# Машинное обучение (Machine Learning)

## Регрессионные модели

Уткин Л.В.

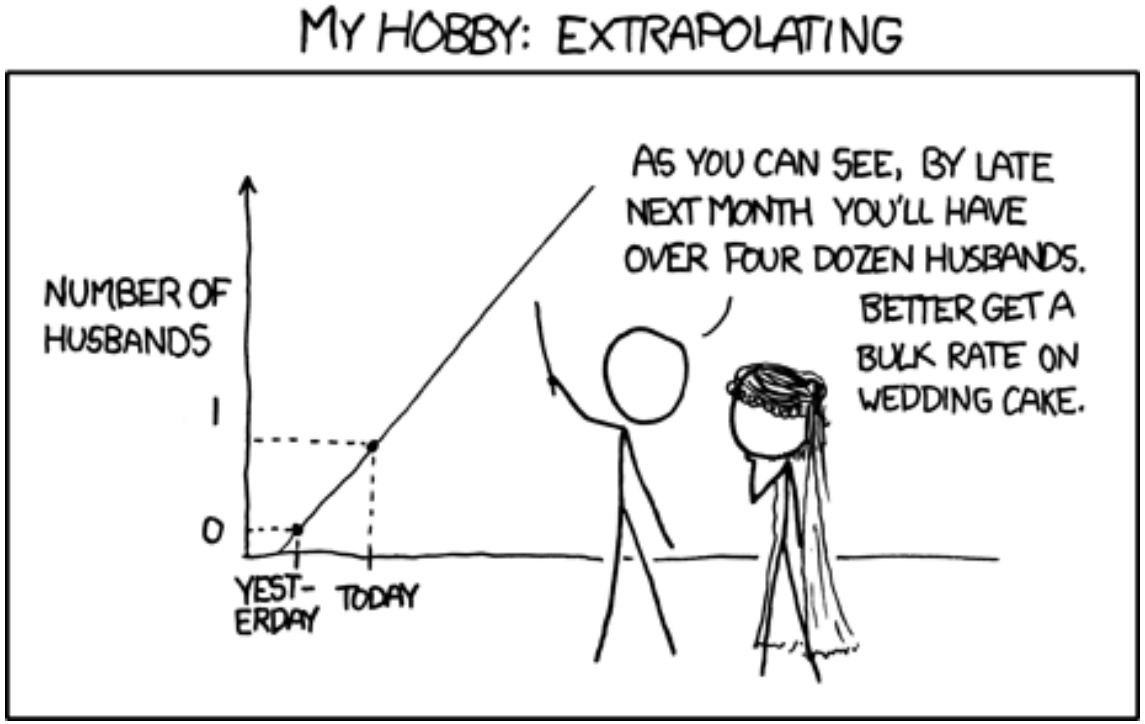
ПОЛИТЕХ
Санкт-Петербургский политехнический университет Петра Великого

# Содержание

*Презентация является компиляцией и заимствованием материалов из замечательных курсов и презентаций по машинному обучению:*

***К.В. Воронцова, А.Г. Дьяконова, Н.Ю. Золотых, С.И. Николенко, Andrew Moore, Lior Rokach, Matthias Schmid, Rong Jin, Cheng Li, Luis F. Teixeira, Alexander Statnikov и других.***

# Регрессия




https://www.explainxkcd.com/wiki/index.php/605:_Extrapolating

# Линейные регрессионные модели

# Линейная регрессионная модель

- Обучающая выборка:
  $\mathbf{S} = \{(\mathbf{x}_1, \widehat{y}_1), (\mathbf{x}_2, \widehat{y}_2), ..., (\mathbf{x}_n, \widehat{y}_n)\}$, $\mathbf{x}_j = (x_{j,1}, ..., x_{j,m})$
- Линейная модель: $f(\mathbf{x}, \mathbf{b}) = b_0 + b_1 X_1 + ... + b_m X_m$
- Оценки:

$$\widehat{y}_i = f(\mathbf{x}_i, \mathbf{b}) + \epsilon = b_0 + b_1 x_{i,1} + ... + b_m x_{i,m} + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Задача: найти “наилучшую” линейную функцию $f(\mathbf{x}, \mathbf{b})$, аппроксимирующую $\mathbf{S}$
- Задача: или найти $\mathbf{b} = (b_0, b_1, ..., b_m)^{\mathrm{T}}$.

# Линейная регрессионная модель

| $y$ | $=$ | $b_0 + b_1 X_1 + \ldots + b_m X_m$ |
|---|---|---|
| Зависимая переменная | | Свободные переменные |
| Dependent variable | | Independent variables |
| Outcome variable | | Predictor variables |
| Response variable | | Explanatory variables |

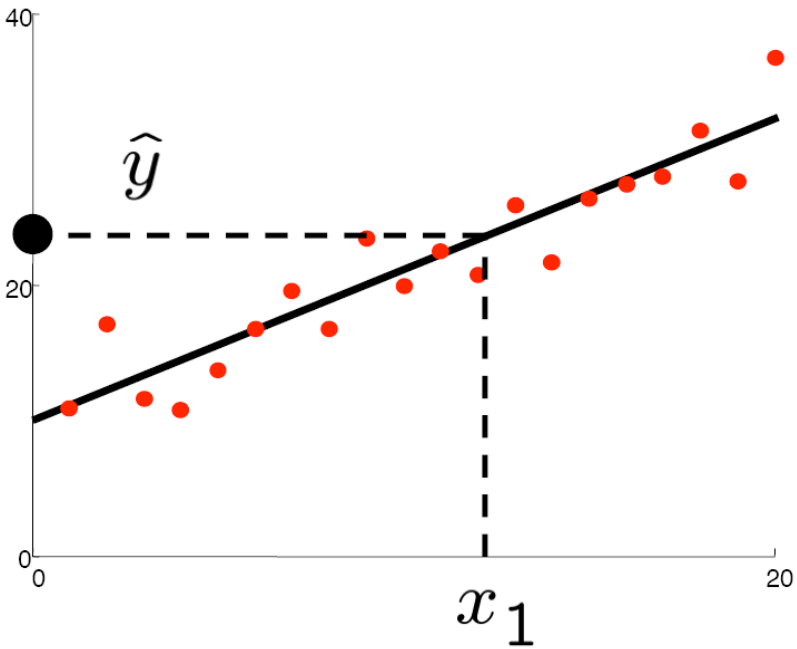
# Линейная регрессионная модель - матричная форма

