CPSC 540: Machine Learning

Message Passing

Mark Schmidt

University of British Columbia

Winter 2018

Last Time: Monte Carlo Methods

- If we want to approximate expectations of random functions,

$$\underbrace{\mathbb{E}[g(x)] = \sum_{x \in \mathcal{X}} g(x)p(x)}_{\text{discrete } x} \quad \text{or} \quad \underbrace{\mathbb{E}[g(x)] = \int_{x \in \mathcal{X}} g(x)p(x)dx,}_{\text{continuous } x}$$

  the Monte Carlo estimate is

$$\mathbb{E}[g(x)] \approx \frac{1}{n}\sum_{i=1}^{n} g(x^i),$$

  where the x^i are independent samples from $p(x)$.

- We can use this to approximate marginals,

$$p(x_j = c) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}[x_j^i = c].$$

Exact Marginal Calculation

- In typical settings Monte Carlo has sublinear converngece like stochastic gradient.
  - Speed affected is measured by variance of samples.
    - If all samples look the same, it converges quickly.
    - If samples look very different, it can be painfully slow.

- We can sometimes avoid Monte Carlo and compute univariate marginals exactly:
  - Markov chains with discrete or Gaussian probabilities.

- In the discrete case, we're giving $p(x_1)$ and we can compute $p(x_2)$ using:

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

- We can repeat this calculation to obtain $p(x_3)$ and other subsequent marginals.

Exact Marginal Calculation

- Recursive marginal formula is called the Chapman-Kolmogorov (CK) equations:

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

- CK equations for one x_j costs $O(k)$.
- To compute $p(x_j)$ for all k states costs $O(k^2)$.
  - Can be written as matrix-vector product with $k \times k$ transition probabilities matrix.

- With CK equations, computing all marginals costs $O(dk^2)$.

Continuous-State Markov Chains

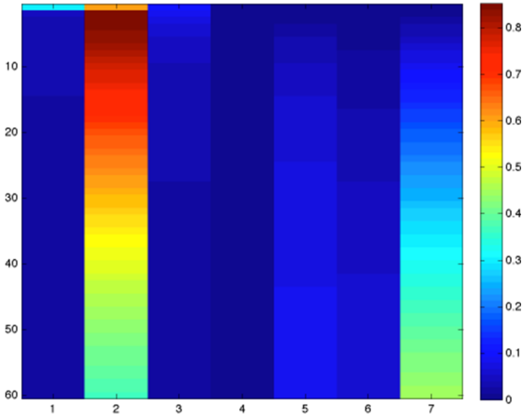
- The CK equations also apply if we have continuous states:

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

- 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.
  - So we can write $p(x_j)$ in closed-form in terms of mean and variance.
- If the probabilities are non-Gaussian, usually can't represent $p(x_j)$ distribution.
  - You are stuck using Monte Carlo or other approximations.

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 year and each column c is a state.

Stationary Distribution

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

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

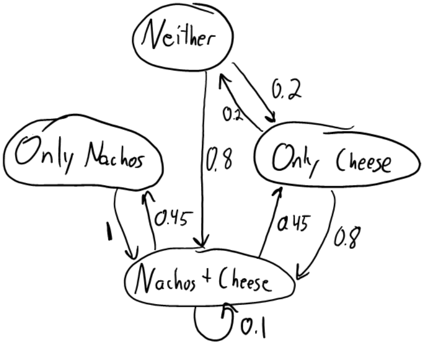
- "The probabilities don't change across time" (also called "invariant" distribution).

- 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.

State Transition Diagram

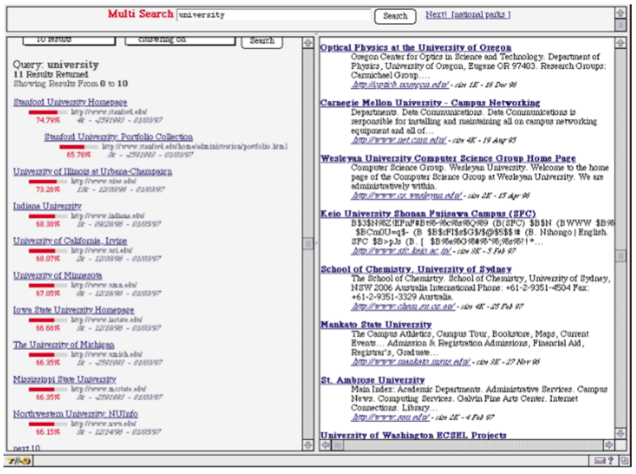
- State transition diagrams are common for visualizing homogenous Markov chains:



- Each node is a state, each edge is a non-zero transition probability.
- Cost of CK equations is only $O(z)$ if you have only z edges.

