

Глава 1

Петафлопс-дни

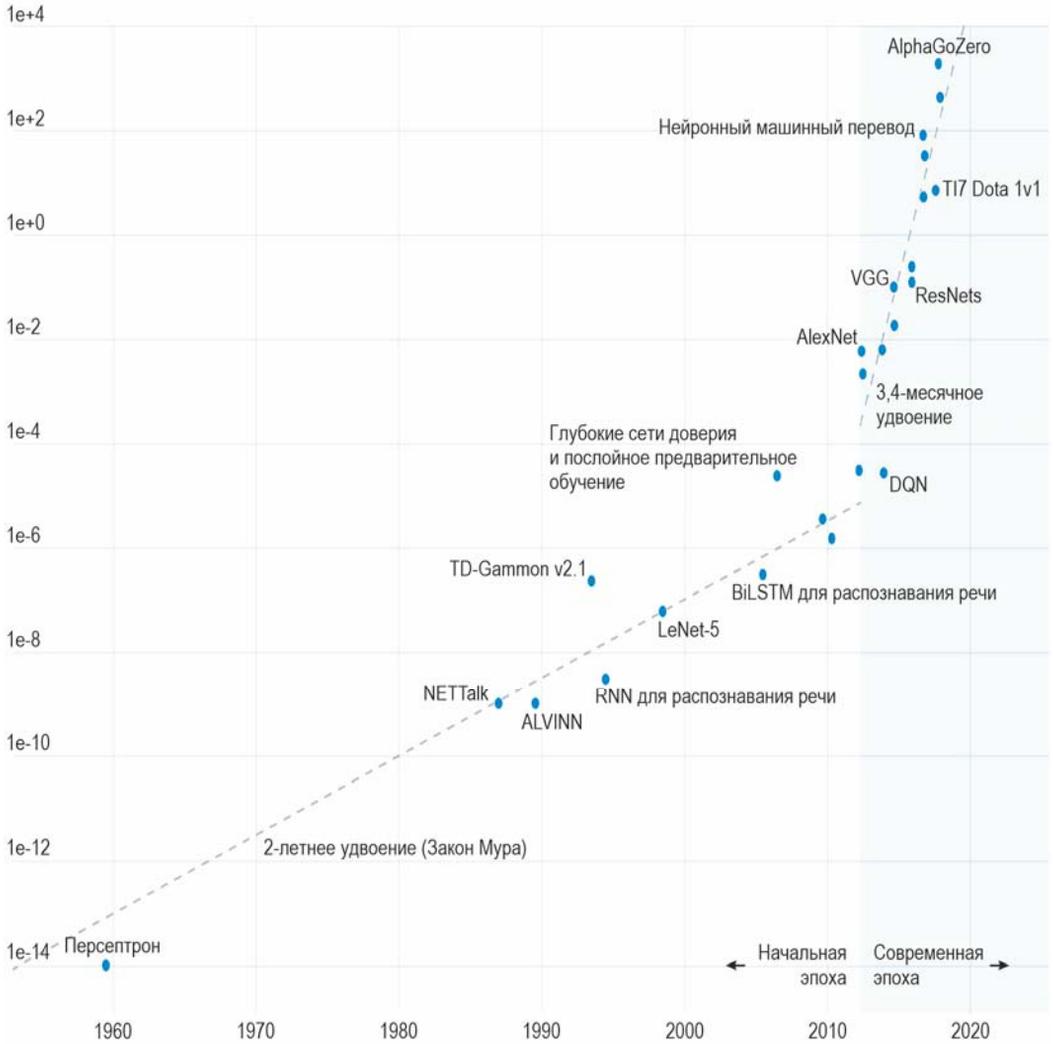


Рис. 1.1. График, демонстрирующий затруднения при настройке гиперпараметров из-за неуклонного роста вычислительных затрат на обучение больших нейросетей



Рис. 1.2. Структура задачи оптимизации «черного ящика», где мы неоднократно запрашиваем значения функции в разных местах, чтобы найти глобальный оптимум

