СЭМПЛИРОВАНИЕ

Сергей Николенко

НИУ ВШЭ — Санкт-Петербург
29 сентября 2017 г.

Random facts:

- 29 сентября 61 г. до н.э. Гней Помпей отметил свой третий триумф за победу над пиратами, конец Митридатовых войн, а также свой 45-й день рождения
- 29 сентября 1922 г. из Петрограда отплыл пароход «Обербургомистр Хакен» с Бердяевым, Франком и Ильиным на борту -- знаменитый «философский пароход»
- 29 сентября 1897 г. в Буэнос-Айресе на премьере спектакля «Креольский суд» было впервые исполнено танго

Сэмплирование в графических моделях

- Вариант выборки с отклонением можно применить к направленным графическим моделям.
- Сэмплировать без evidence – тривиально.
- Сэмплировать с evidence можно так: сделаем сэмпл, если наблюдаемые переменные не сошлись, выкинем.
- Для ненаправленных не так просто, да и для направленных не сработает, если наблюдаемых много.

ПРОБЛЕМЫ

- Как и у предыдущего алгоритма, у выборки с отклонением начинаются проблемы в больших размерностях.
- Суть проблемы та же, что в предыдущем случае, а выражается она в том, что c будет очень большим (экспоненциальным от n), и почти все сэмплы будут отвергаться.

