RL: Многорукий бандит

Многорукий бандит

Многорукий бандит

Многорукий бандит: простая модель

Простой случай: нет различных "состояний", только N действий

Exploration: выяснить, какое действие в целом лучше;
сделать как можно меньше плохих действий

Многорукий бандит: контекст

Упрощенный MDP с одним шагом

Почему бандиты: здесь проще объяснять математику,
формулы примерно на 50% короче
(обобщении MDP далее)

Что такое контекстный бандит

Упрощенный MDP с одним шагом

Примеры:
- Баннерная реклама
- Рекомендации
- Медицинское лечение

В основном это одношаговый MDP, где
- $G(s,a) = r(s,a)$
- $Q(s,a) = E\ r(s,a)$
- Все формулы на 50% короче

Как измерить exploration

Идеи?

Как измерить exploration

Плохая идея: по звучанию названия

Хорошая идея: по $$$, которые она вам принесла/потеряла

Regret политики $\pi(a|s)$:

Рассмотрим оптимальную политику, $\pi^*(a|s)$

Regret = сумма за время обучения [оптимальная – ваша]

$$\eta = \sum_t \underset{s,a\sim\pi^*}{\mathrm{E}} r(s,a) - \underset{s,a\sim\pi}{\mathrm{E}} r(s,a)$$

Конечный горизонт: t<max_t

Бесконечный горизонт: t -> inf

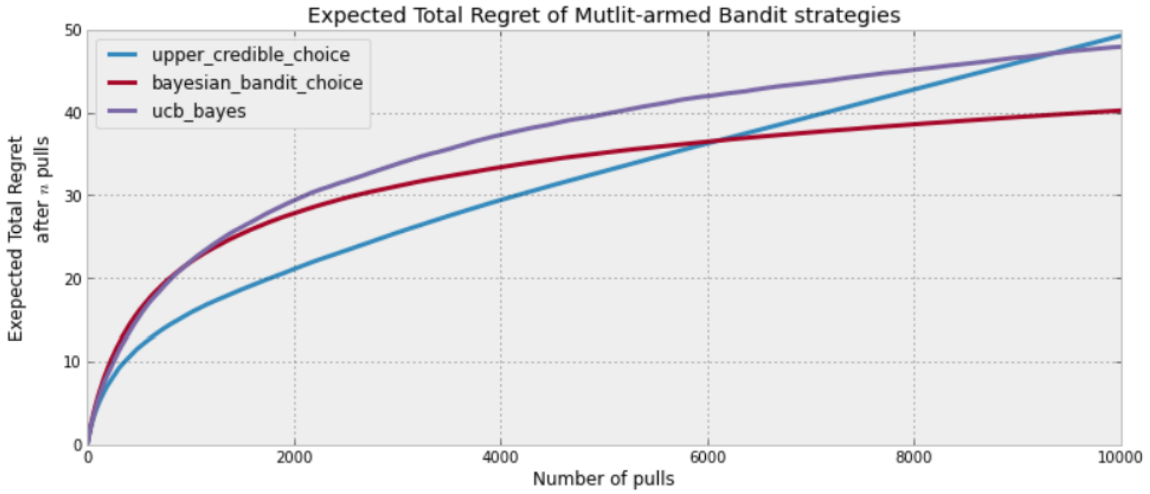
Как измерить exploration

Плохая идея: по звучанию названия

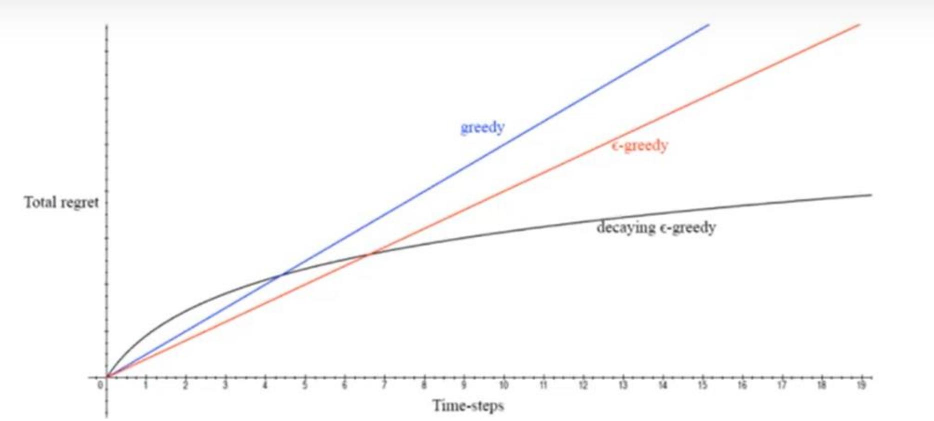
Хорошая идея: по $$$, которые она вам принесла/потеряла

Regret политики $\pi(a|s)$:

Regret за попытку = оптимально - ваша



Линейные и сублинейные потери



Если алгоритм не содержит исследовательскую составляющую – полные потери будут линейны

Если алгоритм содержит постоянную исследовательскую часть – полные потери тоже будут линейны

Возможно ли достичь сублинейных потерь?

Exploration Vs Exploitation

Стратегии исследований

- **ε - жадная :**
  - С вероятностью ε принять равномерно случайное действие
  - В противном случае принять оптимальное действие
- Соотношение Больцмана
  - Выбирать действие пропорционально преобразованному q значению

$$P(a) = softmax\left(\frac{Q(s,a)}{std}\right)$$

- Оптимистическая инициализация

  \- начинать с высокого начального Q(s,a) для всех состояний/действий

  \- хорошо подходит для табличных алгоритмов, трудно аппроксимировать

Стратегии исследований

- **ε -** жадная **:**
  - С вероятностью ε принять равномерно случайное действие
  - В противном случае принять оптимальное действие

Таким образом, если мы используем ε жадную стратегию с ε=0.25, что мы можем сказать о величине regret?

$$\eta = \sum_t \mathop{\mathrm{E}}_{s,a\sim\pi^*} r(s,a) - \max_{s,a\sim\pi} r(s,a)$$

Стратегии исследований

- **ε - жадная :**
  - С вероятностью ε принять равномерно случайное действие
  - В противном случае принять оптимальное действие

Таким образом, если мы используем ε жадную стратегию с ε=0.25, что мы можем сказать о величине regret?

