In [2]:

```
import IPython.display
IPython.display.display_latex(IPython.display.Latex(filename="macros.tex"))
```

Регрессия

Имеем:
(X, Y) тренировочную выборку размера N, количество фич M
$|\hat{Y}| = \mathbb{R}$

knn

решение принимается как среднее от ответов соседей

decision trees критерий качества:

$$D = \frac{1}{l} \sum_{i=1}^{l} \left(y_i - \frac{1}{l} \sum_{i=1}^{l} y_i\right)^2$$

- l - число объектов в листе Минимизируем дисперсию вокруг среднего, ищем признаки, разбивающие выборку таким образом, что значения целевого признака в каждом листе примерно равны.

Как будем предсказывать? -> Среднее по всем объектам в листе.

Линейная регрессия

Имееем:

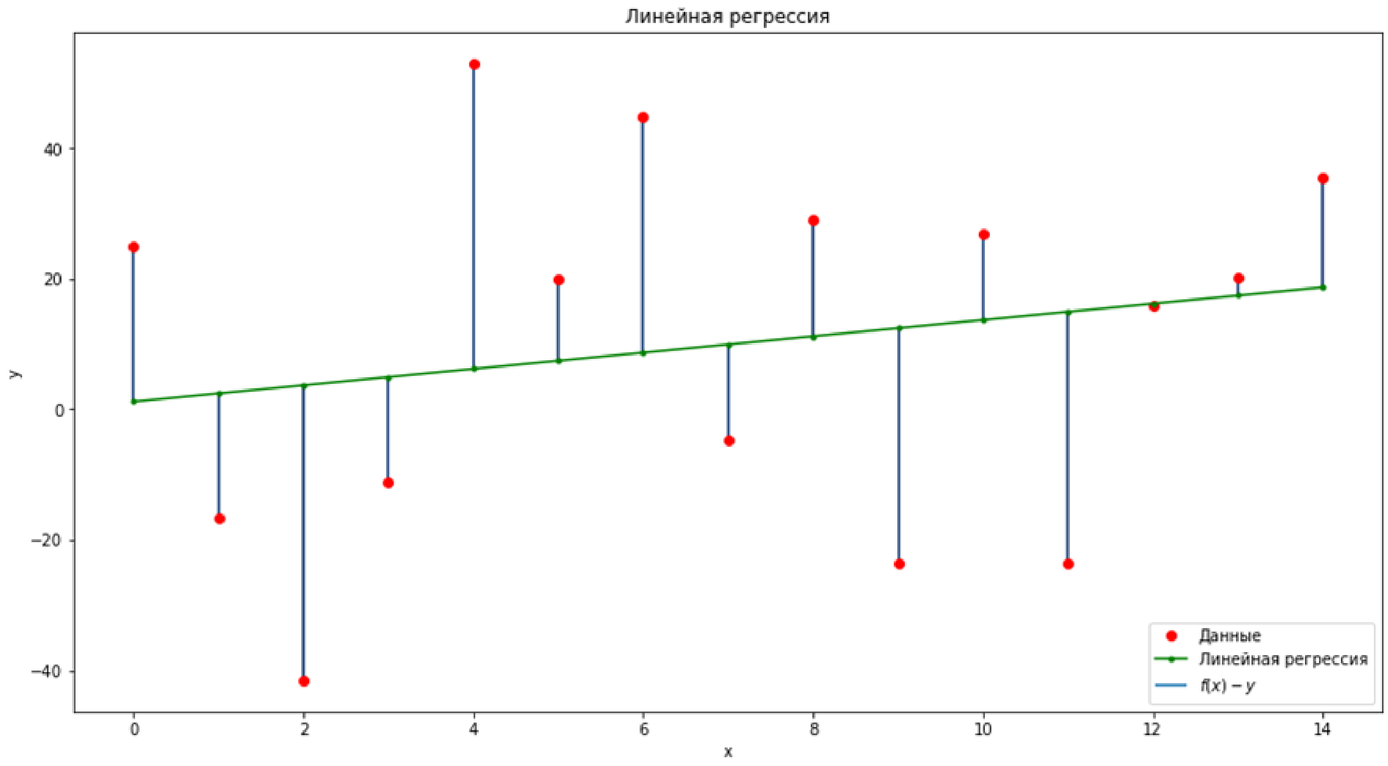
$$\left[X \,\middle|\, Ones\right] * \theta = Y_{predict}$$