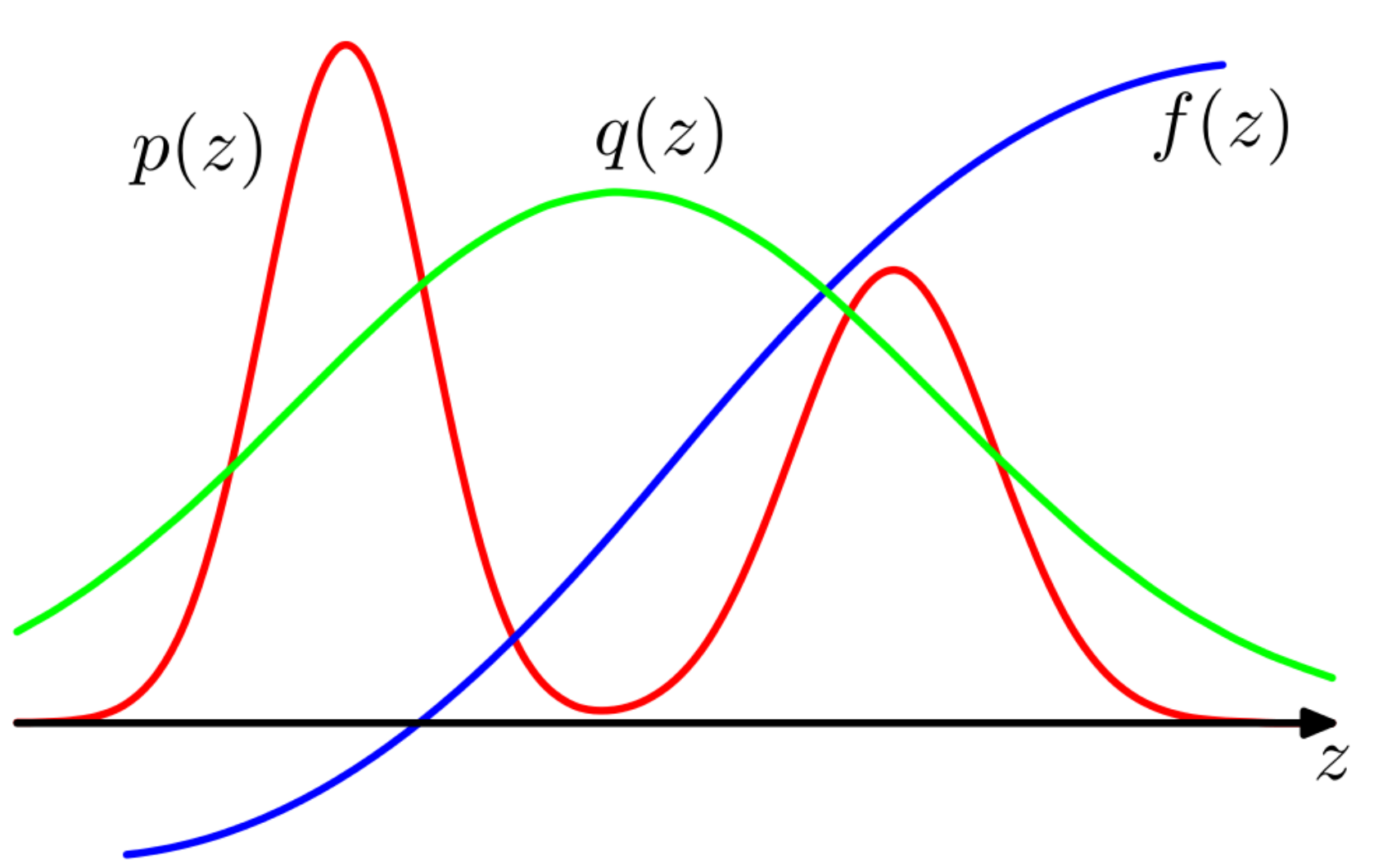
СУТЬ МЕТОДА

- Выборка по значимости — importance sampling.
- Мы решаем только вторую задачу, а не первую.
- То есть нам нужно брать сэмплы, при этом желательно попадая в зоны, где функция p^* имеет большие значения.

СУТЬ МЕТОДА

- Предположим, что у нас есть какое-то другое распределение вероятностей q (точнее, q^*), попроще, и мы умеем брать его сэмплы.
- Тогда алгоритм такой: сначала взять выборку по q^*, а затем перевзвесить её так, чтобы получилась всё-таки выборка по p^*.

ВЫБОРКА ПО ЗНАЧИМОСТИ



вывод

- Мы хотим

$$\mathrm{E}[f] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} =$$

$$\approx \frac{1}{L}\sum_r \frac{p(\mathbf{x}^{(r)})}{q(\mathbf{x}^{(r)})}f(\mathbf{x}^{(r)}).$$

- $w_r = p(\mathbf{x}^{(r)})/q(\mathbf{x}^{(r)})$ – веса, с которыми входят сэмплы, но все сэмплы остаются в множестве.

ВЫВОД

- Если у нас не p и q, а p^* и q^*, и $p = \frac{1}{Z_p} p^*$, $q = \frac{1}{Z_q} q^*$, то

$$\mathrm{E}[f] = \int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \frac{Z_q}{Z_p} \int f(\mathbf{x}) \frac{p^*(\mathbf{x})}{q^*(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \approx$$

$$\approx \frac{Z_q}{Z_p} \frac{1}{R} \sum_{r=1}^{R} \frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})} f(\mathbf{x}^{(r)}),$$

и Z_q / Z_p можно оценить из тех же сэмплов:

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q} \int p^*(\mathbf{x}) d\mathbf{x} = \int \frac{p^*(\mathbf{x})}{q^*(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \approx \frac{1}{R} \sum_{r=1}^{R} \frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})}.$$

ВЫВОД

- Получаем такой алгоритм:
  1. Взять сэмплы $\{x^{(r)}\}_{r=1}^{R}$ по распределению q^*.
  2. Рассчитать веса

$$w_r = \frac{p^*(\mathbf{x}^{(r)})/q^*(\mathbf{x}^{(r)})}{\sum_m p^*(\mathbf{x}^{(m)})/q^*(\mathbf{x}^{(m)})}.$$

  3. Оценить функцию по формуле

$$\mathrm{E}[f] \approx \frac{1}{R}\sum_{r=1}^{R} w_r f(\mathbf{x}^{(r)}).$$

ОБСУЖДЕНИЕ

- Зачем нужно q? Чем это лучше равномерного распределения?

ОБСУЖДЕНИЕ

- Зачем нужно q? Чем это лучше равномерного распределения?
- Проще говоря, распределение q должно помочь выбрать те участки, на которых имеет смысл сэмплить r.
- Если q хорошее, то может помочь, а если плохое, может только навредить.
- Но есть и более фундаментальные проблемы.

ПРОБЛЕМЫ

- Во-первых, сэмплер q не должен быть слишком узким.
- Например, если сэмплер гауссиановский с небольшой вариацией, то пики r далеко от центра q вообще никто не заметит.

ПРОБЛЕМЫ

- Во-вторых, может случиться, что все сэмплы будут напрочь убиты небольшим количеством сэмплов с огромными весами. Это плохо.
- Чтобы показать, как это бывает, давайте перейдём в многомерный случай.

ПРОБЛЕМЫ

- Пусть есть равномерное распределение r на единичном шаре и сэмплер q — произведение гауссианов с центром в нуле:

$$p(x) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{1}{2\sigma^2}\sum_i x_i^2}.$$

Упражнение. Найдите среднее и дисперсию расстояния $r^2 = \sum_i x_i^2$ точки, взятой по этому распределению.

ПРОБЛЕМЫ

- Ответ на упражнение: расстояние будет $N\sigma^2 \pm \sqrt{2N}\sigma^2$ (распределение будет похоже на гауссовское).
- Значит, почти все сэмплы лежат в «типичном множестве», кольце расстоянием около $\sigma\sqrt{N}$ от нуля.

