Системы искусственного интеллекта

Линейная регрессия со множеством переменных. Классификация. Логистическая регрессия

Кулагин Максим Алексеевич
Кафедра
«Управление и защита информации»

2019, Максим Кулагин
e-mail: maksimkulagin06@yandex.ru

План лекции

- ✔ Линейная регрессия со множеством переменных
- ✔ Метод градиентного спуска для нескольких переменных. Масштабирование признаков. Выбор скорости обучения
- ✔ Полиномиальная регрессия
- ✔ Нормальные уравнения
- ✔ Классификация. Логистическая регрессия
- ✔ Граница решения
- ✔ Стоимостная функция для логистической регрессии
- ✔ Многоклассовая классификация на основе логистической регрессии. Подходы «один против всех» и «один против одного»

Линейная регрессия с одной переменной

✔ Тренировочное множество данных (скажем, всего m)

Площадь (фут^2) – x	**Цена в 1000-х ($) – y**
2104	460
1416	232
1534	315
852	178
…	…

Обозначения: **m** = число обучающих примеров
x = «входная» переменная / свойства
y = «выходная» переменная / «метка»
($\mathbf{x}^{(i)}$, $\mathbf{y}^{(i)}$) = i-й обучающий пример (строка)

Линейная регрессия с одной переменной

✔ Тренировочное множество данных (скажем, всего m)

Площадь (фут^2) – x	Цена в 1000-х ($) – y
2104	460
1416	232
1534	315
852	178
…	…

Обозначения: **m** = число тренировочных примеров
x = «
y = «
($x^{(i)}$, $y^{(i)}$) = i-й тренировочный пример

ая / свойства
ная / «метка»

Гипотеза h выглядит так: $h_Q(x) = Q_0 + Q_1 x$

Линейная регрессия со множеством переменных

✔ Тренировочное множество данных (скажем, всего m)

Площадь (фут^2), x_1	Число комнат, x_2	Число этажей, x_3	Возраст дома (год), x_4	Цена в 1000-х ($), y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Обозначения: **n** = число свойств/признаков/дескрипторов
$\mathbf{x}^{(i)}$ = «вход»/свойства i-го тренировочного примера ($\mathbf{x}^{(i)}$, $\mathbf{y}^{(i)}$)
$\mathbf{x}_j^{(i)}$ = j-е свойство i-го тренировочного примера ($\mathbf{x}^{(i)}$, $\mathbf{y}^{(i)}$)
$\mathbf{y}^{(i)}$ = «выходная» переменная / «метка» i-го тренировочного примера ($\mathbf{x}^{(i)}$, $\mathbf{y}^{(i)}$)

Линейная регрессия со множеством переменных

✔ Тренировочное множество данных (скажем, всего m)

Площадь (фут^2), x_1	Число комнат, x_2	Число этажей, x_3	Возраст дома (год), x_4	Цена в 1000-х ($), y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Обозначения: **n** = число свойств/признаков/дескрипторов

$(x^{(i)}, y^{(i)})$

$y^{(i)})$

ного

примера $(x^{(i)}, y^{(i)})$

Гипотеза h выглядит так:

$h_Q(x) = Q^T x = Q_0 + Q_1 x_1 + Q_2 x_2 + Q_3 x_3 + Q_4 x_4$, здесь

Q и x векторы-столбцы размерности n + 1

Градиентный спуск для линейной регрессии со множеством переменных

repeat until convergence
{
$$Q_j = Q_j - \alpha \frac{\partial}{\partial Q_j} J(Q); \qquad (j = 0, \ldots, n)$$
}

✔ Вычислив производные получим

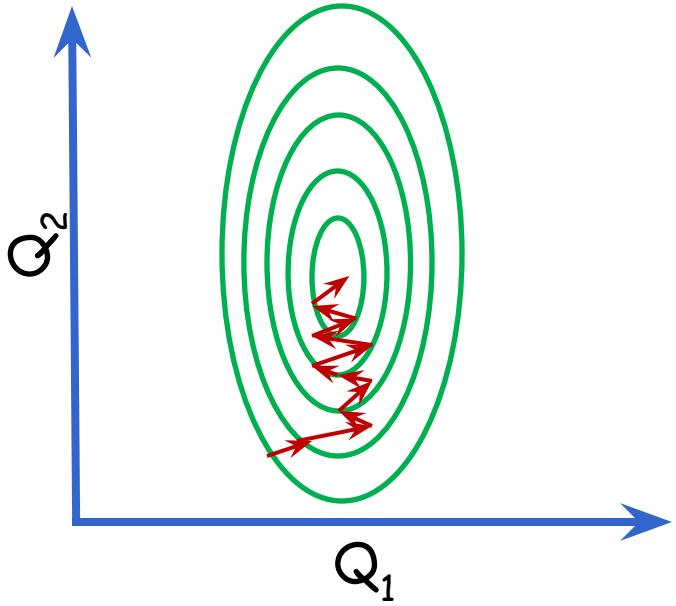
repeat until convergence
{
$$Q_j = Q_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_Q(x^{(i)}) - y^{(i)})\, x_j^{(i)};$$
}

параметры Q обновляются одновременно

$$h_Q(x) = Q^T x = Q_0 + Q_1 x_1 + Q_2 x_2 + \ldots + Q_n x_n$$

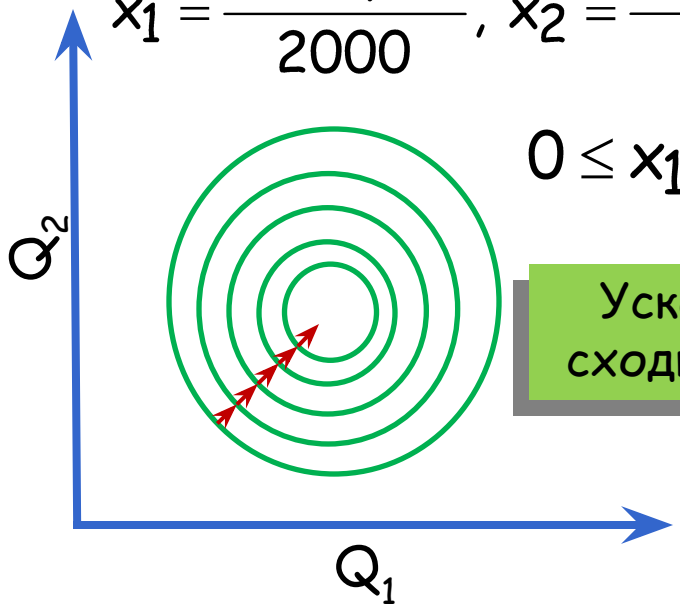
Градиентный спуск на практике!

- ✔ Масштабирование признаков
  - ✔ Идея: привести все свойства к одному и тому же масштабу
  - ✔ Пример. Пусть x_1 – площадь (0-2000 фут^2), x_2 – число комнат (1-5)



$$x_1 = \frac{\text{площадь}}{2000},\ x_2 = \frac{\#\,\text{комнат}}{5},$$

$$0 \le x_1, x_2 \le 1$$



Ускоряем сходимость!

- Нормализация на математическое ожидание
  - Идея: замена x_j на, $x_j - \mu_j$ с целью создания у свойств нулевого среднего

- Нормализация на математическое ожидание и масштабирование свойств приводят к следующей замене:

$$x_j \leftarrow \frac{x_j - \mu_j}{S_j}$$

Обычно в качестве S_j выбирается либо величина среднеквадратического отклонения свойства, либо разница между max и min значениями свойства на тренировочном множестве

Нормализация на мат. ожидание и масштабирование не применяются к свойству x_0!

