Методы регуляризации в машинном обучении

Фатахов Т. М.
Сивошенко А. Ю.
Группа 5030102/20201

Зачем нужна регуляризация

Основная проблема: переобучение модели на тренировочных данных.

Причины:

- слишком гибкая модель;
- много признаков;
- шум в данных;
- коррелированные признаки.

Цель регуляризации:
ограничить сложность модели, чтобы она лучше работала на новых данных.

Основная идея регуляризации

Регуляризация добавляет в функцию потерь штраф за «сложность» модели.

Общая формула: $Loss = Error + \alpha \cdot Penalty$

- α — степень penalization
- Penalty — мера «размера» весов
- результат — более стабильная, устойчиво обученная модель

Компромисс Bias–Variance

- При регуляризации ошибка смещения (bias) растёт
- Но дисперсия модели (variance) снижается
- Итоговая ошибка на тесте часто уменьшается

Регуляризация = управляемый компромисс между bias и variance.

L2-регуляризация (Ridge)

Penalty: сумма квадратов весов.

Эффект:

- сглаживает коэффициенты;
- делает модель стабильнее;
- хорошо работает при мультиколлинеарности;
- *не* зануляет признаки полностью.

Использование: когда признаки сильно коррелированы.

L1-регуляризация (Lasso)

Penalty: сумма модулей весов.

Эффект:

- делает модель разреженной;
- может занулять ненужные признаки;
- выполняет отбор признаков автоматически;
- может быть нестабильной при коррелированных данных.

Использование: когда нужна модель с малым числом важных признаков.

Elastic Net

Комбинация L1 и L2:

$$Penalty = \alpha(\beta \sum |w| + (1 - \beta) \sum w$$

Эффект:

- устойчивость Ridge + разреженность Lasso;
- лучше работает, когда много коррелированных признаков.

Использование: практический универсальный вариант.

Регуляризация в других моделях

Деревья решений:

- ограничение глубины
- минимальный размер листа
- pruning

Нейронные сети:

- Dropout
- BatchNorm
- Early stopping

Логистическая регрессия: L1 и L2 аналогично линейной.

Настройка численного эксперимента

Мы генерируем данные с:

- линейной зависимостью;
- шумом;
- двумя сильно коррелированными признаками X1 и X2.

Сравниваем модели:

- Linear Regression
- Ridge
- Lasso
- Elastic Net

Код: генерация данных

```python
np.random.seed(42)
n = 200

# Два сильно коррелированных признака
X1 = np.random.randn(n)
X2 = X1 + np.random.normal(scale=0.1, size=n)   # почти копия X1

# Ещё один независимый признак
X3 = np.random.randn(n)

# Целевая переменная
y = 3*X1 - 2*X2 + 0.5*X3 + np.random.normal(scale=0.5, size=n)

# Матрица признаков
X = np.vstack([X1, X2, X3]).T

# Разделение на train / test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
```

Код: обучение моделей

```python
models = {
    "Linear": LinearRegression(),
    "Ridge": Ridge(alpha=1.0),
    "Lasso": Lasso(alpha=0.1),
    "ElasticNet": ElasticNet(alpha=0.1, l1_ratio=0.5)
}

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    results[name] = {
        "Coefficients": model.coef_,
        "MSE": mean_squared_error(y_test, y_pred)
    }

```

Сравнение коэффициентов

Таблица коэффициентов:

- Linear: коэффициенты нестабильны, «скачут»
- Ridge: коэффициенты сглажены
- Lasso: занулило признак X2
- Elastic Net: баланс между L1 и L2

Вывод: регуляризация уменьшает нестабильность весов.

Таблица Коэффициентов

===	Коэффициенты моделей		===	
	Linear	Ridge	Lasso	ElasticNet
X1	4.240840	1.954071	0.906470	0.755596
X2	-3.158143	-0.920198	0.000000	0.166973
X3	0.554793	0.542790	0.426735	0.456483

Сравнение качества моделей (MSE)

Почему Ridge лучший:

- сопротивляется мультиколлинеарности
- не удаляет нужные признаки (как Lasso)
- снижает variance сильнее всего