ПРОБЛЕМЫ

- Тогда большинство сэмплов q будут лежать в интервале

$$\frac{1}{(2\pi\sigma^2)^{n/2}} 2^{-\frac{N}{2} \pm \frac{\sqrt{2N}}{2}},$$

  и ненулевые веса будут иметь значения порядка

$$(2\pi\sigma^2)^{n/2} 2^{\frac{N}{2} \pm \frac{\sqrt{2N}}{2}}.$$

- Это значит, что максимальный вес будет относиться к среднему примерно как $2^{\sqrt{2N}}$, а это очень много.

СЭМПЛИНГ В ГРАФИЧЕСКИХ МОДЕЛЯХ

- Варианты выборки по значимости для направленных графических моделей:
  - uniform sampling – фиксируем evidence, выбираем остальные равномерно, вес у сэмпла получается просто $p(\mathbf{x})$, потому что он автоматически сходится с evidence;
  - likelihood weighted sampling – фиксируем evidence, выбираем остальные от родителей к детям из условного распределения $p(x_i \mid \text{pa}(x_i))$, где $\text{pa}(x_i)$ уже зафиксированы; вес тогда будет

$$r(\mathbf{x}) = \prod_{x_i \notin E} \frac{p(x_i \mid \text{pa}(x_i))}{p(x_i \mid \text{pa}(x_i))} \prod_{x_i \in E} \frac{p(x_i \mid \text{pa}(x_i))}{1} = \prod_{x_i \in E} p(x_i \mid \text{pa}(x_i)).$$

ЗАКЛЮЧЕНИЕ

- Если размерность большая, то у выборки по значимости есть две большие проблемы.
- Во-первых, чтобы получить разумные сэмплы, нужно уже заранее выбрать q так, чтобы оно хорошо аппроксимировало p.
- Во-вторых, даже если их получить, часто может так случиться, что веса у некоторых сэмплов будут слишком велики.
- В общем, для случая многих размерностей это не очень хороший метод.

МАРКОВСКИЕ МЕТОДЫ МОНТЕ-КАРЛО

ОБЩАЯ ИДЕЯ

- Суть алгоритма похожа на выборку с отклонением, но есть важное отличие.
- Распределение q теперь будет меняться со временем, зависеть от текущего состояния алгоритма.
- Как и прежде, нужно распределение q, точнее, семейство $q(x'; x^{(t)})$, где $x^{(t)}$ — текущее состояние.
- Но теперь q не должно быть приближением p, а должно просто быть каким-нибудь сэмплируемым распределением (например, сферический гауссиан).
- Кандидат в новое состояние x' сэмплируется из $q(x'; x^{(t)})$.