Regret растет линейно с течением времени!

Агент всегда действует не оптимально из-за ε

Exploration во времени

- **Идея:**

  - Если вы хотите сходиться к оптимальной политике, вам нужно постепенно сокращать exploration

- **Пример:**

Инициализируем ε = 0.5 и затем постепенно будем уменьшать это значение

- Если ε → 0, то в пределе это жадная стратегия
  - С нестационарными средами надо быть осторожными

Оптимистичная инициализация

- Простая и практичная идея: инициализируем *Q(a)* наивысшим значением
- Обновляем полезности действия по методу МК

$$Q_t(a_t) = Q_t(a_t) + \frac{1}{N_t(a_t)}(r_t - Q_{t-1})$$

Это поощряет систематическое исследование на ранних стадиях, однако проблема постоянного выбора субоптимального выбора действия остается

- Жадный алгоритм + оптимистичная инициализация также имеет линейные общие потери (total regret)
- ε-жадный алгоритм + оптимистичная инициализация также имеет линейные общие потери

Сколько необходимо сделать случайных удачных действий сделать, чтобы:

- ❑ Оказать медицинскую помощь
- ❑ Управлять роботом
- ❑ Оптимизировать продажи

Сколько необходимо сделать случайных удачных действий сделать, чтобы:

- Оказать медицинскую помощь
- Управлять роботом
- Оптимизировать продажи

Люди учатся не используя ε жадную стратегию

Байесовские методы:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(E[r]||data)$$

https://habr.com/ru/company/ods/blog/325416/

Нижняя граница

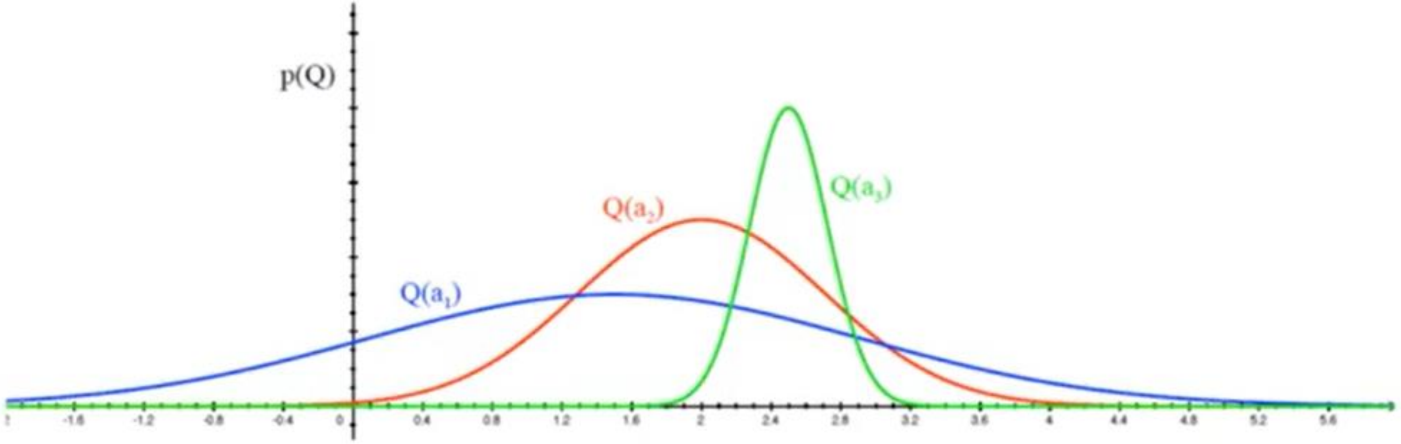
- Производительность любого алгоритма для задачи бандитов определяется сходством между оптимальной рукой и другими руками
- Сложными являются задачи, в которых похожие руки имеют разные полезности
- Формально это можно описать через расхождение Δ_a и сходство распределений $KL(R^a||R^{a^*})$

Теорема (Lia and Robbins)

В асимптоте полные потери зависят от количества шагов по крайней мере логарифмически

$$\lim_{t\to\infty} L_t \geq \log t \sum_{a|\Delta_a>0} \frac{\Delta_a}{KL(R^a||R^{a^*})}$$

Оптимизм в неопределенности (Optimism in face of uncertainty)

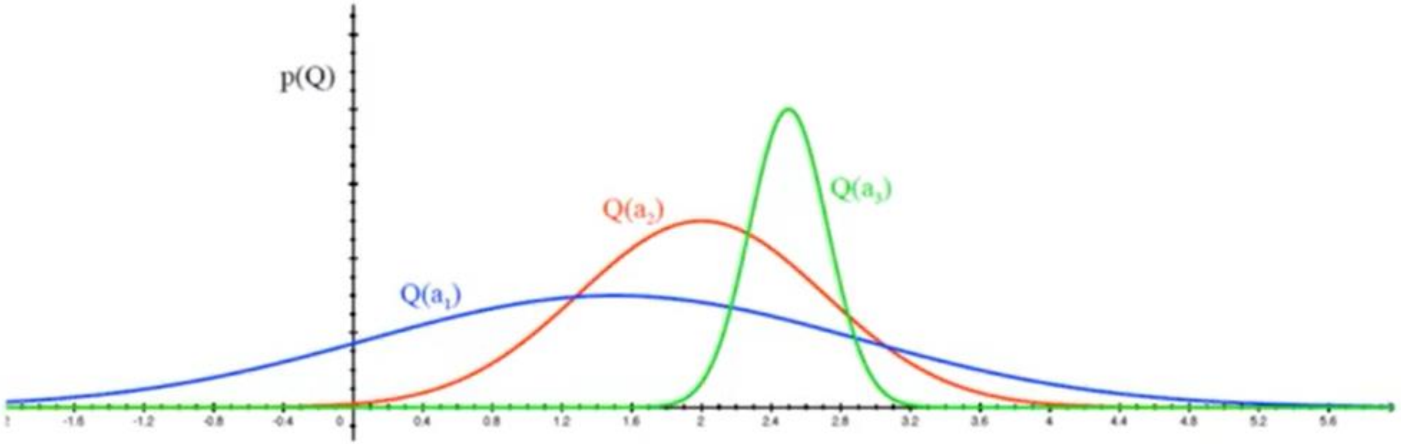


- Какое действие выбрать
- Чем больше мы сомневаемся в полезности действия, тем

большее значение имеет исследование этого действия.

