ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Логическая регрессия

Аннотация к занятию: обучающиеся познакомятся с тем, как вычислять вероятность принадлежности тому или иному классу, введут понятие логистической регрессии и запишут её в матричном виде. Обсудят алгоритм линейной классификации в целом, а также запишут формулу для работы алгоритма логистической регрессии. Узнают, как обучать логистические регрессии и, в частности, поговорят о логистической функции потерь.

Цель занятия: сформировать у обучающихся представление о логической регрессии. Познакомить с алгоритмом линейной классификации, с формулой для работы алгоритма логистической регрессии.

Задачи занятия:

- разобрать задачи линейной классификации;
- познакомить обучающихся с логической регрессией;
- записать критерий качества;
- познакомить с логистической функцией потерь;
- узнать, как обучать логистическую регрессию.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	Дорогие друзья, здравствуйте. В новом модуле мы изучим линейную классификацию и поговорим о логистической регрессии. Начнём мы с некоторого замечания о мотивации построения алгоритмов классификации, позволяющих предсказывать вероятность принадлежности классам. Этот вопрос в курсе мы уже обсуждали, но повторить будет полезно. Действительно, в некоторых ситуациях бывает интересно получить в качестве ответа не только метку класса, но и оценку вероятности отнесения объекта к каждому из классов. Например, это бывает полезно при медицинской диагностике. Представьте, что вам нужно определить, есть ли у пациента конкретная болезнь. Это задача классификации на два класса, но гораздо информативнее будет узнать вероятность болезни. При 60% лучше отправить пациента на дополнительные анализы, а при 95% — немедленно начинать лечение. Предсказание вероятности легко превратить в предсказание класса: если вероятность первого класса больше 50%, предсказываем его, если меньше, предсказываем другой класс.	Для справки: Логистическая регрессия — это метод классификации, используемый для прогнозирования значения категориальной зависимой переменной на основе связи с одной или несколькими независимыми переменными, которые, предположительно, имеют логистическое распределение. Если зависимое значение имеет только два возможных значения (успех или неудача), логистическая регрессия будет двоичной

Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	7 мин.	Сегодня мы обсудим, как вычислять вероятность принадлежности тому или иному классу, введём понятие логистической регрессии и запишем её в матричном виде. **Вопрос для обсуждения** Что такое классификация? **Возможные ответы обучающихся** Разделение по определенным характеристикам, признакам. Верно, пока что сконцентрируемся на случае обычной классификации объектов на, скажем, два класса. Рассмотрим задачу предсказания выживших пассажиров «Титаника». В этой задаче нам необходимо по признакам пассажира предсказать, спасся он или нет. В нашем распоряжении — обучающая выборка с размеченными объектами. Обозначим выживших пассажиров классом +1, погибших — −1. Для простоты будем считать, что у каждого объекта есть всего два признака: возраст — действительное число, и пол — метка 0 или 1. Представим нашу выборку в координатах, соответствующих этим признакам. Объекты первого класса на картинке отмечены синим, объекты минус первого класса — красным. Наша цель — разбить плоскость на два множества: синее и красное. **Вопрос для обсуждения** Как вы думаете, как будет работать алгоритм в данном примере?	Подробнее: https://ml-handbook.ru/chapters/model_evaluation/intro

		Возможные ответы обучающихся При попадании объекта в синее множество алгоритм решит, что объект относится к первому классу (то есть пассажир выжил), при попадании в красное — к минус первому (то есть пассажир погиб). Отлично. Верно.	
Изучение нового материала	55 мин.	**Вопрос для обсуждения** Какой самый простой способ разделить плоскость на две такие части? **Возможные ответы обучающихся** Прямая. Естественно, это прямая. Посмотрим на какую-нибудь разделяющую прямую с уравнением $Ax_1+Bx_2+C = 0$. Вот она на графике. Здесь x_1 и x_2 — это названия осей абсцисс и ординат. Для всех точек из синей полуплоскости выражение Ax_1+Bx_2+C принимает значения одного знака. Нам удобно будет считать, что там все значения будут именно положительными. Соответственно, в красной полуплоскости Ax_1+Bx_2+C всегда будет отрицательным. Тогда алгоритм y с крышечкой от (x_1, x_2) можно записать очень коротко: $a(x_1,x_2) = \mathrm{sign}(Ax_1 + Bx_2 + C)$ — просто знак соответствующей линейной комбинации признаков x_1 и x_2.	Для справки:

		Обратите внимание, что расстояния от точки (x_1, x_2) до нашей прямой вычисляется по формуле модуль $Ax_1 + By_1 + C$ делить на корень из суммы квадратов A и B. Чем больше расстояние до разделяющей прямой, тем больше мы можем быть уверены в результате работы алгоритма. Таким образом, величину $Ax_1 + Bx_2 + C$, то есть числитель нашей дроби, можно воспринимать как степень уверенности алгоритма в классификации точки (x_1, y_1). В самом деле, знаменатель дроби для фиксированной прямой не будет зависеть от конкретной точки x_1, x_2. Итак, если эта степень уверенности большая положительная, то объект почти наверняка относится к классу +1, а если большая по модулю отрицательная, то, скорее всего, к классу −1. Наконец, если дробь чуть больше или чуть меньше нуля, то алгоритму не хватает данных для принятия уверенного решения. Как вы понимаете, единственное, что нам осталось сделать, — это правильно подобрать коэффициенты A, B и C. **Вопрос для обсуждения** Что же значит «правильно»? Когда мы говорили про линейную регрессию, нам на помощь пришёл метод наименьших квадратов или, иначе говоря, метод минимизации эмпирического риска с квадратичной функцией потерь — MSE. В этом модуле мы проведём для задачи классификации похожее рассуждение, в котором, тем не менее, будут серьёзные отличия.	https://habr.com/ru/post/173821/

		Сначала поставим эту задачу в многомерном случае, когда признаков у объектов может быть произвольное количество. Каждый объект описывается вектором $x = (x_1, x_2, \ldots, x_n)$ из n компонент. Чтобы не возиться со свободным членом, мы сразу добавим ко всем объектам фиктивный признак, тождественно равный 1. И у нас снова имеется вектор весов $k = (k_0, k_1, \ldots, k_n)$, здесь k_0 выступает в роли свободного члена. Решающее правило для классификации объекта выглядит так: $a(x) = \mathrm{sign}$(скалярного произведения векторов x,k), то есть единица, если соответствующая сумма больше или равна нулю, и минус единица, если она меньше нуля. Мы уже упоминали, что часто результат классификации сам по себе бесполезен, а значение имеет вероятность принадлежности объекта к одному из классов, определяемая алгоритмом. Логично было бы при написании функции потерь тем или иным образом использовать эту самую вероятность. Проблема заключается в том, что единственная мера принадлежности к классу 1, которая есть в нашем распоряжении, — это скалярное произведение x,k. Эта величина может принимать сколь угодно большие и сколь угодно малые значения и, следовательно, на роль вероятности никак не подходит. Ну что же, попытаемся решить эту проблему. Если сама величина скалярного произведения x,k не подходит на роль вероятности класса 1, то, может быть, удастся как-то преобразовать эту величину, чтобы получить число от 0 до 1? То есть мне нужна какая-нибудь функция f(z)	Для справки: В задачах машинного обучения и искусственного интеллекта важную роль играют вероятностные соображения. Вероятность объясняет меру изменения любого конкретного события или результата, который должен произойти, и используется для увеличения шансов наступления любого конкретного результата.