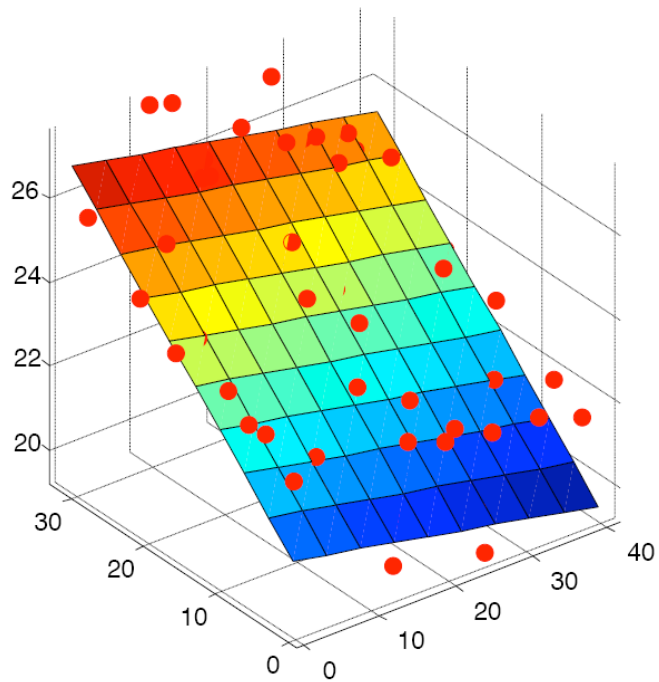
$$\begin{aligned}
\widehat{y}_1 &= b_0 + b_1 x_{1,1} + \ldots + b_m x_{1,m} + \epsilon_1 \\
\widehat{y}_2 &= b_0 + b_1 x_{2,1} + \ldots + b_m x_{2,m} + \epsilon_2 \\
&\cdots \\
\widehat{y}_n &= b_0 + b_1 x_{n,1} + \ldots + b_m x_{n,m} + \epsilon_n
\end{aligned} \quad \Rightarrow \quad \mathbf{Y} = \mathbf{X}\mathbf{b} + \epsilon$$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,m} \\ 1 & x_{2,1} & & x_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x_{n,1} & \cdots & x_{n,m} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_0 \\ b_1 \\ \cdots \\ b_m \end{pmatrix}$$

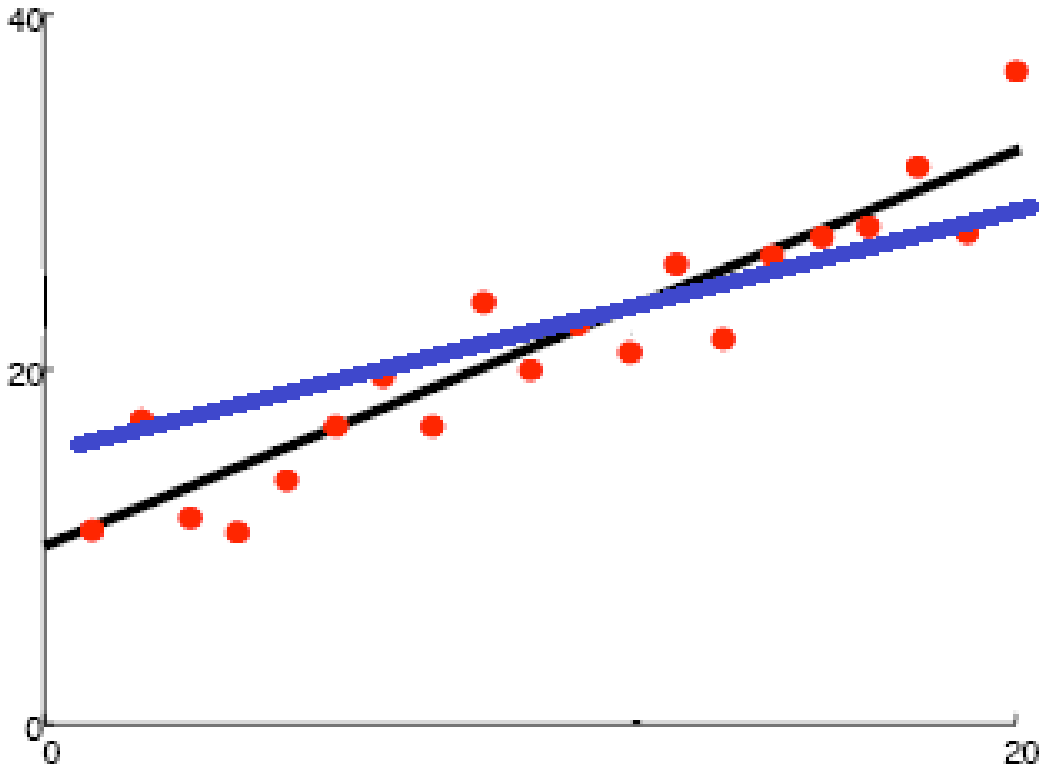
# Линейная регрессионная модель

$$\widehat{y}_i = b_0 + b_1 x_{i,1} + \epsilon, \qquad \widehat{y}_i = b_0 + b_1 x_{i,1} + b_1 x_{i,2} + \epsilon$$

