Машинное обучение: обзор целевых функций

И. Куралёнок, Н. Поваров

Яндекс

СПб, 2013

Задача на сегодня

Строить варианты целевой функции на заданную тему.

Для этого нам понадобится:

- узнать чем отличается измерение от оптимизации;
- понять какие существуют подходы к построению целевой функции;
- научиться строить целевые функции для заданных примеров (это уже ДЗ).

Пример

Вахтер хочет понять кого пускать в парадную. Он хочет минимизировать свою работу (больше спать) по:

- проверке входящих;
- разборкам с жильцами/руководством;
- уборке/проветриванию.

Для этого ему надо проверять входящих (думать). Однако, минимизировать “время сна” напрямую очень сложно. Наша задача помочь бедному вахтеру.

Суть проблемы

Если мы понимаем чего хотим: $\mathcal{M}(F_0)(X)$ (линейка позволяющая измерить конкретное решение), то задачу оптимизации можно переписать так:

$$\max_{\mathcal{T}} \mathcal{M}\left(\arg\max_{F} \mathcal{T}(F, L)\right)(T)$$

Если выборка не смещена по параметрам оптимизации, то K.O. говорит нам:

$$\mathcal{M} \equiv \arg\max_{\mathcal{T}} \mathcal{M}\left(\arg\max_{F} \mathcal{T}(F, L)\right)(T)$$

Однако, все не так просто.

Про вахтера в новых обозначениях

$\mathcal{M}$ — время сна;

F — способы проверки входящих;

$\mathcal{T}$ — способы оценить проверку входящих.

Например в качестве F может выступать оценка вероятности того, что “клиент” — проблемный. Тогда $\mathcal{T}$ может быть как собственно время сна, так и средняя ошибка предсказания по результату работы функции F.

Проблема в построении

Что может быть “не так” в очевидном решении:

- $\mathcal{M}$ может быть неудобна для оптимизации (кусочно-постоянная, например);
- сложно гарантировать несмещенность по параметрам оптимизации;
- сложно собирать данные в терминах $\mathcal{M}$;

Поэтому все еще актуально решать исходную задачу:

$$\max_{\mathcal{T}} \mathcal{M}\left(\arg\max_{F} \mathcal{T}(F, L)\right)(T)$$

Как можно подойти к построению $\mathcal{T}$

Можно исходить из трех соображений:

$\mathcal{T} \equiv \mathcal{M}$: усреднение $\mathcal{M}$ по всему доступному опыту;

$\arg\max_F \mathcal{T}(F, L) = \arg\max_F \mathcal{M}(F, L)$:

- регрессия по "очкам": введем для каждого наблюдения стоимость, и будем ее приближать по $\mathcal{T}$;
- принцип максимальной энтропии;
- принцип минимального описания;

$\max_{\mathcal{T}} \mathcal{M}\left(\arg\max_F \mathcal{T}(F, L)\right)(T)$: вероятностное моделирование происходящего: как можно получить $\mathcal{M}$ из удобного $\mathcal{T}$.

Средние значения

Будем оценивать каждое наблюдение. Хотим улучшить результат в среднем.

$$F_0 = \arg\max_F \frac{1}{n} \sum_i \mathfrak{m}(F(x_i), y_i)$$

Поделили большую $\mathcal{M}$ на много маленьких $\mathfrak{m}$.

+ по наблюдениям делить естественно;
— надо следить за независимостью наблюдений;
— работает только для ситуаций когда нет ∞ потерь/приобретений;

Средние значения бывают разные

Последнюю проблему можно решить с помощью других средних:

геометрическое : $\sqrt[n]{\prod_i x_i}$;

гармоническое : $\frac{n}{\sum_i \frac{1}{x_i}}$;

Разные только пространства усреднения

$$A(\{x_i\}) = f^{-1}\left(\frac{1}{n}\sum_i f(x_i)\right)$$

В этих терминах все средние отличаются лишь отображением f;

арифметическое : $f(x) = x$;

геометрическое : $f(x) = \log x$;

гармоническое : $f(x) = \frac{1}{x}$;

С точки зрения оптимизации, если функция f монотонная, то мы все можем отбросить и оптимизировать только

$$\max \sum_i f(x_i)$$

Почему это работает

Пусть задана метрика $\mathcal{M}$, хотим получить решение F, которое ее максимизирует. Введем дополнительную интерпретацию наблюдений $s(y_i) \in \mathbb{R}$. Будем предсказывать $s(y_i)$:

$$F_0 = \arg\min_F \sum_i \|F(x) - s(y_i)\|_{l_q})$$

- формирование $s(y)$ — отдельная проблема;
- регрессия известная задача;
- все гладко[1], выпукло и удобно для оптимизации.

