Машинное обучение: Сэмплирование

Кураленок И.Е.

Яндекс

22 марта 2012 г.

Содержание

Понятие сэмплирования

Семплирование – метод исследования множества путем анализа его подмножеств.

Применяется:

- множество слишком велико для перебора;
- каждое дополнительное измерение дорого;
- предварительный анализ;

Алгоритм сэмплирования

1. Понять какое множество мы изучаем
2. Осознать что из этого множества мы можем измерить
3. Определить количество измерений
4. Разработать план сэмплирования
5. Провести сэмплирование

Типы сэмплирования

Вероятностное сэмплирование

$$p(x), \forall x : p(x) > 0 \tag{1}$$

Например: попробуем посчитать соотношение мужчин/женщин

Невероятностное сэмплирование

$$p(x), \exists x : p(x) = 0 \tag{2}$$

Например: попробуем посчитать безработных в рабочее время

Без возвращений

С возвращениями

Виды сэмплирования

- Вероятностное сэмплирование
  - Простое вероятностное
  - Систематическое
  - Стратифицированное (oversampling!)
  - Пропорциональное
  - Кластерное
- Невероятностное сэмплирование
  - Опрос ближайших
  - Панельное сэмплирование

Как выбрать нужное?

Надо учитывать:

- природа и размер возможного сампла;
- наличие дополнительной информации о элементах;
- необходимая точность измерений;
- точность отдельных измерений в сэмплировании;
- стоимость измерений;

Возвращаясь к ML

$$F_0 = \underset{F}{\operatorname{argmax}} p(F|X) \quad (3)$$

+ если известны вероятности можно попробовать посэмплировать решения;
- не определено пространство F;
- неясно как устроить обход;

Иногда все просто

$$F_0 = \operatorname*{argmax}_{F \in \{f_i\}_{i=1}^n} p(F|X) \tag{4}$$

1. введем порядок обхода;
2. переберем все возможные решения;
3. составим взвешенное решение/выберем лучшее;

Но чаще все непросто

$$F_0 = \underset{F \in \{f_i\}_{i=1}^{\infty}}{\operatorname{argmax}} p(F|X) \tag{5}$$

1. введем порядок обхода;
2. применим систематическое сэмплирование;
3. составим взвешенное решение/выберем лучшее;

Случайное блуждание

Чтобы построить порядок обхода можно воспользоваться такой схемой:

$$F = F\left(x, \lambda\right), \qquad \lambda \in \mathbb{R}^n \tag{6}$$

$$F_t = F\left(x, \lambda_t\right) \tag{7}$$

$$\lambda_{t+1} = \lambda_t + \xi \quad C\left(\lambda_{t+1} | \{\lambda_i\}_0^t\right)) \tag{8}$$

- будем блуждать по пространству параметров;
- необходимо определить:
  1. способ сделать шаг;
  2. условие принятия этого шага;

Случайное блуждание

На что стоит обратить внимание при построении блуждания:

- размерность λ может быть меньше чем кажется;
- ограничения на λ существенно осложняют процедуру:

Некоторые виды случайного блуждания

- множество фиксированных шагов $\xi \sim U\left(\{\xi_i\}_1^m\right)$;
- гауссовское $\xi_i \sim N(\mu, \sigma^2)$;
- самозависимое;
- etc.

Simple hill climbing

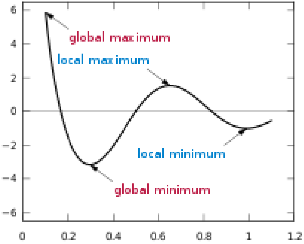
$$\xi \sim U\left(\{\xi_i\}\right), \xi_{ii} = \omega, \xi_{ij} = 0, j \neq i, \tag{9}$$

$$C(\lambda_{t+1}|\lambda_t) = \frac{p(F(\lambda_{t+1})|X)}{p(F(\lambda_t)|X)} > 1 \tag{10}$$

Свойства:

- простой;
- быстро сходится;
- зависим от выбора начальной точки.

Random-restart (shotgun) hill climbing



Проблемы:

- сходится в локальный максимум;
- может долго сходиться, если начало далеко от максимума;
- аллеи.

$\Rightarrow$ Можно рестартить hill climbing из разных начальных точек

Интуиция

- наверное нельзя всегда ходить “по шерсти”;
- хорошо бы обойти все пространство;
- скорость движения должна меняться.

$\Rightarrow$ Markov Chain Monte-Carlo (MCMC)

Metropolis-Hastings алгоритм

$$p(\lambda|\lambda_t) \tag{11}$$

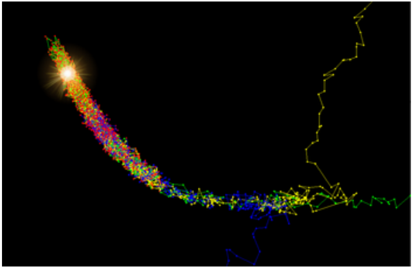
$$\alpha = \frac{p(F(\lambda_t)|X)p(\lambda_t|\lambda)}{p(F(\lambda)|X)p(\lambda|\lambda_t)} \tag{12}$$

$$\psi \sim U(0,1) \tag{13}$$

$$C(\lambda_{t+1}|\lambda_t) = \begin{cases} 1, & \alpha \geq \psi \\ 0 \end{cases} \tag{14}$$

Например $p(\lambda|\lambda_t) \sim N(\lambda_t|\sigma^2 E)$

Свойства



- Доказано:
  - обходит все пространство;
  - это действительно взвешенное сэмплирование;

$\Rightarrow$ точно придем в максимум!

Проблема только с тем, что придем за бесконечное время

Алгоритм Гиббса

В Метрополисе есть проблема: все зависит от $p(\lambda|\lambda_t)$.

$$i \sim U(1, \dots, n) \tag{15}$$

$$p(\xi|\lambda_t) \tag{16}$$

$$\lambda_i = \lambda_{t_i} + \xi, \tag{17}$$

$$\lambda_j = \lambda_{t_j}, j \neq i \tag{18}$$

Как можно построить $p(F|X)$

Если MSE, то все просто:

$$p(\lambda|X) = \frac{e^{-c\|F(\lambda|X)-Y\|_2}}{Z} \tag{19}$$

$$Z = \int_\lambda e^{-c\|F(\lambda|X)-Y\|} d\lambda \tag{20}$$

Если максимизируем, то надеемся задрать Y так, чтобы Z был определен.