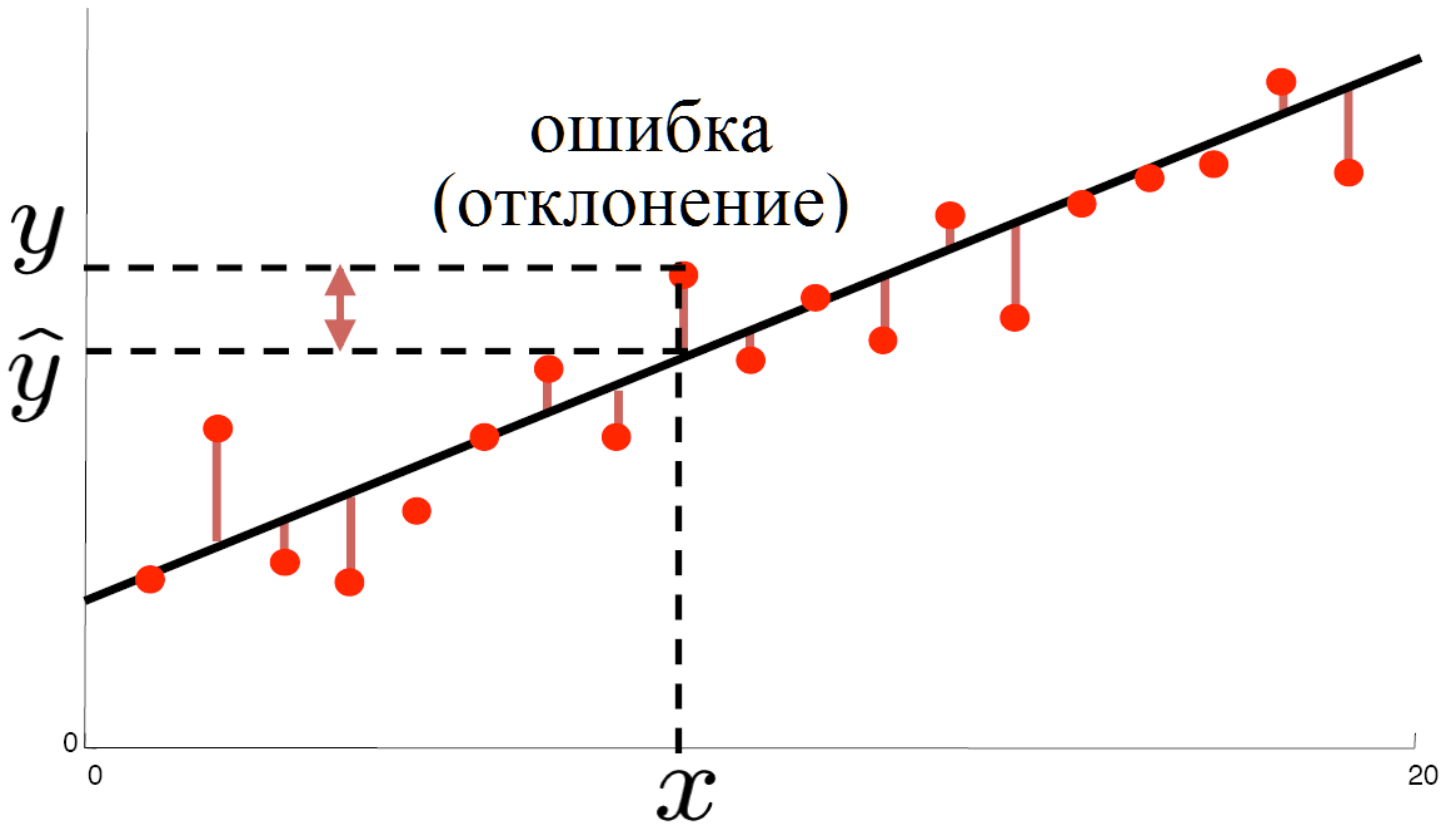


# Какая модель лучше?



# Линейная регрессионная модель



# Линейная регрессионная модель - эмпирический функционал риска

$$E(\mathbf{b}) = \frac{1}{2N}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i, \mathbf{b}))^2$$

$$= \frac{1}{2N}\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_{i,1} + \ldots + b_m x_{i,m}))^2 \to \min_{\mathbf{b}}$$

Матричная форма:

$$E(\mathbf{b}) = \frac{1}{2N}(\mathbf{Y} - \mathbf{X}\mathbf{b})^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{b}) \to \min_{\mathbf{b}}$$

МНК - метод наименьших квадратов

Как найти коэффициенты $\mathbf{b}$? Производные по всем $b_0, b_1, \ldots, b_m$.

# Решение задачи определения параметров

$$E(\mathbf{b}) = \frac{1}{2N}(\mathbf{Y} - \mathbf{X}\mathbf{b})^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{b}) \to \min_{\mathbf{b}}$$

$$\frac{\partial E(\mathbf{b})}{\partial b_k} = \sum_{i=1}^{n} \left(b_0 + b_1 x_{i,1} + \ldots + b_m x_{i,m} - y_i\right) x_{i,k}$$

$$= \left(\sum_{i=1}^{n} \mathbf{b}\mathbf{x}_i - \sum_{i=1}^{n} y_i x_{i,k}\right) = 0, \;\; k = 0, \ldots, m.$$

Здесь $\mathbf{x}_i = (1, x_{i,1}, \ldots, x_{i,m})$

$$\mathbf{b} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$$

# Некоторые проблемы с линейной регрессией

- Что делать, если размерность переменных больше, чем количество наблюдений?
- МНК не работает, так как система уравнений имеет бесконечное число решений
- **Основная идея**: ограничить множество решений путем ограничений на множество параметров $\mathbf{b}$

# Гребневая регрессия

- Пусть $f(\mathbf{x}, \mathbf{b}) = b_0 + b_1 x_1 + ... + b_m x_m = b_0 + \sum_{i=1}^{m} b_i x_i$
- Ограничим возможные большие коэффициенты $\mathbf{b}$ условием $\sum_{i=1}^{m} b_i^2 < C$
- $\sum_{i=1}^{m} b_i^2 = \|\mathbf{b}\|^2$ - Эвклидова норма
- Гребневая регрессия (**ridge regression**):

$$\mathbf{b} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} \left( y_i - b_0 - \sum_{j=1}^{m} b_j x_{ij} \right)^2$$

при ограничении

$$\sum_{j=1}^{m} b_j^2 < C$$

# Гребневая регрессия (двойственная форма)

- Используя метод множителей Лагранжа, получим эквивалентную задачу

$$\mathbf{b} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} \left( y_i - b_0 - \sum_{j=1}^{m} b_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{m} b_j^2$$

- Это задача квадратичной оптимизации
- Второе слагаемое - штраф, зависящий от $\|\mathbf{b}\|^2 = \sum_{j=1}^{m} b_j^2$

# Решение задачи оптимизации

- Производная по $\mathbf{b}$ приравнивается 0:

$$(\mathbf{Y} - \mathbf{X}\mathbf{b})^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{b}) + \lambda \mathbf{b}^{\mathrm{T}}\mathbf{b}$$
$$= \mathbf{b}^{\mathrm{T}}[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda I]\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Y} - \mathbf{Y}^{\mathrm{T}}\mathbf{X}\mathbf{b} + \mathbf{Y}^{\mathrm{T}}\mathbf{Y} = 0$$

- Решение

$$\mathbf{b}_{\text{гребн}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda I)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$$

# Лассо (Tibshirani, 1996)

- Least Absolute Shrinkage and Selection Operator - LASSO
- Заменим Эвклидову норму $\|\mathbf{b}\|^2$ нормой $L_1$:

$$\mathbf{b} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} \left( y_i - b_0 - \sum_{j=1}^{m} b_j x_{ij} \right)^2$$

при ограничении

$$\sum_{j=1}^{m} |b_j| < C$$

- Двойственная форма:

$$\mathbf{b} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} \left( y_i - b_0 - \sum_{j=1}^{m} b_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{m} |b_j|$$

