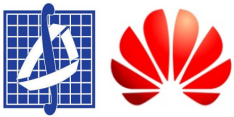
Введение в искусственный интеллект. Машинное обучение

Тема: Эмпирический риск и стохастический градиентный спуск

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

кафедра Математической Теории Интеллектуальных Систем



План

1. Эмпирический риск и его минимизация
2. Разделяющая поверхность
3. GD и SGD

Обучение по прецедентам

- X – множество описаний объектов, Y – множество допустимых ответов
- Неизвестная целевая зависимость: отображение $y^* : X \to Y$
- Конечная обучающая выборка: $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, т.ч. $y_i = y^*(x_i)$
- **Задача обучения по прецедентам** состоит в том, чтобы построить алгоритм $a : X \to Y$, который приближал бы целевую зависимость y^* как на обучающей выборке X^m, так и на всём множестве X
- **Эмпирический риск** – это средняя величина ошибки a на X^m
- **Метод минимизации эмпирического риска** – это общий подход к решению широкого класса задач обучения по прецедентам (задачи классификации и регрессии)

Эмпирический риск – определения

Функция потерь $L(y, y')$

Характеризует величину отклонения ответа $y = a(x)$ от правильного ответа $y' = y^*(x)$ на объекте $x \in X$

Множество алгоритмов $A = \{a : X \to Y\}$

В этом множестве будет вестись поиск отображения, приближающего неизвестную целевую зависимость

Эмпирический риск

Функционал качества, характеризующий среднюю ошибку алгоритма a на выборке X^m: $R(a, X^m) = \frac{1}{m} \sum_{i=1}^{m} L(a(x_i), y^*(x_i))$

Минимизация эмпирического риска

Минимизация эмпирического риска (М.Э.Р.)

В заданном множестве алгоритмов A найти алгоритм, минимизирующий эмпирический риск:

$$a = \arg\min_{a \in A} R(a, X^m)$$

Достоинство М.Э.Р.

Конструктивный и универсальный подход, позволяющий сводить задачу обучения к задачам численной оптимизации

Недостаток М.Э.Р.

Явление переобучения на обучающей выборке X^m, которое возникает практически всегда при использовании метода М.Э.Р., поскольку критерием качества является ошибка **на этой же выборке** (решение: для оценки менять выборку) .

Примеры функций потерь

Задача классификации

- Пороговая функция $L(y, y') = [y \neq y']$
- Функция разрывна $\Rightarrow$ минимизация эмпирического риска – это задача комбинаторной оптимизации $\Rightarrow$ во многих практически важных случаях сводится к поиску максимальной совместной подсистемы в системе неравенств (число неравенств совпадает с число объектов обучения m) и является NP-полной

Задача регрессии

Квадратичная функция потерь $L(y, y') = (y - y')^2$

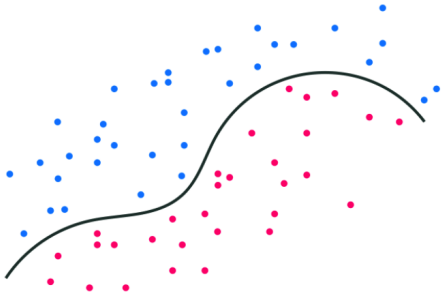
Время для вопросов

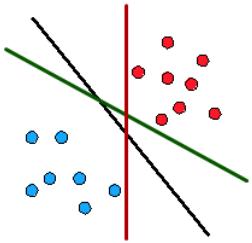
Разделяющая поверхность

- Функцию потерь для задачи классификации $L(y, y') = [y \neq y']$ нельзя продифференцировать (и, значит, эффективно минимизировать)
- Для подключения известного дифференциального аппарата вводятся два понятия — разделяющая поверхность и аппроксимация Э.Р.
- Рассмотрим задачу бинарной классификации: $X \to Y$, $Y = \{+1, -1\}$ на обучающей выборке $X^m = (x_i, y_i)_{i=1}^m$
- Будем алгоритм искать в виде $a(x, w) = \operatorname{sign} g(x, w)$, где $g(x, w)$ - дискриминантная функция, а w - вектор параметров
- $g(x, w) = 0$ - **разделяющая поверхность** (граница между классами); тогда ошибка классификации $a(x_i, w) \neq y_i \Leftrightarrow y_i g(x_i, w) < 0$.

