Markov Chain Monte Carlo Methods

Christoph Teichmann Antoine Venant

Goals

Teaching Goals

- Problem of Bayesian Inference
- Markov Chain Monte Carlo
- Metropolis-Hasting Technique
- Gibbs Technique

Motivation

Our Model From Last Session

We were discussing how to learn a language model:

- Bigram model for text
- Probabilities are hidden variables
- Dirichlet Prior for probabilities

Can we improve this model?

Hidden States

Assume the words are generated from hidden states

- $h \in H$ hidden tags
- $w \in L$ words
- P_w distribution over words give hidden tags (one per tag)
- P_h distribution over hidden tags given previous tag (one per tag) and initial tag
- P_w and P_h have Dirichlet Prior

$$P(w_1, w_2, \dots) = P(P_h)P(P_w)P_h(h_0) \prod_{i \in 1,\dots} P(w_i|h_i)P_h(h_i|h_{i-1})$$

Make This Concrete

- $L = \{\text{Mary, sees, something, ., John}\}$
- $H = \{1, 2\}$
- how many probability distributions/densities are we thinking about?

Reminder: Dirichlet $= P(P_x) = \frac{\prod_{o \in O} P_x(o)_o^{\alpha}}{B(\boldsymbol{\alpha})}$

Make This Concrete

- $L = \{\text{Mary}, \text{sees}, \text{something}, ., \text{John}\}$
- $H = \{1, 2\}$
- how many probability distributions/densities are we thinking about?
- we need 2 word given state, 2 state given state, 1 initial state + priors for each $\rightarrow$ 10
- α for all parameters is 0.5 except:
- $\alpha^1_{Mary} = 1$, $\alpha^2_{sees} = 1$ (symmetry breaking)

$$\text{Reminder: Dirichlet} = P(P_x) = \frac{\prod_{o \in O} P_x(o)^{\alpha}_{o}}{B(\boldsymbol{\alpha})}$$

Reasoning Based on the Model

Assume corpus: $C =$ "Mary sees something"

- What is posterior probability $P(P_h^{exam}|C)$, with:
  - $P_h^{exam}(1|1) = 0.1$
  - $P_h^{exam}(2|1) = 0.9$
  - $P_h^{exam}(1|2) = 0.6$
  - $P_h^{exam}(2|1) = 0.4$
  - $P_h^{exam}(1|s) = 0.8$
  - $P_h^{exam}(2|s) = 0.2$
- What is $P(h_2 = 2)$?
- What is . . .

We Can Use Knowledge of Dirichlet

Easy if we know the tags, e.g.:

"Mary:1 sees:2 something:1"

$$P(P_h^{exam}|T,C) = \frac{\prod_{i\in\{1,2\}} P_h^{exam}(i|1)^{\alpha_i^1}}{B(\boldsymbol{\alpha}^1)} \times \frac{\prod_{i\in\{1,2\}} P_h^{exam}(i|2)^{\alpha_i^2}}{B(\boldsymbol{\alpha}^2)} \times \frac{\prod_{i\in\{1,2\}} P_h^{exam}(i|s)^{\alpha_i^s}}{B(\boldsymbol{\alpha}^s)}$$

Where $\alpha_1^s = 1.5$, $\alpha_2^1 = 1.5$, $\alpha_1^2 = 1.5$ and other αs for hidden tags are still 0.5

But We do not Know the Hidden Tags

$$
\begin{aligned}
P(P_h^{exam}|C) &= \sum_T P(P_h^{exam}|T,C)P(T|C) \\
&= \sum_T P(P_h^{exam}|T,C)\frac{P(C|T)P(T)}{P(C)}
\end{aligned}
$$

Typical Problems in Bayesian Learning / Inference

$$P(P_h^{exam}|C) = \overbrace{\sum_T}^{\substack{\text{annoying}\\\text{sum}}} P(P_h^{exam}|T, C)\frac{P(C|T)P(T)}{\underbrace{\substack{\text{normalization factor}\\\text{(secretly annoying sum)}}}_{P(C)}}$$

We need a generic fix for this problem!

Use Representatives

When you want to know what people think, you do not ask everyone, you ask a few representatives.

When you have an annoying some over T you do not consider every possible asignment of tags, you only consider a few representative ones.

What Does it Mean to be Representative

Let us make our problem more general:

$$P(P_h^{exam}|C) = \underbrace{\sum_T}_{\text{sum over variable}} \underbrace{P(P_h^{exam}|T,C)}_{\text{function of variable}} \underbrace{\frac{P(C|T)P(T)}{P(C)}}_{\text{probability of variable}}$$

$$\overbrace{\sum_V}^{\text{sum over variable}} \overbrace{f(V)}^{\text{function of variable}} \overbrace{P(V)}^{\text{probability of variable}}$$

Expected value problem – Many problems in Bayesian Learning/Inference can be formulated as expected value problems

The Law of Large Numbers

The law of large numbers can be formulated as follows:

- Produce sequence $V_1, V_2, \ldots$
- $P(V_i = v)$ given by $P(V)$ from our expected value
- Then $\lim_{n\to\infty} \sum_n \frac{1}{n} f(v_i) = \sum_V f(V)P(V)$

What Does it Mean to be Representative?

