $p(x_j \mid x_{j-1})$ for each j.
  - This makes sense for digits data, but probably not for the rain data.

- For discrete x_j we could use a tabular parameterization,

$$p(x_j = c \mid x_{j=1} = c') = \theta^j_{c,c'}.$$

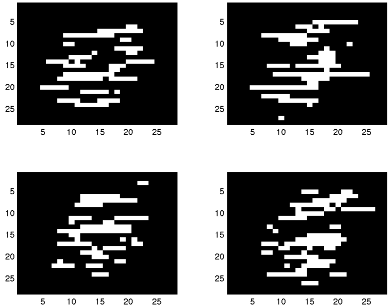
- MLE under this parameterization is given by

$$\theta^j_{c,c'} = \frac{(\text{number of transitions from } c' \text{ to } c \text{ starting at } (j-1))}{(\text{number of times we saw } c' \text{ at position } (j-1))},$$

- Such inhomogeneous Markov chains include independent models as special case:
  - If we set $p(x_j \mid x_{j-1}) = p(x_j)$ for all j we get product of independent model.

Density Estimation for MNIST Digits

- Samples from an inhomogeneous Markov chain fit to digits:



- We have correlations between adjacent pixels in rows and position information.
  - But isn't capturing long-range dependencies or dependency between rows.
  - Later we'll discuss graphical models which address this.

Training Markov Chains

- Some common setups for fitting the parameters Markov chains:
  1. We have one long sequence, and fit parameters of a homogeneous Markov chain.
     - Here, we just focus on the transition probabilities.
  2. We have many sequences of different lengths, and fit a homogeneous chain.
     - And we can use it to model sequences of any length.
  3. We have many sequences of same length, and fit an inhomgeneous Markov chain.
     - This allows "position-specific" effects.
  4. We use domain knowledge to guess the initial and transition probabilities.
     - Here we would be interested in inference in the model.

Fun with Markov Chains

- Markov Chains "Explained Visually":
  http://setosa.io/ev/markov-chains
- Snakes and Ladders:
  http://datagenetics.com/blog/november12011/index.html
- Candyland:
  http://www.datagenetics.com/blog/december12011/index.html
- Yahtzee:
  http://www.datagenetics.com/blog/january42012/
- Chess pieces returning home and K-pop vs. ska:
  https://www.youtube.com/watch?v=63HHmjlh794

Outline

Inference in Markov Chains

- Given a Markov chain model, these are the most common inference tasks:
  1. Sampling: generate sequences that follow the probability.
  2. Marginalization: compute probability of being in state c at time j.
  3. Stationary distribution: probability of being in state c as j goes to ∞.
     - Usually for homogeneous Markov chains.
  4. Decoding: compute assignment to the x_j with highest joint probability.
     - Usually for inhomogeneous Markov chains (important for supervised learning).
  5. Conditioning: do any of the above, assuming $x_j = c$ for some j and c.
     - For example, "filling in" missing parts of the sequence.

Ancestral Sampling

- To sample dependent random variables we can use the chain rule of probability,

$$p(x_1, x_2, x_3, \ldots, x_d) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2, x_1) \cdots p(x_d \mid x_{d-1}, x_{d-2}, \ldots, x_1).$$

- The chain rule suggests the following sampling strategy:
  - Sample x_1 from $p(x_1)$.
  - Given x_1, sample x_2 from $p(x_2 \mid x_1)$.
  - Given x_1 and x_2, sample x_3 from $p(x_3 \mid x_2, x_1)$.
  - ...
  - Given x_1 through x_{d-1}, sample x_d from $p(x_d \mid x_{d-1}, x_{d-2}, \ldots x_1)$.

- This is called ancestral sampling.
  - It's easy if (conditional) probabilities are simple, since sampling in 1D is usually easy.
  - But may not be simple, binary conditional j has 2^j values of $\{x_1, x_2, \ldots, x_j\}$.

Ancestral Sampling Examples

- For Markov chains the chain rule simplifies to

$$p(x_1, x_2, x_3, \ldots, x_d) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_d \mid x_{d-1}),$$

- So ancestral sampling simplifies too:
  1. Sample x_1 from initial probabilities $p(x_1)$.
  2. Given x_1, sample x_2 from transition probabilities $p(x_2 \mid x_1)$.
  3. Given x_2, sample x_3 from transition probabilities $p(x_3 \mid x_2)$.
  4. ...
  5. Given x_{d-1}, sample x_d from transition probabilities $p(x_d \mid x_{d-1})$.