Application: PageRank

- Web search before Google:



Query: **university**
11 Results Returned
Showing Results From **0** to **10**

Stanford University Homepage
http://www.stanford.edu/
74.79%

Stanford University: Portfolio Collection
http://www.stanford.edu/home/administration/portfolio.html
65.78%

University of Illinois at Urbana-Champaign
http://www.uiuc.edu/
73.26%

Indiana University
http://www.indiana.edu/
68.38%

University of California, Irvine
http://www.uci.edu/
68.07%

University of Minnesota
http://www.umn.edu/
67.05%

Iowa State University Homepage
http://www.iastate.edu/
66.66%

The University of Michigan
http://www.umich.edu/
66.35%

Mississippi State University
http://www.msstate.edu/
66.35%

Northwestern University: NUInfo
http://www.nwu.edu/
66.15%

next 10

Optical Physics at the University of Oregon
Oregon Center for Optics in Science and Technology. Department of Physics, University of Oregon, Eugene OR 97403. Research Groups: Carmichael Group....
- size 1K - 16 Dec 96

Carnegie Mellon University - Campus Networking
Departments. Data Communications. Data Communications is responsible for installing and maintaining all on campus networking equipment and all of...
http://www.net.cmu.edu/ - size 4K - 19 Aug 95

Wesleyan University Computer Science Group Home Page
Computer Science Group. Wesleyan University. Welcome to the home page of the Computer Science Group at Wesleyan University. We are administratively within.
- size 2K - 15 Apr 96

Keio University Shonan Fujisawa Campus (SFC)
B3N%Z!EFnF#Bt%-%c%s%Q%9 (B(SFC) BN (BWWW $B%
$BCm0U=q$- (B BrFIsG$/$@$5$$!# (B. Nihongo | English.
SFC $B>pJs (B. [$B%a%G%#%"%;%s%?!*...
http://www.sfc.keio.ac.jp/ - size 3K - 5 Feb 97

School of Chemistry, University of Sydney
The School of Chemistry. School of Chemistry, University of Sydney, NSW 2006 Australia International Phone: +61-2-9351-4504 Fax: +61-2-9351-3329 Australia.
- size 4K - 25 Feb 97

Mankato State University
The Campus Athletics, Campus Tour, Bookstore, Maps, Current Events... Admission & Registration Admissions, Financial Aid, Registrar's, Graduate...
- size 3K - 27 Nov 96

St. Ambrose University
Main Index: Academic Departments. Administrative Services. Campus News. Computing Services. Galvin Fine Arts Center. Internet Connections. Library...
- size 2K - 4 Feb 97

University of Washington ECSEL Projects

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

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.
    - Can be easily parallelized.
    - Achieves a linear convergence rate.

- More recent works have shown coordinate optimization can be faster.

Application: Game of Thrones

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

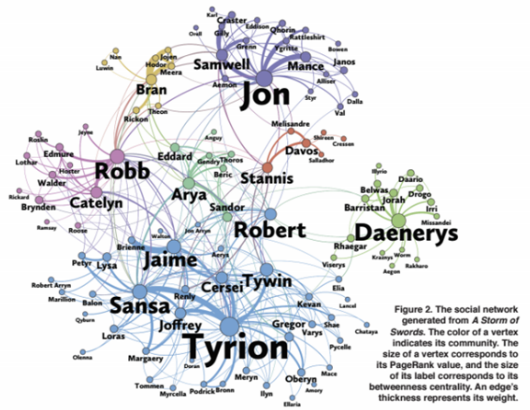


Figure 2. The social network generated from *A Storm of Swords*. The color of a vertex indicates its community. The size of a vertex corresponds to its PageRank value, and the size of its label corresponds to its betweenness centrality. An edge's thickness represents its weight.