		от этого выражения, которая преобразует произвольное действительное число в число на отрезке от нуля до единицы. Эта функция будет отображает меру принадлежности классу 1 в вероятность принадлежности классу 1. **Вопрос для обсуждения** Давайте тогда подумаем, какими свойствами должна обладать f(z)? ● Во-первых, f(z) должна монотонно возрастать. В самом деле, чем дальше точка находится от прямой «вглубь» класса 1, тем больше должна быть вероятность принадлежности к классу 1. ● Во-вторых, f(0) должно быть равно ½, ведь в случае, если точка лежит на прямой, классы 1 и −1 должны быть равноправны, то есть вероятность должна равняться ½. ● Наконец, при стремлении аргумента к бесконечности вероятность должна стремиться к 1, а при стремлении к минус бесконечности — к нулю. Это нужно для того, чтобы все промежуточные значения вероятности между 0 и 1 были достижимы. Конечно, есть множество функций, которые удовлетворяют этим условиям. Одна из них — это функция сигмоида, которая вычисляется по формуле \sigma(z) = 1 делить на {1 + e^{-z}}. Вот, кстати, её график. Несложно убедиться, что она подходит под все условия, которые мы с вами перечислили: в нуле она равна ½ (проверьте это по формуле), на плюс и минус бесконечности стремится к единице и к нулю. Ну и, конечно, она монотонна.	Для справки: Сигмоида — это гладкая монотонная возрастающая нелинейная функция, имеющая форму буквы «S», которая часто применяется для «сглаживания» значений некоторой величины.

		Замечательно! **Вопрос для обсуждения** Как мы сможем вычислять вероятность принадлежности к классу? **Возможные ответы учеников** Для этого нужно всего лишь применить функцию сигмоиды к отступу, то есть к нашему скалярному произведению вектора признаков на вектор весов. Это даст нам вероятность принадлежности к классу 1. Чтобы вычислить вероятность принадлежности к классу −1, необходимо вычесть это значение из единицы. Алгоритм линейной классификации, при котором вероятность принадлежности к классам вычисляется по такой формуле, называется логистическая регрессия. Что ж, это всё замечательно, но пока не даёт нам ответа на вопрос, как находить оптимальные коэффициенты k и b. Это мы сделаем на следующем уроке, а пока что на примере «Титаника» покажем, как вычислять вероятность принадлежности к классам для логистической регрессии. Обратите внимание, что k_1, скорее всего, должно быть положительно, поскольку женщины выживают с большей вероятностью, чем мужчины, а k_2 должно быть отрицательно, так как молодые выживают чаще пожилых. Мы вычисляем для каждого объекта значение	Для справки: https://habr.com/ru/company/io/blog/265007/

		$k_0 + k_1x_1 + k_2x_2$, получаем на каждом объекте значение отступа. Если значение отступа положительно, то алгоритм предсказывает, что пассажир выживет. Теперь мы применяем к значениям отступов функцию сигмоиды и получаем значение вероятности выживания, то есть вероятность того, что у равно 1. Обратите внимание, вероятность больше ½ — в том и только в том случае, если отступ положительный. Теперь давайте запишем алгоритм логистической регрессии в общем матричном виде. Обозначим через X нашу матрицу объекты-признаки: она состоит из ell строк (по количеству объектов в выборке) и n столбцов (по количеству признаков). Как обычно, элементы матрицы X нумеруются двумя индексами: верхним и нижним, верхний индекс — это номер объекта, нижний индекс — это номер признака. У нас есть вектор весов k, состоящий из n элементов, — по количеству признаков, а также есть вектор ответов y, состоящий из ell элементов, — по количеству объектов. Вообще-то было бы правильно нумеровать игреки верхними индексами, но в этом случае неоднозначности нет, поэтому мы будем для простоты нумеровать нижними. Тогда в матричном виде алгоритм логистической регрессии записывается следующим образом: вектор ответов алгоритма y с крышечкой равен вектору из сигмоид, применённых к скалярным произведениям x^1	