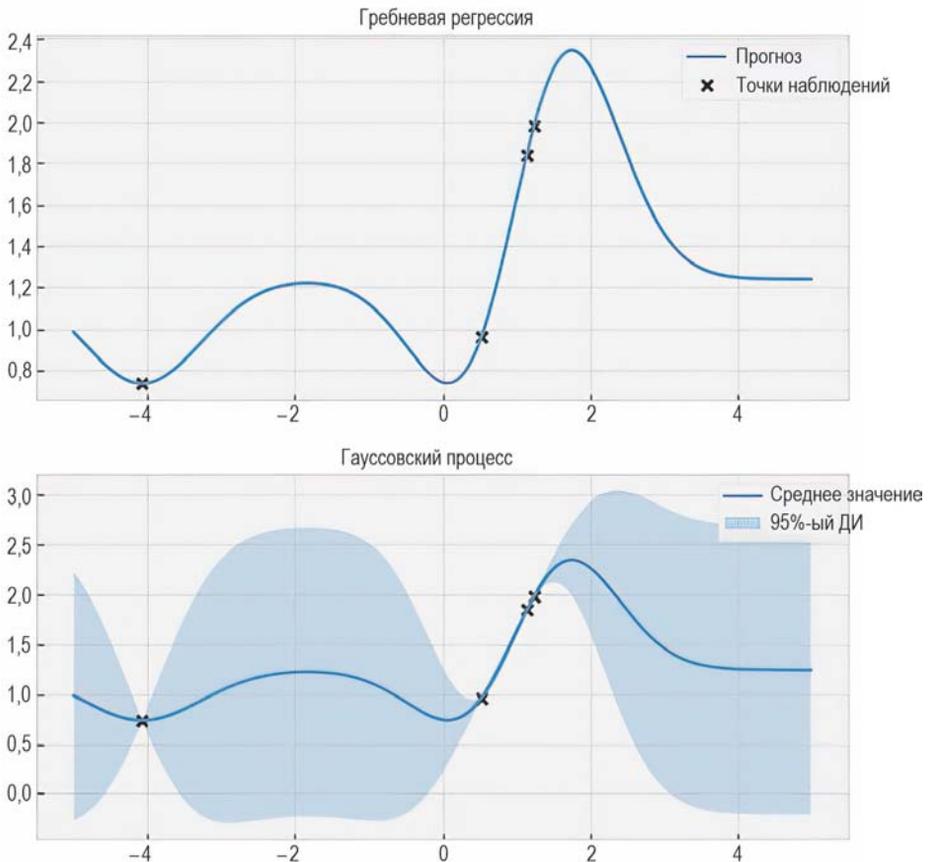


Рис. 1.3. Не-байесовские модели, такие как гребневые регрессоры, дают точечные оценки, в то время как гауссовские процессы создают распределения вероятностей в качестве прогнозов: таким образом, гауссовские процессы предлагают калиброванную количественную оценку неопределенности, которая является важным фактором при принятии решений с высоким риском