`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 adds probability α of jumping to a random page.
- Weaker sufficient conditions for existene and uniqueness ("ergodic"):
    1. "Irreducible" (doesn't get stuck in part of the graph).
    2. "Aperiodic" (probability of returning to state isn't on fixed intervals).

Outline

Decoding: Maximizing Joint Probability

- Decoding in density models: finding x with highest joint probability:

$$\operatorname*{argmax}_{x_1,x_2,\dots,x_d} p(x_1,x_2,\dots,x_d).$$

- For CS grad student $(d = 60)$ the decoding is “industry” for all years.
  - The decoding often doesn’t look like a typical sample.
  - The decoding can change if you increase d.

- Decoding is easy for independent models:
  - We can just optimize each x_j independently.
  - For example, with four variables we have

$$\max_{x_1,x_2,x_3,x_4}\{p(x_1)p(x_2)p(x_3)p(x_4)\} = \left(\max_{x_1} p(x_1)\right)\left(\max_{x_2} p(x_2)\right)\left(\max_{x_3} p(x_3)\right)\left(\max_{x_4} p(x_4)\right).$$

- Can we also maximize the marginals to decode a Markov chain?

Example of Decoding vs. Maximizing Marginals

- Consider the "plane of doom" 2-variable Markov chain:

$$X = \begin{bmatrix} \text{“land”} & \text{“alive”} \\ \text{“land”} & \text{“alive”} \\ \text{“crash”} & \text{“dead”} \\ \text{“explode”} & \text{“dead”} \\ \text{“crash”} & \text{“dead”} \\ \text{“land”} & \text{“alive”} \\ \vdots & \vdots \end{bmatrix}.$$

- Initial probabilities are given by

$$p(x_1 = \text{“land”}) = 0.4, \quad p(x_1 = \text{“crash”}) = 0.3, \quad p(x_1 = \text{“explode”}) = 0.3,$$

and x_2 is "alive" iff x_1 is "land".

Example of Decoding vs. Maximizing Marginals

- Initial probabilities are given by

$$p(x_1 = \text{“land”}) = 0.4, \quad p(x_1 = \text{“crash”}) = 0.3, \quad p(x_1 = \text{“explode”}) = 0.3,$$

  and x_2 is “alive” iff x_1 is “land”.

- If we apply the CK equations we get

$$p(x_2 = \text{‘’alive”}) = 0.4, \quad p(x_2 = \text{“dead”}) = 0.6,$$

  so maximizing the marginals $p(x_j)$ independently gives (“land”, “dead”).
    - This actually has probability 0.

- Decoding considers the joint assignment to x_1 and x_2 maximizing probaiblity.
    - In this case it's (“land”, “alive”), which has probability 0.4.

Distributing Max across Product

- Note that decoding can't be done forward in time as in CK equations.
  - We need to optimize over all k^d assignments to all variables.
  - Even if $p(x_1 = 1) = 0.99$, the most likely sequence could have $x_1 = 2$.

- Fortunately, the Markov property makes the max simplify:

$$\begin{aligned}\max_{x_1,x_2,x_3,x_4} p(x_1,x_2,x_3,x_4) &= \max_{x_1,x_2,x_3,x_4} p(x_4 \mid x_3)p(x_3 \mid x_2)p(x_2 \mid x_1)p(x_1)\\ &= \max_{x_4}\max_{x_3}\max_{x_2}\max_{x_1} p(x_4 \mid x_3)p(x_3 \mid x_2)p(x_2 \mid x_1)p(x_1)\\ &= \max_{x_4}\max_{x_3}\max_{x_2} p(x_4 \mid x_3)p(x_3 \mid x_2)\max_{x_1} p(x_2 \mid x_1)p(x_1)\\ &= \max_{x_4}\max_{x_3} p(x_4 \mid x_3)\max_{x_2} p(x_3 \mid x_2)\max_{x_1} p(x_2 \mid x_1)p(x_1),\end{aligned}$$

where we're using that $\max_i \alpha a_i = \alpha \max_i a_i$ for non-negative α.