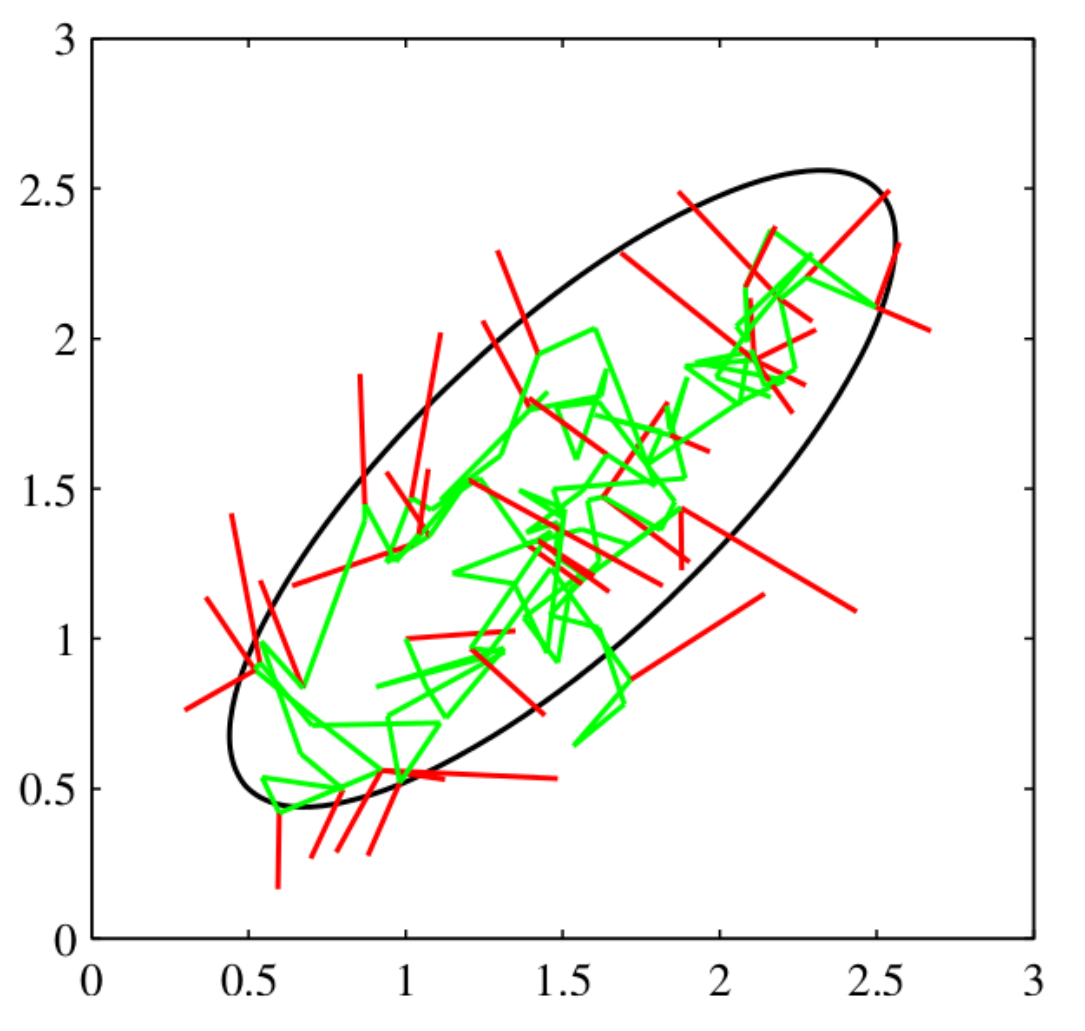
АЛГОРИТМ

- Очередная итерация начинается с состояния $x^{(i)}$.
- Выбрать x' по распределению $q(x'; x^{(i)})$.
- Вычислить
$$a = \frac{p^*(x')}{p^*(x^{(i)})} \frac{q(x^{(i)}; x')}{q(x'; x^{(i)})}.$$
- С вероятностью a (1, если $a \geq 1$) $x^{(i+1)} := x'$, иначе $x^{(i+1)} := x^{(i)}$.

ОБСУЖДЕНИЕ

- Суть в том, что мы переходим в новый центр распределения, если примем очередной шаг.
- Получается этакий random walk, зависящий от распределения p^*.
- $\frac{q(x^{(i)};x')}{q(x';x^{(i)})}$ для симметричных распределений (гауссиана) равно 1, это просто поправка на асимметрию.
- Отличие от rejection sampling: если не примем, то не просто отбрасываем шаг, а записываем $x^{(i)}$ ещё раз.

пример блуждания [bishop]



ОБСУЖДЕНИЕ

- Очевидно, что $x^{(i)}$ — отнюдь не независимы.
- Независимые сэмплы получаются только с большими интервалами.
- Поскольку это random walk, то если большая часть q сосредоточена в радиусе ϵ, а общий радиус p^* равен D, то для получения независимого сэмпла нужно будет минимум... сколько?

Упражнение. Рассмотрим одномерное случайное блуждание, где на каждом шаге с вероятностью $1/2$ точка движется влево или вправо на единицу длины. Какое ожидаемое расстояние точки от нуля после T шагов?

ОБСУЖДЕНИЕ

- Ответ на упражнение: ожидаемое расстояние будет $\sqrt{T}$.
- Значит, нам потребуется где-то $\left(\frac{D}{\epsilon}\right)^2$ шагов (и это оценка снизу).
- Хорошие новости: это верно для любой размерности. То есть времени надо много, но нет катастрофы при переходе к размерности 1000.

КОГДА РАЗМЕРНОСТЬ ВЕЛИКА

- Когда размерность большая, можно не сразу все переменные изменять по $q(x'; x)$, а выбрать несколько распределений q_j, каждое из которых касается части переменных, и принимать или отвергать изменения по очереди.
- Тогда процесс пойдёт быстрее, чаще принимать изменения будем.