Разделяющая поверхность: вариативность

- Наиболее простой вид разделяющей поверхности — прямая (гиперплоскость)
- Однако разделяющая поверхность может быть нелинейной
- Разделяющая поверхность может быть не одна





Аппроксимация эмпирического риска

- Через понятие разделяющей поверхности можем переопределить ошибку классификации: $a(x_i, w) \neq y_i \Leftrightarrow y_i g(x_i, w) < 0$
- Но нужно еще ввести аппроксимацию самого Э.Р. — $\tilde{R}$ со свойствами:
  1. $\tilde{R}$ — дифференцируема
  2. $\tilde{R}$ — верхняя граница для Э.Р. R (чтобы минимизация $\tilde{R}$ подразумевала и минимизацию R)
- М.Э.Р.: $R(a, X^m) = \frac{1}{m}\sum_{i=1}^{m}[y_i g(x_i, w) < 0] \leq \tilde{R}(a, X^m) = \frac{1}{m}\sum_{i=1}^{m} L(y_i g(x_i, w))$, где новая функция потерь $L(y_i g(x_i, w))$ - невозрастающая и неотрицательная аппроксимация функции $[y_i g(x_i, w) < 0]$, т.ч.: $L(y_i g(x_i, w)) \geq [a(x_i, w) \neq y_i]$

Упражнение. Зачем нужны свойства невозрастаемости и неотрицательности L?
Замечание. В дальнейшем будем предполагать, что мы работаем сразу с аппроксимаций Э.Р. $\tilde{R}$, поэтому знак $\tilde{}$ будем опускать.

Вероятностный смысл минимизации аппроксимированного Э.Р.

Рассмотрим принцип максимизации правдоподобия, или MLE (Maximum Likelihood Estimation).

- Параметрическая модель плотности распределения $p(x, y|w)$
- Максимизация логарифма правдоподобия

$$L(w, X^m) = \ln \prod_{i=1}^{m} p(x_i, y_i|w) = \sum_{i=1}^{m} \ln p(x_i, y_i|w) \to \max_w$$

- Минимизация аппроксимированного Э.Р.

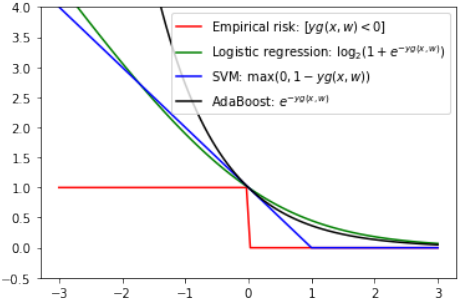
$$R(w, X^m) = \frac{1}{m} \sum_{i=1}^{m} L(y_i g(x_i, w)) \to \min_w$$

Вывод. Эти два принципа эквивалентны при $L(y_i g(x_i, w)) = -\ln p(x_i, y_i|w)$ (коэффициент $\frac{1}{m}$ не влияет на вывод).

Об аппроксимации

- Рассмотрим аппроксимацию функции ошибки на обучающем примере: $L(y_i g(x_i, w)) \geq [y_i g(x_i, w) < 0]$
- В дальнейшем будем рассматривать в основном достаточно **гладкие** (непрерывные и дифференцируемые) функции $L(y_i g(x_i, w))$
- Некоторые аппроксимации способны улучшать **обобщающую способность** классификатора
- Непрерывные аппроксимации позволяют применять известные **численные методы оптимизации** для настройки весов w (например, градиентные методы и методы выпуклого программирования)

Примеры аппроксимации функции $[yg(x, w) < 0]$:



Время для вопросов

Классический градиентный спуск

Задача: минимизировать аппроксимированный Э.Р. (выбор алгоритма осуществляется по w):

$$R(w) = \frac{1}{m}\sum_{i=1}^{m} L(y_i g(x_i, w)) = \frac{1}{m}\sum_{i=1}^{m} L_i(w) \to \min_w$$

Численная оптимизация методом градиентного спуска

- $w^{(0)} :=$ начальное приближение
- $w^{(t+1)} := w^{(t)} - \eta \cdot \nabla R(w^{(t)})$ - итерация алгоритма
- η - градиентный шаг

Проблема: сложно считать в условиях большого количества объектов в обучающей выборке.