Markov Chain Toy Example: CS Grad Career

- "Computer science grad career" Markov chain:
  - Initial probabilities:

State	Probability	Description
Industry	0.60	They work for a company or own their own company.
Grad School	0.30	They are trying to get a Masters or PhD degree.
Video Games	0.10	They mostly play video games.

  - Transition probabilities (from row to column):

From\to	Video Games	Industry	Grad School	Video Games (with PhD)	Industry (with PhD)	Academia	Deceased
Video Games	0.08	0.90	0.01	0	0	0	0.01
Industry	0.03	0.95	0.01	0	0	0	0.01
Grad School	0.06	0.06	0.75	0.05	0.05	0.02	0.01
Video Games (with PhD)	0	0	0	0.30	0.60	0.09	0.01
Industry (with PhD)	0	0	0	0.02	0.95	0.02	0.01
Academia	0	0	0	0.01	0.01	0.97	0.01
Deceased	0	0	0	0	0	0	1

- So $p(x_t = \text{"Grad School"} \mid x_{t-1} = \text{"Industry"}) = 0.01$.

Example of Sampling x_1

- Initial probabilities are:
  - 0.1 (Video Games)
  - 0.6 (Industry)
  - 0.3 (Grad School)
  - 0 (Video Games with PhD)
  - 0 (Academia)
  - 0 (Deceased)
- So initial CDF is:
  - 0.1 (Video Games)
  - 0.7 (Industry)
  - 1 (Grad School)
  - 1 (Video Games with PhD)
  - 1 (Academia)
  - 1 (Deceased)

- To sample the initial state x_1:
  - First generate a uniform number u, for example $u = 0.724$.
  - Now find the first CDF value bigger than u, which in this case is "Grad School".

Example of Sampling x_2, Given x_1 = "Grad School"

- So we sampled x_1 = "Grad School".
  - To sample x_2, we'll use the "Grad School" row in transition probabilities:

From\to	Video Games	Industry	Grad School	Video Games (with PhD)	Industry (with PhD)	Academia	Deceased
Video Games	0.08	0.90	0.01	0	0	0	0.01
Industry	0.03	0.95	0.01	0	0	0	0.01
Grad School	0.06	0.06	0.75	0.05	0.05	0.02	0.01
Video Games (with PhD)	0	0	0	0.30	0.60	0.09	0.01
Industry (with PhD)	0	0	0	0.02	0.95	0.02	0.01
Academia	0	0	0	0.01	0.01	0.97	0.01
Deceased	0	0	0	0	0	0	1

Example of Sampling x_2, Given $x_1 =$ "Grad School"

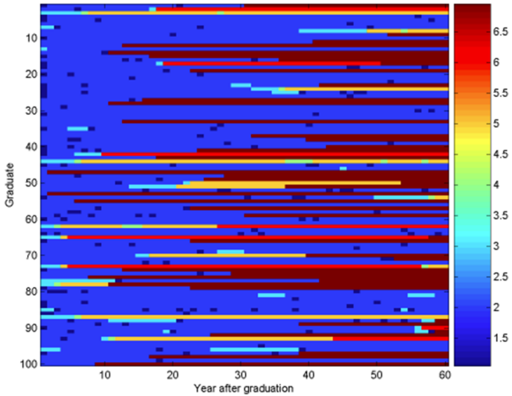
- Transition probabilities:
  - 0.06 (Video Games)
  - 0.06 (Industry)
  - 0.75 (Grad School)
  - 0.05 (Video Games with PhD)
  - 0.02 (Academia)
  - 0.01 (Deceased)

- So transition CDF is:
  - 0.06 (Video Games)
  - 0.12 (Industry)
  - 0.87 (Grad School)
  - 0.97 (Video Games with PhD)
  - 0.99 (Academia)
  - 1 (Deceased)

- To sample the second state x_2:
  - First generate a uniform number u, for example $u = 0.113$.
  - Now find the first CDF value bigger than u, which in this case is "Industry".

Markov Chain Toy Example: CS Grad Career

- Samples from "computer science grad career" Markov chain:



- State 7 ("deceased") is called an absorbing state (no probability of leaving).
- Samples often give you an idea of what model knows (and what should be fixed).

Ancestral Sampling with Blocks of Variables

- We sometimes factorize variables in terms of blocks of variables, as in

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1, x_2)p(x_3, x_4 \mid x_1, x_2)p(x_5, x_6 \mid x_1, x_2, x_3, x_4).$$

- With this factorization ancestral sampling takes the form
  1. Sample x_1 and x_2 from $p(x_1, x_2)$.
  2. Given x_1 and x_2, sample x_3 and x_4 from $p(x_3, x_4 \mid x_2, x_1)$.
  3. Given $x_{1:4}$, sample x_5 and x_6 from $p(x_5, x_6 \mid x_1, x_2, x_3, x_4)$.

