### Слайд 1

Сначала я расскажу про линейную классификацию: то есть как применять линейные модели к задачам классификации.

### Слайд 2

И начну о самом простом виде классификации — бинарной классификации, где ответы принимают значения из множества −1 и +1, то есть всего 2 возможных значения. Чтобы работать с той или иной моделью, нужно уметь отвечать на 3 вопроса: первый — это как мы измеряем качество, как устроен функционал ошибки; второй — как устроено семейство алгоритмов, то есть то множество алгоритмов, из которого мы выбираем наилучший с точки зрения функционала; и третий — это как мы обучаем алгоритм, то есть выбираем лучший из семейства с точки зрения функционала ошибки. Сейчас я расскажу о семействе алгоритмов, а в далее— о том, как измерять их ошибки и как обучать эти алгоритмы.

### Слайд 3

Про линейную регрессию уже был же доклад. Линейные классификаторы устроены очень похоже. В линейной регрессии мы складывали все признаки с весами, при этом вес при j-том признаке обозначали как wj-тое, и после этого суммирования проявляли еще свободный коэффициент w0, который также называется сдвигом. В случае с регрессией нас эта формула полностью устраивала, поскольку она принимала вещественные значения, теперь же алгоритм должен возвращать бинарные значения: −1 или +1. Чтобы добиться этого, можно просто взять знак от этого выражения, именно это и будет видом линейного классификатора.

### Слайд 4

Заметим, что формула не очень однородная, в ней есть свободный коэффициент, чтобы убрать его, давайте просто добавим еще один признак выборки, константный признак, который на каждом объекте принимает значение 1, единичный признак. В этом случае свободный коэффициент уже не нужен, его роль будет выполнять вес при этом константном признаке, и формула приобретает такое значение. Заметим, что по сути такая взвешенная сумма признаков — это скалярное произведение вектора весов на вектор признаков, и значит итоговый вид линейного классификатора — это знак скалярного произведения w на x, этой формулой мы и будем пользоваться дальше.

### Слайд 5

Давайте разберемся, какой геометрический смысл у линейного классификатора. В случае с линейной регрессией мы обсуждали, что это по сути приближение зависимости ответа от признаков с помощью прямой или гиперплоскости. Что же это будет в случае с классификацией? Для этого давайте запишем то, что стоит под функцией знака — скалярное произведение w на x, и приравняем к 0, получим такое уравнение. Давайте нарисуем вектор весов w и будем искать все такие точки x, которые удовлетворяют этому уравнению, то есть все такие точки, для которых скалярное произведение этой точки, этого вектора на w = 0. Равенство нулю скалярного произведения означает, что угол между этими векторами = 90 градусов, то есть что они перпендикулярны. Получается, что точки, удовлетворяющие этому уравнению — это все векторы, ортогональные вектору весов. Из аналитической геометрии известно, что это множество представляет собой плоскость. Получается, что уравнение задает плоскость, более того, с одной стороны от этой плоскости значение скалярного произведения будет больше 0, а с другой стороны от плоскости — меньше 0. Получается, что линейный классификатор проводит гиперплоскость в пространстве признаков, и все объекты, которые с одной стороны, относят к классу +1, а те, которые с другой стороны — к классу −1.

### Слайд 6

В случае с двумя признаками это выглядит как-то так: у нас есть выборка, мы проводим разделяющую прямую, и все объекты, которые с одной стороны, относим к классу −1, все, которые с другой стороны, относим к классу +1.

### Слайд 7

Заметим, что линейные классификаторы вычисляют значение скалярного произведения, которое имеет вещественное значение, а затем берет только знак, отбрасывая часть информации. При этом, наверное, само значение скалярного произведения тоже имеет смысл. Действительно, оказывается, что если мы возьмем модуль этого скалярного произведения и отнормируем его, то есть поделим на норму вектора весов, то это выражение будет равно расстоянию от точки x до гиперплоскости, которая задается вектором нормали, вектором весов w. Получается, что линейный классификатор сначала измеряет расстояние от точки до гиперплоскости со знаком и дальше смотрит лишь на знак, то есть на то, с какой стороны от гиперплоскости лежит эта точка.

### Слайд 8

Так мы приходим к очень важному понятию в линейной классификации — к понятию отступа. Отступом называется выражение вида: скалярное произведение вектора весов на объект, умноженное на истинный ответ на этом объекте, который, напомню, равен +1 и −1. Какой смысл у этого выражения? Давайте обратим внимание: если скалярное произведение имеет положительный знак и истинный ответ равен +1 — это верная классификация и произведение скалярного произведения на истинный ответ будет больше 0. Если скалярное произведение меньше 0, и истинный ответ равен −1, то это тоже будет правильная классификация, и их произведение снова будет больше 0. Если же знак ответа и знак скалярного произведения противоположные, то классификация будет ошибочной и знак отступа будет меньше 0. Получается, что отступ — это некоторая величина, которая характеризует корректность ответа. Если отступ больше 0, то алгоритм дает корректный ответ, если отступ меньше 0 — алгоритм дает некорректный ответ. При этом само абсолютное значение отступа свидетельствует о расстоянии от точки до разделяющей гиперплоскости. Принято считать, что если точка находится рядом с разделяющей гиперплоскостью, то классификация неуверенная, наш алгоритм сомневается, к какому классу относить ее, если же точка находится далеко от разделяющей гиперплоскости, то классификация уверенная, при этом, если алгоритм прав, то он просто уверен в этом, все хорошо, если же алгоритм ошибается, и при этом отступ по модулю очень большой, это означает, что алгоритм очень сильно ошибается в классификации этого объекта, возможно, этот объект является выбросом и никак не вписывается в нашу модель, или же алгоритм не подходит для решения этой задачи.