		с k, x^2 с k, и так далее, x^ℓ с k. Наконец, обратим внимание, что вектор из таких скалярных произведений — это просто матрица X умножить матрично на вектор-столбец k, по аналогии с тем, что мы имели для линейной регрессии. Получается, что y с крышечкой вычисляется как сигмоида от произведения матрицы X на вектор k (здесь, конечно, подразумевается, что сигмоида, применённая к вектору, применяется к каждой компоненте вектора по отдельности). Ура, мы записали логистическую регрессию в матричном виде. Такая матричная запись, как и в случае линейной регрессии, значительно облегчает реализацию этих алгоритмов, поскольку, как правило, все вычислительные движки, вроде библиотеки NumPy, используют реализованное матричное произведение. Теперь наша цель — обучить алгоритм логистической регрессии, то есть подобрать такое значение вектора k, при котором алгоритм будет как можно более правильно классифицировать объекты обучающей выборки. Перед вами на слайде таблица из трёх колонок. Первая колонка — y_true — это истинные значения целевой переменной. Вторая колонка — это вероятность класса −1, предсказанная алгоритмом, третья колонка — вероятность класса 1, предсказанная алгоритмом. Скажем, для первого объекта мы видим, что данный пассажир выжил (поскольку y_true равно +1), при этом алгоритм предсказывает ему выживание с вероятностью 0,231, что нам не очень нравится, хочется вероятность выше. Для третьего объекта нас всё	Подробнее: https://vc.ru/dev/72964-obuchenie-logisticheskoy-regressii

		устраивает: правильный ответ −1 и вероятность класса −1 высокая: 0,786. Сформулируем это соображение в общем виде. Если у нас есть объект класса i (здесь i равно либо 1, либо −1), то наш алгоритм тем лучше работает на данном объекте, чем больше вероятность того, что y = i. Соответствующее значение мы получаем из таблицы на слайде. Соответственно, наша цель — подобрать коэффициенты k, для которых выделенная вероятность будет максимально высокой. Понятное дело, что для одного объекта мы бы смогли подобрать веса так, чтобы нужная вероятность была высокой. Проблема в том, чтобы сделать это для каждого объекта. В общем случае мы не сможем подобрать коэффициенты так, чтобы правильная вероятность была высокой на всех объектах. Давайте тогда вместо этого будем пытаться максимизировать произведение всех этих вероятностей. Можно было бы, конечно, максимизировать не произведение, а, скажем, сумму вероятностей, но это будет работать хуже по причинам из математической статистики, которые мы оставим за рамками нашего с вами курса. Итак, наша цель — максимизировать произведение выделенных красным вероятностей, которое мы и записали в строчку. Наконец, последнее действие, которое мы делаем с нашим выражением — это взятие логарифма от этого произведения. Поскольку логарифм, как вы помните, монотонная функция, то максимизация произведения	

		будет равносильна максимизации логарифма произведения. Логарифм, кстати, можно брать по любому основанию, большему единицы, но мы будем брать натуральный логарифм, то есть по основанию е. **Вопрос для обсуждения** Что такое логарифм произведения? **Возможные ответы учеников** Логарифм произведения — это, конечно, сумма логарифмов. Запишем это. Теперь мы видим, что все эти логарифмы на самом деле будут отрицательными: в самом деле, это логарифмы от чисел меньше единицы (ведь это вероятности!). Наконец, последнее, что мы делаем, это домножаем это выражение на минус единицу и делим на количество слагаемых — в данном случае, на 4. Получается выражение, которое мы обозначаем буквой L. Сумму логарифмов мы хотели максимизировать, поэтому L мы должны минимизировать. Вот эту самую величину L и называют логистической функцией потерь для данной выборки. Поскольку минимизация среднего равносильна минимизации суммы, мы часто будем опускать деление на количество объектов. В начале запишем критерий качества работы алгоритма логистической регрессии, а затем перейдём к логистической функции потерь.	Для справки: Функция потерь — функция, которая в теории статистических решений характеризует потери при неправильном принятии решений на основе наблюдаемых данных.