We will try to generate a sequence as if each v_i was drawn from $P(V)$

But how do we do this?

For Simple Cases

For simple distributions (categorial) solutions exist based on pseudo-random number generators.

But we have a hard case here!

Markov Chains

Markov Chain Trick

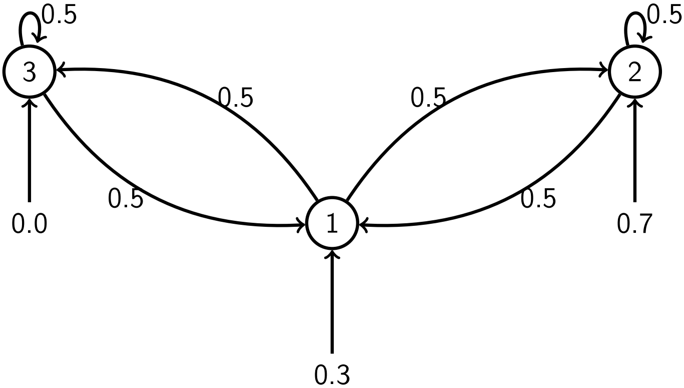
- Produce a Markov Chain (later)
- Produce sequence $V_1, V_2, \dots$ from Markov Chain
- Ensure certain conditions
- Then $\lim_{n \to \infty} \sum_n \frac{1}{n} f(v_i) = \sum_V f(V)P(V)$

What is a Markov Chain

- Set of states V – will be variables of interest, can be infinite (but we assume discrete)
- Initial Probability $S(V_0)$
- Transition Probability $T(V_i|v_{i-1})$

Example

How to create $v_1, v_2, \ldots$ from this:



What are the Magic Requirements?

What do we need so: $\lim_{n\to\infty} \sum_n \frac{1}{n} f(v_i) = \sum_V f(V)P(V)$?

Magic Requirement 1: Invariant Distribution

Invariant distribution $I(V)$ of chain $= P(V)$

What is invariant distribution?

Magic Requirement 1: Invariant Distribution

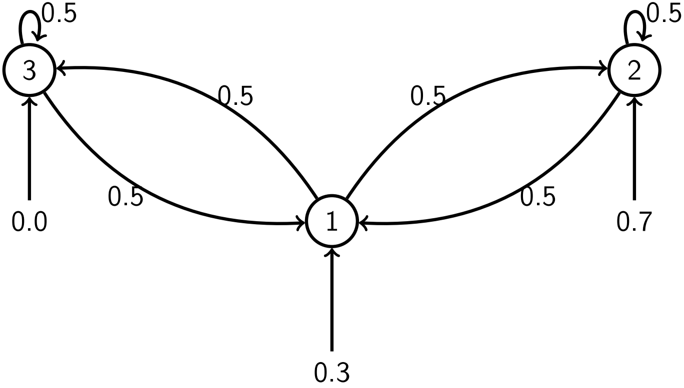
Invariant distribution $I(V)$ of chain $= P(V)$

$I(V)$ s.t. for all v: $I(v) = \sum_{v' \in V} T(v|v')I(v')$

Initial probability does not matter

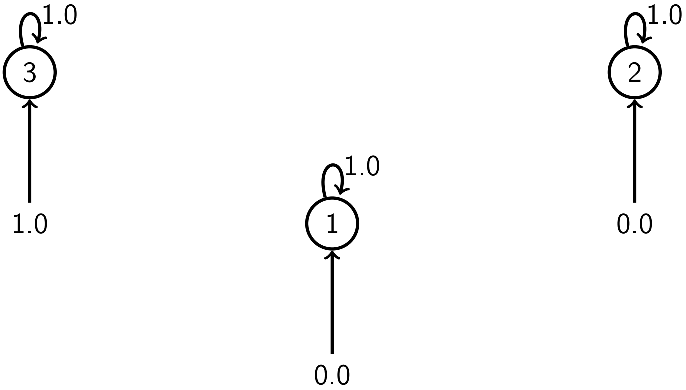
Example

$$I(1) = I(2) = I(3) = \frac{1}{3} \text{ – Show!}$$



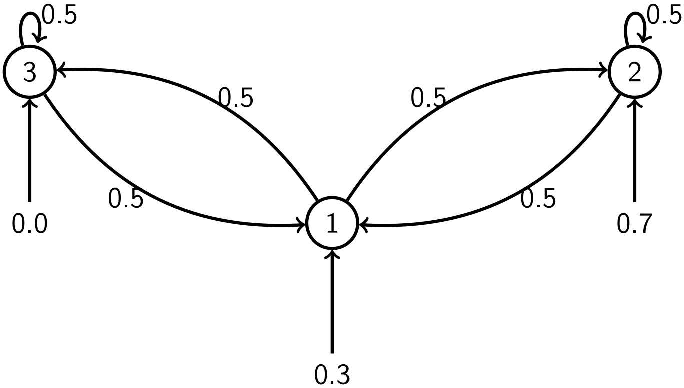
Magic Requirement 2: Irreducibility

Irreducible: Same $I(v)$, no magic!



