

Генетические алгоритмы

И. Куралёнок, Н. Поваров

Яндекс

СПб, 2013

Содержание

- 1 Генетические алгоритмы
- 2 Свойства генетических алгоритмов
- 3 Differential evolution
- 4 No free lunch theorem

Идея

Кажется природа решает ту же задачу.

⇒ Можно подсмотреть механизмы

⇒ Популяция позволяет говорить о "производных"

⇒ Внутри популяции можно исследовать зависимости и использовать их

⇒ Просто объяснять, но нужен специальный "масонский" язык

Отсебятена: генетические алгоритмы скорее язык описания перебора.

Этапы генетического алгоритма

1. Выбрать начальную популяцию
2. Измерить "приспособленность" особей
3. Повторить до сходимости
 1. Выбрать представителей популяции для размножения
 2. Породить новых особей с помощью скрещивания и мутаций
 3. Измерить приспособленность "детей"
 4. Заменить наименее приспособленных "детьми"

Выбор начальной позиции

Как любые переборные методы генетика существенно зависима от начальных позиций.

Можно так:

- Сводят пространство решений в $[0, 1]^n$ и берут равномерно рандомные точки
- Сводят к $0, 1^k$ с помощью бинаризации и берут орты
- Делают предварительное сэмплирование

Выбор особей для размножения

Выбор особей - ключевой момент!

Можно так:

- Выбирать с вероятностью, пропорциональной "приспособленности"
- Предварительно делать shuffle "приспособленности"
- От "приспособленности" зависит очередность
- Можно выбирать равномерно из популяции (по классике нельзя)
- Любые комбинации :)

Скрещивание

Хочется сохранить лучшие черты, однако какие лучшие не ясно. Значит устроим случайный процесс!

- Потомок имеет связанные "куски" ДНК родителей:
 - n-point crossover;
 - cut'n'splice;
 - равномерное скрещивание.
- Родители влияют на направление развития
- Много родителей

Мутация

Использование только скрещивания может привести к вырождению популяции (genetic drift) *Rightharrow* мутация. Мутация очень похожа на фазу выбора новой точки в МСМС/случайном блуждании.

- Элементарные строковые преобразования
- Инвертирование бита
- Случайная мутация компонента

Мутация vs Скрещивание

Мутация - способ "вылезти" из локальных экстремумов, скрещивание - глубину "влезания" в экстремумы. Есть несколько лагерей:

- Скрещивание круче, так как быстро бежит. Мутация нужна только, чтобы убедиться, что других хороших направлений нет.
- Мутация круче, так как обходит всё. Скрещивание только немного ускоряет процесс.

Замещение элементов популяции потомками

- Меняем % "худших" стариков
- Вводим "penalty" за старость и меняем только если потомок лучше
- Старикам здесь не место

Основные pros и cons

Генетику легко сделать, но сложно понять как она работает.

Хорошо:

- очень много вариантов отобразить наше понимание область (любой успех легко объяснить :));
- легко понять непрофессионалу;
- легко программируется;
- в отличии от сэмплирования можно применять методы первого порядка, применяя в качестве градиента разности элементов популяции.

Плохо:

- Когда не работает непонятно за что хвататься;
- Результаты зачастую не повторяются на другом множестве(так как незя понять сложность модели);
- большое поле для псевдонауки.

Алгоритм Differential Evolution

$$\arg \max_{\lambda \in \mathbb{R}^n} F(\lambda)$$

- 1 Выбрать начальную популяцию рандомом
- 2 До сходимости, для каждого элемента популяции $x \in P$:
 - 1 выбрать $a, b, c \in P$ чтобы все отличались;
 - 2 $k \sim U(1..n)$;
 - 3 $y = (y_i)$ для каждого i
 - 1 $r \sim U((0, 1))$
 - 2 $y_i = \begin{cases} a_i + F(b_i - c_i), & i = k | r < C \\ y_i = x_i \end{cases}$
 - 4 Заменить старый элемент новым, если новые лучше.
- 3 Выбрать лучшего представителя получившейся популяции.

Постановка вопроса

Целых 2 занятия мы обходим пространство решений. Может быть есть лучший из всех методов обхода? В качестве аналогии:

Ресторан: способ обхода пространства решений

Блюдо: конкретная задача, требующая оптимизации

Цена: значение целевой функции на обойденной части пространства

Theorem (No free lunch theorem)

Все рестораны в среднем одинаково дороги

NFL: формальная формулировка

$$\begin{aligned}d_m &= \{(d_m^x(1), d_m^y(1)), \dots, (d_m^x(m), d_m^y(m))\} \\f &: \mathcal{X} \rightarrow \mathcal{Y} \\ \mathcal{F} &= \mathcal{Y}^{\mathcal{X}} \\ p(f) &= \frac{1}{\mathcal{F}}\end{aligned}$$

Theorem (David Wolpert and William G. Macready (1997))

Для любых двух способов обхода a_1 и a_2 :

$$\begin{aligned}\sum_f p(d_m^y|f, m, a_1) &= \sum_f p(d_m^y|f, m, a_2) \\ \sum_f p(d_m^y|f_0, M, m, a_1) &= \sum_f p(d_m^y|f_0, M, m, a_2)\end{aligned}$$

NFL: следствия

- Нам всем хватит работы :)
- Полоса белая, полоса черная
- Нужно искать близкие задачи

Домашнее задание

Кто согласен с основным утверждением лекции, что генетика это только альтернативный способ описания перебора:

- Фитим $\ln(x)$
- Датасет старый
- Полином ищем генетикой

Кто не согласен:

- Доводы
- Примеры