[1] для некоторых q :)

Виды невязки l_q

Невязку можно интерпретировать по-разному и в зависимости от интерпретации подбирать q:

$$l_q(x, b) = \|x - b\|_{l_q} = \left(\sum_i \|x_i - b_i\|^q \right)^{\frac{1}{q}}$$

Все чуть менее гладко, чем хотелось бы, но и такие штуки оптимизируются с помощью односторонних градиентов.

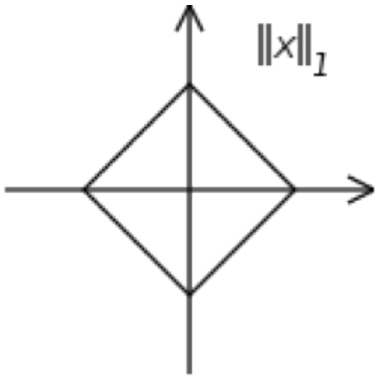
Экстремальные случаи q

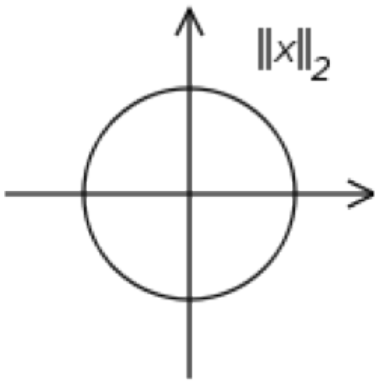
Особенно интересны экстремальные случаи q:

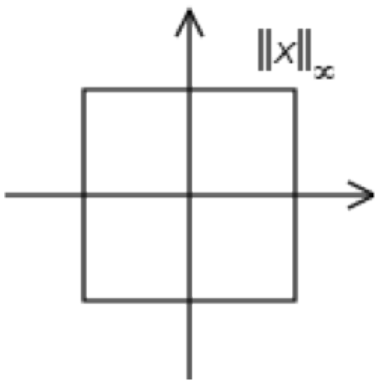
$$l_0(x, b) = \sum_i I\{x = b\}$$
$$l_\infty(x, b) = max|x - b|$$

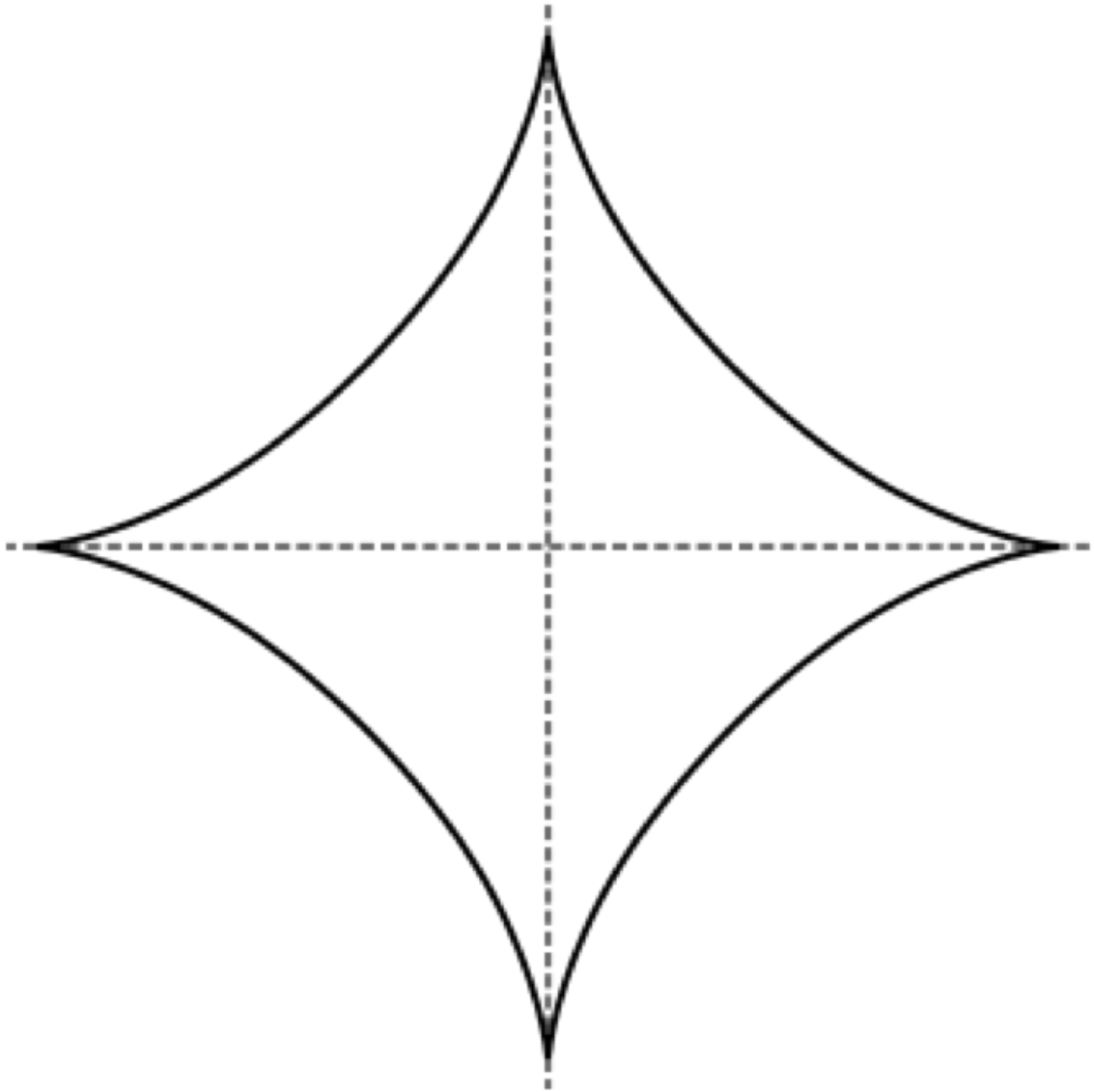
Очень понятный физический смысл, но с гладкостью беда: оптимизация l_0 NP-hard.