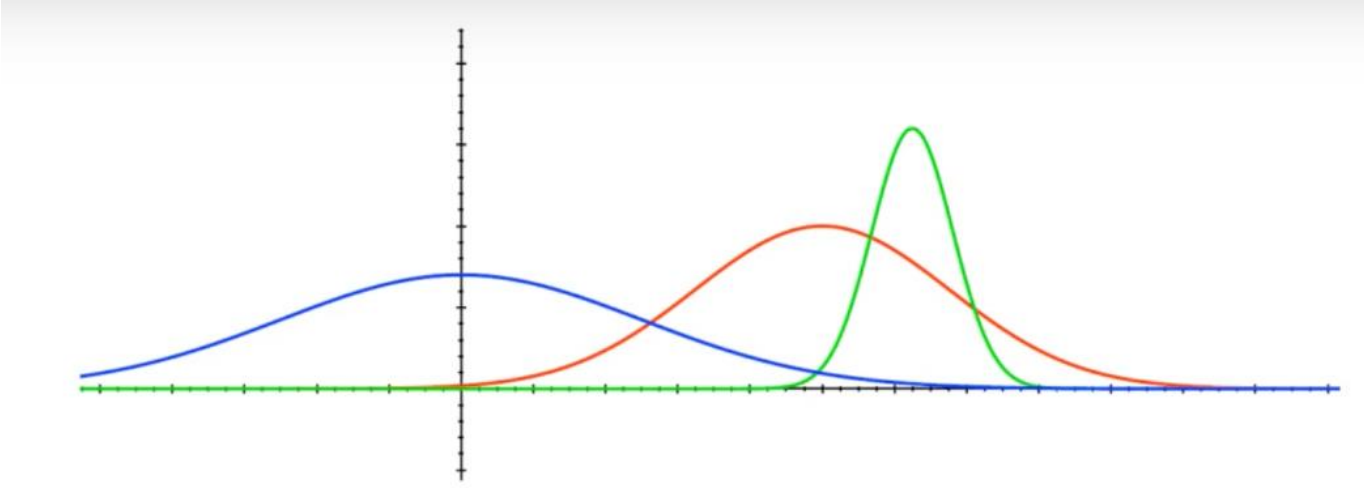
- Это действие может оказаться наилучшим

Оптимизм в неопределенности



- Вычислим 95% верхнюю доверительную границу для каждого бандита
- Выбрать действие по наибольшей доверительной границе
- Настройка: изменить 95% на больше/меньше

Оптимизм в неопределенности



- Изменения после выбора синего действия
- Неопределенность стала ниже
- По этому с большей вероятностью выберем другое действие
- Будем это учитывать, пока не придем к лучшему действию

Верхняя доверительная граница (Upper Confidence Bound – UCB)

- Будем оценивать верхнюю доверительную границу $\hat{U}_t(a)$ для каждого действия
- При этом будем добиваться того, чтобы $Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ выполнялось с высокой вероятностью
- Это зависит от количества $N_t(a)$, сколько раз было выбрано это действие:
  - Небольшое значение $N_t(a)$ – высокая $\hat{U}_t(a)$ (оценка полезности не определена)
  - Большое значение $N_t(a)$ – низкая $\hat{U}_t(a)$ (оценка полезности точна)
- Выбираем действие с учетом верхней доверительной границы (UCB):

$$a_t = \operatorname*{argmax}_{a \in A} \hat{Q}_t(a) + \hat{U}_t(a)$$

Неравенство Хёфдинга

❑ Теорема (неравенство Хёфдинга)

Пусть $X_1, \dots, X_t$ *- независимые и одинаково распределенные случайные величины в интервале [0, 1] и* $\bar{X}_t = \frac{1}{\tau}\sum_{\tau=1}^{t} X_\tau$ *- выборочное среднее, тогда:*

$$P[E[X] > \hat{X}_t + u] \leq e^{-2tu^2}$$

- Применяем неравенство Хёфдинга к вознаграждениям бандита при условии выбора действия *a*:

$$P[Q[a] > \hat{Q}_t(a) + U_t(a)] \leq e^{-2N_t(a)U_t(a)^2}$$

Вычисление верхних доверительных границ

- Выберем вероятность p, с которой истинное значение превышает UCB
- Найдем для нее $\hat{U}_t(a)$:

$$e^{-2N_t(a)U_t(a)^2} = p$$

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

- Будем уменьшать р так, чтобы мы наблюдали больше вознаграждений, например $p = t^{-4}$
- Это гарантирует, что мы будем выбирать оптимальные действия при $t \to \infty$:

$$U_t(a) = \sqrt{\frac{2\log t}{N_t(a)}}$$

Алгоритм USB1

- В итоге получаем алгоритм UCB1 для многорукого бандита:
- Найдем для нее $\widehat{U}_t(a)$:

$$a_t = \underset{a \in A}{\operatorname{argmax}}\, Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

- Теорема:

В пределе алгоритм UCB имеет логарифмические полные потери:

$$\lim_{t \to \infty} L_t \geq 8 \log t \sum_{a | \Delta_a > 0} \Delta_a$$

Байесовские бандиты

- До этого мы предполагали, что у нас нет представления о распределении R
  - За исключением границ на вознаграждения
- Байесовские бандиты используют априорные знания о вознаграждении *p[R]*
- В них вычисляется апостериорное распределение вознаграждений $p[R|h_t]$, где $h_t = a_1, r_1, \dots, a_{t-1}, r_{t-1}$ - история
- Мы можем использовать апостериорное распределение для управления исследованием среды:
  - Для вычисления верхних доверительных границ (байесовский UCB)
  - Применяя соответствие вероятностей (Thompson sampling)
- В итоге мы можем получить лучшую производительность (если априорные знания достаточно точны)

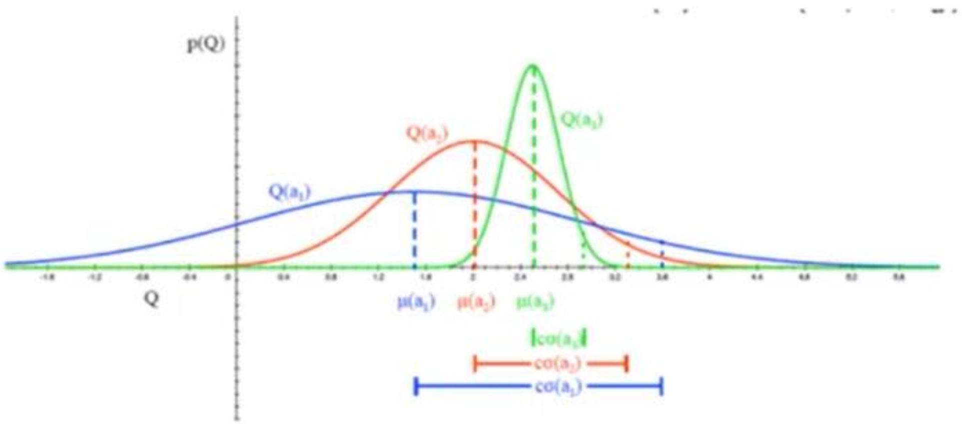
Пример Байесовского UCB: независимые гауссианы

- Предположим, что распределение вознаграждений гауссово:

$R^a(a) = N(r; \mu_a, \sigma_a^2)$

- Вычислим апостериорное гауссово распределение на μ_a и σ_a^2:

$p[\mu_a, \sigma_a^2 | h_t] \propto p[\mu_a, \sigma_a^2] \prod_{t|a_t=a} N(r_t, \mu_a, \sigma_a^2)$



- Выберем действие, максимизирующее стандартное отклонение от *Q(a):*

$$a_t = \underset{a \in A}{\operatorname{argmax}}\, \mu_a + \frac{c\sigma}{\sqrt{N(a)}}$$

Соответствие вероятностей

- Сопоставление вероятностей позволяет выбрать действие в соответствии с вероятностью того, что оно является оптимальным:

$$\pi(a|h_t) = P[Q(a) > Q(a'), \forall a' \neq a|h_t]$$

- Сопоставление вероятностей аналогично оптимизму в неопределенности: неопределенные действия имеют более высокую вероятность быть оптимальными
- Основная сложность – аналитически посчитать из апостериорного распределения

Томпсоновская выборка

- ТВ реализует сопоставление вероятностей::

$$\pi(a|h_t) = P[Q(a) > Q(a'), \forall a' \neq a|h_t] = E_{R|h_t}\left[1\left(a = \operatorname*{argmax}_{a \in A} Q(s)\right)\right]$$

- Используем байесовское правило для вычисления апостериорного распределения $p[R|h_t]$
- Проводим выборку из апостериорного распределения вознаграждения *R*
- Вычисляем полезность действия $Q(a) = E[R^a]$
- Выбираем действие максимизируеющее полезность на выборке:

$$a_t = \operatorname*{argmax}_{a \in A} Q(a)$$

- ТВ позволяет достичь границы Лаи-Роббинса

Полезность информации

- Исследование среды полезно, т.к. это дает новую информацию о среде. Как можно измерить это количество информации?
  - Каким количеством вознаграждения агент может пожертвовать за эту информацию до принятия решения.
  - Долгосрочные вознаграждения после получения информации – немедленное вознаграждения
- Информационная добавка выше для неопределенных ситуаций, поэтому имеет мсысл исследовать эти ситуации больше
- Если мы знаем полезность информации, мы сможем решить дилемму исследования и использования оптимально

Пространство информационных состояний

- Мы рассматриваем бандитов как одношаговую задачу принятия решений
- На нее можно взглянуть с точки зрения последовательного принятия решения
- На каждом шаге у нас есть информационное состояние $\tilde{s}$:
  - $\tilde{s} = f(h_t)$ – статистика истории,
  - Аккумуляция всей раннее поступившей информации
- Каждое действие приводит к переходу от состояния $\tilde{s}$ к состоянию $\tilde{s}'$ (за счет добавления новой информации) с вероятностью $\tilde{P}^{a}_{\tilde{s},\tilde{s}'}$.
- Это приводит к определению МППР в расширенном информационном пространстве состояний:

$$\widetilde{M} = \langle \tilde{S}, A, \tilde{P}, R, \gamma \rangle$$

Пример бернуллиевские бандиты

- Рассмотрим бернуллиевского бандита, для которого $R^a = B(\mu_a)$
- Проигрыш или выигрыш игры с вероятностью μ_a
- Мы хотим найти информационное состояние $\tilde{s} = \langle \alpha, \beta \rangle$, где:
  - α_a – количество применения действия a с получения вознаграждения 0
  - β_a - количество применения действия a с получения вознаграждения 1