Перед вами схема получения вероятности класса +1 по объекту x. Отметим, что вероятность класса −1 равна единице минус $P_k(x)$.

Наша цель — обучить алгоритм логистической регрессии, то есть подобрать такое значение вектора k, при котором алгоритм будет как можно правильнее классифицировать объекты обучающей выборки. Перед вами таблица из трёх колонок.

- Первая колонка — y_true — это истинные значения целевой переменной.
- Вторая колонка — это вероятность класса −1, предсказанная алгоритмом.
- Третья колонка — вероятность класса 1, предсказанная алгоритмом.

Скажем, для первого объекта мы видим, что данный пассажир выжил (поскольку y_true равно +1). При этом алгоритм предсказывает ему выживание с вероятностью 0,231. Хочется вероятность повыше. Значения для третьего объекта нас полностью устраивают: правильный ответ — −1, и вероятность класса −1 высокая: 0,786.

Вопрос для обсуждения
Кто сможет сформулировать мысль в общем виде?

Возможные ответы учеников
Если у нас есть объект класса i (здесь i равно либо 1, либо −1), то наш алгоритм тем лучше работает на данном объекте, чем больше вероятность того, что y = i.

		Соответствующее значение мы получаем из таблицы на слайде. Соответственно, наша цель — подобрать коэффициенты k, для которых выделенная вероятность будет максимально высокой. Для одного объекта мы бы быстро подобрали веса так, чтобы нужная вероятность повысилась. Проблема в том, чтобы сделать это для каждого объекта. В общем случае мы не сможем подобрать коэффициенты так, чтобы правильная вероятность была высокой на всех объектах. Вместо этого попытаемся максимизировать произведение всех этих вероятностей. Конечно, можно было бы максимизировать не произведение, а, скажем, сумму вероятностей, но это бы сработало хуже по причинам из математической статистики — мы оставим их за рамками курса. Итак, нужно максимизировать произведение выделенных красным вероятностей, которое мы и записали в строчку. Последнее, что мы сделаем с нашим выражением, — возьмём логарифм от этого произведения. Поскольку логарифм — монотонная функция, то максимизация произведения будет равносильна максимизации логарифма произведения. Логарифм, кстати, можно	

брать по любому основанию больше единицы, но мы будем брать натуральный логарифм, то есть по основанию e.

Логарифм произведения — это сумма логарифмов, что мы и запишем. Теперь видно, что все эти логарифмы на самом деле будут отрицательными: это логарифмы от чисел меньше единицы (ведь это вероятности!).

Домножим выражение на минус единицу и разделим на количество слагаемых — в данном случае, на 4. Получается выражение, которое мы обозначим буквой L. Сумму логарифмов мы хотели максимизировать, поэтому L мы должны минимизировать.

Величину L и называют логистической функцией потерь для данной выборки. Поскольку минимизация среднего равносильна минимизации суммы, мы часто будем опускать деление на количество объектов.

Итак, давайте ещё раз зафиксируем и запишем эту функцию потерь в общем виде.
Для одного объекта с номером i мы максимизируем $P_k(x)$, если y^i равно 1, и минимизируем, если y^i равно 0.

Для всей выборки мы можем записать произведение всех вероятностей $P_k(x^i)$ по всем объектам первого класса, а также произведение $(1-P_k(x^i))$ по всем объектам минус первого класса. Большая буква П в формуле обозначает произведение всех сомножителей, указанных после этой буквы — это обозначение,