Стохастический градиентный спуск

Алгоритм стохастического градиентного спуска

- Инициализация весов w
- Инициализация Э.Р. $R := \frac{1}{m}\sum_{i=1}^{m} L_i(w)$

Итерации

- Выбор объекта $x_i \in X^m$ (например, случайным образом)
- Вычисление ошибки на данном объекте: $\varepsilon_i = L_i(w)$
- Шаг градиентного спуска: $w := w - \eta \cdot \nabla L_i(w)$
- Вычисление сглаженного Э.Р.: $R := (1 - \lambda)R + \lambda\varepsilon_i$

Замечание: параметр сглаживания $\lambda \in [0, 1]$ (можно использовать, например, 0.1).

Вариативность SGD

Инициализация

- $w_j = 0 \quad \forall j = 1, \ldots, n$ (где n - число весов)
- $w_j = rand(-\frac{1}{2n}, \frac{1}{2n})$
- Предобучение на другой обучающей выборке

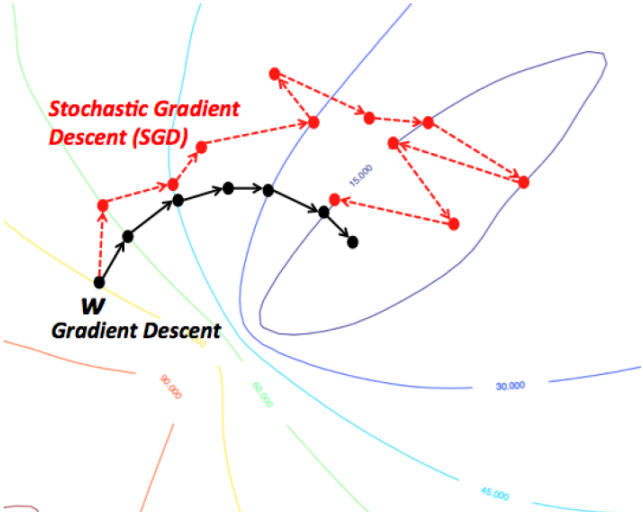
Порядок выбора объектов x_i

- Случайная перетасовка: попеременно брать объекты разных классов
- Чаще брать объекты с большой ошибкой (маленькое значение $y_i g(x_i, w)$)
- Чаще брать объекты с большой неуверенностью (маленькое значение $|y_i g(x_i, w)|$)

Критерий остановки

- Исчерпали лимит по числу шагов
- Значение Э.Р. либо весов перестало меняться

Визуализации градиентных методов



Пакетный (mini-batch) SGD

Пакетный SGD

Идея: на каждом шаге использовать более надежную оценку градиента не на одном примере, а на нескольких

Итерации

- Выбор подмножества объектов мощности $1 < k < m$: $J = \{i_1, \ldots, i_k\}$
- Вычисление ошибки на этих объектах: $L_{i_1}(w^{(t)}), \ldots, L_{i_k}(w^{(t)})$
- Шаг градиентного спуска: $w^{(t+1)} := w^{(t)} - \eta \cdot \frac{1}{k} \sum_{j=1}^{k} \nabla_w L_{i_j}(w^{(t)})$

Выбор шага SGD

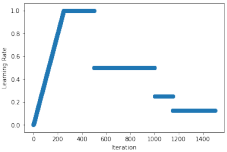
Способы управления градиентным шагом в SGD

- Уменьшать (например, делить на 2×10) каждые N итераций;
- Уменьшать (например, делить на 2×10) каждые N итераций, когда значение Э.Р. перестало существенно меняться за последние K шагов;
- Использование стратегии "разогрева"[1]
- Использование косинусного[2] / линейного[3] закона (вместо дискретных делений) изменения
- Использование цикличности[4]

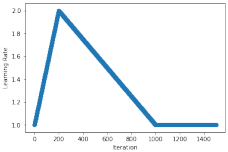
[1] Goyal, Priya, et al. "Accurate, large minibatch sgd: Training imagenet in 1 hour." 2017
[2] Loshchilov, Ilya, and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts." 2016
[3] Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." 2018
[4] Smith, Leslie N. "Cyclical learning rates for training neural networks." 2017