Decoding with Memoization

- The Markov property writes decoding as a sequence of max problems:

$$\max_{x_1,x_2,x_3,x_4} p(x_1,x_2,x_3,x_4) = \max_{x_4}\max_{x_3} p(x_4 \mid x_3)\max_{x_2} p(x_3 \mid x_2)\max_{x_1} p(x_2 \mid x_1)p(x_1),$$

but note that we can't just "solve" $\max_{x_1}$ once because it's a function of x_2.

  - Instead, we'll memoize solution $M_2(x_2) = \max_{x_1} p(x_2 \mid x_1)p(x_1)$ for all x_2,

$$\max_{x_1,x_2,x_3,x_4} p(x_1,x_2,x_3,x_4) = \max_{x_4}\max_{x_3} p(x_4 \mid x_3)\max_{x_2} p(x_3 \mid x_2)M_2(x_2).$$

- Now we memoize solution $M_3(x_3) = \max_{x_2} p(x_3 \mid x_2)M_2(x_2)$ for all x_3,

$$\max_{x_1,x_2,x_3,x_4} p(x_1,x_2,x_3,x_4) = \max_{x_4}\max_{x_3} p(x_4 \mid x_3)M_3(x_3).$$

- And defining $M_4(x_4) = \max_{x_3} p(x_4 \mid x_3)M_2(x_3)$ the maximum value is given by

$$\max_{x_1,x_2,x_3,x_4} p(x_1,x_2,x_3,x_4) = \max_{x_4} M_4(x_4).$$

Example: Decoding the Plane of Doom

- We have $M_1(x_1) = p(x_1)$ so in "plane of doom" we have

$$M_1(\text{"land"}) = 0.4, \quad M_1(\text{"crash"}) = 0.3, \quad M_1(\text{"explode"}) = 0.3.$$

- We have $M_2(x_2) = \max_{x_1} p(x_2 \mid x_1) M_1(x_1)$ so we get

$$M_2(\text{"alive"}) = 0.4, \quad M_2(\text{"dead"}) = 0.3.$$

- $M_2(2) \neq p(x_2 = 2)$ because we needed to choose either "crash" or "explode".
- We maximize $M_2(x_2)$ to find that optimal decoding ednds with "alive".
  - We now need to backtrack to find the state that lead to "alive", giving "land".

Viterbi Decoding

- What is $M_j(x_j)$ in words?
    - "Probability of most likely length-j sequence ending in x_j (ignoring future)".

- The Viterbi decoding algorithm (special case of dynamic programming):
    1. Set $M_1(x_1) = p(x_1)$ for all x_1.
    2. Compute $M_2(x_2)$ for all x_2, store value of x_1 leading to the best value of each x_2.
    3. Compute $M_3(x_3)$ for all x_3, store value of x_2 leading to the best value of each x_3.
    4. ...
    5. Maximize $M_d(x_d)$ to find value of x_d in a decoding.
    6. Bactrack to find the value of x_{d-1} that lead to this x_d.
    7. Backtrack to find the value of x_{d-2} that lead to this x_{d-1}.
    8. ...

- Computing all $M_j(x_j)$ given all $M_{j-1}(x_{j-1})$ costs $O(k^2)$.
    - Total cost is only $O(dk^2)$ to search over all k^d paths.

Application: Voice Photoshop

- Application: Adobe VoCo uses Viterbi as part of synthesizing voices:

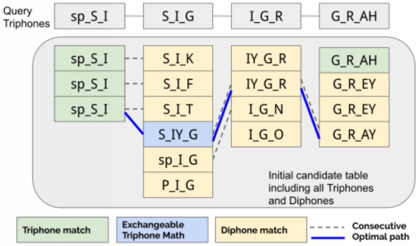


Fig. 7. Dynamic triphone preselection. For each query triphone (top) we find a candidate set of good potential matches (columns below). Good paths through this set minimize differences from the query, number and severity of breaks, and contextual mismatches between neighboring triphones.

`http://gfx.cs.princeton.edu/pubs/Jin_2017_VTI/Jin2017-VoCo-paper.pdf`

- `https://www.youtube.com/watch?v=I3l4XLZ59iw`

Summary

- 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.
- Decoding is task of finding most probable x.
- Viterbi decoding allow efficient decoding with Markov chains.
- Next time: weakening the Markov assumption.