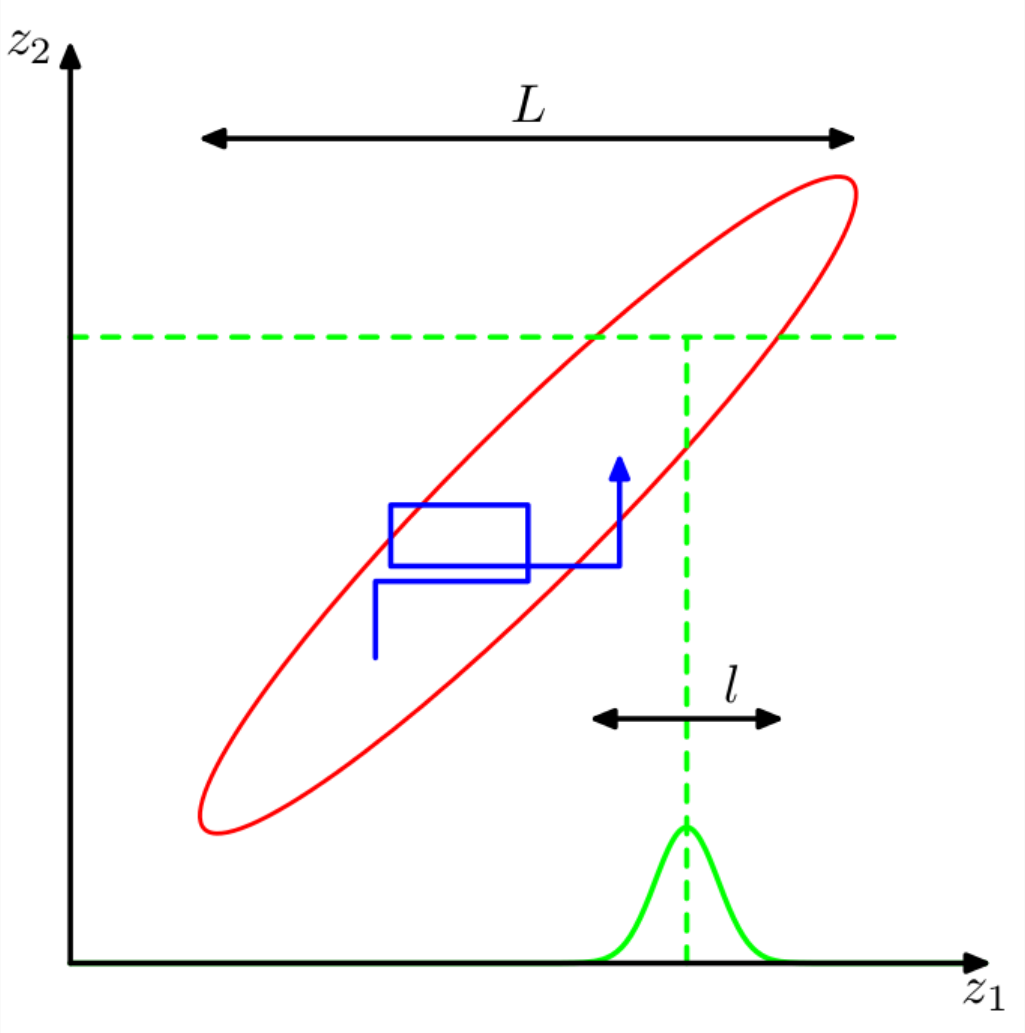
ИДЕЯ СЭМПЛИРОВАНИЯ ПО ГИББСУ

- Пусть размерность большая. Что делать?
- Давайте попробуем выбирать сэмпл не весь сразу, а покомпонентно.
- Тогда наверняка эти одномерные распределения окажутся проще, и сэмпл мы выберем.

НА ДВУХ ПЕРЕМЕННЫХ

- Пусть есть две координаты: x и y. Начинаем с (x^0, y^0).
- Выбираем x^1 по распределению $p(x|y = y^0)$.
- Выбираем y^1 по распределению $p(y|x = x^1)$.
- Повторяем.

пример [bishop]



ОБЩАЯ СХЕМА

- В общем виде всё то же самое: x_i^{t+1} выбираем по распределению

$$p(x_i | x_1^{t+1}, \dots, x_{i-1}^{t+1}, x_{i+1}^t, \dots, x_n^t)$$

  и повторяем.
- Это частный случай алгоритма Метрополиса (для распределений $q(\mathbf{x}'; \mathbf{x}) = p(x_i' \mid \mathbf{x}_{-i})$, и вероятность принятия получится 1 – упражнение).
- Поэтому сэмплирование по Гиббсу сходится, и, так как это тот же random walk по сути, верна та же квадратичная оценка.

ОБСУЖДЕНИЕ

- Нужно знать $p(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$. Это, например, особенно легко знать в байесовских сетях.
- Как будет работать сэмплирование по Гиббсу в байесовской сети?
- Для сэмплирования по Гиббсу не нужно никаких особенных предположений или знаний. Можно быстро сделать работающую модель, поэтому это очень популярный алгоритм.
- В больших размерностях может оказаться эффективнее сэмплить по несколько переменных сразу, а не по одной.

МАРКОВСКИЕ ЦЕПИ

- Марковская цепь задаётся начальным распределением вероятностей $p^0(x)$ и вероятностями перехода $T(x'; x)$.
- $T(x'; x)$ — это распределение следующего элемента цепи в зависимости от следующего; распределение на $(t+1)$–м шаге равно
$$p^{t+1}(x') = \int T(x'; x) p^t(x) dx.$$
- В дискретном случае $T(x'; x)$ — это матрица вероятностей $p(x' = i | x = j)$.