При масштабировании и нормализации свойств на этапе обучения, требуется выполнять аналогичные операции на этапе предсказания для нового входа x!

Градиентный спуск на практике!

- Отладка. Как убедиться в том, что градиентный спуск работает корректно?
  - J(Q) должна уменьшаться после каждой итерации!

- Как выбрать скорость обучения α?
  - Если α маленькое, то градиентный спуск может быть медленным
  - Если α большое, то градиентный спуск может проскочить минимум. Алгоритм может не сходиться или даже расходиться

Полиномиальная регрессия

- Предскажем цену на дом с использованием следующей гипотезы:

$$h_Q(x) = Q^T x = Q_0 + Q_1 \text{ (длина дома)} + Q_2 \text{ (ширина дома)}$$

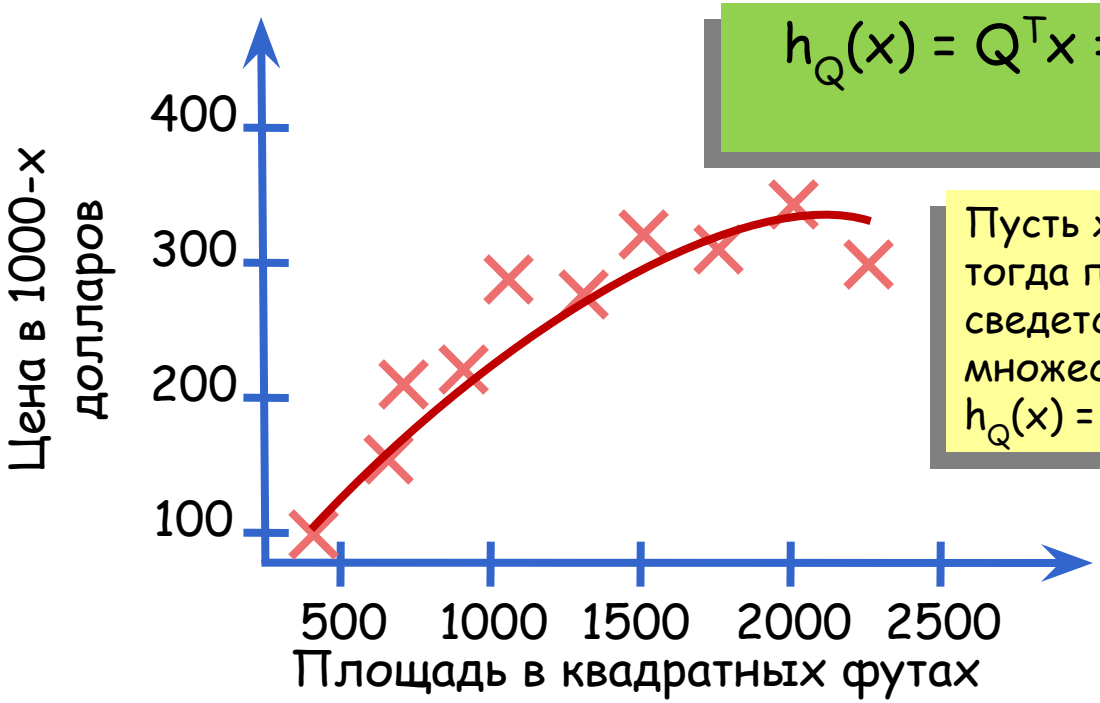
- На основе свойств «длина дома» и «ширина дома», можно построить новое свойство «площадь дома» = «длина дома» * «ширина дома» и предсказывать цену так:

$$h_Q(x) = Q^T x = Q_0 + Q_1 \text{ (площадь дома)}$$

Иногда за счет введения новых свойств можно получить более лучшую модель!

Полиномиальная регрессия

- Полиномиальная регрессия – это тот инструмент, который близко связан с выбором новых свойств



$$h_Q(x) = Q^T x = Q_0 + Q_1 (\text{площадь}) + Q_2 (\text{площадь})^2$$

Пусть x_1 = площадь, x_2 = (площадь)^2, тогда полиномиальная регрессия сведется к линейной регрессии со множеством переменных:

$h_Q(x) = Q^T x = Q_0 + Q_1 x_1 + Q_2 x_2$

Аналитическое решение

- Метод аналитического поиска параметров Q
  - Рассмотрим стоимостную функцию J(Q)

$$J(Q_0, Q_1, \ldots, Q_n) = J(Q) = \frac{1}{2m}\sum_{i=1}^{m}(h_Q(x^{(i)}) - y^{(i)})^2,$$

  - Вычислим частные производные J(Q) по Q_0, Q_1, …, Q_n
  - Приравняем полученные производные к нулю

$$\frac{\partial}{\partial Q_j} J(Q) = \ldots = 0, \text{ для всех } j$$

  - Решим систему линейных уравнений относительно Q_0, Q_1, …, Q_n

Нормальные уравнения

✔ Тренировочное множество данных (скажем, всего m = 4)

Площадь (фут^2), x_1	Число комнат, x_2	Число этажей, x_3	Возраст дома (год), x_4	Цена в 1000-х ($), y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}, \quad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}, \quad Q = (X^T X)^{-1} X^T y$$

Масштабирование свойств не нужно!

Когда, что лучше использовать?

- Пусть есть m тренировочных примеров и n свойств

Градиентный спуск

1. Необходим выбор α
2. Необходимо много итераций
3. Работает хорошо даже если n большое ($n = 10^6$)

Для вычисления обратной матрицы в Matlab используем функцию pinv

Нормальные уравнения

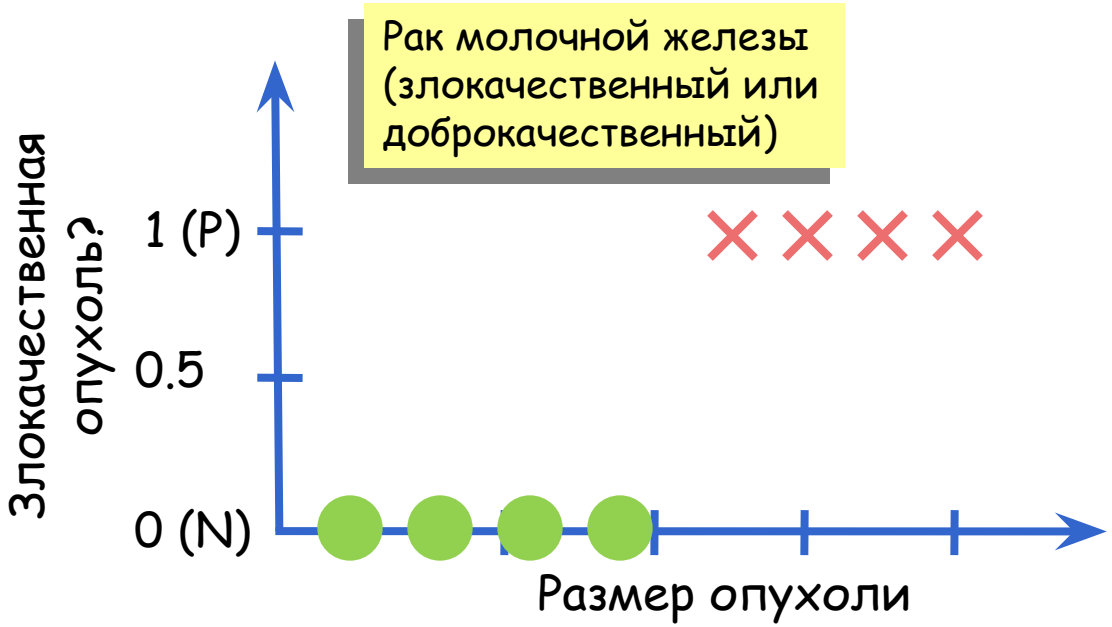
1. Нет необходимости выбирать α
2. Нет необходимости в итерациях
3. Необходимо вычислять $(X^TX)^{-1}$, вычислительная стоимость $O(n^3)$
4. Медленно работает если n большое. Используем если n = 100, 1000, 10000