Linear: 0.2525

Ridge: 0.2155 ← лучшее

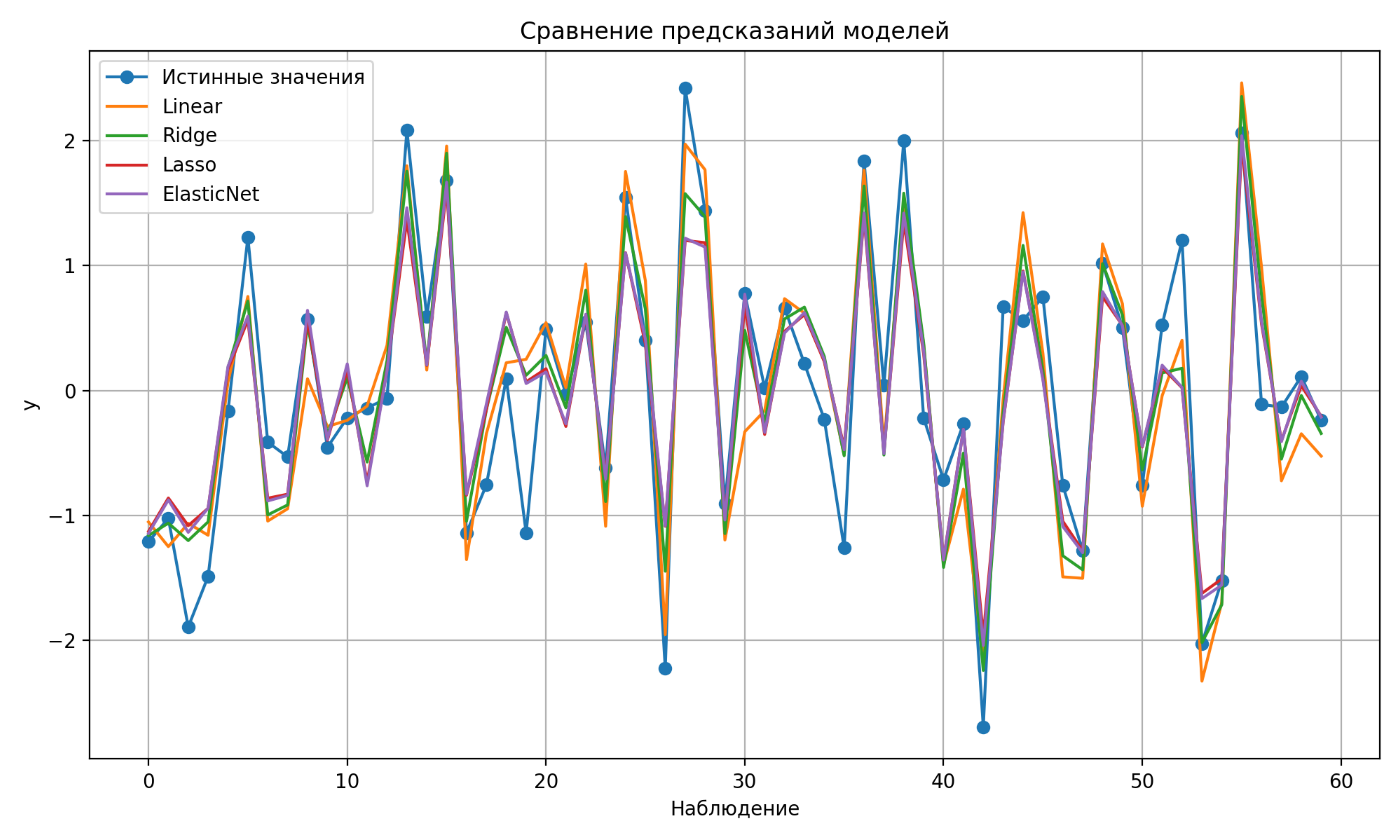
Lasso: 0.2608

ElasticNet: 0.2553

График предсказаний

Интерпретация:

- Ridge наиболее близок к истинным значениям
- Linear часто отклоняется
- Lasso иногда теряет качество из-за зануления X2
- Elastic Net — компромисс



Численный пример вручную: OLS против Ridge

Данные и модель:

- Наблюдения:

  $(x_1, y_1) = (1, 2)$,

  $(x_2, y_2) = (2, 3)$

- Линейная модель:

  $y = \beta_0 + \beta_1 x$

Матричная запись:

$X = [[1\ \ 1],$

$\quad [1\ \ 2]],$

$y = [2, 3]^T$

Обычная линейная регрессия (OLS):

- Оценка:

  $\hat{\beta} = (X^TX)^{-1} X^T y$

- $X^TX = [[2\ \ 3],$

  $\quad [3\ \ 5]]$

- $(X^TX)^{-1} = [[\ 5\ \ -3],$

  $\quad [-3\ \ \ 2]]$

- $X^Ty = [5, 8]^T$

Вычисление $\hat{\beta}$:

$\hat{\beta} = [[\ 5\ \ -3],$

$\quad [-3\ \ \ 2]] \cdot [5, 8]^T$

$= [1, 1]^T$

Итог:

- $\hat{\beta}_0(OLS) = 1$

- $\hat{\beta}_1(OLS) = 1$

- Предсказания:

  $\hat{y}_1 = 1 + 1 \cdot 1 = 2$

  $\hat{y}_2 = 1 + 1 \cdot 2 = 3$

- Ошибки:

  $e_1 = 2 - 2 = 0$

  $e_2 = 3 - 3 = 0$

- $MSE_{(OLS)} = (0^2 + 0^2) / 2 = 0$

Ridge-регрессия ($\lambda = 1$):

- Матрица:

  $X^TX + \lambda I = [[2\ \ 3],$

  $\quad [3\ \ 5]] + [[1\ \ 0],$

  $\quad [0\ \ 1]]$

  $= [[3\ \ 3],$

  $\quad [3\ \ 6]]$

- $\det = 3 \cdot 6 - 3 \cdot 3 = 9$

- $(X^TX + \lambda I)^{-1} = (1/9) \cdot [[\ 6\ \ -3],$

  $\quad [-3\ \ \ 3]]$

Численный пример вручную: OLS против Ridge

Вычисление $\hat{\beta}^R$:

$$\hat{\beta}^R = (X^T X + \lambda I)^{-1} X^T y$$

$$= (1/9) \cdot \begin{bmatrix} 6 & -3 \\ -3 & 3 \end{bmatrix} \cdot [5, 8]^T$$

$$= (1/9) \cdot [6, 9]^T$$

$$= [2/3, 1]^T$$

Итог:

- $\hat{\beta}_0(\text{Ridge}) = 2/3$
- $\hat{\beta}_1(\text{Ridge}) = 1$
- Предсказания:

  $\hat{y}_1^R = 2/3 + 1 \cdot 1 = 5/3 \approx 1.67$

  $\hat{y}_2^R = 2/3 + 1 \cdot 2 = 8/3 \approx 2.67$
- Ошибки:

  $e_1^R = 2 - 5/3 = 1/3$

  $e_2^R = 3 - 8/3 = 1/3$
- $MSE_{(Ridge)} = ((1/3)^2 + (1/3)^2)/2 = (2/9)/2 = 1/9 \approx 0.11$

Сравнение:

- $MSE_{(OLS)} = 0$
- $MSE_{(Ridge)} = 1/9 \approx 0.11$

Эксперимент 1: мультиколлинеарность — выигрывает Ridge

Постановка эксперимента:

- Цель: показать влияние сильной мультиколлинеарности на линейную регрессию.

- Число наблюдений: $n = 200$

- Признаки:

  $X_1 \sim N(0, 1)$

  $X_2 = X_1 + \varepsilon,\ \varepsilon \sim N(0, 0.01^2)$ ($X_2 \approx X_1$, сильная мультиколлинеарность)

  $X_3 \sim N(0, 1)$, независимый

- Истинные коэффициенты:

  $\beta\_true = (3.0, -2.0, 0.5)$

- Целевая переменная:

  $y = 3 \cdot X_1 - 2 \cdot X_2 + 0.5 \cdot X_3 + \xi,\ \xi \sim N(0, 1)$

```
def experiment_multicollinear():
    rng = np.random.RandomState(42)
    n_samples = 200

    X1 = rng.normal(0, 1, size=n_samples)
    X2 = X1 + rng.normal(0, 0.01, size=n_samples)  # сильная мультиколлинеарность
    X3 = rng.normal(0, 1, size=n_samples)

    X = np.vstack([X1, X2, X3]).T
    feature_names = ["X1", "X2", "X3"]

    true_beta = np.array([3.0, -2.0, 0.5])
    y = X @ true_beta + rng.normal(0, 1.0, size=n_samples)

    run_and_report(
        name="Эксперимент 1: сильная мультиколлинеарность (Ridge выигрывает)",
        X=X,
        y=y,
        feature_names=feature_names,
        plot_filename="regularization_multicollinear.png",
    )
```

Эксперимент 1: мультиколлинеарность — выигрывает Ridge

Истинные коэффициенты:

X1 = 3.0

X2 = -2.0

X3 = 0.5

Оценённые коэффициенты (по результатам запуска):

Коэффициенты (без свободного члена):