Свойства марковских цепей: инвариантное распределение

- Не всякая марковская цепь нам подойдёт.
- Во-первых, цепь должна сходиться к распределению, которое нас интересует.
- Это называется *инвариантным распределением*; инвариантное распределение π удовлетворяет

$$\pi(x') = \int T(x'; x)\pi(x)dx.$$

- Нам нужно, чтобы инвариантным распределением нашей цепи было $p(x)$, которое мы хотим сэмплировать.

Свойства марковских цепей: эргодичность

- Ну, и нужно, чтобы собственно сходилось:

$$\forall p^0(x) \quad p^t(x) \longrightarrow \pi(x) \text{ при } t \to \infty.$$

- Какие могут быть примеры неэргодичных цепей?

Свойства марковских цепей: эргодичность

- Ну, и нужно, чтобы собственно сходилось:

$$\forall p^0(x) \quad p^t(x) \longrightarrow \pi(x) \text{ при } t \to \infty.$$

- Какие могут быть примеры неэргодичных цепей?
- В цепи могут быть недостижимые состояния (тогда предел зависит от p^0).
- У цепи может быть период, т.е. предельное распределение может меняться с некоторым периодом (например, по соображениям чётности).

ИЗ ЧЕГО ДЕЛАЮТ МАРКОВСКИЕ ЦЕПИ

- Есть несколько удобных конструкций, с помощью которых можно построить достаточно сложную функцию T, сохраняя её свойства.
- Давайте их рассмотрим.

ИЗ ЧЕГО ДЕЛАЮТ МАРКОВСКИЕ ЦЕПИ: КОНКАТЕНАЦИЯ

- Можно конкатенировать распределения, запуская их друг за другом:
$$T(x', x) = \int T_2(x', x'')T_1(x'', x)dx''.$$
- При этом сохраняется инвариантное распределение (докажите).

ИЗ ЧЕГО ДЕЛАЮТ МАРКОВСКИЕ ЦЕПИ: СМЕСЬ

- Можно смешивать распределения. Если были функции $T_i(x', x)$, то можно ввести новую

$$T(x', x) = \sum_i p_i T_i(x', x), \text{ где } \sum_i p_i = 1.$$

УСЛОВИЕ БАЛАНСА

- Как убедиться, что марковская цепь сходится именно к тому распределению, которое нам нужно?
- Свойство баланса в марковских цепях: для p и T

$$\forall x, x' \quad T(x, x')p(x') = T(x', x)p(x).$$

- Т.е. вероятность того, что мы выберем x и дойдём до x', равна вероятности выбрать x' и дойти до x.
- Такие цепи называются *обратимыми* (reversible).
- Если выполняется условие баланса, то $p(x)$ — инвариантное распределение (докажите!).

МЕТРОПОЛИС--ГАСТИНГС

- Очередная итерация начинается с состояния $x^{(i)}$.
- Выбрать x' по распределению $q(x'; x^{(i)})$.
- Вычислить
$$a(x', x) = \frac{p^*(x')}{p^*(x^{(i)})} \frac{q(x^{(i)}; x')}{q(x'; x^{(i)})}.$$
- С вероятностью $a(x', x)$ (1, если $a \geq 1$) $x^{(i+1)} := x'$, иначе $x^{(i+1)} := x^{(i)}$.

МЕТРОПОЛИС--ГАСТИНГС

- Условие баланса:

$$p(x)q(x;x')a(x',x) = \min(p(x)q(x;x'), p(x')q(x';x)) =$$
$$= \min(p(x')q(x';x), p(x)q(x;x')) = p(x')q(x';x)a(x,x').$$

- Важный параметр – дисперсия распределения q; она задаёт баланс между частым принятием и быстрым перемещением по пространству состояний.