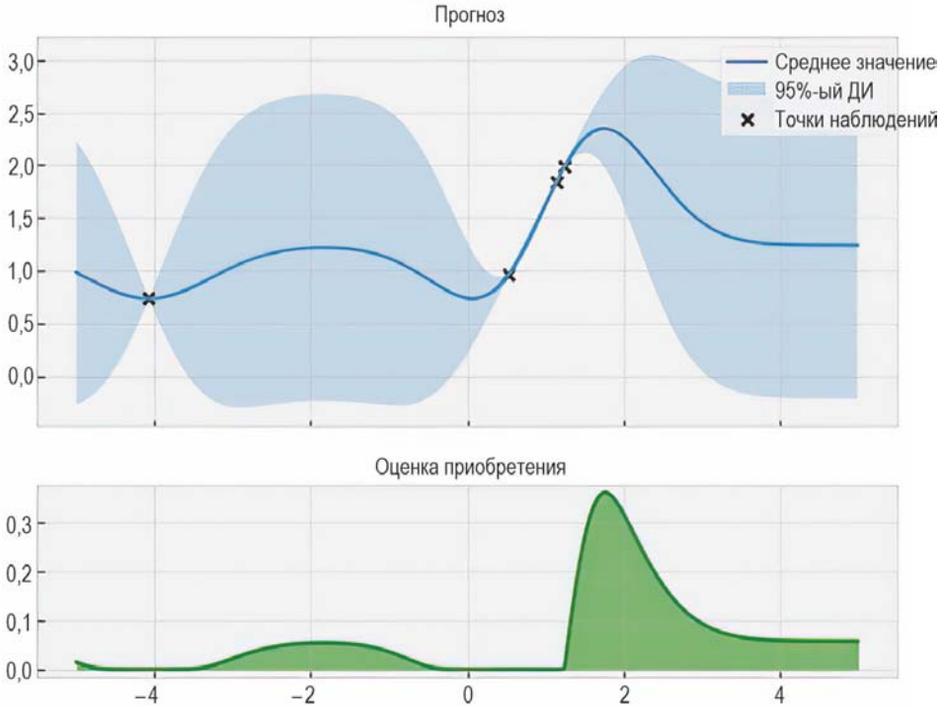


Рис. 1.4. Политика оценивает каждую отдельную точку данных по ее полезности для определения глобального оптимума: предпочтение отдается высокой прогностической ценности (где выигрыш более вероятен), а также высокой неопределенности (когда выигрыш потенциально может быть большим)

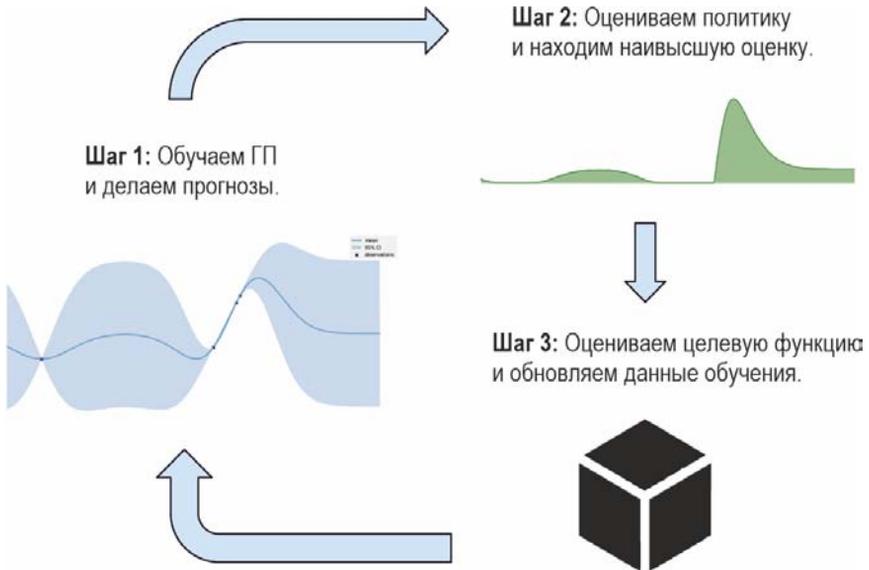


Рис. 1.5. Цикл БО, сочетающий в себе ГП для моделирования и политику принятия решений: этот полный рабочий процесс теперь можно использовать для оптимизации функций «черного ящика»