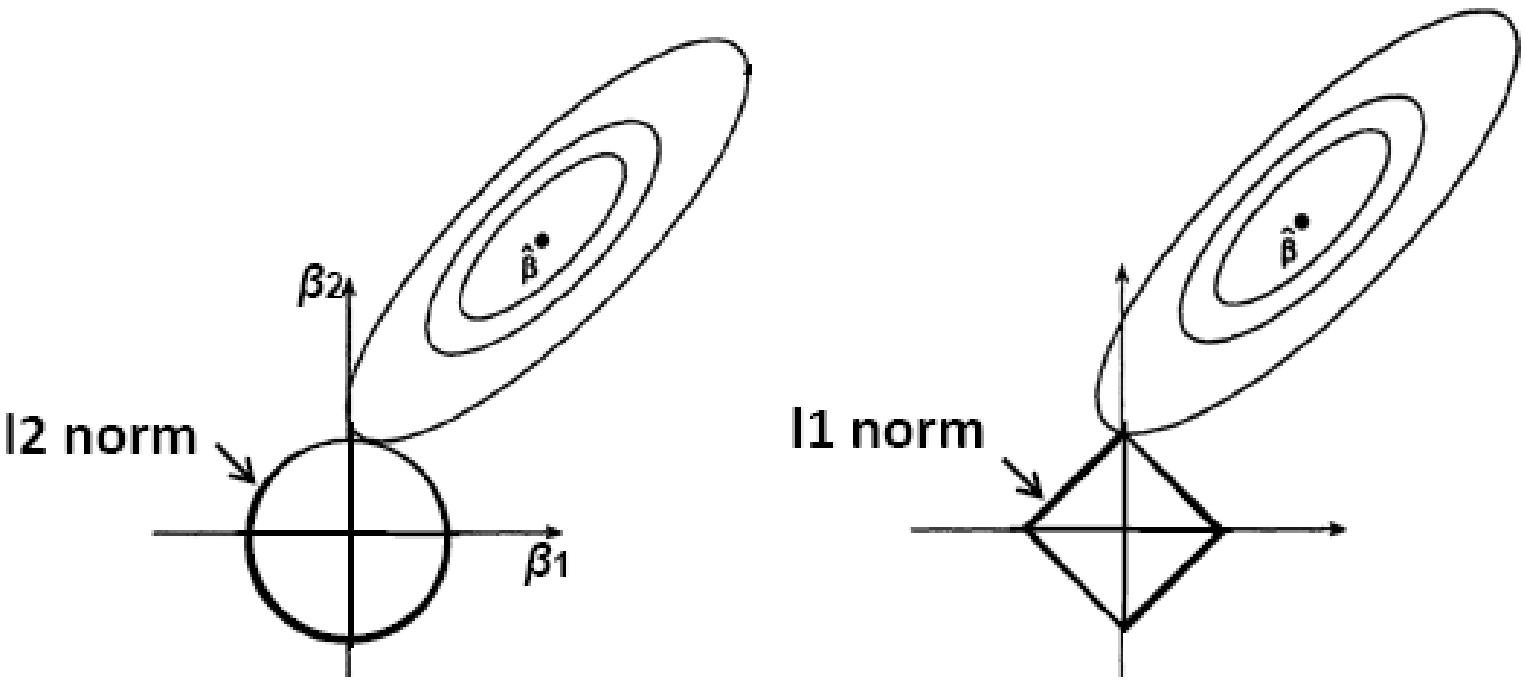
# Лассо (двойственная форма)

- Задача оптимизации больше не квадратическая, но выпуклая

$$\mathbf{b} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} \left( y_i - b_0 - \sum_{j=1}^{m} b_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{m} |b_j|$$

- В отличие от гребневой регрессии, нет аналитического решения
- Efron et al. (2002) предложили эффективный алгоритм *lars* для решения
- **Решение разреженное**

# Лассо и гребневая регрессия



# Эластичные сети

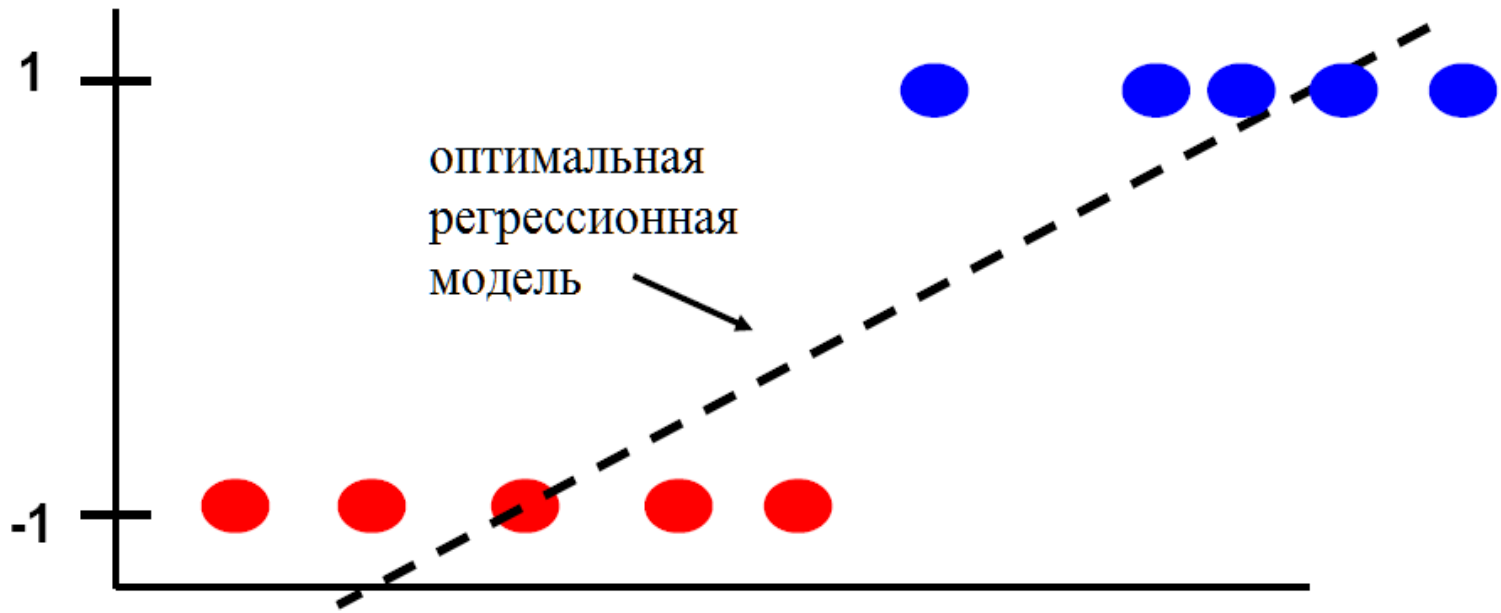
- При помощи Лассо получаем разреженное решение
- При помощи гребневой регрессии имеем слишком “размазанное” решение
- А можно что-то промежуточное?
- Эластичные сети (**elastic net**):

$$\mathbf{b} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n} \left( y_i - b_0 - \sum_{j=1}^{m} b_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{m} b_j^2 + \lambda_2 \sum_{j=1}^{m} |b_j|$$

# Логистическая регрессия

# Логистическая регрессия

- А можно ли использовать регрессию для классификации, т.е., когда $y \in \{0, 1\}$?
- Если $\mathbf{x}_i\mathbf{b} \geq 0$, то $y_i = 1$, иначе $y_i = 0$.