Классификация. Примеры

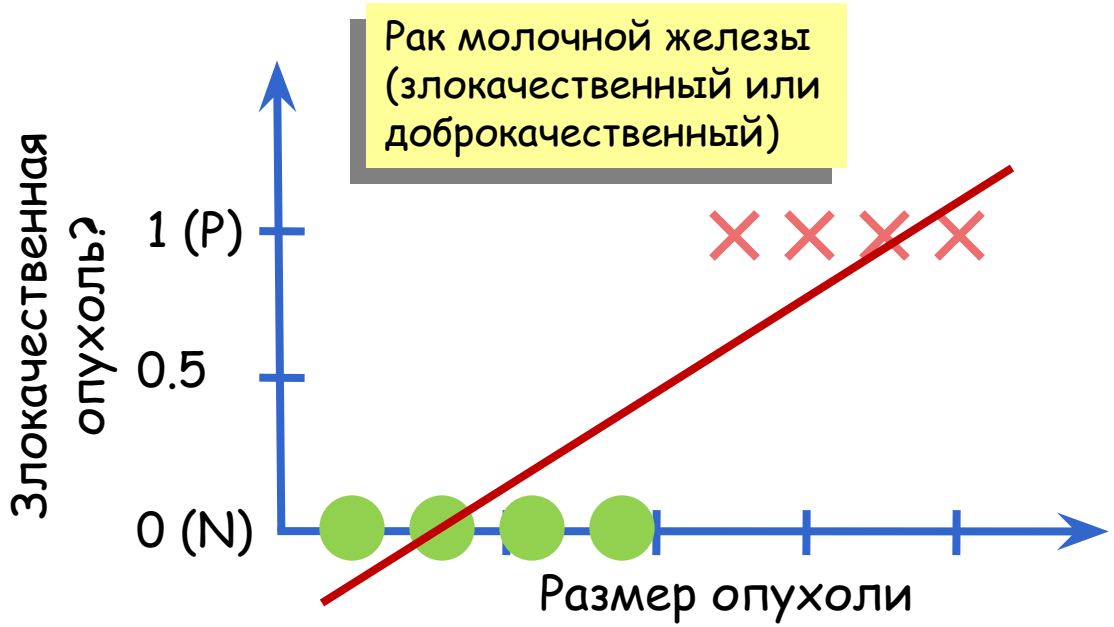
- **Классификация** (предсказание дискретной выходной величины, например, 0 или 1)
- Примеры задач классификации
  - Электронная почта (Email): спам/не спам
  - Онлайн транзакции: мошенничество (да/нет)
  - Опухоль: злокачественная/доброкачественная
  - Видеоаналитика: номер/не номер, пешеход/не пешеход, лицо/не лицо и т.п.
- Далее рассмотрим задачу бинарной классификации!
  - 0: «отрицательный класс» (доброкачественная опухоль)
  - 1: «положительный класс» (злокачественная опухоль)

Снова рассматриваем обучение с учителем!

Классификация

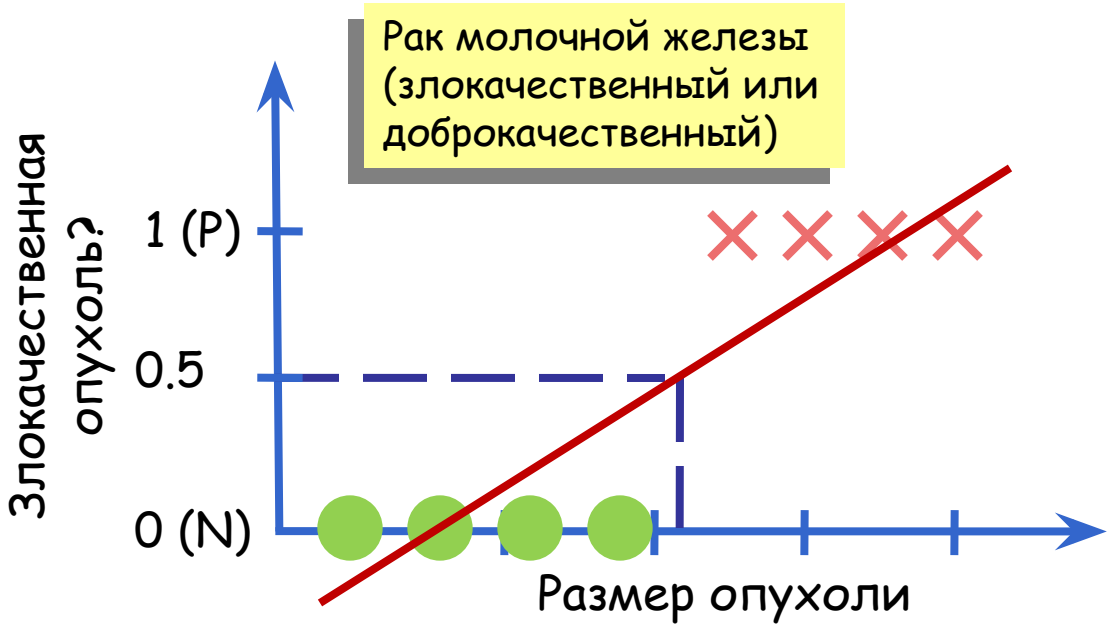


Классификация



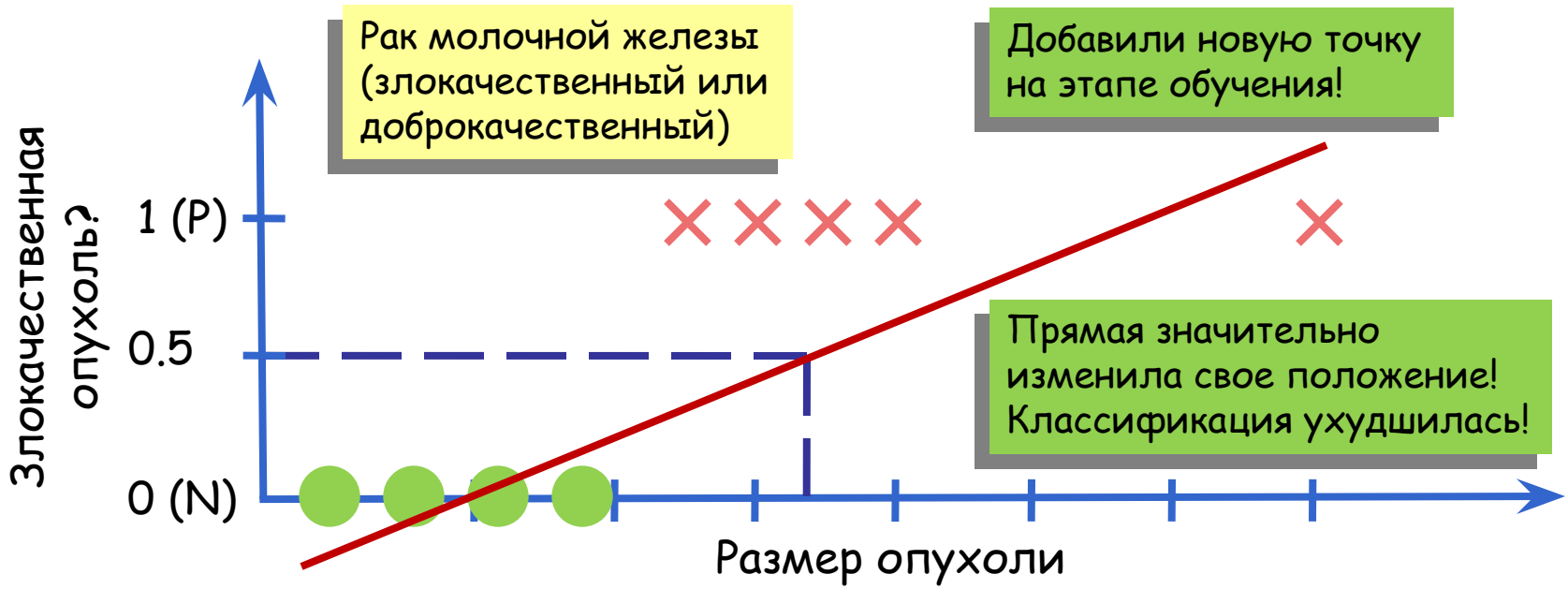
- ✔ Воспользуемся для решения задачи классификации обычной линейной регрессией с одной переменной!