Как выглядят разные q









Подбираем “очки”

$$\max_{s} \mathcal{M} \left(\arg\max_{F} \|F(x) - s(y)\|_{l_q} \right) (T)$$

Обычно первая версия “от балды”. Развиваем итеративно :).

Вспоминая вахтера

Назначим очки:

Результат	Очки
Ничего	0
Громко тусовался	10
Накурил	15
Напачкал	100

Это сильно проще сделать “от балды”, чем считать статистику.

Моделирование вахтера

Попробуем объяснить происходящее, зная как оно бывает:

Местные проблемные только если выпьют;

Не местные бывают:

Приличные не будут ничего плохого делать, пока не выпьют;

Неприличные могут нахамить, могут создать проблемы.

Составим из этой картины мира вероятностную модель, и оптимизируем ее.

Оптимизация вероятностной модели

$$\arg\max_F p(F|X)$$

- В вероятностях проще рассуждать о жизни
- Интересно найти параметры, которые наиболее вероятны при наблюдаемых данных
- Обычно непонятно как это распределение построить напрямую

Почему это работает

Мы строим p таким образом, что она отражает наше понимание о структуре области. По сути мы итеративно напрямую оптимизируем

$$\max_{\mathcal{T}} \mathcal{M}\left(\arg\max_{F} \mathcal{T}(F, L)\right)(T)$$

с использованием натурального интеллекта :).

Байесовские методы

$$p(f|X) = \frac{p(X|f)p(f)}{p(X)} \sim p(X|f)p(f)$$

Да, внизу интеграл, мы надеемся, что его можно взять и он не 0

- p(X|f) правдоподобие
- p(f) априорное знание о семействе

$$F(x) = \int_f p(x|f)p(f|X)df$$

Байесовские методы (практика)

1. Задаем априорное распределение параметров (например N(0,1))
2. Вычисляем вероятность X, считая что точки независимы и одинаково распределены

$$p(X|f) = \prod_i p(x_i|f)$$

3. Получаем распределение из которого можно посамплить
4. Усредняем посампленное

Байесовские методы (свойства)

+ Все честно с точностью до входных данных и построенной модели
+ Можно использовать информацию о предыдущем обучении (задавая prior)
+ Можно понять погрешность предсказания (даже если она не выводится аналитически)

– Все сильно зависит от выбора prior
– Сложная решающая функция
– Необходимо эффективное сэмплирование пространства решений

Максимум апостериори

Байес по простому

- Хочется попроще
- Для оценки ошибок есть бутстраппинг
- Ансамбли можно сделать другими способами и включить в решающую функцию

Чтобы не возиться со сложной F, можно просто взять самое вероятное решение:

$$F(x) = f(x) : p(f|X) > p(g|X), \forall g \in F$$

получим **maximum a posteriori**.