# Функционал риска в регрессии

- Используем в регрессии:

$$E(\mathbf{b}) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i, \mathbf{b}))^2$$

- Функционал риска может давать плохие результаты в классификации, т.к. чем больше отступ или ошибка $(y_i - f(\mathbf{x}_i, \mathbf{b}))^2$, тем хуже, а в классификации положительный отступ - лучше
- Попробуем логарифмический функционал риска

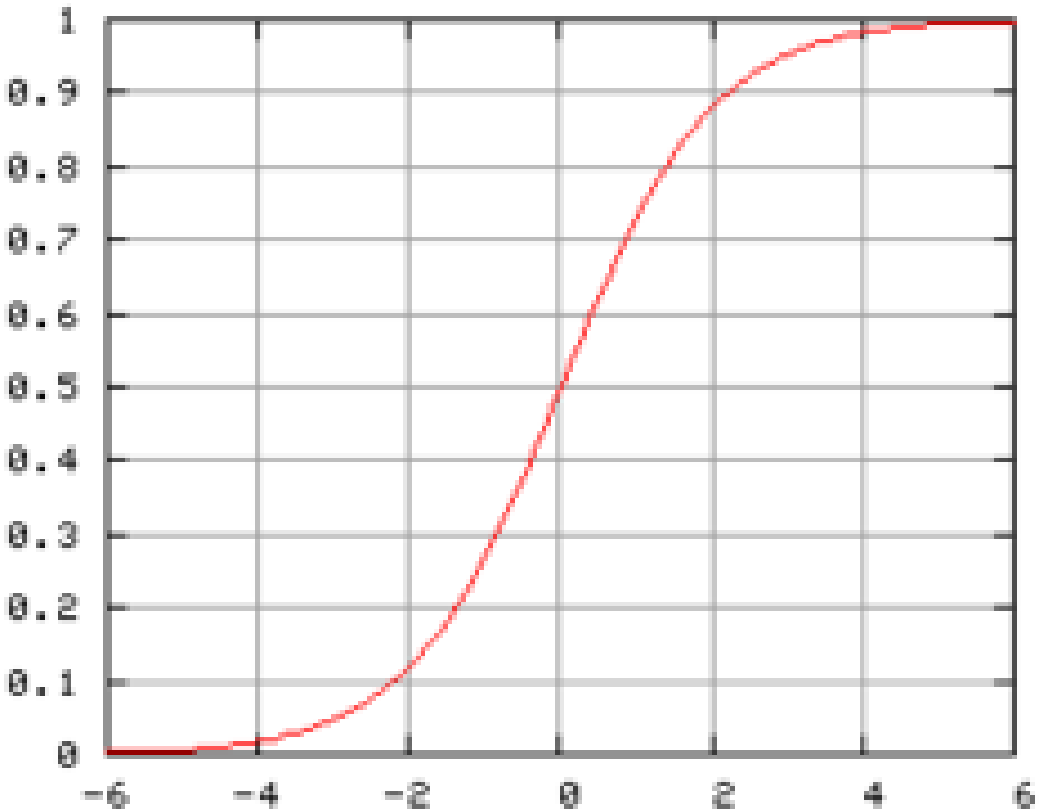
# Логарифмический функционал риска

- $$E(\mathbf{b}) = \sum_{i=1}^{n} \log_2 \left(1 + e^{y_i f(\mathbf{x}_i, \mathbf{b})}\right)$$
- Это эквивалентно замене линейной функции $f(\mathbf{x}, \mathbf{b})$ логистической или сигмоидной функцией (сигмоид)

$$g(z) = \frac{1}{1 + e^{-z}}, \quad z = \mathbf{b}^{\mathrm{T}}\mathbf{x}, \quad 0 \leq g(z) \leq 1$$

# Сигмоид

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



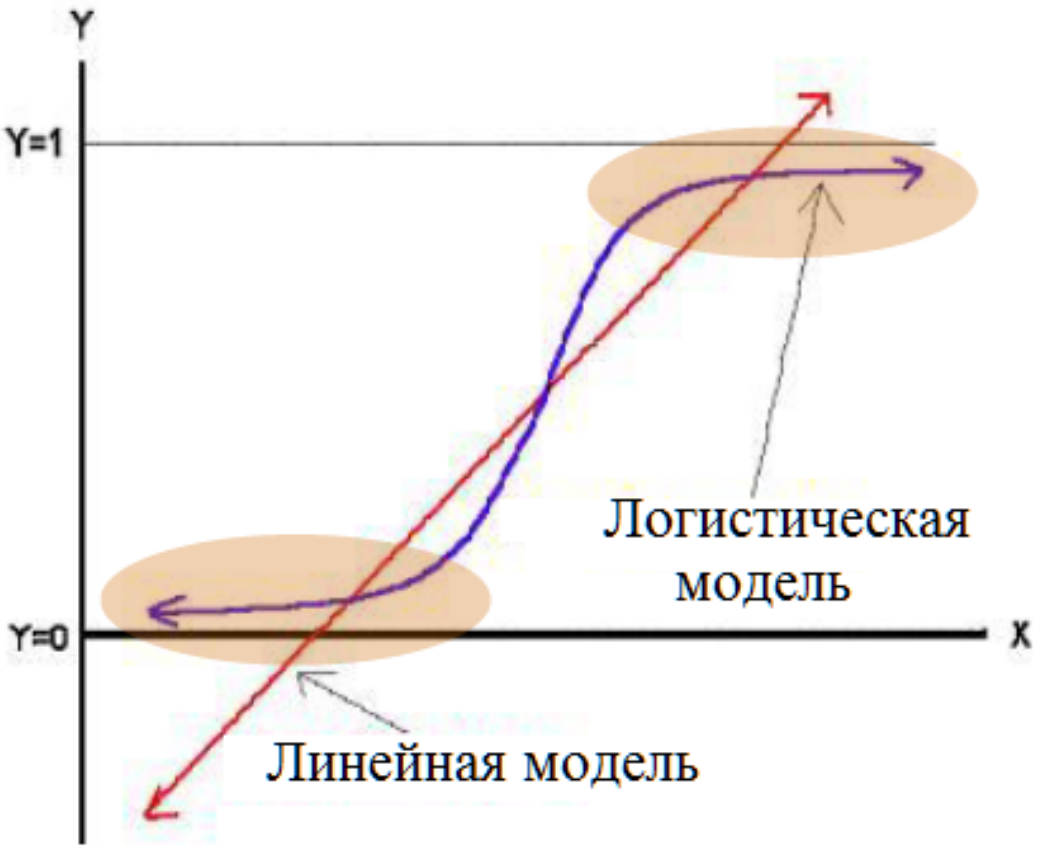
# Классификация

Используя сигмоид, получим вероятности:

$$P(y = 0|\mathbf{x}, \mathbf{b}) = g(\mathbf{b}^{\mathrm{T}}\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{b}^{\mathrm{T}}\mathbf{x}}}$$

$$P(y = 1|\mathbf{x}, \mathbf{b}) = 1 - g(\mathbf{b}^{\mathrm{T}}\mathbf{x}) = \frac{e^{-\mathbf{b}^{\mathrm{T}}\mathbf{x}}}{1 + e^{-\mathbf{b}^{\mathrm{T}}\mathbf{x}}}$$

# Классификация



# Параметры классификации

Метод максимума функции правдоподобия

$$L(y|\mathbf{x},\mathbf{b}) = \prod_{i=1}^{n} \left(1 - g(\mathbf{b}^{\mathrm{T}}\mathbf{x}_i)\right)^{y_i} g(\mathbf{b}^{\mathrm{T}}\mathbf{x}_i)^{(1-y_i)} \to \max_{\mathbf{b}}$$

или логарифм

$$\begin{aligned} L(y|\mathbf{x},\mathbf{b}) &= \sum_{i=1}^{n} y_i \ln\left(1 - g(\mathbf{b}^{\mathrm{T}}\mathbf{x}_i)\right) + (1-y_i)\ln(\mathbf{b}^{\mathrm{T}}\mathbf{x}_i) \\ &= \sum_{i=1}^{n} y_i \mathbf{b}^{\mathrm{T}}\mathbf{x}_i - \ln\left(1 + e^{-\mathbf{b}^{\mathrm{T}}\mathbf{x}_i}\right) \to \max_{\mathbf{b}} \end{aligned}$$

# Параметры классификации

- Решение задачи оптимизации: частные производные по $\mathbf{b}$.
- Проблема: нельзя получить решение в явном виде
- Но: функция производной является вогнутой
- Следовательно можно получить численное решение, например, при помощи градиентного спуска.

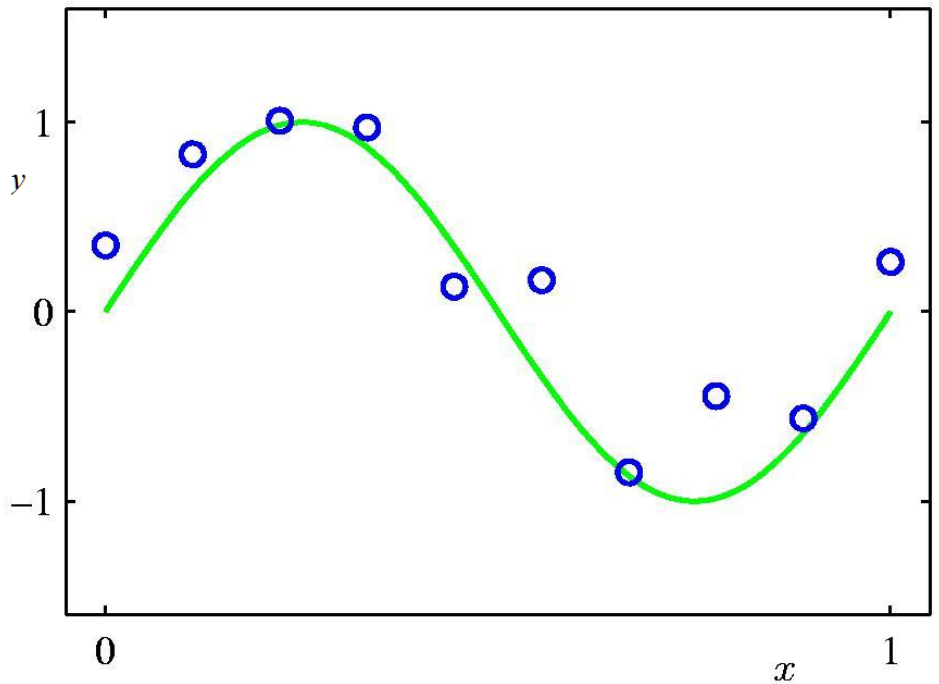
# Логистическая регрессия со сглаживанием

- Целевая функция - отрицательная логарифмическая функция правдоподобия

$$\min_{\mathbf{b}} \left[ -\frac{1}{n} \sum_{i=1}^{n} y_i \mathbf{b}^{\mathrm{T}} \mathbf{x}_i - \ln \left( 1 + e^{-\mathbf{b}^{\mathrm{T}} \mathbf{x}_i} \right) \right] + \lambda_1 \sum_{j=1}^{m} b_j^2 + \lambda_2 \sum_{j=1}^{m} |b_j|$$