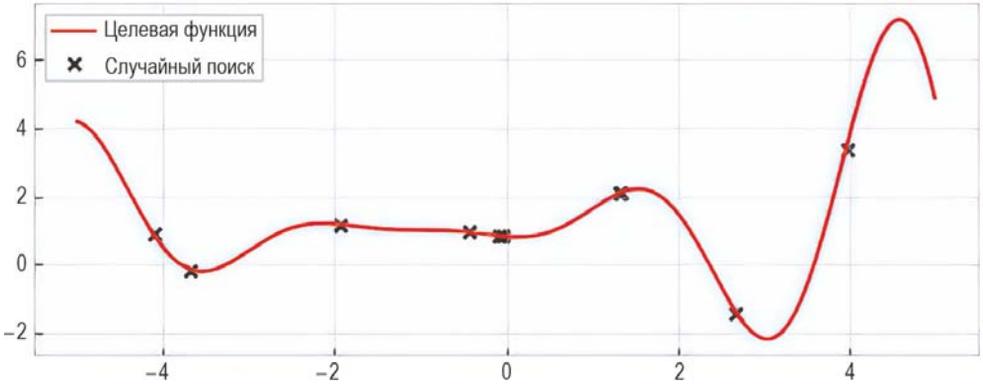


Рис. 1.6. Целевая функция, которую необходимо максимизировать, где случайный поиск тратит ресурсы на бесперспективные регионы

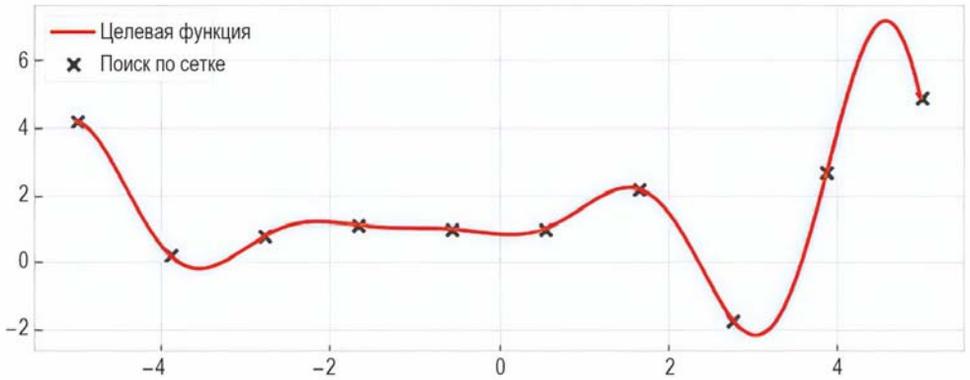


Рис. 1.7. Поиск по сетке по-прежнему неэффективен для сужения подходящего региона