		аналогичное обозначению для суммы. И мы это выражение максимизируем. Мы максимизируем его по k, потому что при обучении мы подбираем именно параметры алгоритма. Обучающая выборка, то есть иксы и игреки, у нас фиксированы. Наконец, логарифмируем и добавляем знак минус. Теперь мы хотим минимизировать минус сумму логарифмов вероятностей верных классов. Наша цель при обучении логистической регрессии — подобрать такое значение весовых коэффициентов k, чтобы минимизировать значение функции потерь, указанное на слайде. Казалось бы, это всё, что нам нужно: задача оптимизации записана, теперь можно искать оптимальные параметры. Но мы пойдём дальше и преобразуем функцию потерь в более удобный вид. Сначала распишем нужный нам минус логарифм вероятности для объекта класса 1. Здесь у нас есть логарифм, применённый к значению сигмоиды от скалярного произведения x и k. Логарифм 1 делить на выражение — это минус логарифм этого выражения, минусы сокращаются, в итоге остаётся просто логарифм от 1 плюс такая вот экспонента. Распишем то же самое для объектов класса −1. Здесь вместо логарифма вероятности первого класса будет логарифм от единицы минус вероятность первого класса. Давайте приведём слагаемые к общему знаменателю в выражении в скобках. Под логарифмом останется частное двух таких выражений.	

		Избавимся от экспоненты в числителе, домножив всё на e в степени скалярное произведение x и k. В итоге получим минус логарифм от дроби 1 делить на 1 плюс экспонента, то есть просто логарифм от 1 + экспонента от скалярного произведения. **Вопрос для обсуждения** Сравните эти два выражения для классов +1 и −1. Чем они отличаются? **Возможные ответы учеников** Они очень похожи, но не совпадают и отличаются минусом в показателе экспоненты. Тогда можно записать общий вид этого выражения: логарифм от 1 + экспонента от минус игрек умножить на скалярное произведение x на k. Заметьте, что если игрек равен 1, то получается ровно функция потерь для случая y = 1, в случае если y = −1, тоже получается нужная функция потерь. Тогда итоговая задача оптимизации может быть переписана так: L(k) равно сумма по i от 1 до ell логарифм от один плюс e в степени минус игрек i-ое на скалярное произведение x^i и k. Уже эту сумму нам необходимо минимизировать по k. Мы справились с этой математикой. Давайте подведём итоги. • Логистическая регрессия — линейный алгоритм классификации. • Логистическая регрессия позволяет вычислять вероятность принадлежности к классам.	

		• Решающая способность логрегрессии ограничена, как и любого линейного алгоритма. • Параметры логистической регрессии — вектор весов k. Обучение логистической регрессии • Подбираем вектор параметров k, решая задачу оптимизации логистической функции потерь: $L(k) = \sum ln(1 + e^{-yi\langle xi,k\rangle}) \rightarrow min\ k$ • Решаем задачу оптимизации с помощью градиентного спуска	
Закрепление изученного материала	10 мин.	**Вопросы для обсуждения:** • Что такое двумерная линейная классификация? • Как вычислять вероятности классов? • Что такое логистическая регрессия? • Расскажите о логистической функции потерь	Педагог организует беседу по вопросам
Этап подведения итогов занятия (рефлексия)	8 мин.	**Вопросы для обсуждения:** • Чему я научился? • С какими трудностями я столкнулся? • Каких знаний мне не хватает для более глубокого понимания изученного материала? • Достиг ли я поставленных целей и задач?	Педагог способствует размышлению обучающихся над вопросами

Информация о домашнем задании, инструктаж по его применению	5 мин.	Дома повторите основные определения, просмотрите математические действия, которые мы с вами сегодня разбирали.	

Рекомендуемые ресурсы для дополнительного изучения:

1. Как легко понять логистическую регрессию. [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/io/blog/265007/.
2. Оценка качества моделей. [Электронный ресурс] – Режим доступа: https://ml-handbook.ru/chapters/model_evaluation/intro.
3. Функция потерь для логистической регрессии. [Электронный ресурс] – Режим доступа: https://russianblogs.com/article/9935657698/.
4. Линейный классификатор. [Электронный ресурс] – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Линейный_классификатор.
5. Обучение логистической регрессии [Электронный ресурс] – Режим доступа: https://vc.ru/dev/72964-obuchenie-logisticheskoy-regressii.