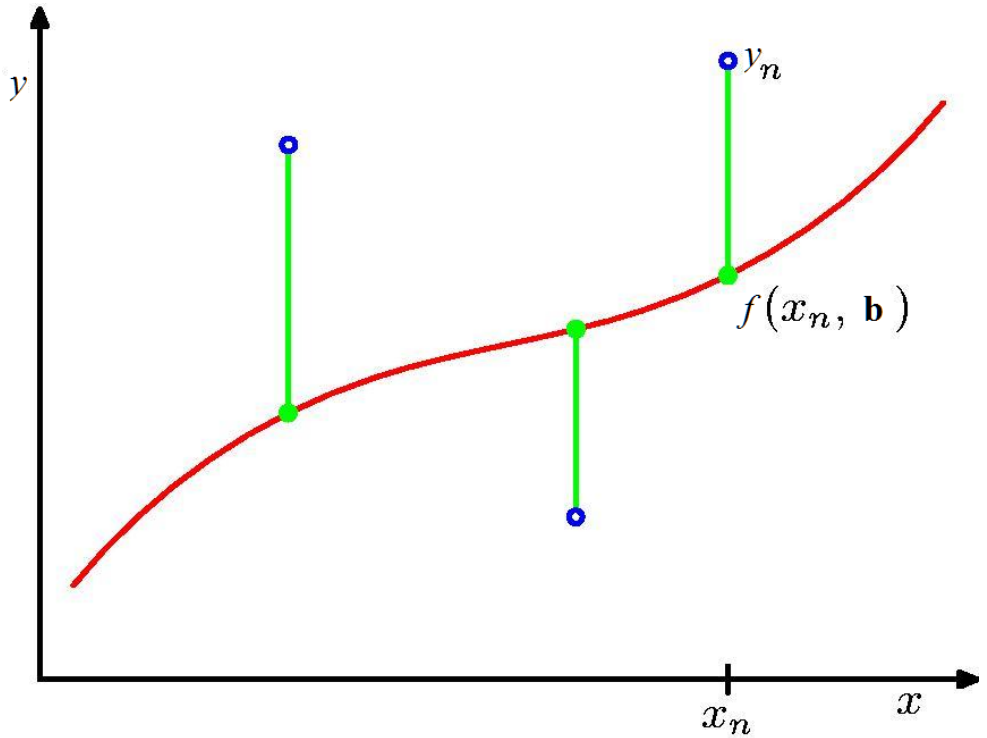
# Нелинейная регрессия

# Полиномиальная регрессионная модель



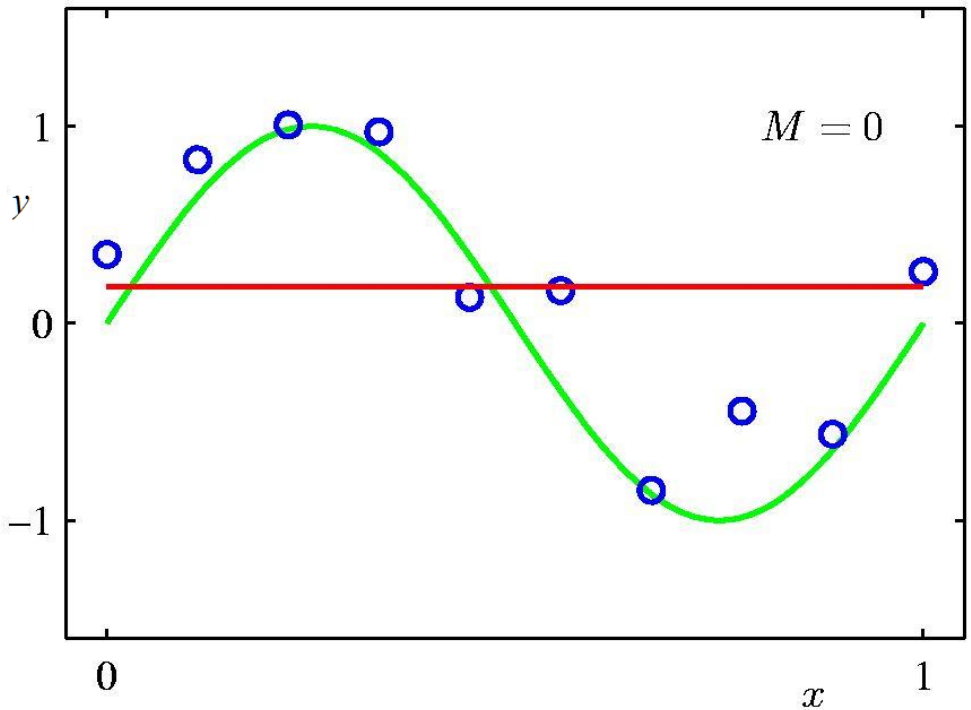

$$f(x, \mathbf{a}) = a_0 + a_1 x + a_2 x^2 + \ldots + a_M x^M = \sum_{i=0}^{M} a_i x^i$$

# Сумма квадратов отклонений

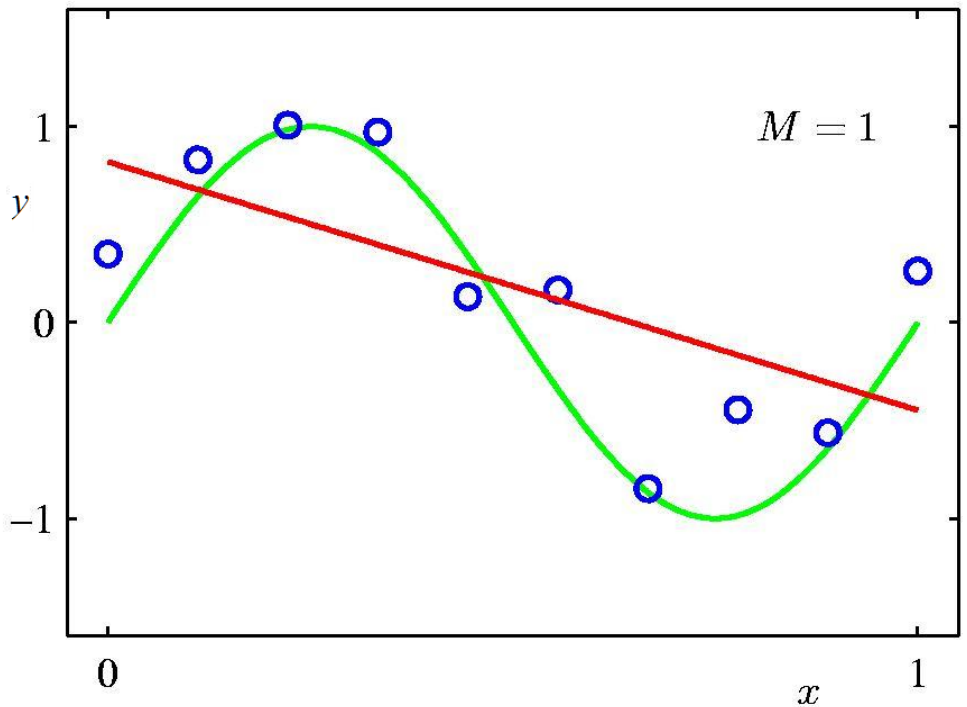



$$E(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{N}\left(f(x_i, \mathbf{a}) - y_i\right)^2$$