Magic Requirement 3: Recurrence

Probability of returning is 1 – always true for finite.



Markov Chain Trick

- Produce a Markov Chain
- Produce sequence $V_1, V_2, \ldots$ from Markov Chain
- If Markove Chain Recurrent, Irreducible, and has Invariant Distribution $I(V) = P(V)$
- Then $\lim_{n \to \infty} \sum_n \frac{1}{n} f(v_i) = \sum_V f(V)P(V)$

  See "Markov Chains" by James Norris

  relevant chapters available online `http://www.statslab.cam.ac.uk/~james/Markov/`)

Invariant Tricks

How the Magic Happens

- Irreducibility and Recurrence – no general trick – often easy
- Correct Invariant Distribution – super hard?

Invariant Distribution Trick 1: Metropolis-Hastings

- Proposal Distribution – $p(V|v_i)$
- Must be easy to draw from!
- e.g. for example: flip one tag at random (table)
- Then accept with probability:

$$T(v_i|v_{i-1}) = \min\left(1, \frac{P(v_i)p(v_{i-1}|v_i)}{P(v_{i-1})p(v_i|v_{i-1})}\right)$$

- Otherwise $v_i = v_{i-1}$

Invariant Distribution Trick 1: Metropolis-Hastings

- Proposal Distribution – $p(V|v_i)$
- Must be easy to draw from!
- e.g. for example: flip one tag at random (table)
- Then accept with probability:

$$T(v_i|v_{i-1}) = \min\left(1, \frac{P(v_i)p(v_{i-1}|v_i)}{P(v_{i-1})p(v_i|v_{i-1})}\right)$$

- Otherwise $v_i = v_{i-1}$

Show that indeed $I(V) = P(V)$!

Invariant Distribution Trick 1: Metropolis-Hastings

- Proposal Distribution – $p(V|v_i)$
- Must be easy to draw from!
- e.g. for example: flip one tag at random (table)
- Then accept with probability:

$$T(v_i|v_{i-1}) = \min\left(1, \frac{P(v_i)p(v_{i-1}|v_i)}{P(v_{i-1})p(v_i|v_{i-1})}\right)$$

- Otherwise $v_i = v_{i-1}$

No problem with

$$P(v) = \frac{f(v)}{\text{super complicated normalizer}}$$

Main Problem: Where to get $p(v_i|v_{i-1})$

- Bad $p \rightarrow$ lot of proposals that will be rejected
- Lot of rejection $\rightarrow$ takes forever to converge
- Often need just the right p for a given problem

$$T(v_i|v_{i-1}) = \min\left(1, \frac{P(v_i)p(v_{i-1}|v_i)}{P(v_{i-1})p(v_i|v_{i-1})}\right)$$

Invariant Distribution Trick 2: Gibbs

- Assume that each $V = \langle t_1, \ldots, t_n \rangle$
- E.g., hidden tags in our language model
- Pick a position i at random (or systematically $\rightarrow$ harder to prove)
- We want to change only t_i
- Pick it according to $P(t_i | t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n)$

Show that this is case of Metropolis-Hastings

Invariant Distribution Trick 2: Gibbs

- Assume that each $V = \langle t_1, \ldots, t_n \rangle$
- E.g., hidden tags in our language model
- Pick a position i at random (or systematically $\rightarrow$ harder to prove)
- We want to change only t_i
- Pick it according to $P(t_i' | t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n)$

$$P(t_i' | t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) = \frac{P(t_1, \ldots, t_i', \ldots, t_n)}{\sum_{\bar{t}_i} P(t_1, \ldots, \bar{t}_i, \ldots, t_n)}$$

Invariant Distribution Trick 2: Gibbs

- Assume that each $V = \langle t_1, \ldots, t_n \rangle$
- E.g., hidden tags in our language model
- Pick a position i at random (or systematically $\rightarrow$ harder to prove)
- We want to change only t_i
- Pick it according to $P(t_i' | t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n)$

$$P(t_i' | \text{rest}) = \frac{f(t_1, \ldots, t_i', \ldots, t_n)}{\substack{\text{super} \\ \text{complicated} \\ \text{normalizer}} \sum_{\bar{t}_i} P(t_1, \ldots, \bar{t}_i, \ldots, t_n)}$$

Invariant Distribution Trick 2: Gibbs

- Assume that each $V = \langle t_1, \ldots, t_n \rangle$
- E.g., hidden tags in our language model
- Pick a position i at random (or systematically $\rightarrow$ harder to prove)
- We want to change only t_i
- Pick it according to $P(t_i' | t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n)$

$$P(t_i' | \text{rest}) = \frac{f(t_1, \ldots, t_i', \ldots, t_n)}{\text{super}^2 \text{ complicated normalizer } \sum_{\bar{t}_i} f(t_1, \ldots, \bar{t}_i, \ldots, t_n)}$$

Main Problem: Stuck Locally

- For many problems values of t_i very dependent
- Need to really make sure that Irreducible
- Likely to get stuck in certain locations for long time
- Two box example (table)

Finish

Let us work through Gibbs sampling for our example.