Решение задачи в информационном пространстве бандитов

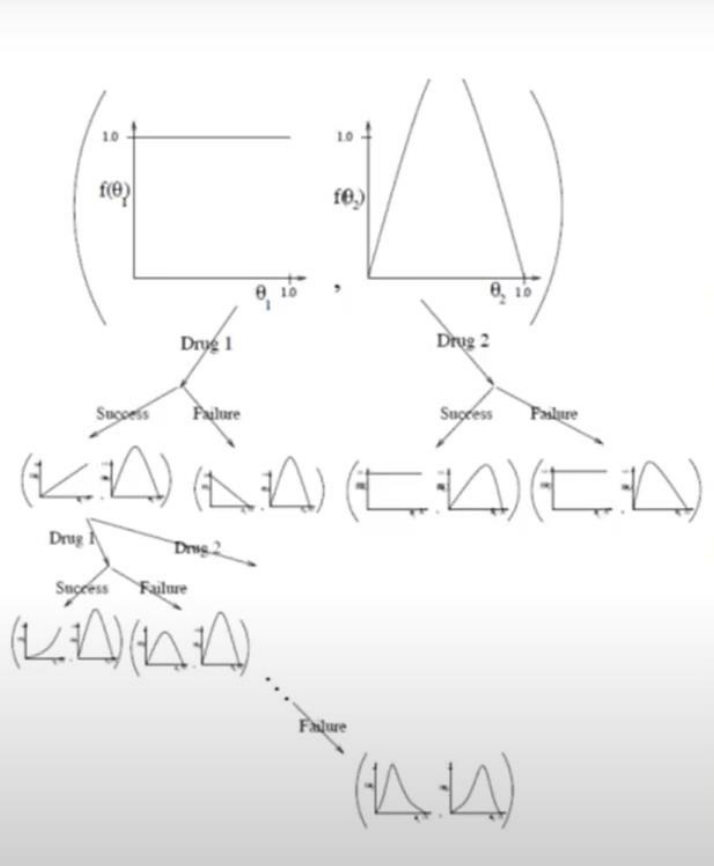
- В результате мы получим бесконечный МППР на множестве информационных состояний
- Мы можем найти решений этого МППР с помощью обучения с подкреплением:
  - Безмодельные методы, например, Q-learning (Duff, 1994)
  - Байесовские модели в обучении с подкреплением, например, индексы Джиттинса (Gittins, 1979) – адаптивное по Байесу обучение с подкреплением (Bayes-adaptive RL), позволяет найти оптимальное по Байесу решение дилеммы исследования/использования с учетом априорного распределения

Адаптивные по Байесу бернуллиевские бандиты

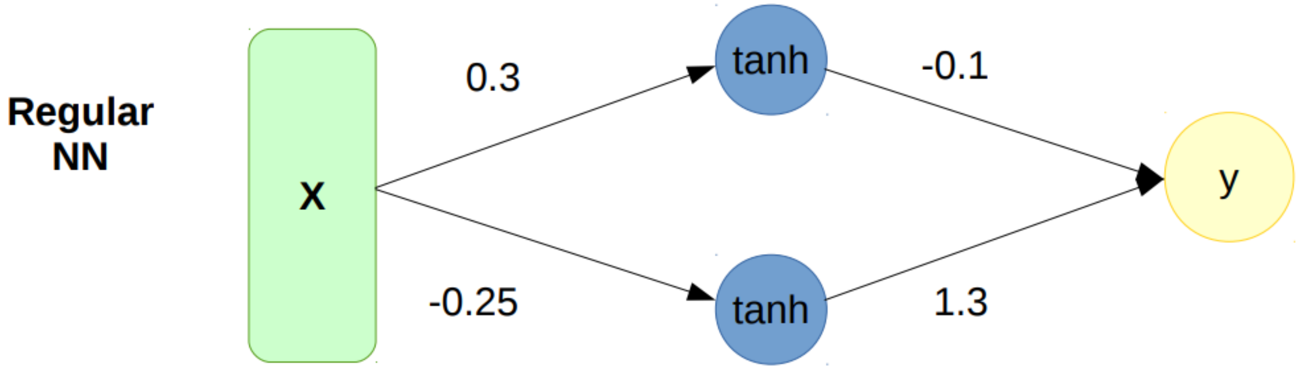
- Начинаем с априорного распределения $Beta(\alpha_a, \beta_a)$ по функции вознаграждения R^a. Каждый раз, когда выбирается действие a, обновляем апостериорное распределения для R^a:
  - $Beta(\alpha_a + 1, \beta_a), \text{если}\, r = 0$
  - $Beta(\alpha_a, \beta_a + 1), \text{если}\, r = 1$
- Таким образом мы определяем функцию переходов $\tilde{P}$ для адаптивного по Байесу МППР
- Информационное состояние $\langle \alpha_a, \beta_a \rangle$ соответствует модели вознаграждения $Beta(\alpha_a, \beta_a)$
- Каждый переход соответствует байесовскому обновлению модели