Классификация



- Пусть порог классификатора $h_Q(x)$ находится в точке 0.5:
  - Если $h_Q(x) \geq 0.5$, то предсказываем «1»
  - Если $h_Q(x) < 0.5$, то предсказываем «0»

Классификация



- Пусть порог классификатора $h_Q(x)$ находится в точке 0.5:
  - Если $h_Q(x) \geq 0.5$, то предсказываем «1»
  - Если $h_Q(x) < 0.5$, то предсказываем «0»

Классификация

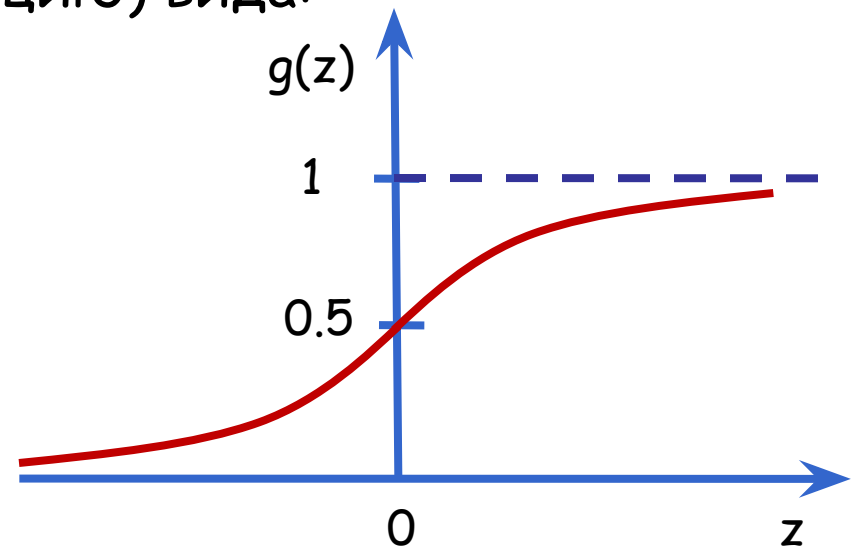
- ✔ Проблемы классификации на основе линейной регрессии с одной переменной
  - ✔ Выход (y) задачи бинарной классификации должен принимать значения «0» или «1». В линейной регрессии $h_Q(x)$ может быть > 1 или < 0
  - ✔ Сильная чувствительность гипотезы по отношению к тренировочной выборке
- ✔ Линейная регрессия может работать хорошо для некоторых частных случаев, но в общем классификация на основе нее – это плохая идея!
- ✔ Введем понятие логистической регрессии, как простейшего метода классификации ($0 \le h_Q(x) \le 1$)

Логистическая регрессия

- ✔ Необходимо сделать так, чтобы $0 \le h_Q(x) \le 1$
- ✔ Для решения этой задачи представим гипотезу в следующем виде: $h_Q(x) = g(Q^T x)$
  - ✔ Здесь функция $g(z)$ представляет сигмоидную функцию (логистическую функцию) вида:

$$g(z) = \frac{1}{1 + e^{-z}},$$

$$h_Q(x) = \frac{1}{1 + e^{-Q^T x}}$$



Интерпретация гипотезы в логистической регрессии

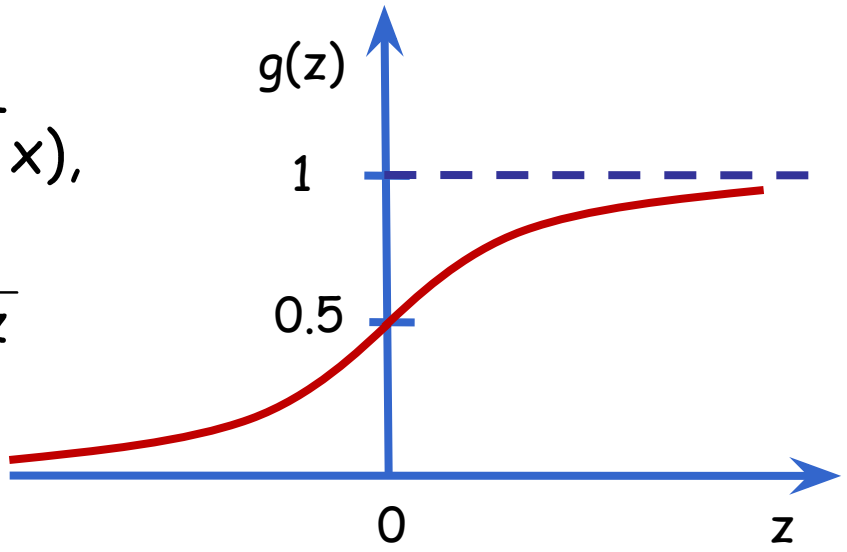
- ✔ $h_Q(x)$ = оценке вероятности того, что y = 1 для входа x
- ✔ Пример: если $x = [x_0, x_1]^T$ = [1, размер опухоли]^T и $h_Q(x) = 0.7$, тогда пациент с 70% шансом имеет злокачественную опухоль
- ✔ Рассматриваемая вероятность P(y = i|x; Q) является условной вероятностью параметризованной Q того, что y = i для заданного x

$$P(y = 0|x; Q) + P(y = 1|x; Q) = 1,$$
$$P(y = 0|x; Q) = 1 - P(y = 1|x; Q)$$

Граница решения (Decision Boundary)