Аналитическое решение

$$\theta = (X^T * X)^{-1} * X^T * Y$$

используя градиентный спуск

$$\theta^{(j+1)} := \theta^{(j)} - \gamma * (X^T * (X * \theta^{(j)} - Y))$$


Линейная регрессия
40
20
0
−20
−40
y
0
2
4
6
8
10
12
14
x
Данные
Линейная регрессия
$f(x) - y$

метрики

MAE (Mean absolute error):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\alpha(x_i) - y_i\right|$$

RMSE (Root mean squared error):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\alpha(x_i) - y_i\right)^2}$$

RMSE – будет меньше для модели с большим количеством малых ошибок, чем для модели с одной большой ошибкой, в то время как для этой же модели MAE может быть одинаковым.

MSE (mean squared error):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \big(\alpha(x_i) - y_i\big)^2$$

Relative absolute error (RAE) Относительная абсолютная ошибка

$$RAE = \frac{\sum_{i=1}^{N} \left|\alpha(x_i) - y_i\right|}{\sum_{i=1}^{N} \left|\overline{\alpha(x_i)} - y_i\right|}$$

Relative squared error (RSE) Относительная квадратичная ошибка

$$RSE = \frac{\sum_{i=1}^{N} \left(\alpha(x_i) - y_i\right)^2}{\sum_{i=1}^{N} \left(\overline{\alpha(x_i)} - y_i\right)^2}$$

Коэффициент детерминации, часто называемый $R2$, представляет прогностическую силу модели как значение между 0 и 1. Ноль означает, что модель случайна (ничего не объясняет); 1 означает, что есть идеальная подгонка.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(\alpha(x_i) - y_i\right)^2}{\sum_{i=1}^{N} \left(\overline{y} - y_i\right)^2}$$

метод максимального правдоподобия

$$p(y_1, y_2, \dots y_N \mid x_1, x_2, \dots x_N, \theta) \to max_\theta$$

$$W = \prod_i p(y_i \mid x_i, \theta) \to max_\theta$$

<u>Гипотеза</u>: $y(x_i) = \alpha(x_i, \theta) + \xi_i$, где ξ_i из $N(0, \sigma_i)$

$$W = \prod_i p(\alpha(x_i, \theta) + \xi_i) \to max_\theta$$

Плотность нормального распределения:

$$p_{N(\mu,\sigma)}(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\Big(- \frac{(x-\mu)^2}{2\sigma^2} \Big)$$

Итого:

$$-log(W) = const + \frac{1}{2}\sum_i \frac{1}{\sigma_i^2}(\xi)^2 = const + \frac{1}{2}\sum_i \frac{1}{\sigma^2}\big(\alpha(x,\theta) - y(x_i)\big)^2 \to min$$

Получили метод наименьших квадратов Ordinary Least Squares(OLS)!

backfitting algorithm

$$\alpha(x) = \sum_i \phi_i(x_i)$$

ϕ_i - не линейные функции.

1) зафиксировать стартовые ϕ_i
2) Пока ER $Q(\alpha, X) = \sum_i (\alpha(x_i) - y_i)^2$ не стабилизировался повторяем:
   for j:
      $z_i = y_i - \sum_{k=1, k \neq i}^{N} (\phi_k(x_k^i))$, for $i \in [1, N]$

      $\phi_j := argmin_{\phi_j} \sum_{i=1}^{N} (\phi_j(x_i) - z_i)^2$

      $Q_j := \sum_{i=1}^{N} (\phi_j(x_i) - z_i)^2$

algorithms with regularisation

Ordinary Least Squares:

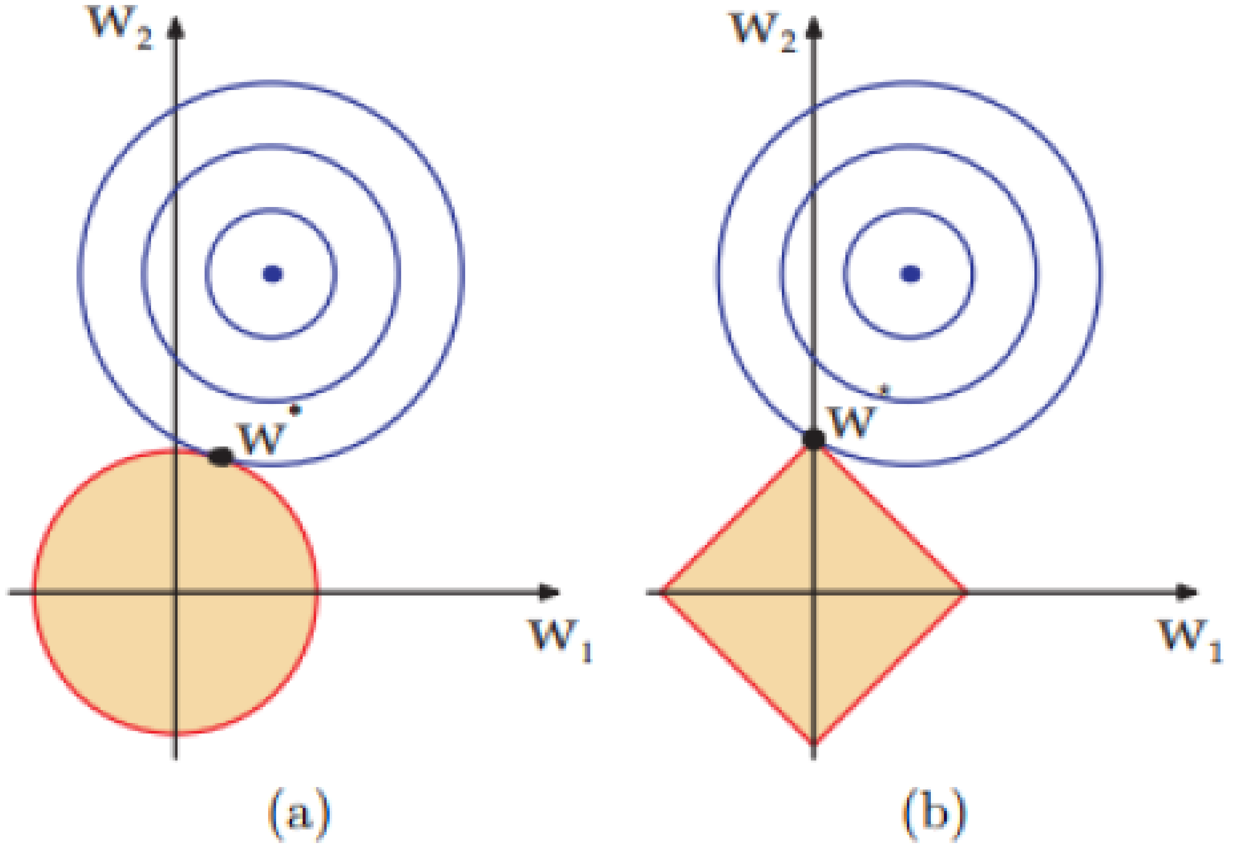
$$MSE \to min$$

LASSO(least absolute shrinkage and selection operator):

$$MSE + \lambda * \|\theta\| \to min$$

Ridge:

$$MSE + \lambda * \|\theta\|^2 \to min$$



w_2
w^*
w_1
(a)
w_2
w^*
w_1
(b)

L2 регуляризация как помошник в борьбе со скоррелированными признаками

С прошлых семинаров и лекций (либо можно прям явно вывести):

$$\nabla_\theta L(X, Y, \theta) = 0 = X^T X\theta - X^T Y$$

$$L_{reg} = L + 0.5(\lambda\theta)^T \lambda\theta$$

$$\nabla_\theta L_{reg}(X, Y, \theta) = \nabla_\theta((X\theta - Y)^T (X\theta - Y) + (\lambda\theta)^T \lambda\theta) =$$
$$X^T X\theta - X^T Y + 0.5\nabla_\theta(\lambda\theta)^T \lambda\theta = (X^T X + \lambda^T \lambda)\theta - X^T Y = 0$$