Адаптивные по Байесу бернуллиевские бандиты

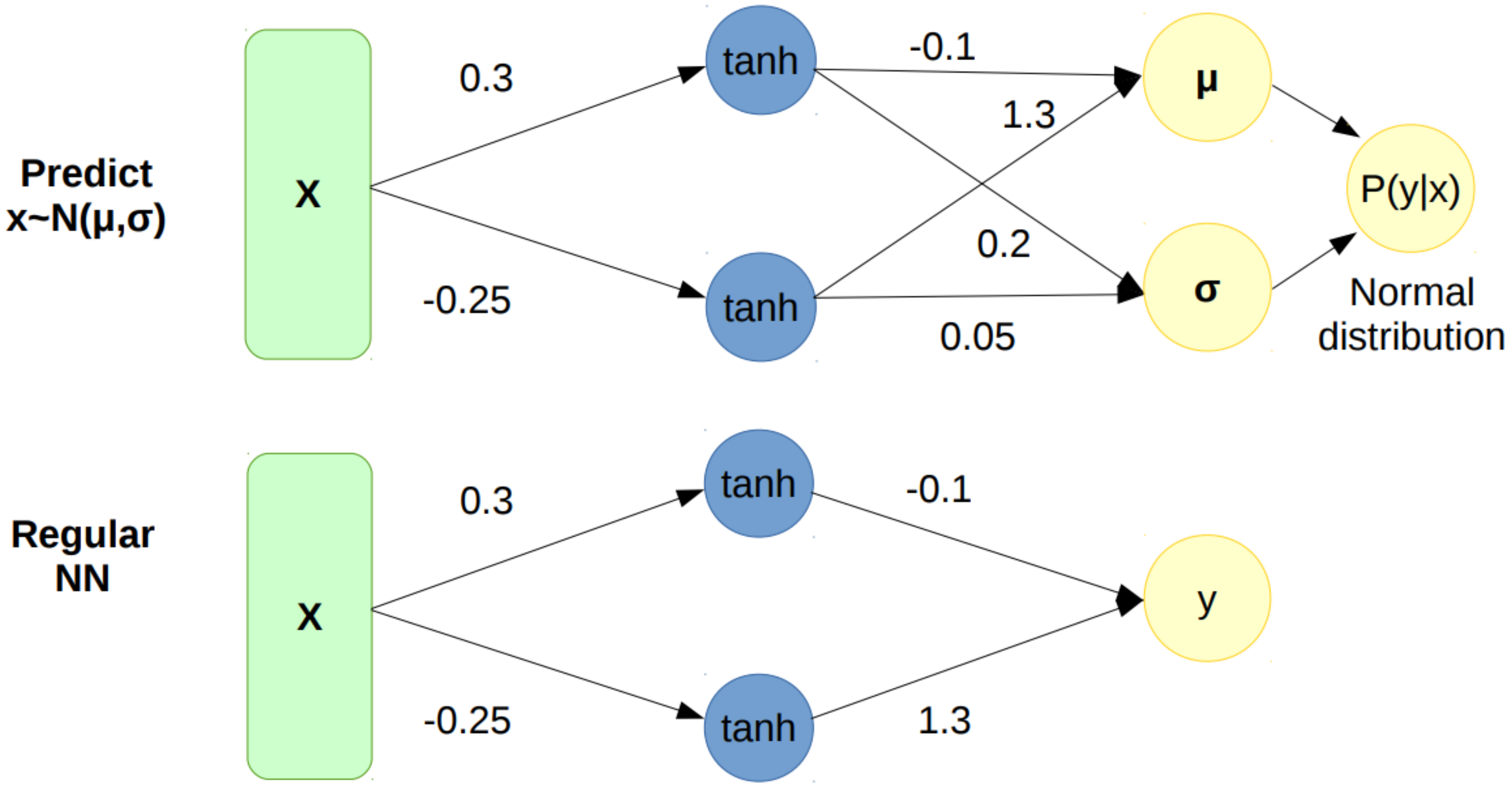
- $Beta(\alpha_a + 1, \beta_a)$, если $r = 0$
- $Beta(\alpha_a, \beta_a + 1)$, если $r = 1$