Метод максимального правдоподобия

(Байес совсем по простому)

- Лень придумывать prior
- Нет информации о предыдущих экспериментах
- Быстро меняющиеся условия

А можно совсем обнаглеть и убрать еще prior, сказав, что все решения одинаково вероятны:

$$p(f) = p(g) \forall f, g$$

$$\begin{aligned} F &= \arg\max_F p(X|f) = \arg\max_F \prod_i p(x_i|f) \\ &= \arg\max_F \sum_i log(p(x_i|f)) = \arg\max_F LL(X, f) \end{aligned}$$

Заметим, что prior не корректен, в случае $|F| = \infty$.

Веса при LL

$$\begin{aligned} F &= \arg\max_F \prod_i (p(x_i|F))^{n(\frac{w_i}{Z}N)} \\ &= \arg\max_F \prod_i (p(x_i|F))^{w_i} \\ &= \arg\max_F \sum_i w_i \log p(x_i|F) \end{aligned}$$

- Важность точек может быть разной
- Введем «вес» для каждой точки
- Будем выбирать точки случайно, с вероятностью пропорциональной весу

Сходимость ММП

- Идентификация (все функции разные);
- Множество функций компактно;
- Функции непрерывны с вероятностью 1;
- Существует мажорирующая интегрируемая D:

$$|lnF(x)| < D(x)$$

$\Longrightarrow$ при увеличении количества точек L сходится

$$sup\|LL(x,F) - LL(x,F_0)\| \xrightarrow{a.s.} 0$$

Асимптотическая нормальность ММП

- Первые две производные L определены:

$$F = F(x, \lambda)$$
$$i_{j,k} = \mu_X(\frac{\partial^2 L}{\partial \lambda_j \partial \lambda_k})$$

- Матрица I не ноль, непрерывная функция лямбды;
- Выполняется консистентность;
- И всё остальное хорошо:

$$\sqrt{n}(\lambda_{mle} - \lambda_0) \xrightarrow{d} \mathcal{N}(0, I^{-1})$$

Принцип максимальной энтропии I

- Много Больцмана;
- Кажется, что энтропия может сама только увеличиваться;
- Если оставить систему в покое, то может быть она прийдёт к максимуму энтропии;
- Будем считать, что система живёт уже давно;
- Найдём такие параметры системы, которые обеспечивают максимальную энтропию, сохраняя априорно заданные параметры.

Принцип максимальной энтропии II

- Выразим априорные свойства в виде ограничений;
- Найдём распределение обладающее максимальной энтропией;
- Когда хочется своего $p(x|I)$ решение будет другое.

$$\sum_i p(x_i|I) f_k(x_i) = f_k^0, k = 1, \ldots, m$$

$$p(x|I) = \frac{1}{Z} e^{\sum_k \lambda_k f_k(x)}$$

$$Z = \sum_i \exp^{\sum_k \lambda_k f_k(x)}$$

$$f_k^0 = \frac{\partial}{\partial \lambda_k} log Z$$

Почему это работает

$\arg\max_F \mathcal{T}(F, L) = \arg\max_F \mathcal{M}(F, L)$
Максимизацией энтропии мы выпиливаем информацию про выборку, оставляя лишь информацию о генеральной совокупности. Так как мы хорошо смоделировали, надеемся, что минимум по метрике и по энтропии в одной точке.

Принцип наименьшего описания

- Формализация бритвы Оккама;
- Колмогоров/Solomonoff;
- Вводим сложность по Колмогорову;
- Находим оптимальное решение;
- По хорошему вероятность = 1.

$$F_0 = \arg \min_{F: p(X|F) \geq \epsilon} C(F)$$

Почему это работает

Те же рассуждения, что и при ПМЭ.

Сглаживание таргета

Когда целевая функция "плохая":

$$F_0 = \arg\max_F UT(X|F) = \arg\max_F \mu_{x \sim p(F)}(UT(x))$$

Чему следовать выбирая таргет

- Чувство прекрасного;
- Возможность применять математику:
  - Скорость вычисления;
  - Дифференцируемость (градиентные методы).
- Наличие интересных внутренних параметров;
- Возможность проверить осмысленность промежуточных результатов.

NB: Чем больше мы в области, тем больше знания мы перенесем в таргет.

Задание на дом

- Придумать таргеты для некоторых задач;
- Точные задачи приведены в файле howto.txt;
- Дедлайн 18 октября.