$$\theta = (X^T X + \lambda^T \lambda)^{-1} X^T Y$$

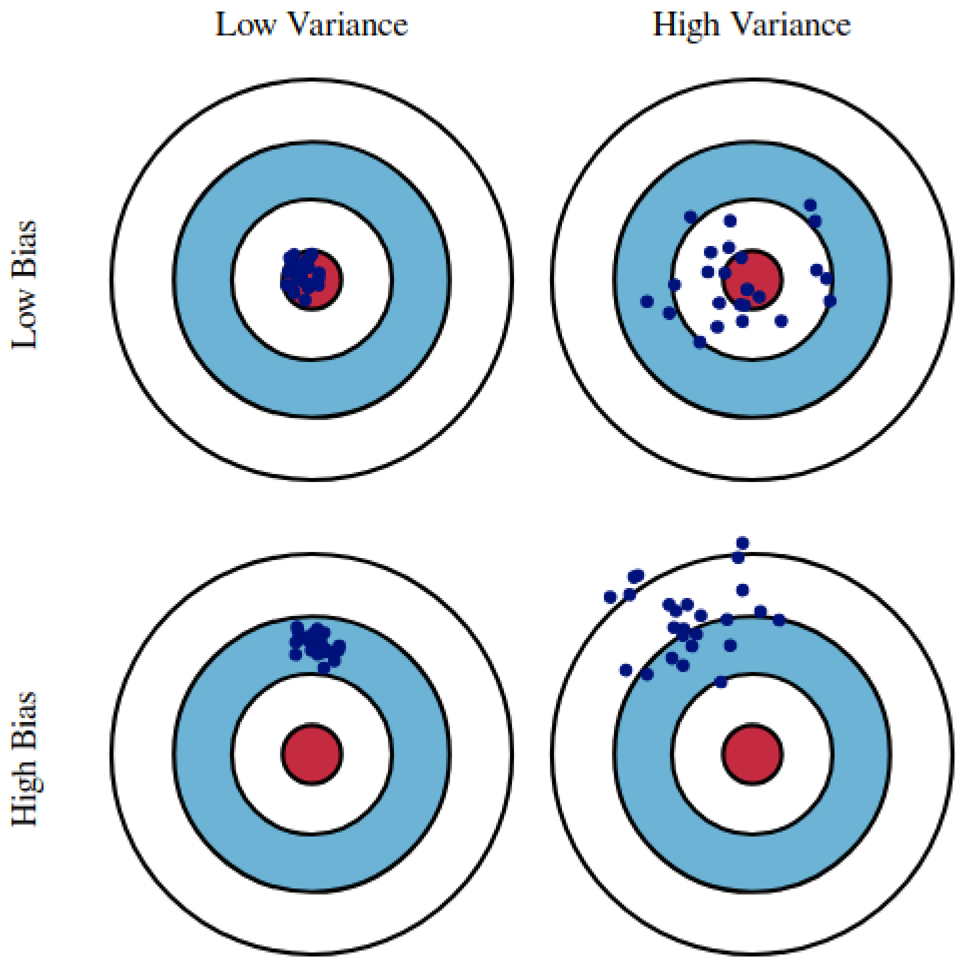
Bias - Variance TradeOff (Разложение ошибки на разброс и смещение)

$Y = f(X) + \epsilon, E[\epsilon] = 0, Var[\epsilon] = \sigma$

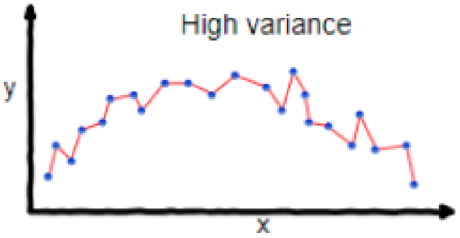
Истинная зависимость: f, наш алгоритм: $\hat{f}$. Давайте найдем среднее значение ошибки на задаче регресии.

$$E[(Y - \hat{f})^2] = E[Y^2 - 2Y\hat{f} + \hat{f}^2] = E[Y^2] - 2E[Y\hat{f}] + E[\hat{f}^2]$$
$$E[Y^2] = E[(f+\epsilon)^2] = f^2 + 2E[\epsilon]f + E[\epsilon^2] = f^2 + \sigma^2$$
$$E[Y\hat{f}] = E[(f+\epsilon)\hat{f}] = E[f\hat{f}] + E[\epsilon\hat{f}] = fE[\hat{f}]$$
$$E[\hat{f}^2] = Var[\hat{f}] + E[\hat{f}]^2$$