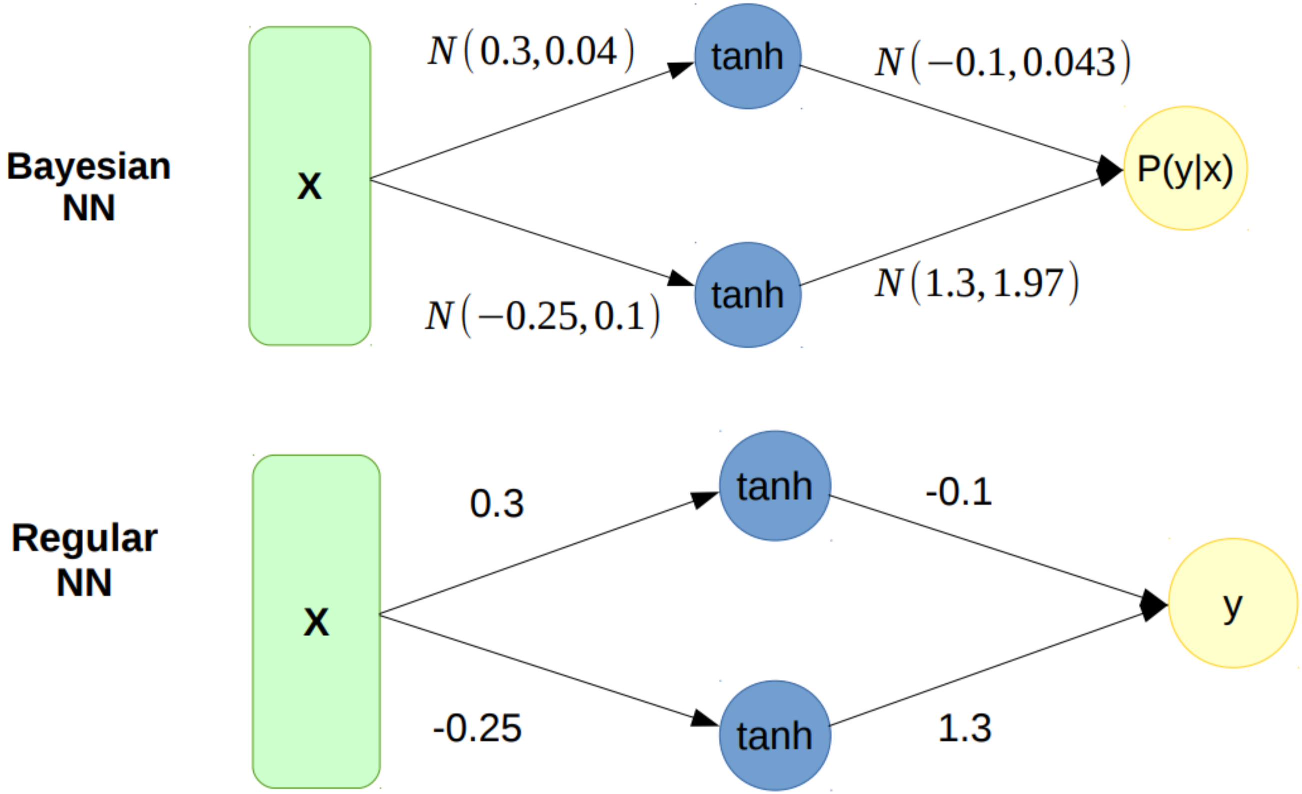
Параметрическая оценка



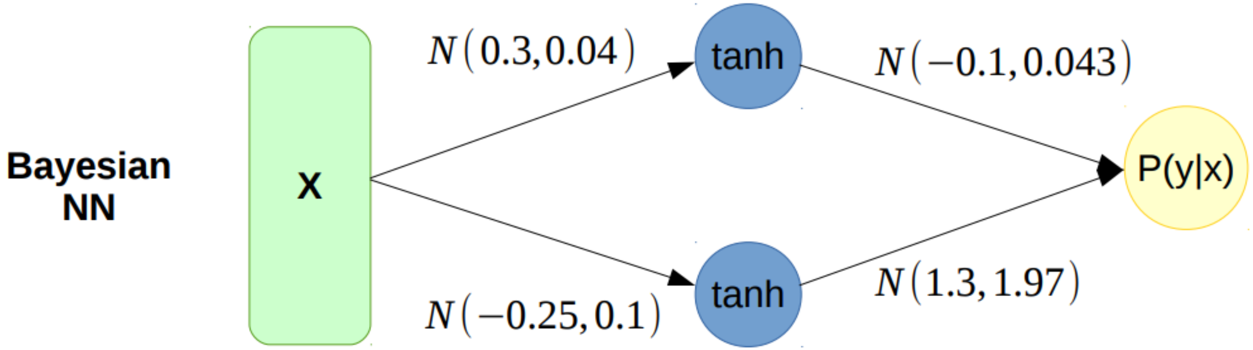
Параметрическая оценка



Байесовские нейронные сети



Байесовские нейронные сети



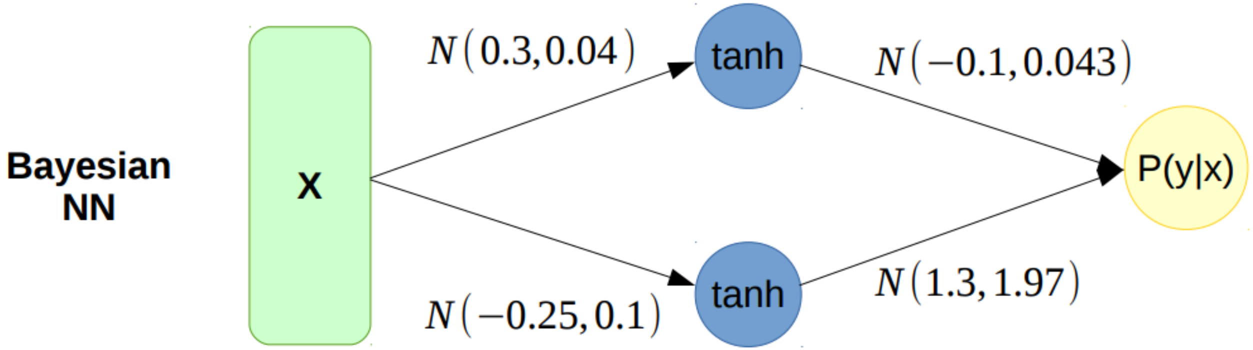
Идея:

- Никаких явных весов
- Поддерживать параметрическое распределение на весах
- Практика: полнофакторное нормальное или аналогичное

$$q(\theta|\varphi:[\mu,\sigma]) = \prod_i N(\theta_i|\mu_i,\sigma_i)$$

$$P(y|x) = E_{\theta\sim q(\theta|\varphi)} P(y|x,\theta)$$

Байесовские нейронные сети



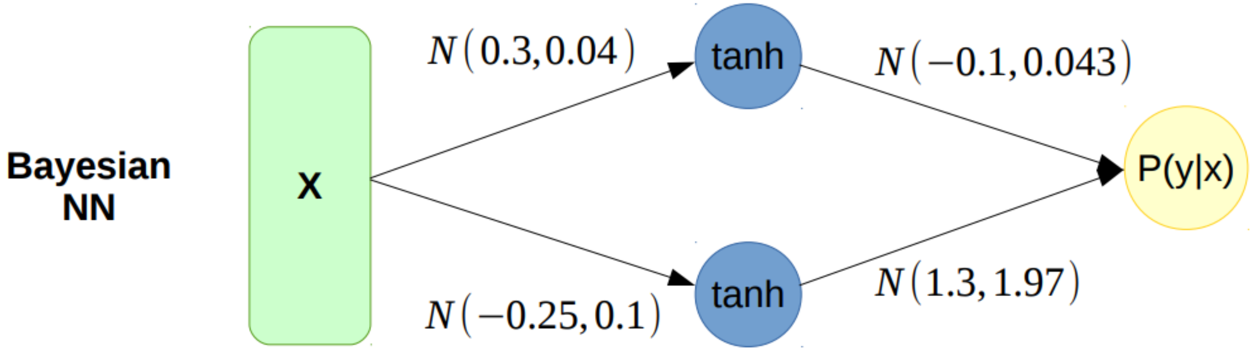
Идея:

- Никаких явных весов
- Поддерживать параметрическое распределение на весах
- Практика: полнофакторное нормальное или аналогичное

$$q(\theta|\varphi\colon[\mu,\sigma]) = \prod_i N(\theta_i|\mu_i,\sigma_i)$$

$$P(y|x) = E_{\theta\sim q(\theta|\varphi)}P(y|x,\theta)$$

Байесовские нейронные сети



Идея:

- Никаких явных весов
- Inference: выборка из распределений весов, предсказание 1 точки
- Чтобы получить распределение, объединить K выборок (например, с помощью гистограммы)
- Да, это означает многократный прогон сети для одного X

$$P(y|x) = E_{\theta \sim q(\theta|\varphi)} P(y|x, \theta)$$

Байесовские нейронные сети

Идея:

- Никаких явных весов
- Поддерживать параметрическое распределение на весах
- Практика: полнофакторное нормальное или аналогичное

$$q(\theta|\varphi:[\mu,\sigma]) = \prod_i N(\theta_i|\mu_i,\sigma_i)$$

$$P(y|x) = E_{\theta\sim q(\theta|\varphi)} P(y|x,\theta)$$

- Выучить параметры этого распределения (трюк репараметризации)
  - Меньшая дисперсия: локальный трюк репараметризации.

$$\varphi = \underset{\varphi}{\operatorname{argmax}}\, E_{x_i,y_i\sim d} E_{\theta\sim q(\theta|\varphi)} P(y_i|x_i,\theta)$$

d - dataset

Хотим получить явные формулы?