### Слайд 9

В случае с классификацией возникает вполне естественный подход. У нас ответов конечное число. Соответственно, можем требовать точного совпадения класса, предсказанного алгоритмом A(Xi), и истинного класса Yi. Соответственно, функционал, который мы получаем, — это доля неправильных ответов, доля ошибочных ответов. Он записывается вот так. Это сумма индикаторов того, что предсказанный класс A(Xi) не совпал с истинным классом Yi. И все это усредняется по всей обучающей выборке.

### Слайд 10

Давайте вспомним, что в прошлый раз мы изучали понятие отступа, который позволяет понять, ошибается или нет алгоритм на данном объекте. Отступ на этом объекте задается как произведение истинного ответа Yi на скалярное произведение вектора весов W на вектор признаков Xi. Если отступ меньше нуля, то алгоритм ошибается на данном объекте. Соответственно, наш функционал долю неправильных ответов можно переписать, как среднее значение индикатора того, что отступ на (i) объекте меньше нуля.

### Слайд 11

Давайте посмотрим, как выглядит функция, которая стоит под знаком суммы. Индикатор того, что отступ меньше нуля. По оси "Икс" отложим отступ Mi, отступ на этом объекте, по оси "Игрек" — значение функции потерь, значение индикатора. Мы видим, что эта функция пороговая. Она равна единице, если отступ меньше нуля, и нулю, если отступ больше нуля.

### Слайд 12

Эта функция является разрывной, у нее разрыв в нуле. Из-за этого ее нельзя оптимизировать градиентными методами. Конечно, можно воспользоваться методами негладкой оптимизации, но они довольно сложные в реализации и не дают гарантии сходимости к локальному оптимуму. Поэтому давайте попробуем как-то изменить задачу, чтобы она стала гладкой.

### Слайд 13

Для этого возьмем индикатор того, что отступ меньше нуля, нашу пороговую функцию потерь. Оценим сверху этот индикатор некоторой гладкой функцией "L с волной", которая также зависит от отступа M. То есть это должна быть такая функция, которая больше или равна единице, если отступ отрицательный, и больше или равна нуля, если отступ положительный. Далее, используя данную верхнюю оценку "L с волной", мы можем оценить весь функционал ошибки, весь функционал доли неверных ответов. Верхняя оценка на этот функционал будет выглядеть так: это среднее значение нашей гладкой функции потерь "L с волной" по всей обучающей выборке.

### Слайд 14

Обратите внимание: в этом случае мы будем минимизировать не долю неправильных ответов, а среднее значение нашей гладкой функции потерь "L с волной". При этом мы надеемся, что если мы приведем к нулю данное среднее значение гладкой функции, то при этом прижмется к нулю и то, что она оценивает сверху, прижмется к нулю доля неправильных ответов. Но при этом, конечно же, нет никаких гарантий, что, минимизируя верхнюю оценку, мы будем точно минимизировать и то, что она оценивает, то есть долю неправильных ответов. Но при этом мы получаем очень удобную, хорошую гладкую задачу минимизации.

### Слайд 15

Давайте рассмотрим несколько примеров таких гладких оценок. Например, это может быть логистическая функция потерь L(M), которая записывается как логарифм, под которым стоит единица плюс экспонента от минус отступа. Она используется в логистической регрессии. Другие примеры — это экспоненциальная функция потерь или кусочно-линейная, которые используются в методе опорных векторов.

### Слайд 16

Вот графики этих функций. Видно, что они все, действительно, оценивают сверху пороговую функцию потерь. При этом все они делают это по-разному. Какие-то имеют экспоненциальный рост, какие-то более медленный темп роста при уменьшении отступа.

### Слайд 17

Давайте возьмем для примера логистическую функцию потерь и запишем функционал для нее. Он будет выглядеть вот так. Мы усредняем значение данной функции. Этот функционал будет гладким. Чтобы понять, как его оптимизировать, давайте поставим вместо отступа его определение. То есть Yi истинный ответ, умноженный на скалярное произведение вектора весов на вектор признаков Xi. Видно, что мы получили гладкий, хороший функционал, у которого легко посчитать градиенты по вектору весов W и осуществлять градиентный спуск или пользоваться любым другим вашим любимым методом оптимизации.

### Слайд 18

Итак, что мы делаем при решении задачи классификации, при обучении линейного классификатора? Мы оцениваем сверху долю неправильных ответов, наш базовый функционал ошибки, с помощью некоторой гладкой функции потерь, например, логистической. И далее минимизируем эту гладкую функцию потерь с помощью любого метода оптимизации — стохастического градиентного спуска, градиентного спуска или чего-то еще. И при этом надеемся, что минимизации данного функционала будет также приводить к минимизации доли неправильных ответов. Кстати, обратите внимание: в случае с логистической функцией потерь, даже если все отступы стали больше нуля, все равно алгоритм градиентной оптимизации будет стремиться увеличивать отступы, то есть увеличивать уверенность классификатора в этих ответах. Это довольно хорошее свойство.

### Слайд 19

**Логистическая регрессия позволяет** не только вычислять классификацию для произвольного объекта x, но и оценивать апостериорные вероятности его принадлежности классам:

с помощью сигмоидальной функции