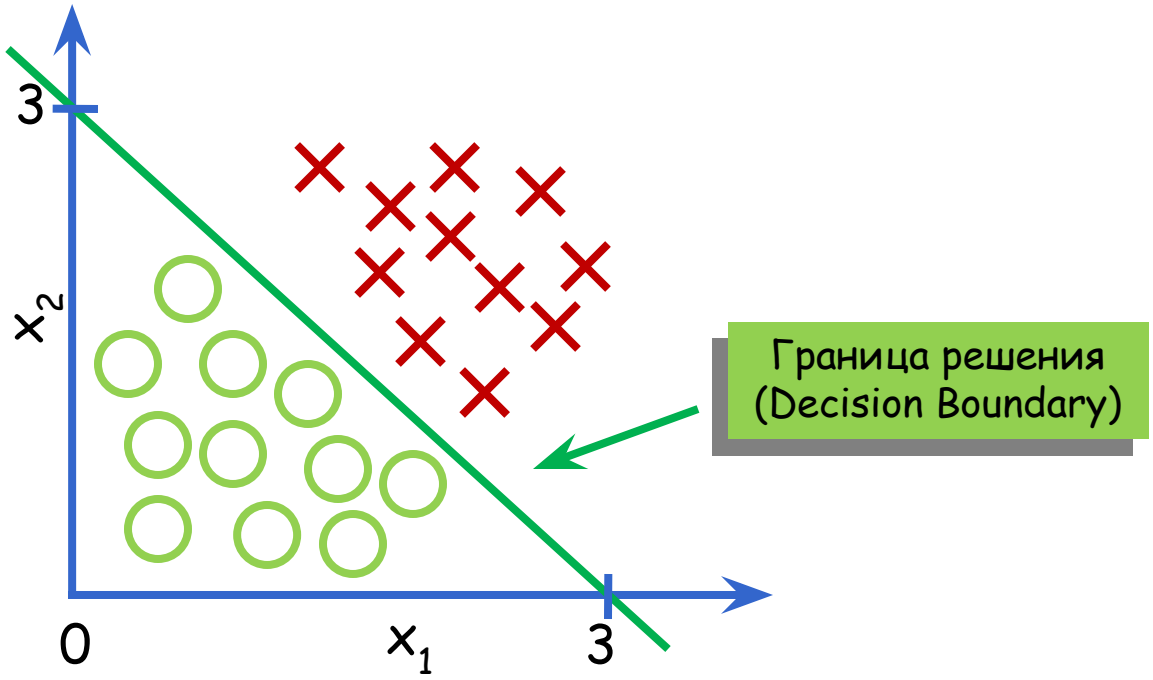
$$h_Q(x) = g(Q^T x),$$

$$g(z) = \frac{1}{1 + e^{-z}}$$



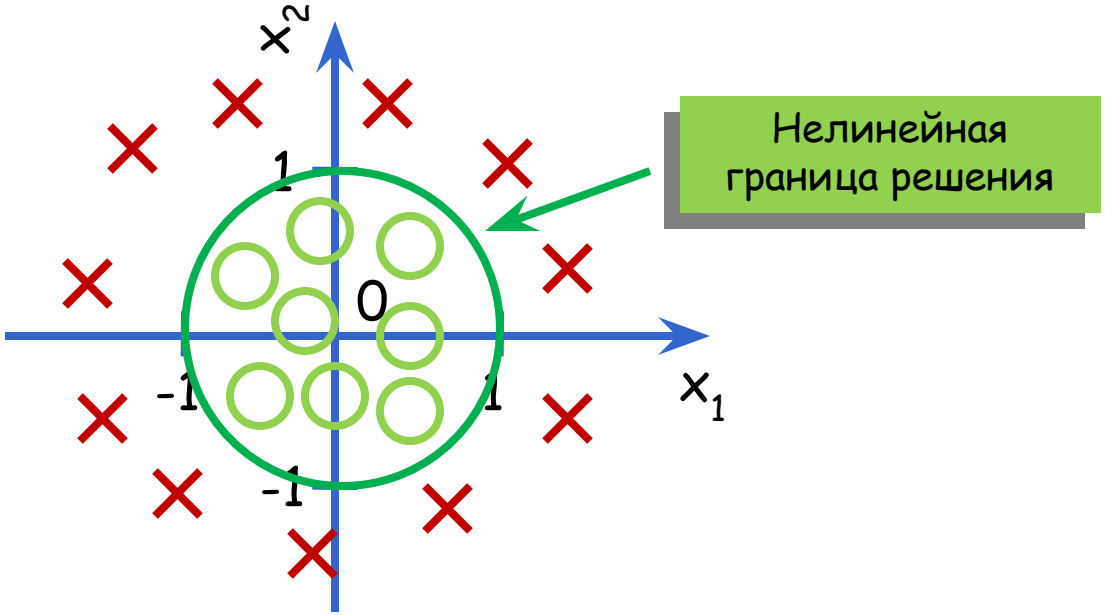
- Пусть порог классификатора $h_Q(x)$ находится в точке 0.5:
  - Если $h_Q(x) \geq 0.5$ ($Q^T x \geq 0$), то предсказываем «1»
  - Если $h_Q(x) < 0.5$ ($Q^T x < 0$), то предсказываем «0»

Граница решения (Decision Boundary)



- Пусть классификатор имеет вид:
  $h_Q(x) = g(Q_0 + Q_1x_1 + Q_2x_2) = g(-3 + x_1 + x_2)$:
- Предсказываем «y = 1» если $-3 + x_1 + x_2 \geq 0$, иначе «y = 0»

Нелинейные границы решения



- ✔ Пусть классификатор имеет вид:
  $h_Q(x) = g(Q_0 + Q_1x_1 + Q_2x_2 + Q_3x_1^2 + Q_4x_2^2) = g(-1 + x_1^2 + x_2^2)$:
- ✔ Предсказываем «y = 1» если $-1 + x_1^2 + x_2^2 \geq 0$, иначе «y = 0»

Стоимостная функция (Cost Function)

- Дана тренировочная выборка $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$, где m – число тренировочных примеров
- Пусть $x \in [x_0, x_1, \ldots, x_n]^T$, $x_0 = 1$, $y \in \{0, 1\}$
- Гипотеза $h_Q(x)$ имеет вид:

$$h_Q(x) = \frac{1}{1 + e^{-Q^T x}}$$

Как определить параметры Q?

Стоимостная функция (Cost Function)

- Дана тренировочная выборка $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$, где m – число тренировочных примеров
- Пусть $x \in [x_0, x_1, \ldots, x_n]^T$, $x_0 = 1$, $y \in \{0, 1\}$
- Гипотеза $h_Q(x)$ имеет вид:

$$h_Q(x) = \frac{1}{1 + e^{-Q^T x}}$$

Как определить параметры Q?

Воспользуемся как и в линейной регрессии стоимостной функцией!

Стоимостная функция (Cost Function)

- Дана тренировочная выборка $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$, где m – число тренировочных примеров
- Пусть $x \in [x_0, x_1, \ldots, x_n]^T$, $x_0 = 1$, $y \in \{0, 1\}$
- Гипотеза $h_Q(x)$ имеет вид:

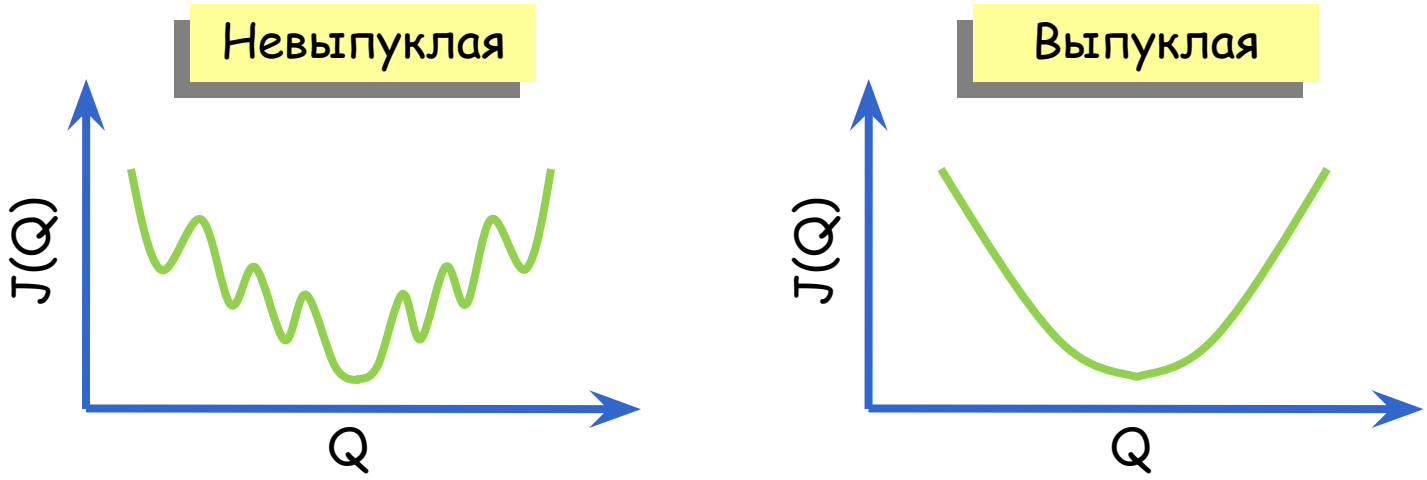
$$h_Q(x) = \frac{1}{1 + e^{-Q^T x}}$$

Как определить параметры Q?

Как задать стоимостную функцию?

Стоимостная функция (Cost Function)

- Выбор стоимостной функции. Вариант первый!
  - Возьмем абсолютно такую же как и в линейной регрессии, помня о том, что гипотеза $h_Q(x)$ задается через сигмоидную функцию
    - Проблема! Стоимостная функция перестает быть выпуклой



Стоимостная функция (Cost Function)

- Выбор стоимостной функции. Вариант второй!
  - Пусть стоимостная функция имеет вид:

$$J(Q) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_Q(x^{(i)}), y^{(i)}),$$

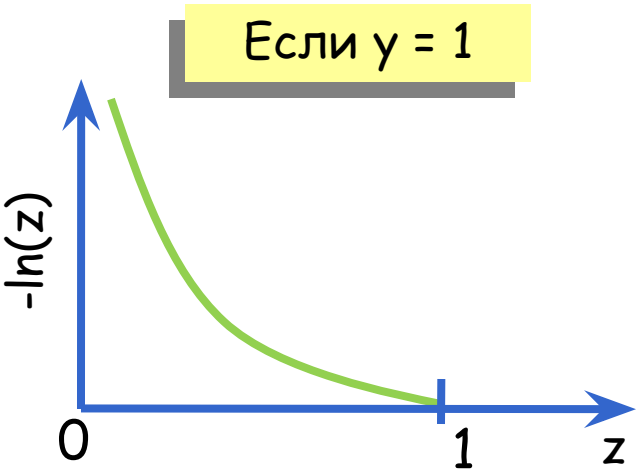
$$Cost(h_Q(x^{(i)}), y^{(i)}) = \begin{cases} -\ln(h_Q(x^{(i)})), \text{ если } y^{(i)} = 1, \\ -\ln(1 - h_Q(x^{(i)})), \text{ если } y^{(i)} = 0 \end{cases}$$

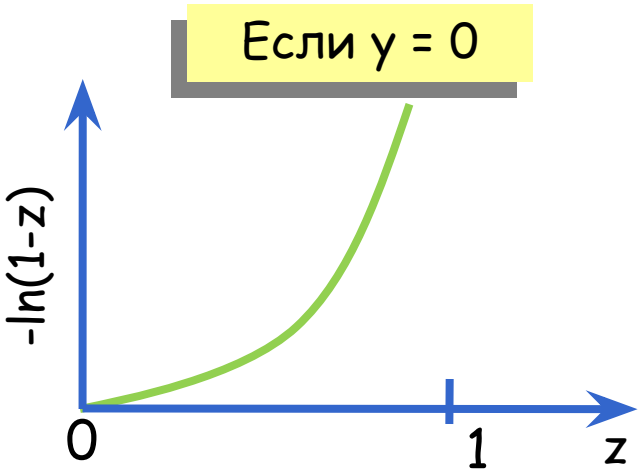
- Заметим, что Cost = 0 если $y^{(i)} = 1$, $h_Q(x^{(i)}) = 1$
  - Если $y^{(i)} = 1$ и $h_Q(x^{(i)}) \to 0$ тогда $Cost \to \infty$
  - Если $h_Q(x^{(i)}) = 0$, но $y^{(i)} = 1$, мы штрафуем алгоритм обучения очень высокой стоимостью!

Стоимостная функция (Cost Function)

- Выбор стоимостной функции. Вариант второй!
- Немного пояснений!

$$Cost(h_Q(x^{(i)}), y^{(i)}) = \begin{cases} -\ln(h_Q(x^{(i)})), \text{ если } y^{(i)} = 1, \\ -\ln(1 - h_Q(x^{(i)})), \text{ если } y^{(i)} = 0 \end{cases}$$





Стоимостная функция (Cost Function)

- Для дальнейшего анализа стоимостную функцию для логистической регрессии удобно представить в виде:

$$J(Q) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ln(h_Q(x^{(i)})) + (1-y^{(i)})\ln(1-h_Q(x^{(i)}))]$$

- Для того, чтобы найти параметры Q, необходимо минимизировать J(Q), например, методом градиентного спуска
- Для того, чтобы выполнить предсказание для нового входного значения x используем

$$h_Q(x) = \frac{1}{1+e^{-Q^T x}}$$

Градиентный спуск для лог. регрессии

repeat until convergence
{
$$Q_j = Q_j - \alpha \frac{\partial}{\partial Q_j} J(Q); \qquad (j = 0, \ldots, n)$$
}

✔ Вычислив производные получим

repeat until convergence
{
$$Q_j = Q_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_Q(x^{(i)}) - y^{(i)}) x_j^{(i)};$$
}

параметры Q обновляются одновременно

Замечание. Градиентный спуск выглядит идентично линейной регрессии, но $h_Q(x)$ задается иначе!

Градиентный спуск для лог. регрессии

repeat until convergence
{
$$Q_j = Q_j - \alpha \frac{\partial}{\partial Q_j} J(Q); \qquad (j = 0, \ldots, n)$$
}

✔ Вычислив производные получим

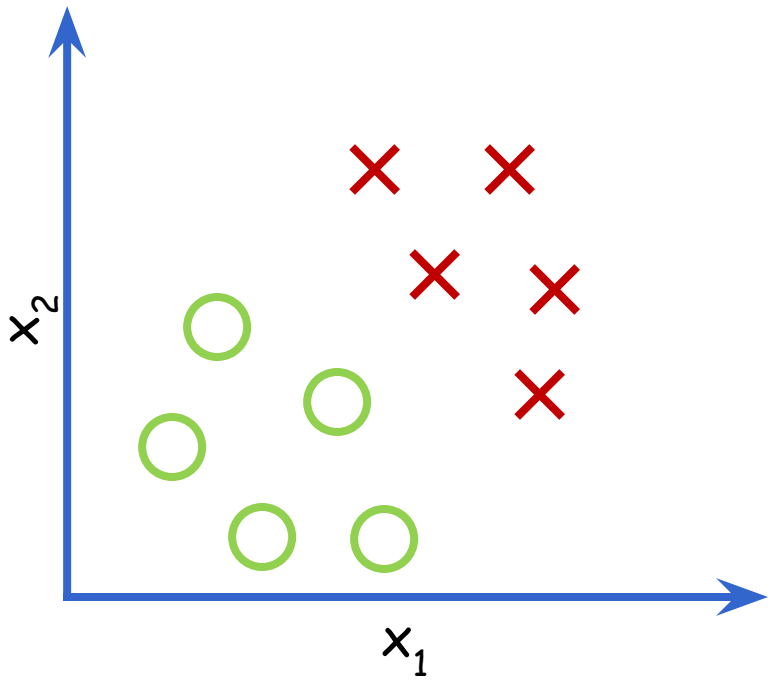
repeat until convergence
{
$$Q_j = Q_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_Q(x^{(i)}) - y^{(i)}) x_j^{(i)};$$
}