Lower bound

$$-KL\big(q(\theta|\varphi)||p(\theta|d)\big) = -\int_{\theta} q(\theta|\varphi)\log\frac{q(\theta|\varphi)}{p(\theta|d)}d\theta -$$

$$-\int_{\theta} q(\theta|\varphi)\log\frac{q(\theta|\varphi)}{\left[\frac{p(d|\theta)p(\theta)}{p(d)}\right]}d\theta = -\int_{\theta} q(\theta|\varphi)\log\frac{q(\theta|\varphi)p(d)}{p(d|\theta)p(\theta)}d\theta =$$

$$-\int_{\theta} q(\theta|\varphi)\left[\log\frac{q(\theta|\varphi)}{p(\theta)} - \log p(d|\theta) + \log p(d)\right]d\theta$$

$$\left[E_{\theta\sim q(\theta|\varphi)}\log p(d|\theta)\right] - KL\big(q(\theta|\varphi)p(\theta)\big) + \log p(d)$$

Loglikelihood -distance to prior +const

Lower bound

$$\varphi = \underset{\varphi}{arg\max}\left(-KL\left(q(\theta|\varphi)||p(\theta|d)\right)\right)$$

$$\underset{\varphi}{\operatorname{argmax}}\left(\left[E_{\theta\sim q(\theta|\varphi)}\log p(d|\theta)\right] - KL\left(q(\theta|\varphi)p(\theta)\right)\right)$$

Можно ли выполнить градиентный метод напрямую?

Трюк с репараметризацией

$$\varphi = \underset{\varphi}{argmax}\left(-KL\left(q(\theta|\varphi)||p(\theta|d)\right)\right)$$

$$\underset{\varphi}{\text{argmax}}\left(\left[E_{\theta\sim q(\theta|\varphi)}\log p(d|\theta)\right] - KL\left(q(\theta|\varphi)p(\theta)\right)\right)$$

Репараметризация

Простая формула
Для нормального q

Можно ли выполнить градиентный метод напрямую?

Правдоподобие BNN

Что означает этот log P(d|..)?

$$E_{\theta\sim N(\theta|\mu_\varphi,\sigma_\varphi)}\log p(d|\theta) = E_{\rho\sim N(0,1)}\log p\left(d|\left(\mu_\varphi + \sigma_\varphi\rho\right)\right)$$

Трюк с репараметризацией

$$\varphi = \underset{\varphi}{arg\max}\left(-KL\left(q(\theta|\varphi)||p(\theta|d)\right)\right)$$

$$\underset{\varphi}{\operatorname{argmax}}\left(\left[E_{\theta\sim q(\theta|\varphi)}\log p(d|\theta)\right] - KL\left(q(\theta|\varphi)p(\theta)\right)\right)$$

Правдоподобие BNN

Что означает этот log P(d|..)?

$$E_{\theta\sim N(\theta|\mu_\varphi,\sigma_\varphi)}\log p(d|\theta) = E_{\rho\sim N(0,1)}\log p\left(d|\left(\mu_\varphi + \sigma_\varphi\rho\right)\right)$$

Использование BNNs

Если вы делаете выборку из BNNs

- Можно выучить ~ произвольное распределение (например, мультимодальное)

- Но это требует многократного запуска сети

- Используйте эмпирические процентили для приоритета исследования

Марковские процессы принятия решений

Наивный подход:

- Вывести апостериорное распределение для Q(s,a)
- Провести UCB или выборку Томпсона по этим Q-значениям.
- Что-нибудь не так?

Марковские процессы принятия решений

- Наивный подход:

- Вывести апостериорное распределение для Q(s,a)

- Провести UCB или выборку Томпсона по этим Q-значениям.

- Агент "жаден" в отношении разведки.

  Он предпочитает предпринять одно неопределенное действие сейчас, чем сделать несколько шагов, чтобы оказаться в неисследованных регионах

Марковские процессы принятия решений

- Наивный подход:

- Вывести апостериорное распределение для Q(s,a)

- Провести UCB или выборку Томпсона по этим Q-значениям.

- Агент "жаден" в отношении разведки.

  Он предпочитает предпринять одно неопределенное действие сейчас, чем сделать несколько шагов, чтобы оказаться в неисследованных регионах

- Увеличение вознаграждения

- Придумайте суррогатное "вознаграждение" за исследование

- Мы "платим" нашему агенту за исследование

- Максимизируем это вознаграждение с помощью (отдельного) RL-агента

Аугментация вознаграждения

- Давайте "заплатим" агенту за разведку!

$$\tilde{r}(s, a, s') = r(s, a, s') + r_{exploration}(s, a, s')$$

Аугментация вознаграждения

- Давайте "заплатим" агенту за разведку!

$$\tilde{r}(s, a, s') = r(s, a, s') + r_{exploration}(s, a, s')$$

Вопрос: любые предложения для суррогата для r для atari

Основная идея UNREAL

- Вспомогательные цели:

  - Управление пикселями: максимизация изменения пикселей в сетке NxN на изображении

  - Управление характеристиками: максимизировать активацию некоторого нейрона в глубине нейронной сети

  - Прогнозирование вознаграждения: предсказать будущее вознаграждение, учитывая историю

article: arxiv.org/abs/1611.05397 blog post:
bit.ly/2g9Yv2A