- For example, in Gaussian discriminant analysis we write

$$p(x^i, y^i) = p(y^i)p(x^i \mid y^i).$$

- Sampling from Gaussian discriminant analysis:
  1. Sample y^i from the categorical distribution $p(y^i)$.
  2. Sample x^i from the multivariate Gaussian $p(x^i \mid y^i)$.

Marginalization and Conditioning

- Given density estimator, we often want to make probabilistic inferences:
  - Marginals: what is the probability that $x_j = c$?
    - What is the probability we're in industry 10 years after graduation?
  - Conditionals: what is the probability that $x_j = c$ given $x_{j'} = c'$?
    - What is the probability of industry after 10 years, if we immediately go to grad school?

- This is easy for simple independent models:
  - We directly model marginals $p(x_j)$, and conditional are marginals: $p(x_j \mid x_{j'}) = p(x_j)$.

- For Markov chains, it is more complicated.
  - Because $p(x_4)$ depends on the values of x_1, x_2 and x_3.
  - And $p(x_4 \mid x_8)$ additionally depends on the values x_5, x_6, x_7, x_8.

Monte Carlo Methods for Markov Chains

- We could use Monte Carlo approximations for inference in Markov chains:
  - Marginal $p(x_j = c)$ is the number of chains that were in state c at time j.
  - Average value at time j, $E[x_j]$, is approximated by average of x_j in the samples.
  - $p(5 \leq x_j \leq 10)$ is approximate by frequency of x_j being between 5 and 10.
    - This makes more sense for continuous states than evaluating equalities.
  - $p(x_j \leq 10, x_{j+1} \geq 10)$ is approximated by number of chains where both happen.
- Monte Carlo works for continuous states too (for inequalities and expectations).

Exact Marginal Calculation

- In typical settings Monte Carlo has slow convergence like stochastic gradient.
  - $O(1/t)$ convergence rate where constant is variance of samples.
    - If all samples look the same, it converges quickly.
    - If samples look very different, it can be painfully slow.

- For discrete-state Markov chains, we can actually compute marginals directly:
  - We're given initial probabilities $p(x_1 = s)$ for all s as part of the definition.
  - We can use transition probabilities to compute $p(x_2 = s)$ for all s:

$$p(x_2) = \underbrace{\sum_{x_1=1}^{k} p(x_2, x_1)}_{\text{marginalization rule}} = \sum_{x_1=1}^{k} \underbrace{p(x_2 \mid x_1)p(x_1)}_{\text{product rule}}.$$

Exact Marginal Calculation

- We can do a similar calculation to compute $p(x_3)$:

$$p(x_3) = \underbrace{\sum_{x_2=1}^{k} p(x_3, x_2)}_{\text{marginalization rule}} = \sum_{x_2=1}^{k} \underbrace{p(x_3 \mid x_2)p(x_2)}_{\text{product rule}}.$$

- So we define $p(x_3)$ in terms of $p(x_2)$.
  - And we defined $p(x_2)$ in terms of $p(x_1)$,

$$p(x_2) = \sum_{x_1}^{k} p(x_2 \mid x_1)p(x_1),$$

so you could compute all values of $p(x_2)$ and then compute $p(x_3)$.

Exact Marginal Calculation

- Recursive formula for maginals at time j:

$$p(x_j) = \sum_{x_{j-1}=1}^{k} p(x_j \mid x_{j-1}) p(x_{j-1}),$$

called the Chapman-Kolmogorov (CK) equations.

- The CK equations can be implemented as matrix-vector multiplication:
  - Define π^j as a vector containing the marginals at time t:

$$\pi_c^j = p(x_j = c).$$

  - Define T^j as a matrix cotaining the transition probabilities:

$$T_{cc'}^j = p(x_j = c \mid x_{j-1} = c').$$

Exact Marginal Calculation

- Implementing the CK equations as a matrix multiplications:

$$T^j \pi^{j-1} = \begin{bmatrix} p(x_j = 1 | x_{j-1} = 1) & p(x_j = 1 | x_{j-1} = 2) & \dots & p(x_j = 1 | x_{j-1} = k) \\ p(x_j = 2 | x_{j-1} = 1) & p(x_j = 2 | x_{j-1} = 2) & \dots & p(x_j = 2 | x_{j-1} = k) \\ p(x_j = k | x_{j-1} = 1) & p(x_j = k | x_{j-1} = 2) & \dots & p(x_j = k | x_{j-1} = k) \end{bmatrix} \begin{bmatrix} p(x_{j-1} = 1) \\ p(x_{j-1} = 2) \\ \vdots \\ p(x_{j-1} = k) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{c=1}^k p(x_j = 1 \mid x_{j-1} = c) p(x_{j-1} = c) \\ \sum_{c=1}^k p(x_j = 2 \mid x_{j-1} = c) p(x_{j-1} = c) \\ \vdots \\ \sum_{c=1}^k p(x_j = k \mid x_{j-1} = c) p(x_{j-1} = c) \end{bmatrix} = \begin{bmatrix} p(x_j = 1) \\ p(x_j = 2) \\ \vdots \\ p(x_j = k) \end{bmatrix} = \pi^j.$$

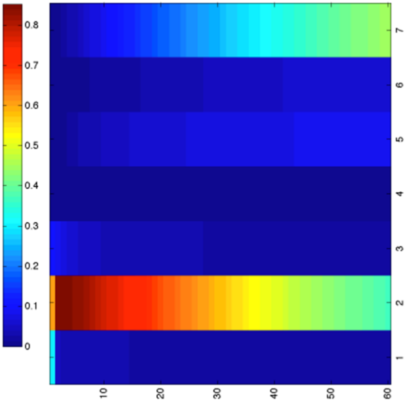
- Cost of multiplying a vector by a $k \times k$ matrix is $O(k^2)$.

- So cost to compute marginals up to time d is $O(dk^2)$.
  - This is fast considering that last step sums over all k^d possible sequences.

$$p(x_d) = \sum_{x_1=1}^k \sum_{x_2=1}^k \cdots \sum_{x_{j-1}=1}^k \sum_{x_{j+1}=1}^k \cdots \sum_{x_{d-1}=1}^k p(x_1, x_2, \dots, x_d).$$

Marginals in CS Grad Career

- CK equations can give all marginals $p(x_j = c)$ from CS grad Markov chain:



- Each row j is a state and each column c is a year.

Continuous-State Markov Chains

- The CK equations also apply if we have continuous states:

$$p(x_j) = \int_{x_{j-1}} p(x_j \mid x_{j-1}) p(x_{j-1}) dx_{j-1},$$

  but this integral may not have a closed-form solution.

- Gaussian probabilities are an important special case:
  - If $p(x_{j-1})$ and $p(x_j \mid x_{j-1})$ are Gaussian, then $p(x_j)$ is Gaussian.
    - Marginal of product of Gaussians.
  - So we can write $p(x_j)$ in closed-form in terms of a mean and variance.
    - Also works states are vectors, with initial/transition following multivariate Gaussian.

- If the probabilities are non-Gaussian, usually can't represent $p(x_j)$ distribution.
  - Gaussian has the special property that it is its own conjugate prior.
  - With other distributions you are stuck using Monte Carlo or other approximations.

Stationary Distribution

- A stationary distribution of a homogeneous Markov chain is a vector π satisfying

$$\pi(c) = \sum_{c'} p(x_j = c \mid x_{j-1} = c')\pi(c').$$

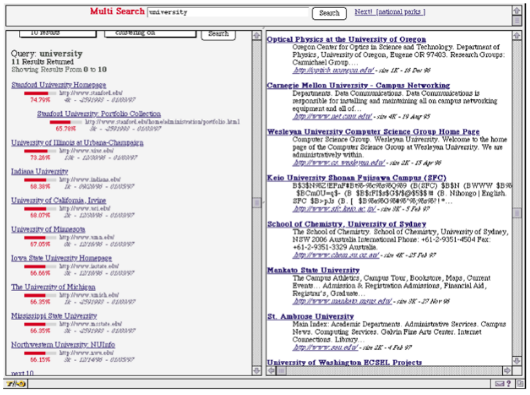
- "Marginal probabilities don't change across time" (forgot about initial state).
  - A stationary distribution is called an "invariant" distribution.
  - Not this does not imply the states converge, just their distribution.

- Under certain conditions, marginals converge to a stationary distribution.
  - $p(x_j = c) \to \pi(c)$ as j goes to ∞.
  - If we fit a Markov chain to the rain example, we have $\pi(\text{“rain”}) = 0.41$.
  - In the CS grad student example, we have $\pi(\text{“dead”}) = 1$.

- Stationary distribution is basis for Google's PageRank algorithm.

Application: PageRank

- Web search before Google:

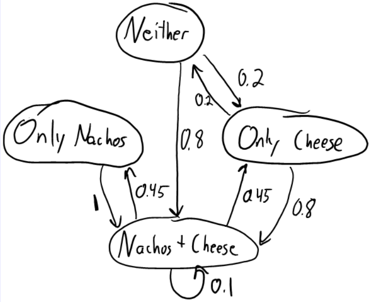


http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

- It was also easy to fool search engines by copying popular websites.