параметры Q обновляются одновременно

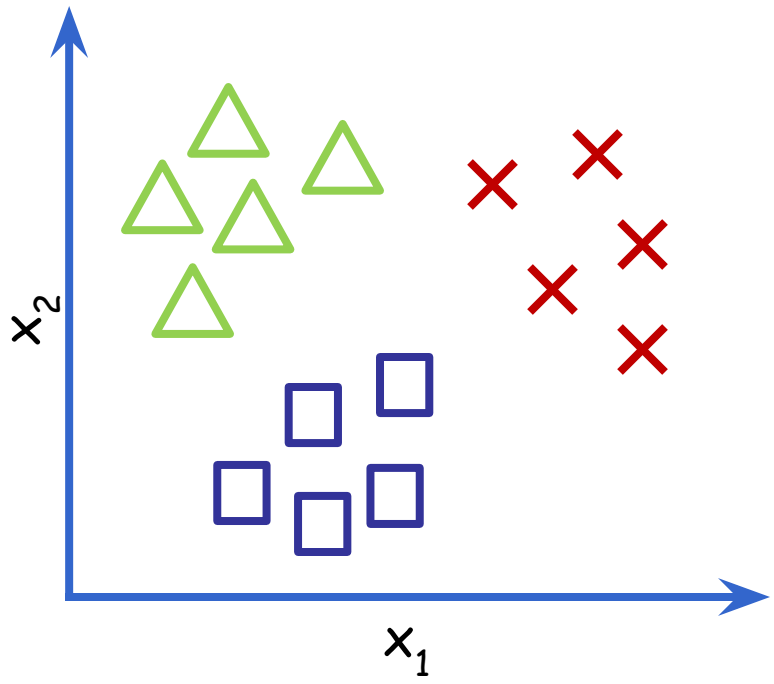
Замечание. В Matlab есть встроенная функция fminunc, позволяющая находить минимум функции нескольких переменных без ограничений. Ее можно использовать вместо вручную написанной Matlab-функции для градиентного спуска (см. лекцию №6 из курса Andrew Ng. Machine Learning (online class), 2012. Stanford University, www.coursera.org/course/ml)!

Многоклассовая классификация

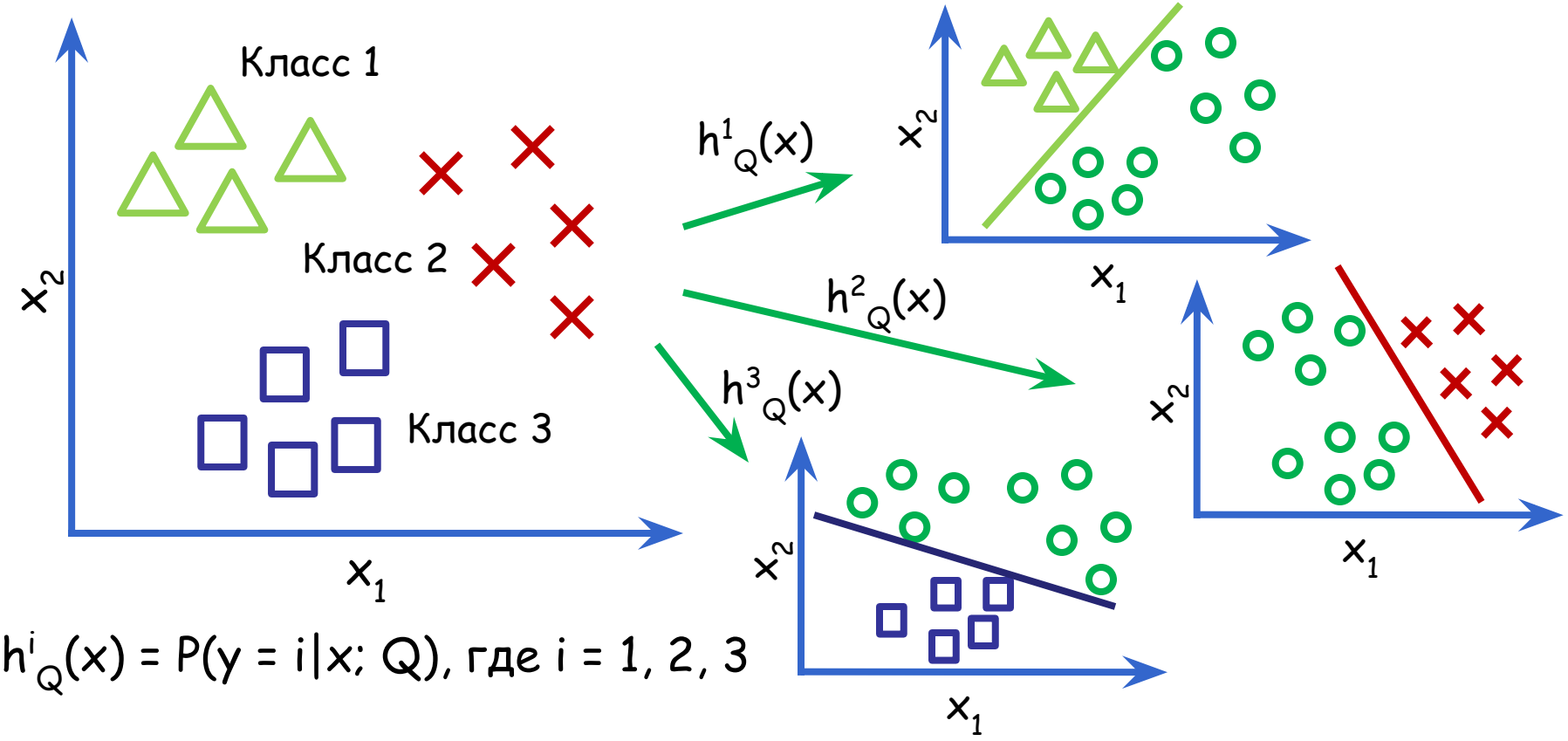
Бинарная классификация



Многоклассовая классификация



Многоклассовая классификация. Подход «один против всех» (One-vs-all)