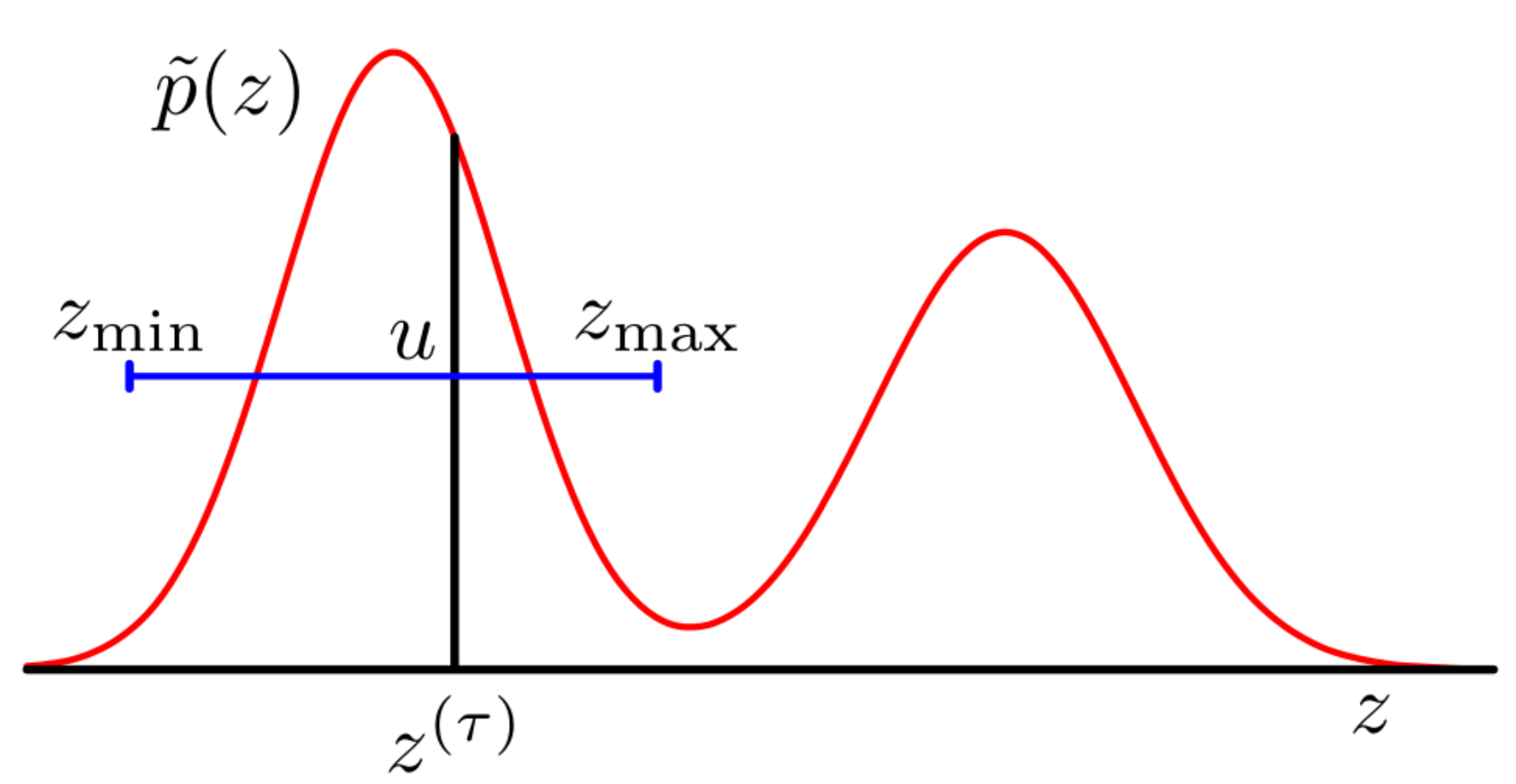
СУТЬ

- Slice sampling — ещё один алгоритм, похожий на алгоритм Метрополиса.
- Это аналог алгоритма Метрополиса, но в нём мы хотим настраивать длину шага («дисперсию») автоматически.

АЛГОРИТМ В ОДНОМЕРНОМ СЛУЧАЕ

- Мы хотим сделать random walk из одной точки под графиком p^* в другую точку под графиком p^*, да так, чтобы в пределе получилось равномерное распределение.
- Вот как будем делать переход $(x, u) \to (x', u')$:
  - Вычислим $p^*(x)$ и выберем u' равномерно из $[0, p^*(x)]$.
  - Сделаем горизонтальный интервал (x_l, x_r) вокруг x.
  - Затем будем выбирать x' равномерно из (x_l, x_r), пока не попадём под график.
  - Если не попадаем, модифицируем (x_l, x_r).
- Осталось понять, как сделать (x_l, x_r) и как его потом модифицировать.

SLICE SAMPLING



ДОПОЛНЕНИЯ К АЛГОРИТМУ

- Исходный выбор (x_l, x_r):
  - Выбрать r равномерно из $[0, \epsilon]$.
  - $x_l := x - r$, $x_r := x + (\epsilon - r)$.
  - Раздвигать границы на ϵ, пока $p^*(x_l) > u'$ и $p^*(x_r) > u'$.
- Модификация (x_l, x_r): Если x' лежит выше p^*, сокращаем интервал до x'.