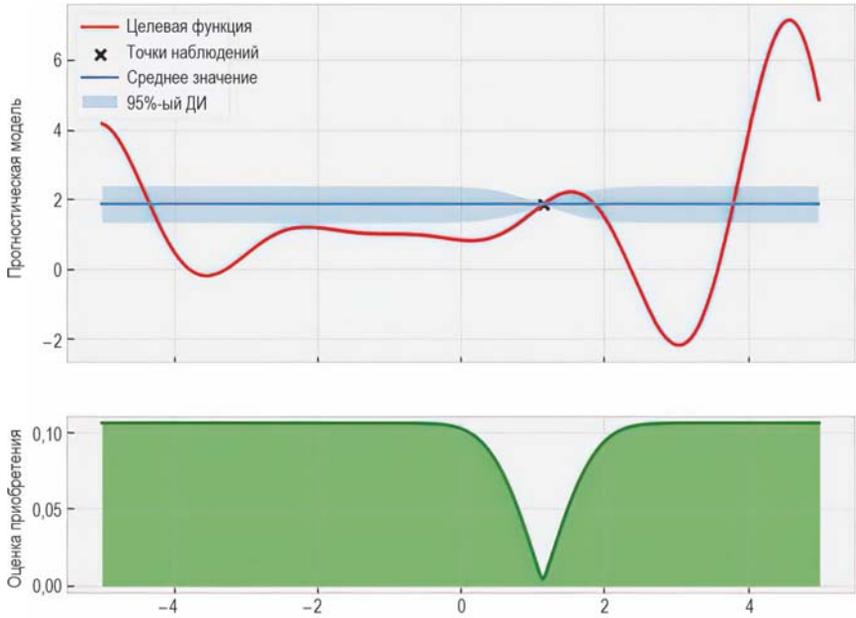


Рис. 1.8. Начало БО, аналогичное началу случайного поиска

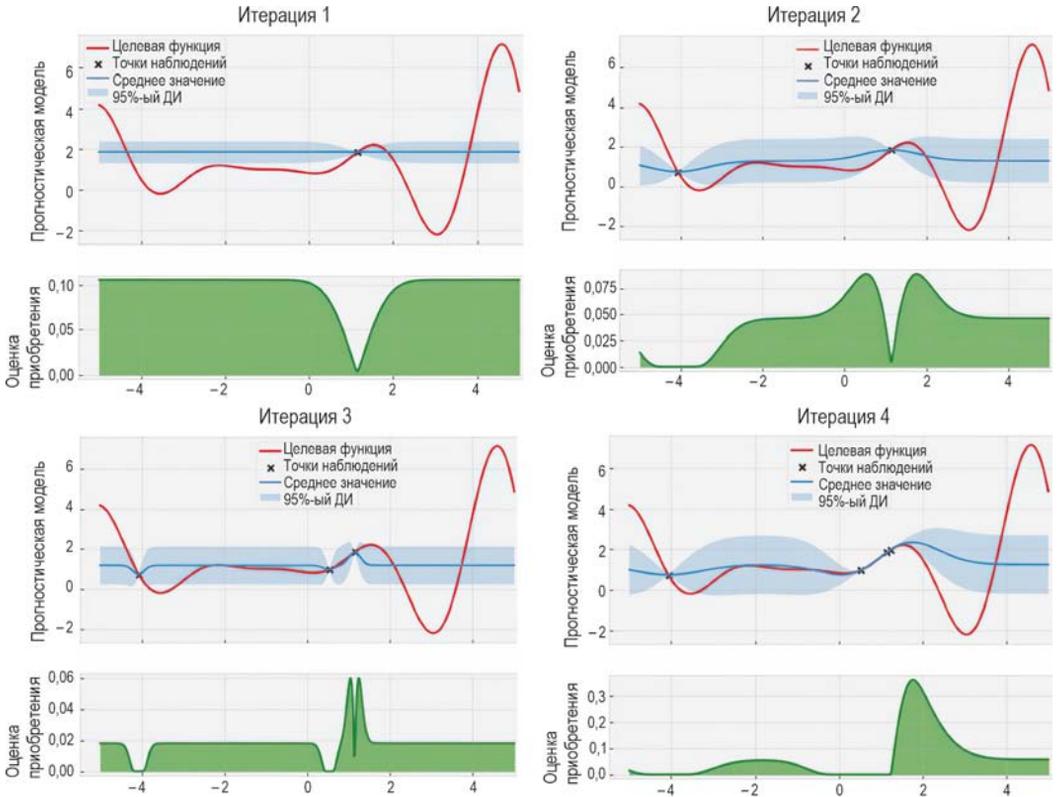


Рис. 1.9. После четырех запросов определен второй лучший оптимум

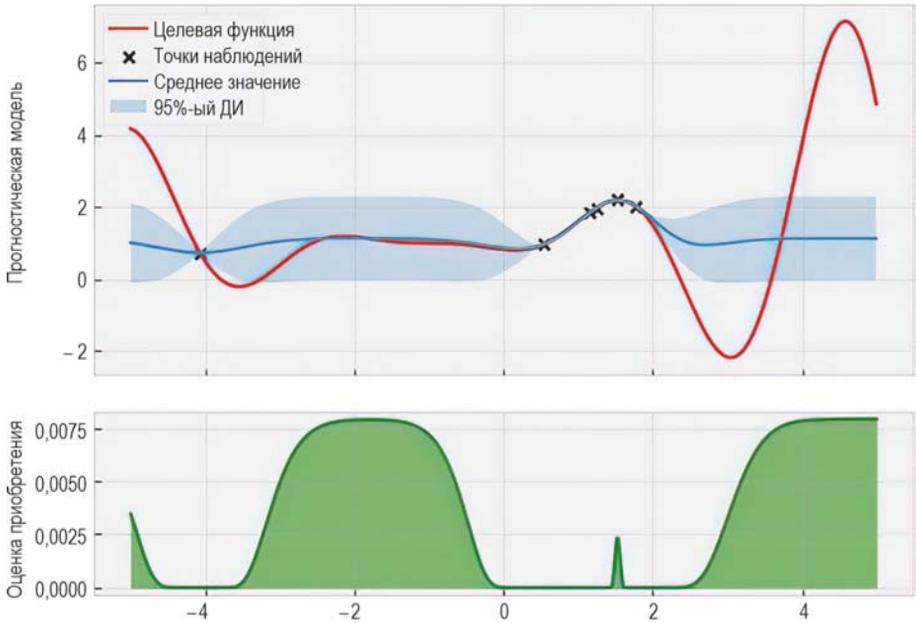


Рис. 1.10. После достаточного изучения локального оптимума политика предлагает рассмотреть другие области

