Практическая работа №2

“Линейные методы”

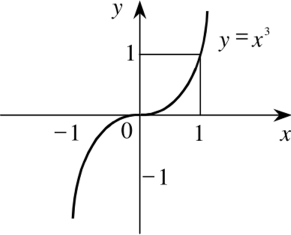
Цель: Изучить алгоритмы линейной регрессии. Научиться обучать алгоритмы линейной регрессии. Освоить основы теории статистического обучения.

Теоретический материал

Линейная регрессия

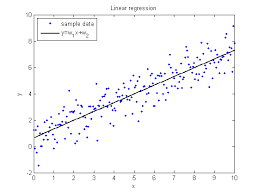
Начнем изучение с основных понятий:

Функциональная зависимость – зависимость при которой каждой зависимой переменной соответствует одна независимая переменная.



Регрессионная зависимость – зависимость где каждой зависимой переменной.

Или если записывать математическое уравнение:



Термин линейная регрессия относится к модели следующей формы:

Когда D = 1 тогда модель принимает форму и называют ее **простой линейной регрессией**:

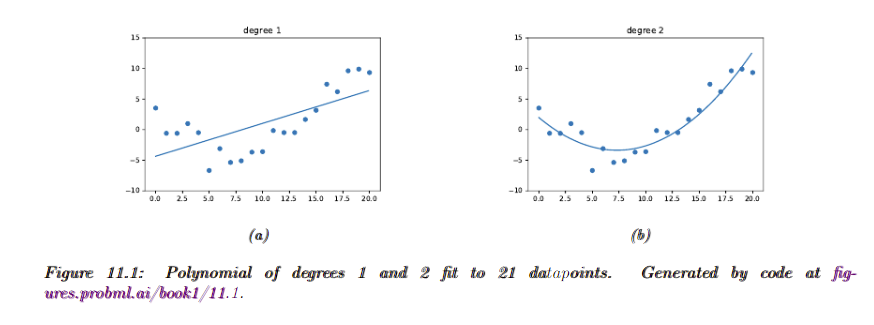
В ином случае **множественной** регрессией.

Когда где J > 1 тогда регрессия называется **многомерной** в ином случае **одномерной**.

Но в общем случае мы ничего не добьемся, применив простую линейную регрессию (мало что можно описать одной линией). Поэтому мы можем применить **нелинейные трансформации к входным данным** (линейные трансформации — это умножение, сложение). Тогда функция примет вид:

Что важно **модель остается линейной** даже при условии того что экстрактор выдает нелинейные выходы.

В качестве простого примера нелинейно трансформации рассмотрим **полиномиальную линейную регрессию.** Которая определяется как:



Запомним этот момент, что существует некая функция извлечения признаков. Это очень часто применяемая фишка (используется для обобщения линейных моделей) в машинном обучении.

Забегая вперед напишу главное различие между машинным обучением и глубоким обучением. В машинном обучении **инженер** выделяет признаки. В глубоком обучении **модель** выделяет признаки.

Модель линейной регрессии решает задачу регрессии (как не странно). Модель описывается как параметрическое семейство функций распределения вероятностей:

Если предположить, что тэта является константным значением (для простоты), то можно вывести формулу максимизирую которую мы получим хорошую модель. Функция максимального правдоподобия становится:

Мы видим, что NLL (Negative log likelihood) пропорционален MSE. Поэтому максимизирую оценку максимального правдоподобия мы будем минимизировать MSE.

.

Наш набор данных будет представлять собой

Определим функцию потерь или ошибки - . Такая функция потерь сильнее уменьшает маленькие ошибки и квадратично увеличивает большие ошибки. Эмпирический риск, когда используем квадратичную ошибку называется MSE (Mean squared error).

Иногда b не пишут учитывая, что w = [b, w1, w2, …, wn], x = [1, x1, x2, …, xn].

w – вектор параметров моделей. Еще о нем можно думать, как о коэффициентах 'важности' признаков.

b – смещение нашей гиперплоскости.

Алгоритм будет уменьшать ошибку эмпирического риска для того что бы алгоритм смогу аппроксимировать эмпирические данные. Но не стоит забывать, что уменьшение эмпирического риска не даёт гарантий на уменьшение ошибки обобщения.

Увеличив количество полиномов (добавив многочленов) в входном векторе мы можем аппроксимировать большее количество функций.

Как вы могли понять увеличение количества полиномов увеличивает шанс переобучения. Это связано с увеличением количества гипотез и соответственное ненужного заучивания данных.

Теперь попробуем аппроксимировать некоторую заданную функцию. К примеру, возьмем функцию f(x) = 4x – 2 + noise.

noise – это некий шум в данных. Предполагается что он появляется из-за различных причин. Например, из-за ошибок измерения, игнорирование других зависимостей. В нашем случае это будет обычный шум Гауса (со средним 0 и стандартным отклонением 1).

Логистическая регрессия

Логистическая регрессия решает задачу классификация. То, что она называется регрессией это просто рудимент прошлого. Модель ее выглядит вот так:

Где:

,