Машинное обучение: Генетические алгоритмы

Кураленок И.Е.

Яндекс

30 марта 2012 г.

Содержание

Идея

Кажется природа решает ту же задачу.

- $\Rightarrow$ Можно подсмотреть механизмы.
- $\Rightarrow$ Популяция позволяет говорить о “производных”.
- $\Rightarrow$ Внутри популяции можно исследовать зависимости и использовать их.
- $\Rightarrow$ Просто объяснять, но нужен специальный “массонский” язык.

Отсебятена: генетические алгоритмы, скорее язык описания перебора.

Этапы генетического алгоритма

1. Выбрать начальную популяцию
2. Измерить “преспособленность” особей
3. Повторить до сходимости
   1. Выбрать представителей популяции для размножения
   2. Породить новых особей с помощью скрещивания и мутаций
   3. Измерить преспособленность “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 незя).
- 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;
  - cun’n’splice;
  - равномерное скрещивание.
- Родители влияют на направление развития.
- Много родителей.

Мутация

Использование только скрещивания может привести к вырождению популяции (genetic drift) $\Rightarrow$ мутация. Мутация очень похожа на фазу выбора новой точки в MCMC/случайном блуждании.

- Элементарные строковые преобразования.
- Инвертирование битов.
- 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 все. Скрещивание только немного ускоряет процесс.

Замещение элементов популяции потомками

- Меняем $x\%$ "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 области (любой успех легко объяснить :));
- легко понять непрофессионалу;
- легко программируется;
- в отличии от семплирования 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);
- боольшое поле для псевдонауки.

Алгоритм Differential Evolution

$$\underset{\lambda \in \mathbb{R}^n}{\operatorname{argmax}} F(\lambda)$$

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. $k \sim U(1 \dots n)$
   3. $y = (y_i)$, для каждого i
      1. $r \sim U((0, 1))$
      2. $y_i = \begin{cases} a_i + F(b_i - c_i), i = k \mid r < C \\ y_i = x_i \end{cases}$
   4. заменить старый элемент новым, если новый лучше.
3. Выбрать лучшего представителя получившейся популяции.

No free lunch theorem (NFL)

Нет “золотой пули” среди алгоритмов оптимизации.