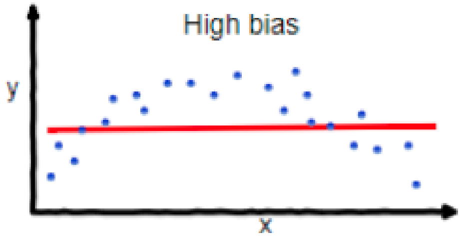
$$E[(Y - \hat{f})^2] = f^2 + \sigma^2 - 2fE[\hat{f}] + Var[\hat{f}] + E[\hat{f}]^2 = \sigma^2 + Var[\hat{f}] + (f - E[\hat{f}])^2$$
$$E[(Y - \hat{f})^2] = \sigma^2 + Var[\hat{f}] + Bias[\hat{f}]^2$$



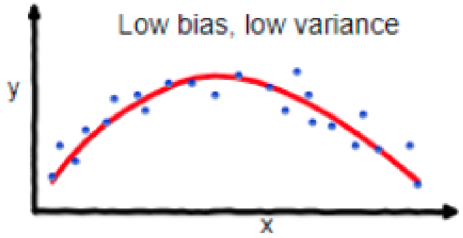
Low Variance
High Variance
Low Bias
High Bias



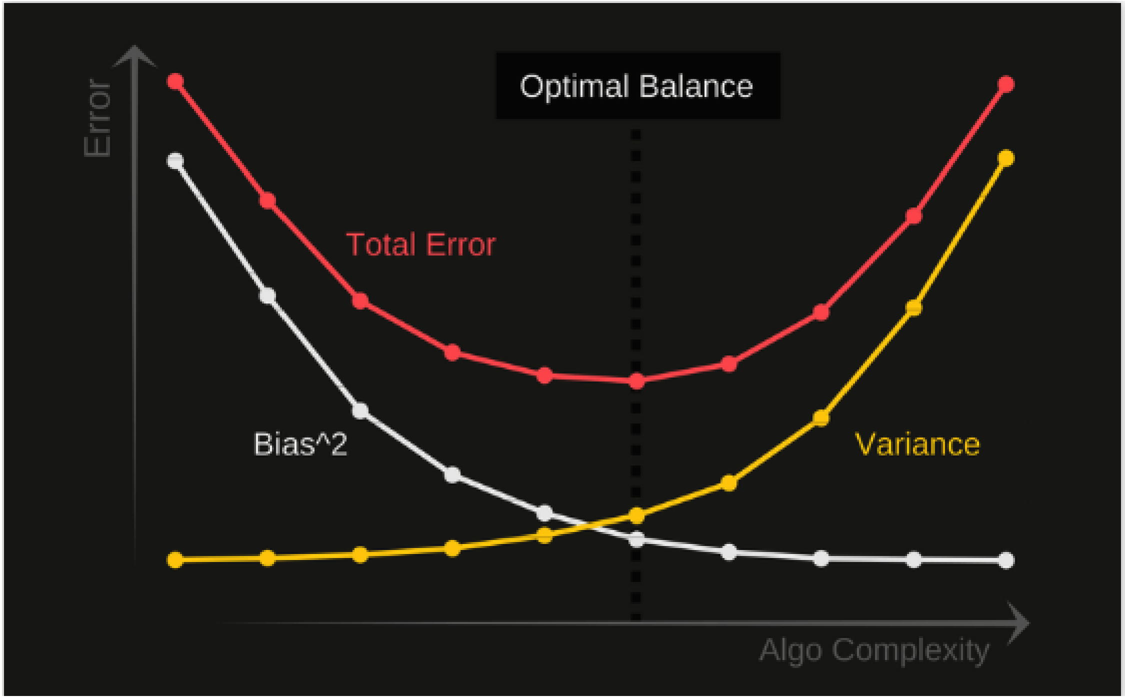
overfitting



underfitting



Good balance



Optimal Balance
Error
Total Error
Bias^2
Variance
Algo Complexity