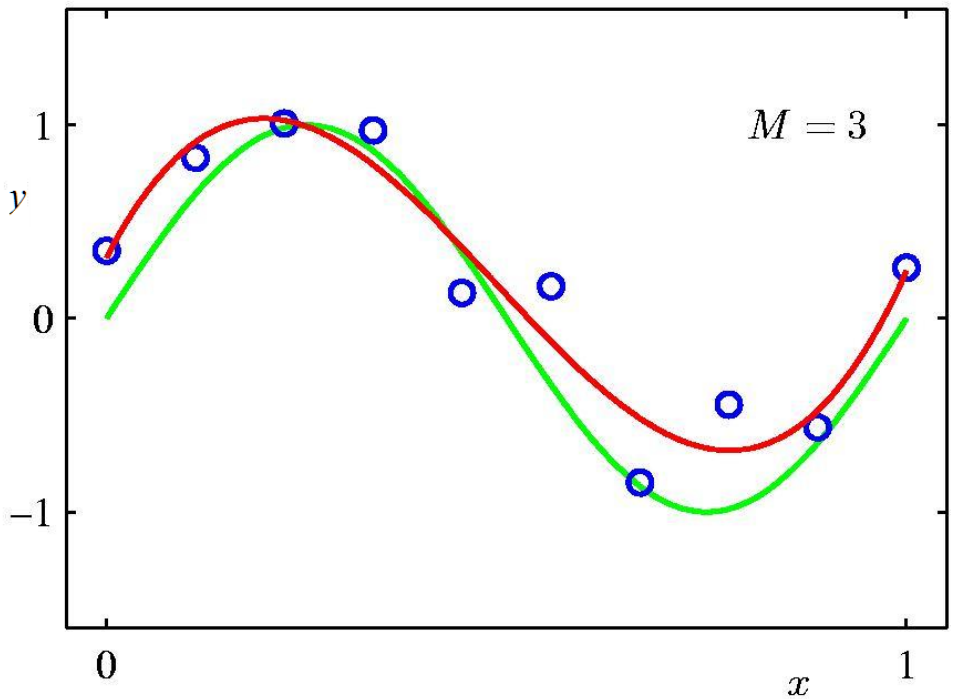
# Полином 0-ой степени



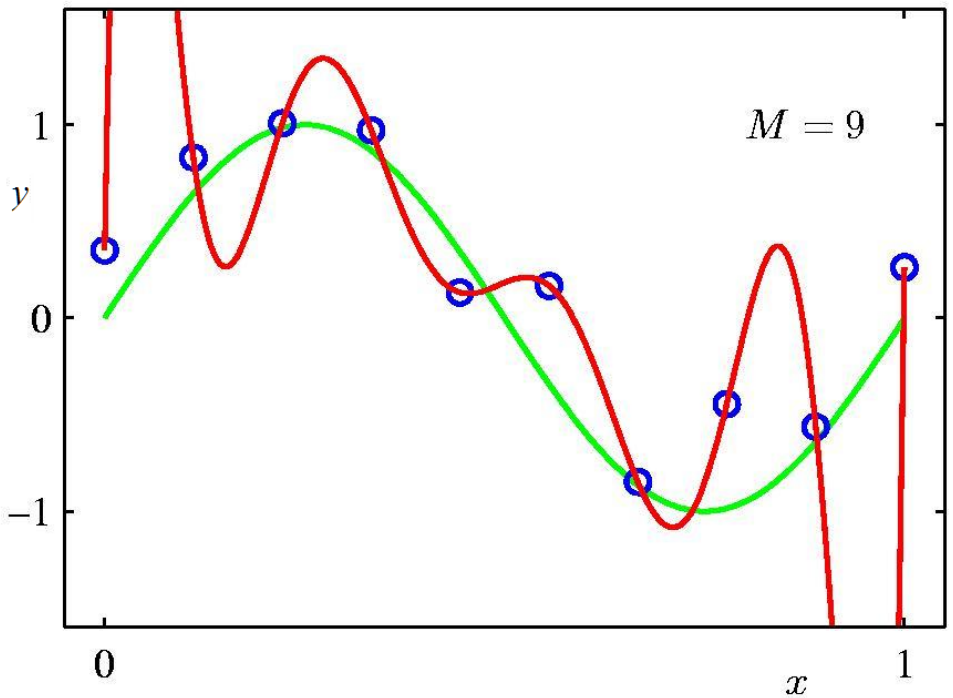
# Полином 1-ой степени



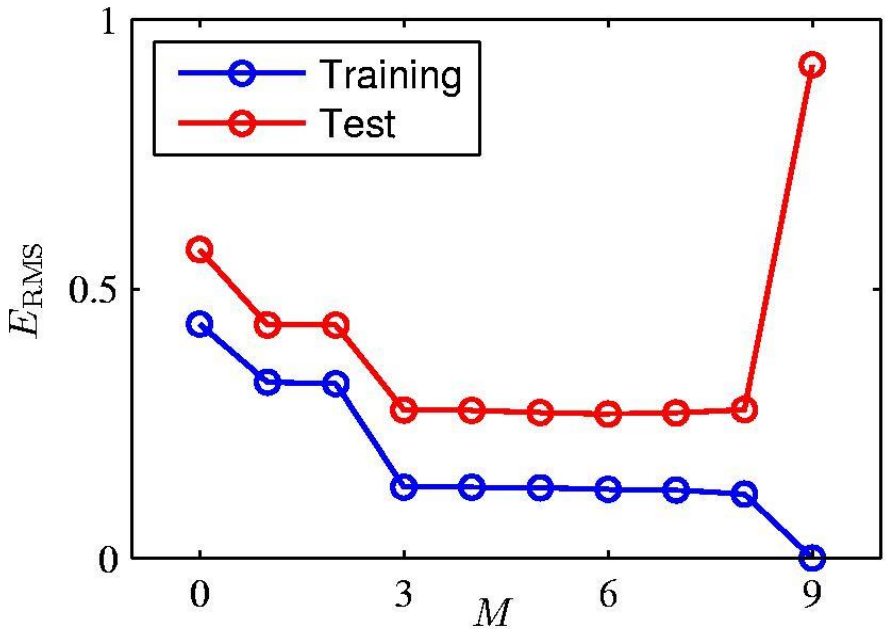
# Полином 3-ей степени



# Полином 9-ой степени



# Зависимость ошибки от степени полинома (переобучение)




Среднеквадратическая ошибка $E_{RMS} = \sqrt{2E(\mathbf{a})/n}$

# Регуляризация (сглаживание)

Штрафуем возможные большие коэффициенты полинома $f(x, w)$

$$E(w) = \frac{1}{2}\sum_{i=1}^{N}(f(x_i, w) - y_i)^2 + \frac{\lambda}{2}\|w\|^2$$
$$= \frac{1}{2}\sum_{i=1}^{N}(f(x_i, w) - y_i)^2 + \frac{\lambda}{2}\sum_{k=1}^{m} w_k^2$$

# Программная реализация в R

- https://cran.r-project.org/web/views/MachineLearning.html
- Package '**glmnet**', практически все модели: гребневая регрессия, метод Лассо, эластичные сети, логистическая регрессия
- Package '**lars**', метод Лассо
- Функция '**lm()**', обычная линейная регрессия
- Функция '**polyGC()**', полиномиальная регрессия

# Вопросы

?