State Transition Diagram

- State transition diagrams are common for visualizing homogenous Markov chains:



$$P = \begin{bmatrix} 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0 & 1 \\ 0.2 & 0 & 0 & 0.8 \\ 0 & 0.45 & 0.45 & 0.1 \end{bmatrix}$$

- Each node is a state, each edge is a non-zero transition probability.
  - For web-search, each node will be a webpage.
- Cost of CK equations is only $O(z)$ instead of $O(k^2)$ if you have only z edges.

Application: PageRank

- Wikipedia's cartoon illustration of Google's PageRank:
  - Large face means higher rank.

https://en.wikipedia.org/wiki/PageRank

- "Important webpages are linked from other important webpages".
- "Link is more meaningful if a webpage has few links".

Application: PageRank

- Google's PageRank algorithm for measuring the importance of a website:
  - Stationary probability in "random surfer" Markov chain:
    - With probability α, surfer clicks on a random link on the current webpage.
    - Otherwise, surfer goes to a completely random webpage.

- To compute the stationary distribution, they use the power method:
  - Repeatedly apply the CK equations.
  - Iterations are faster than $O(k^2)$ due to sparsity of links.
    - Transition matrix is "sparse plus rank-1" which allows fast multiplication.
  - Can be easily parallelized.

Application: Game of Thrones

- PageRank can be used in other applications.
- "Who is the main character in the Game of Thrones books?"

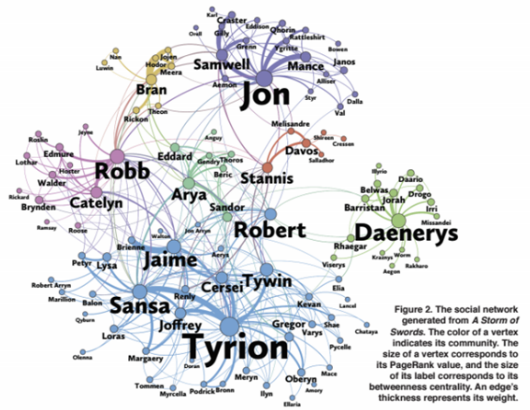


Figure 2. The social network generated from *A Storm of Swords*. The color of a vertex indicates its community. The size of a vertex corresponds to its PageRank value, and the size of its label corresponds to its betweenness centrality. An edge's thickness represents its weight.

http://qz.com/650796/mathematicians-mapped-out-every-game-of-thrones-relationship-to-find-the-main-character

Existence/Uniqueness of Stationary Distribution

- Does a stationary distribution π exist and is it unique?

- A sufficient condition for existence/uniqueness is that all $p(x_j = c \mid x_{j'} = c') > 0$.
  - PageRank satisfies this by adding probability $(1-\alpha)$ of jumping to a random page.

- Weaker sufficient conditions for existence and uniqueness is ergodicity:
  1. "Irreducible" (doesn't get stuck in part of the graph).
  2. "Aperiodic" (probability of returning to state isn't on fixed intervals).

Summary

- Homogeneous Markov chains: same transition probabilities across time.
  - Allows sequences of different lengths.
  - Have more data to estimate transition parameters.
- Inhomogeneous Markov chains: transition probabilities can vary.
  - Allows modeling time-specific probabilities.
- Ancestral sampling generates samples from multivariate distributions.
  - Use chain rule of probability, sequentially sample variables from conditionals.
- Chapman-Kolmogorov equations compute exact univariate marginals.
  - For discrete or Gaussian Markov chains.
- Stationary distribution of homogenous Markov chain.
  - Marginals as time goes to ∞.
  - Basis of Google's PageRank method.

- Next time: voice Photoshop.

Label Propagation as a Markov Chain Problem

- Semi-supervised label propagation method has a Markov chain interpretation.
  - We have $n + t$ states, one for each [un]labeled example.
- Monte Carlo approach to label propagation ("adsorption"):
  - At time $t = 0$, set the state to the node you want to label.
  - At time $t > 0$ and on a labeled node, output the label.
    - Labeled nodes are absorbing states.
  - At time $t > 0$ and on an unlabeled node i:
    - Move to neighbour j with probability proportional w_{ij} (or $\bar{w}_{ij}$).
- Final predictions are probabilities of outputting each label.
  - Nice if you only need to label one example at a time (slow if labels are rare).
  - Common hack is to limit random walk time to bound runtime.