	Linear	Ridge	Lasso	ElasticNet
X1	26.328	0.726	0.998	0.537
X2	-25.163	0.396	0.000	0.491
X3	0.610	0.598	0.483	0.512

Свободные члены (intercept):

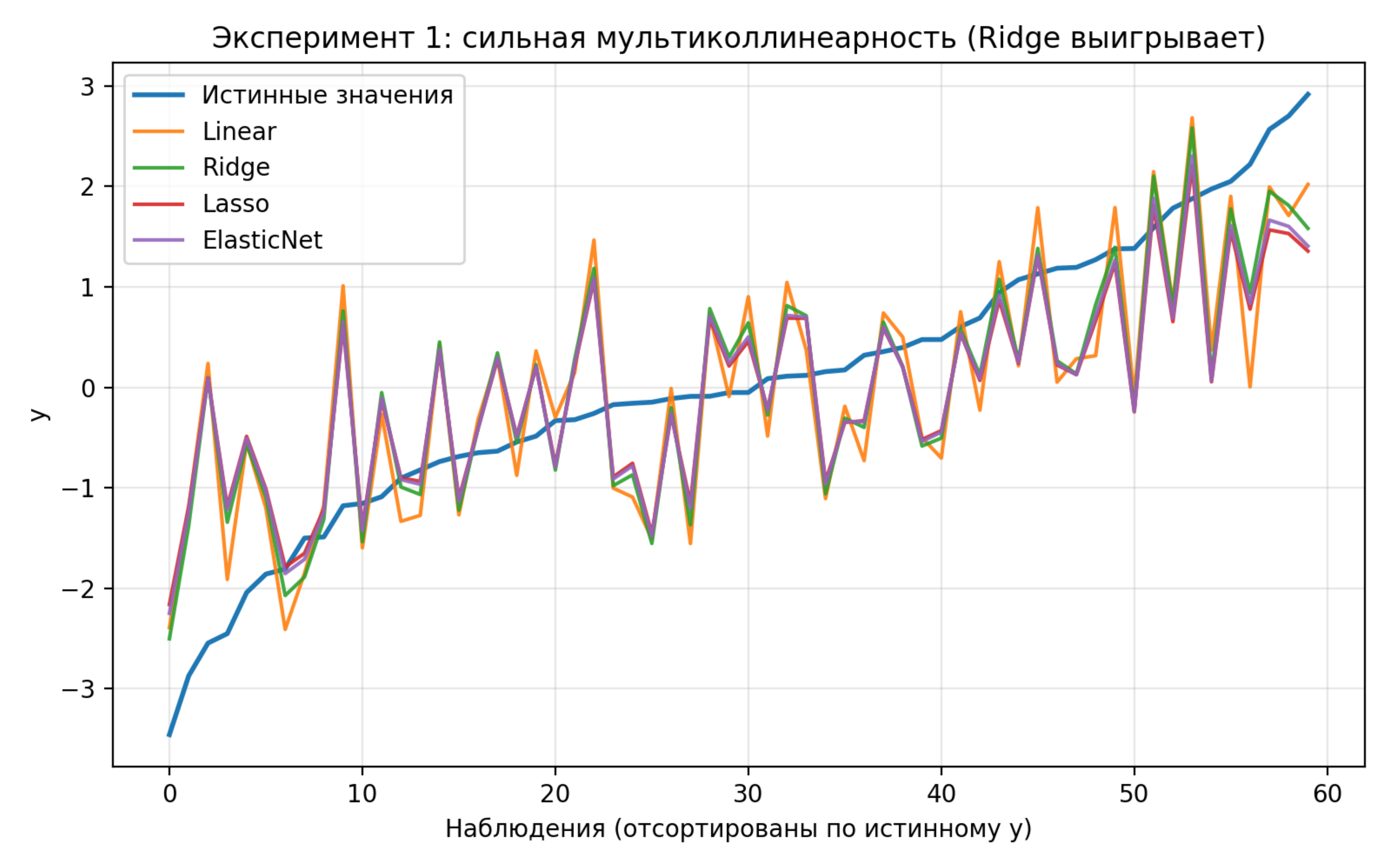
- Linear ≈ 0.028
- Ridge ≈ 0.008
- Lasso ≈ -0.013
- ElasticNet ≈ -0.008

MSE на тестовой выборке:

- Ridge = 0.8750
- ElasticNet = 0.8783
- Lasso = 0.8864
- Linear = 1.0099

Число ненулевых коэффициентов:

- Linear = 3
- Ridge = 3
- Lasso = 2
- ElasticNet = 3



Эксперимент 1: сильная мультиколлинеарность (Ridge выигрывает)
Истинные значения
Linear
Ridge
Lasso
ElasticNet
y
Наблюдения (отсортированы по истинному y)

Эксперимент 2: разрежная high-dimensional модель — выигрывает Lasso

Постановка эксперимента:

- Цель: показать преимущество Lasso при разрежной истинной модели и большом числе признаков.