Иллюстрация выбора шага SGD

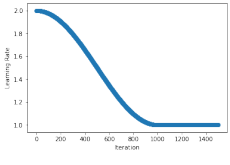
Разогрев



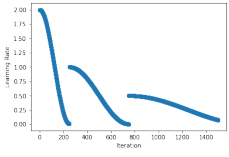
Треугольник



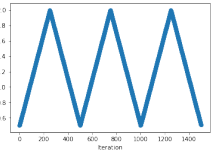
cos



cos + деление



Цикл



О переобучении

Причины переобучения

- Маленькая обучающая выборка;
- Большое число признаков;
- Неинформативные (шумовые/зависимые) признаки.

Проявление переобучения

- Резкое увеличение нормы w (настройка на конкретные признаки);
- Большая разница в ошибке на тестовой и обучающей выборках.

Борьба с переобучением

- Уменьшение норм весов (**регуляризация**);
- Процедура кросс-валидации;
- Ранняя остановка обучения.

MAP

Рассмотрим принцип максимума апостериорной вероятности, или **MAP** (Maximum A Posteriori Probability).
Дано:

- Параметрическая модель плотности распределения $p(x, y|w)$
- Априорная информация о плотности распределения параметров модели $p(w)$
  Например, параметрическое семейство априорных распределений $p(w; h)$, где h — неизвестная и неслучайная величина (гиперпараметр).

Тогда:

- Вероятность по формуле Байеса $p(w|X^m) = \frac{p(X^m|w)p(w;h)}{p(X^m)} \propto p(X^m|w)p(w; h)$
- Максимизируем логарифм

$$-L(w, X^m) = \ln p(w|X^m) = \sum_{i=1}^{m} \ln p(x_i, y_i|w) + \ln p(w; h) \to \max_{w,h}$$

Вероятностный смысл простой регуляризации

$$-L(w, X^m) = \sum_{i=1}^{m} \ln p(x_i, y_i|w) + \ln p(w; h) \to \max_{w,h}$$

Т.о., если вероятностное семейство распределения весов — нормальное, т.е. $p(w; h) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{||w-\mu||^2}{2\sigma^2}}$ с зафиксированными $\mu = 0$ и σ, то

$$\sum_{i=1}^{m} \ln p(x_i, y_i|w) + \ln p(w) \to \max_{w} \Leftrightarrow -\sum_{i=1}^{m} \ln p(x_i, y_i|w) + \frac{\tau}{2}||w||^2 \to \min_{w}$$

L_2-регуляризация и SGD

Рассмотрим введение квадратичного штрафа за увеличение нормы весов в функционал Э.Р.:

$$R_\tau(w, X^m) = R(w, X^m) + \frac{\tau}{2}||w||^2 \to \min_w$$

Тогда градиент Э.Р.: $\nabla R_\tau(w, X^m) = \nabla R(w, X^m) + \tau w$,
А градиентный шаг: $w^{(t+1)} = (1 - \tau\eta)w^{(t)} - \eta\nabla R(w^{(t)}, X^m)$.

Подбор параметра регуляризации τ

- Больше значение τ - больше штрафа за переобучение (но сходимость медленнее!)
- Методом скользящего контроля (cross-validation);

Замечание. В англоязычной литературе зачастую данная техника называется “Weight Decay” (из-за того, что веса линейно уменьшаются каждый шаг), а сам коэффициент WD обычно задается явно (и равен в обозначениях выше $\tau\eta$).

Плюсы и минусы SGD

Плюсы

- Легко реализуется на практике;
- Легко обобщается на любые алгоритмы и функции потерь;
- Возможно онлайн до-обучение (для нового x_i);
- Необязательно использовать все объекты x_i.

Минусы

- На практике возможны расходимость / медленная сходимость;
- Локальные минимумы!!!
- Подбор шага градиента, условия остановки неочевидны.

Takeaway notes

1. Эмпирическим риском измеряем качество классификатора
2. На практике используется аппроксимационный эмпирический риск
3. Градиентный спуск – алгоритм оптимизации первого порядка
4. SGD – практическая версия GD
5. L_2-Регуляризация возникает из-за принципа MAP и изменяет коэффициенты для SGD

Время для вопросов