Задание №1.3. К вопросу о линейности регрессии

На первых двух лекциях мы рассмотрели, как построить и обучить линейную модель данных для решения задачи регрессии. В этом упражнении мы познакомимся с тем, как строить и обучать нелинейную модель данных с помощью такого приема как **трансформация признаков** (feature mapping). Мы также увидим ряд ограничений этого подхода, с которыми в будущем нам предстоит разобраться.

Вопрос №1. Моделирование с помощью полиномов 3-й степени [1.2 балла]

Пусть у нас есть обучающая выборка $\{(\mathrm{x}^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})})\}_{i=1}^{m}$, где $\mathrm{x}^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})} \in \mathbb{R}$. Мы бы хотели обучить на этих данных гипотезу, представляющую собой полином 3-й степени: $\mathrm{h}_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$. Ключевое наблюдение здесь состоит в том, что гипотеза $\mathrm{h}_{\theta}(x)$ по-прежнему остается линейной по своим параметрам θ, несмотря на то, что она перестает таковой быть по входным данным x. Это позволяет нам использовать линейную регрессию для решения указанной задачи следующим образом.

Пусть $\phi: \mathbb{R} \to \mathbb{R}^4$ – функция, отображающая входные данные x в 4-х мерный вектор:

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \in \mathbb{R}^4.$$

Для удобства будем вместо $\phi(x)$ использовать вектор $\hat{x} \in \mathbb{R}^4$, где $\hat{x}^{(i)} = \phi(x^{(i)})$. Далее сконструируем новую обучающую выборку $\{(\hat{x}^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})})\}_{i=1}^{m}$, заменив исходные признаки x на новые вектора $\hat{x}$. Мы видим, что подгонка нелинейной гипотезы[1] $\mathrm{h}_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$ к исходной обучающей выборке эквивалентна подгонке линейной гипотезы $\mathrm{h}_{\theta}(\hat{x}) = \theta_0 + \theta_1 \hat{x}_1 + \theta_2 \hat{x}_2 + \theta_3 \hat{x}_3$ к новой выборке, потому что

$$\mathrm{h}_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 = \theta_0 + \theta_1 \phi(x)_1 + \theta_2 \phi(x)_2 + \theta_3 \phi(x)_3 = \theta^T \hat{x}.$$

Иными словами, мы можем использовать линейную регрессию на новой обучающей выборке для нахождения параметров $\theta_0, \dots, \theta_3$.

Пожалуйста, выпишите:

1. функцию потерь $J(\theta)$ линейной регрессии на новом датасете,
2. правило обновления вектора весов для пакетного градиентного спуска, используемого для обучения модели.

Терминология: данный метод в машинном обучении часто называют трансформацией признаков, при котором функция ϕ отображает исходные признаки в новое пространство признаков. Для того чтобы различать старые и новые признаки, будем называть x атрибутами, а $\phi(x)$ – признаками (хотя в разной литературе эти два понятия могут использоваться взаимозаменяемо).

Вопрос №2. Задача на программирование: кубическая регрессия [1.2 балла]

Для этого задания мы будем использовать обучающую выборку из следующих файлов: train.csv, valid.csv и test.csv. Каждый файл содержит два столбца: x и y. В терминологии, описанной выше, x –

[1] В машинном обучении часто используют слово «подгонка» как синоним «обучение», в том смысле, что мы подгоняем модель под данные (fit a model to some dataset – подогнать модель под данные).

это атрибут (в данном случае одномерный), а y – результат (метка[2]). Ваша задача – реализовать линейную регрессию с помощью *нормальных уравнений*, используя трансформацию исходного атрибута в полином третьей степени. В файле featuremap.py приведен стартовый код.

Создайте точечную диаграмму обучающей выборки и отобразите на ней обученную гипотезу в виде плавной кривой.

Примечание: Предположим, что $\hat{X}$ – это матрица трансформированной обучающей выборки. Иногда вы можете столкнуться с проблемой необратимости матрицы $\hat{X}^T\hat{X}$. Для численно стабильной реализации кода всегда используйте решатель numpy.linalg.solve для получения параметров напрямую вместо того, чтобы явно вычислять обратную матрицу, а затем умножать ее на $\hat{X}^T y$.

Совет: Возможно, вы захотите использовать xlim и ylim, чтобы установить разумные ограничения для отображаемых на графике областей.

Вопрос №3. Задача на программирование: регрессия полиномом k-той степени [1.2 балла]

Теперь мы распространим вышеизложенную идею на полиномы k-той степени, взяв $\phi: \mathbb{R} \to \mathbb{R}^{k+1}$ следующего вида

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ \cdots \\ x^k \end{bmatrix} \in \mathbb{R}^{k+1}.$$

Следуйте процедуре, описанной в предыдущем вопросе, и реализуйте алгоритм для k = 3, 5, 10, 20. Постройте график и включите кривые гипотез для каждого значения k, раскрасив их в разные цвета. Добавьте легенду в график, чтобы указать, какой цвет соответствует какому значению k.

Понаблюдайте, как меняется подгонка гипотезы к обучающему набору по мере увеличения k. Кратко прокомментируйте свои наблюдения.

Вопрос №4. Задача на программирование: другие трансформации признаков [1.2 балла]

Возможно, вы заметили, что для корректного моделирования данных требуется относительно высокая степень k, и это связано с тем, что обучающая выборка не может быть адекватно объяснена (т.е. аппроксимирована) полиномами низкой степени. Визуализируя данные, вы, возможно, поняли, что y можно хорошо аппроксимировать синусоидой. Фактически, данные для упражнения сгенерированы случайной функцией $y = \sin(x) + \xi$, где ξ – случайный шум с нормальным распределением. Обновите карту признаков ϕ, включив в него синусоидальное преобразование следующим образом:

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ \cdots \\ x^k \\ \sin(x) \end{bmatrix} \in \mathbb{R}^{k+2}.$$

С помощью обновленной карты признаков обучите модели линейной регрессии для значений k = 0, 1, 2, 3, 5, 10, 20 и отобразите на графике полученные кривые гипотез.

[2] В задачах машинного обучения с учителем значения y иногда называются метками, так как с их помощью размечаются «правильные ответы» на данных.

Сравните полученные модели с предыдущими и кратко прокомментируйте заметные различия.

Вопрос №5. Переобучение сложных моделей на небольших данных [1.2 балла]

В заключительном вопросе мы рассмотрим задачу прогнозирования на небольшом наборе данных с гораздо меньшим количеством примеров, представленном в следующем файле: small.csv.

Мы проанализируем, что происходит, когда количество признаков начинает превышать количество примеров в обучающей выборке. Запустите свой алгоритм на этом небольшом наборе данных, используя следующую карту признаков:

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ \cdots \\ x^k \end{bmatrix} \in \mathbb{R}^{k+1}$$

с $k = 1,2,5,10,20$.

Постройте график полученных кривых гипотез. Понаблюдайте, как меняется подгонка под обучающую выборку по мере увеличения k. Прокомментируйте то, что вы наблюдаете.

Замечание: Явление, которое вы наблюдаете, когда модель начинает очень хорошо предсказывать саму обучающую выборку, но в остальных точках начинает вести себя неадекватно, называется переобучением (overfitting). Интуитивно можно считать, что, когда объем имеющихся у вас данных невелик по сравнению с выразительной способностью модели, это приводит к переобучению.

Грубо говоря, функция гипотезы становится "очень гибкой", и ее можно легко заставить проходить через все примеры обучающей выборки, пусть даже и не очень естественным образом. Модель начинает объяснять шумы в данных, которые объяснять не следовало бы. Это приводит к снижению обобщающей способности модели.