- Число наблюдений: $n = 40$

- Число признаков: $p = 40$

- Число информативных признаков: $k = 5$

- Признаки:

  $X \in \mathbb{R}^{40\times40}$, элементы $\sim N(0, 1)$

- Истинные коэффициенты:

  $\beta_1,\ldots,\beta_5$ случайны в $[0.5, 2.0]$ по модулю, со случайным знаком

  $\beta_6,\ldots,\beta_{40} = 0$

- Целевая переменная:

  $y = X\beta + \xi,\ \xi \sim N(0, 1)$

```python
def experiment_sparse_highdim():
    rng = np.random.RandomState(42)
    n_samples = 40
    n_features = 40
    n_informative = 5

    X = rng.normal(size=(n_samples, n_features))

    beta = np.zeros(n_features)
    beta[:n_informative] = (
        rng.uniform(0.5, 2.0, size=n_informative)
        * rng.choice([-1, 1], size=n_informative)
    )

    y = X @ beta + rng.normal(0, 1.0, size=n_samples)

    feature_names = [f"x{i + 1}" for i in range(n_features)]

    run_and_report(
        name="Эксперимент 2: разрежная high-dimensional модель (Lasso выигрывает)",
        X=X,
        y=y,
        feature_names=feature_names,
        plot_filename="regularization_sparse.png",
    )
```

Эксперимент 2: разрежная high-dimensional модель — выигрывает Lasso

Истинные коэффициенты (первые 10):

x1 ≈ -1.710

x2 ≈ -1.189

x3 ≈ 0.578

x4 ≈ 1.679

x5 ≈ -0.802

x6 = 0.000

x7 = 0.000

x8 = 0.000

x9 = 0.000

x10 = 0.000

Оценённые коэффициенты (фрагмент, первые 10 признаков):

	Linear	Ridge	Lasso	ElasticNet
x1	-1.073	-1.005	-1.361	-1.233
x2	-0.950	-0.841	-1.286	-1.172
x3	-0.389	-0.291	0.000	0.000
x4	0.724	0.757	0.865	0.791
x5	-0.838	-0.870	-0.837	-0.889
x6	-0.312	-0.318	-0.247	-0.226
x7	-0.499	-0.413	-0.003	-0.178
x8	0.638	0.563	0.460	0.477
x9	0.426	0.352	0.000	0.096
x10	-0.411	-0.289	-0.006	-0.000

Свободные члены (intercept):