СВОЙСТВА

- В алгоритме Метрополиса нужно было выбирать размер шага. И от него всё зависело квадратично.
- А тут размер шага подправляется сам собой, и эта поправка происходит за линейное время (а то и логарифм).
- В задачах с большой размерностью нужно сначала выбрать (случайно или совпадающими с осями) направление изменения y, а потом проводить алгоритм относительно параметра α в распределении $p^*(x + \alpha y)$.

ИДЕЯ

- Рассмотрим ситуацию, когда вероятность можно записать как $p(x) = \frac{1}{Z} e^{-E(x)}$.
- Во многих таких случаях можно вычислить не только $E(x)$, но и градиент $\nabla E(x)$.
- Такую информацию хотелось бы использовать.

ГАМИЛЬТОНОВА МЕХАНИКА

- Займёмся матфизикой: рассмотрим механическую систему.
- Состояние системы описывается обобщёнными координатами q и обобщёнными моментами p (векторные переменные).
- Её общая энергия $H(q, p, t) = V(q, t) + K(p, t)$, где V — потенциальная, K — кинетическая.

- Тогда система будет описываться гамильтоновыми уравнениями
$$\dot{p} = -\frac{\partial H}{\partial q}, \qquad \dot{q} = \frac{\partial H}{\partial p}.$$
- Гамильтонова механика — это, конечно, то же самое, что лагранжева, но вместо уравнений второго порядка на n переменных получаются уравнения первого порядка на $2n$ переменных.
- Важные для нас свойства: в течение эволюции системы
  1. значение гамильтониана H остаётся постоянным;
  2. объём любой области в пространстве переменных (p, q) сохраняется.

СУТЬ

- Гамильтонов метод Монте-Карло — это вариация метода Метрополиса.
- Пространство поиска $\mathbf{x}$ расширяется *моментами* $\mathbf{p}$.
- Благодаря законам сохранения гамильтонова динамика оставляет постоянным совместное распределение $p(\mathbf{x}, \mathbf{p})$; применяя эволюцию вдоль гамильтониана, можно ходить далеко по пространству состояний, не меняя распределение; а потом делать несколько «обычных» (гиббсовских, например) шагов, которые уже будут менять H.

СУТЬ

- Введём *гамильтониан* $H(\mathbf{x}, \mathbf{p}) = E(\mathbf{x}) + K(\mathbf{p})$, где K — кинетическая энергия, например $K(\mathbf{p}) = \frac{\mathbf{p}^T\mathbf{p}}{2}$.
- Теперь блуждание осуществляется двумя способами: первый случайно блуждает по пространству моментов (по Гиббсу, например).
- А второй шаг пытается сэмплировать совместную вероятность

$$p_H(\mathbf{x}, \mathbf{p}) = \frac{1}{Z_H} e^{-H(\mathbf{x},\mathbf{p})} = \frac{1}{Z_H} e^{-E(\mathbf{x})} \frac{1}{Z_H} e^{-K(\mathbf{p})}.$$

- Потом можно будет просто отбросить K и получить сэмплы для $e^{-E(\mathbf{x})}$, потому что тут всё так хорошо разделяется.

СУТЬ

- Мы хотим построить траекторию в пространстве $(\mathbf{x}, \mathbf{p})$, на которой H остаётся постоянным, а затем по методу Метрополиса либо принять, либо отклонить этот сэмпл.
- Понятно, что $\dot{\mathbf{x}} = \mathbf{p}$, а гамильтоновы уравнения нам говорят, что
$$\dot{\mathbf{p}} = -\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}}.$$

СУТЬ

- Осталось это проинтегрировать. Для этого можно использовать leapfrog technique приближённого интегрирования:

$$
\begin{aligned}
p_i(t+\tfrac{\tau}{2}) &= p_i(t) - \tfrac{\tau}{2}\,\frac{\partial E}{\partial x_i}\Big|_{\mathbf{x}(t)}, \\
x_i(t+\tau) &= x_i(t) + \tfrac{\tau}{m_i} p_i(t+\tfrac{\tau}{2}), \\
p_i(t+\tau) &= p_i(t+\tfrac{\tau}{2}) - \tfrac{\tau}{2}\,\frac{\partial E}{\partial x_i}\Big|_{\mathbf{x}(t+\tau)}.
\end{aligned}
$$

- Дополнительные «половинные» шаги позволяют добиться погрешности второго порядка по τ.

СУТЬ

- Алгоритм делает m leapfrog шагов, потом по методу Метрополиса принимает или отвергает получившуюся точку (проекцию на $\mathbf{x}$).
- То есть если мы можем подсчитывать ∇E, а не только E, мы можем включить эту информацию в наш random walk.
- В результате он будет двигаться более-менее в правильном направлении, и пройденное расстояние $\sqrt{n}$ превратится в n (доказывать уж не будем).

СПАСИБО!

Спасибо за внимание!