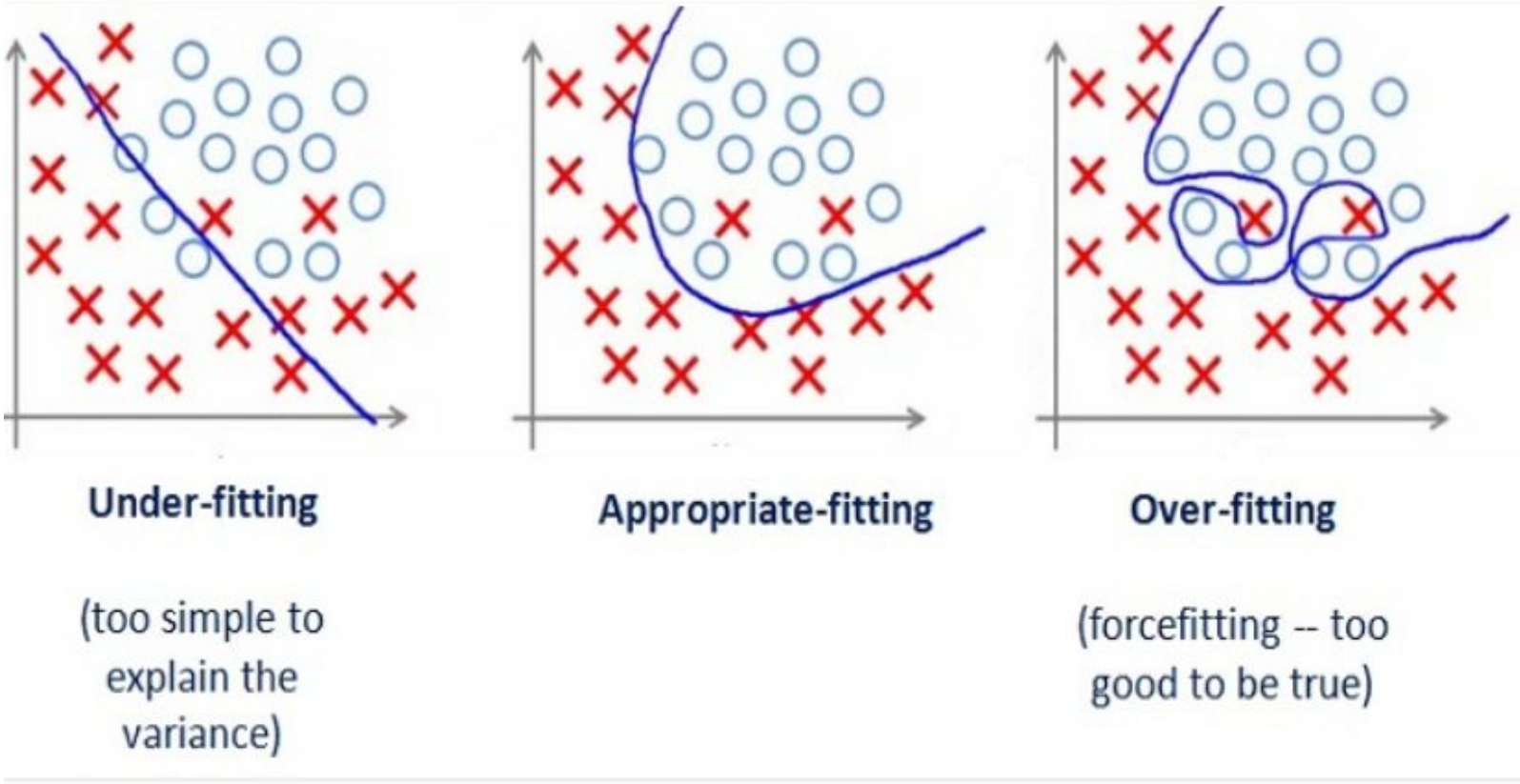


$h^i_Q(x) = P(y = i|x; Q)$, где $i = 1, 2, 3$

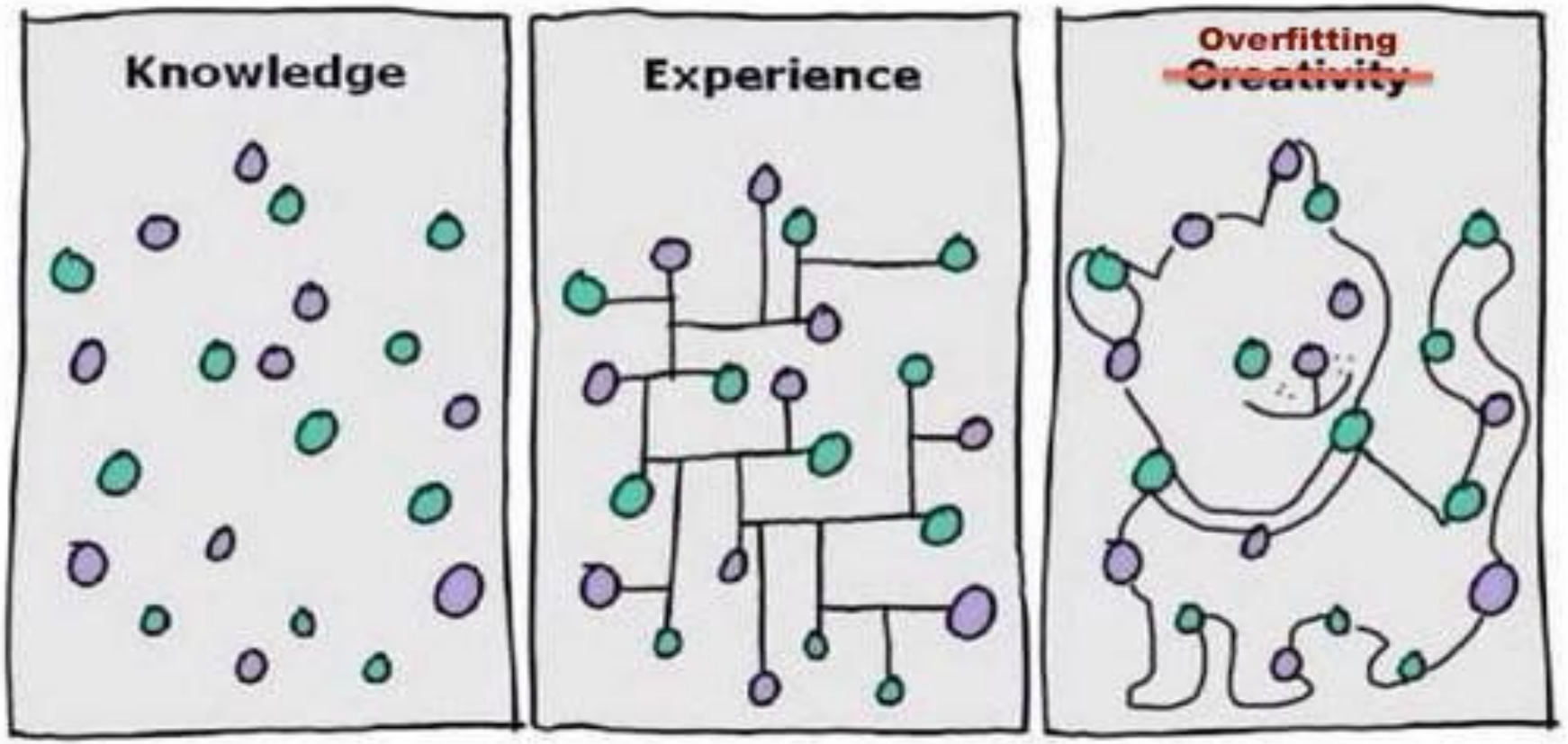
Многоклассовая классификация. Подход «один против всех» (One-vs-all)

- ✔ Обучаем классификаторы основанные на логистической регрессии $h^i_Q(x)$ для каждого i-го класса для того, чтобы предсказать вероятность y = i
- ✔ Для нового входа x выполнить предсказание и выбрать класс i с максимальным значением $h^i_Q(x)$
- ✔ Возможной альтернативой решения задачи многоклассовой классификации может являться подход «один против одного» (One-vs-one)
  - ✔ Обучаем логистическую регрессию для каждой пары классов
  - ✔ Каждый классификатор голосует за классы
  - ✔ Выбираем класс с наибольшим числом голосов

Обучение и переобучение



Обучение и переобучение



Благодарности

- ✔ В лекции использовались материалы курса:
  - ✔ Andrew Ng. Machine Learning (online class), 2012. Stanford University, www.coursera.org/course/ml

Куррикулум витте Эндрю здесь: http://ai.stanford.edu/~ang/