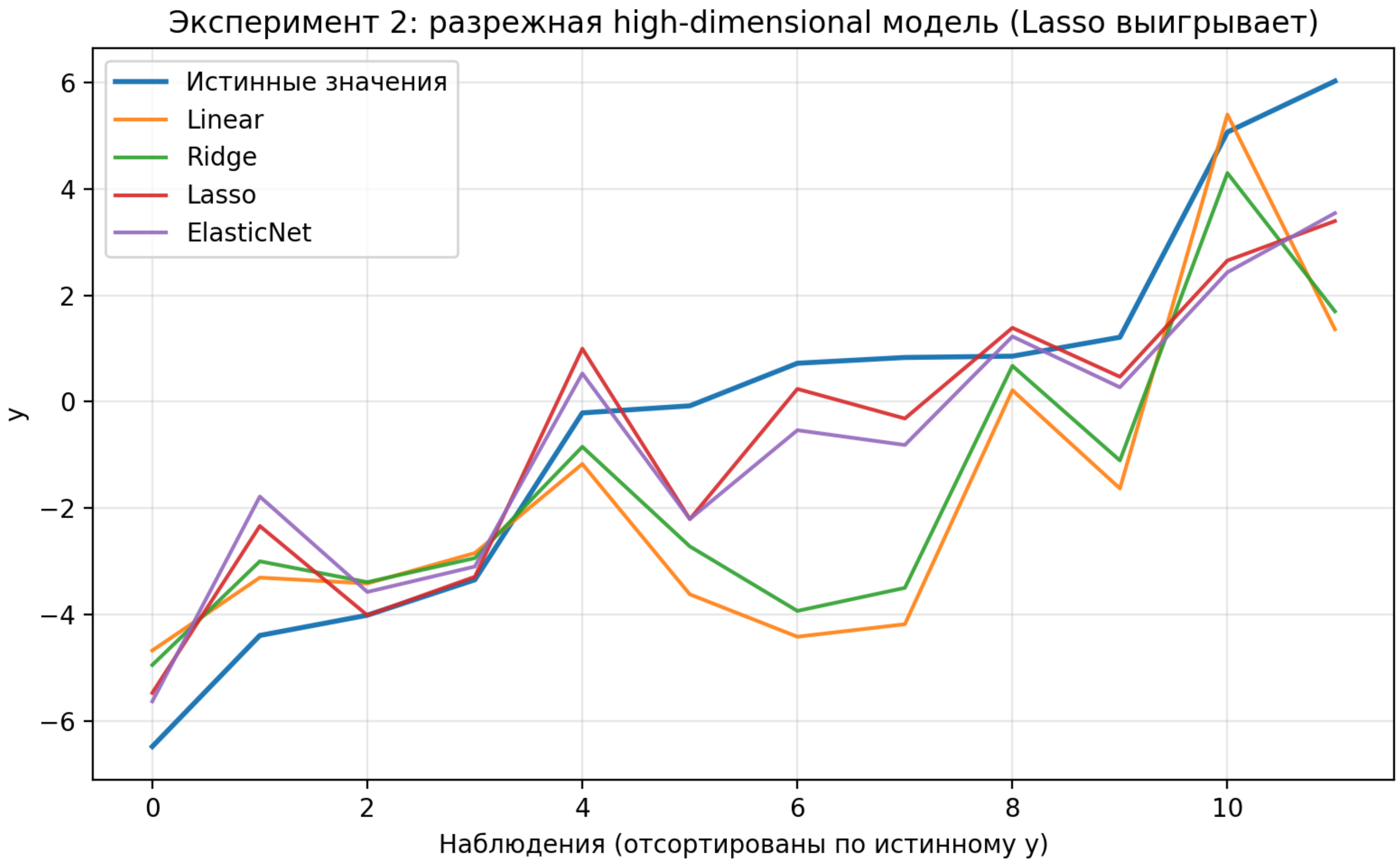
- Linear ≈ -0.455
- Ridge ≈ -0.418
- Lasso ≈ 0.084
- ElasticNet ≈ 0.017

MSE на тестовой выборке:

- Lasso = 2.2002
- ElasticNet = 2.6088
- Ridge = 6.4535
- Linear = 8.3788

Число ненулевых коэффициентов:

- Linear = 40
- Ridge = 40
- Lasso = 19
- ElasticNet = 24


Эксперимент 2: разрежная high-dimensional модель (Lasso выигрывает)
Истинные значения
Linear
Ridge
Lasso
ElasticNet
y
6
4
2
0
−2
−4
−6
0
2
4
6
8
10
Наблюдения (отсортированы по истинному y)

Сводная таблица: когда какой метод регуляризации выигрывает

Сценарий 1: численный пример вручную (2 точки, 1 признак)

- Данные идеальны, лежат точно на прямой.

- Мультиколлинеарности нет, признаков мало, шума нет.

- Результаты:

  MSE(OLS) = 0

  MSE(Ridge) = 1/9 ≈ 0.11

- Лучший метод: OLS (без регуляризации).

Сценарий 2: мультиколлинеарность (Эксперимент 1)

- Данные: $X_2 \approx X_1$ (сильная мультиколлинеарность), 3 признака.

- Результаты MSE:

  Ridge = 0.8750

  ElasticNet = 0.8783

  Lasso = 0.8864

  Linear = 1.0099

- Лучший метод: Ridge (L2-регуляризация), близок по качеству Elastic Net.

Сценарий 3: разрежная high-dimensional модель (Эксперимент 2)

- Данные: 40 признаков, реально важны только 5, остальные шум.

- Результаты MSE:

  Lasso = 2.2002

  ElasticNet = 2.6088

  Ridge = 6.4535

  Linear = 8.3788

- Число ненулевых коэффициентов:

  Linear = 40

  Ridge = 40

  Lasso = 19

  ElasticNet = 24

- Лучший метод: Lasso (L1-регуляризация) по MSE и по разреженности.

Общие выводы

Регуляризация — ключевой инструмент борьбы с переобучением

Ridge стабилен при коррелированных признаках

Lasso делает автоматический отбор признаков

Elastic Net сочетает плюсы обоих

В нашем эксперименте Ridge показал лучшую точность