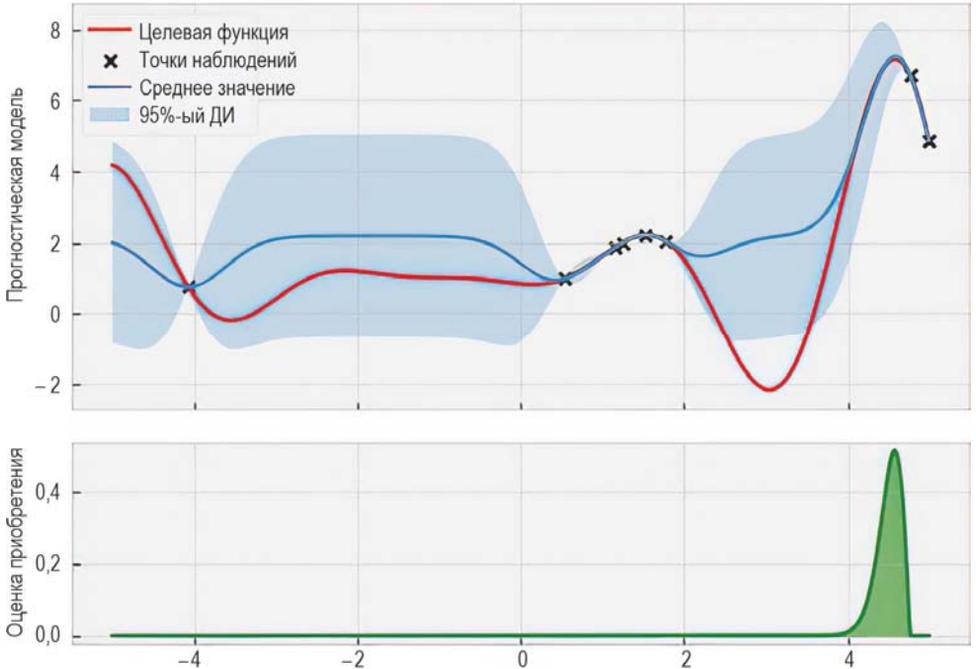


Рис. 1.11. БО успешно игнорирует большую область слева

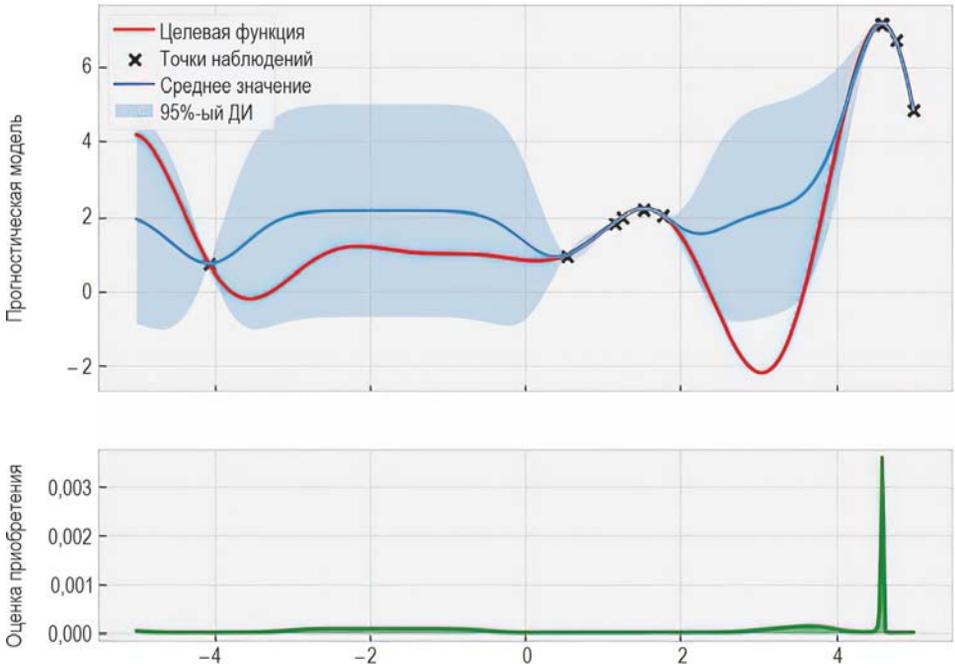


Рис. 1.12. В конце поиска БО нашла глобальный оптимум

Глава 2

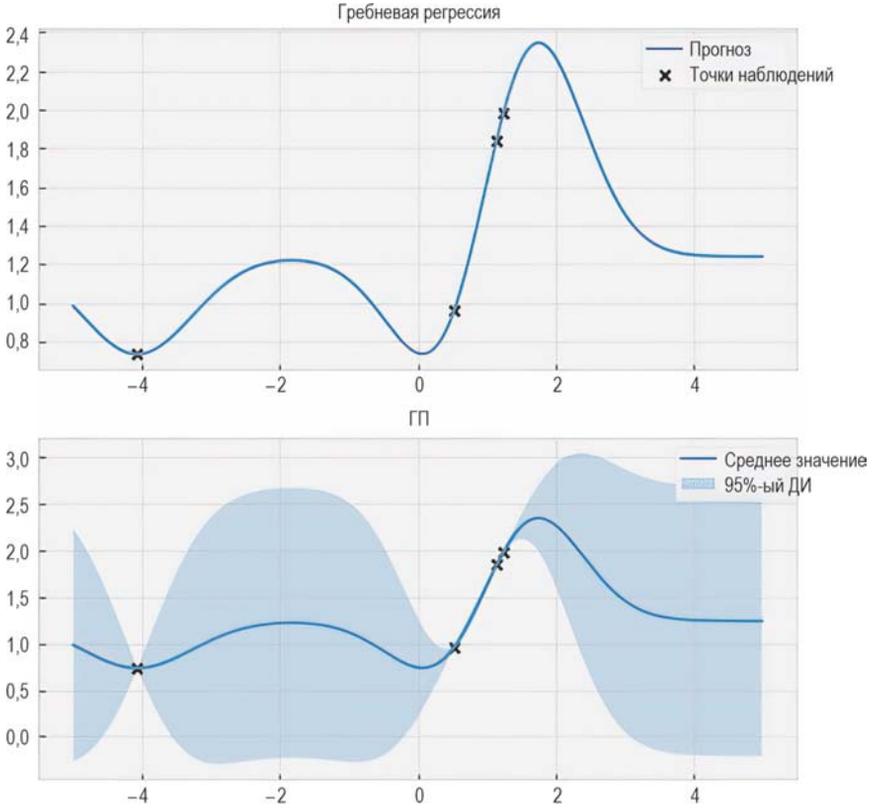


Рис. 2.1. Сравнение прогнозов гребневой регрессии и ГП: несмотря на то что среднее значение прогноза ГП аналогично прогнозу гребневой регрессии, он также предлагает ДИ, который указывает на уровень неопределенности (серая область)



Рис. 2.2. Теорема Байеса, которая предоставляет метод для обновления убеждений об интересующей величине, представленной как вероятностное распределение случайной переменной: прежде чем наблюдать какие-либо данные, имеется априорное убеждение относительно величины X; после обновления данных мы получаем апостериорное убеждение об этой величине

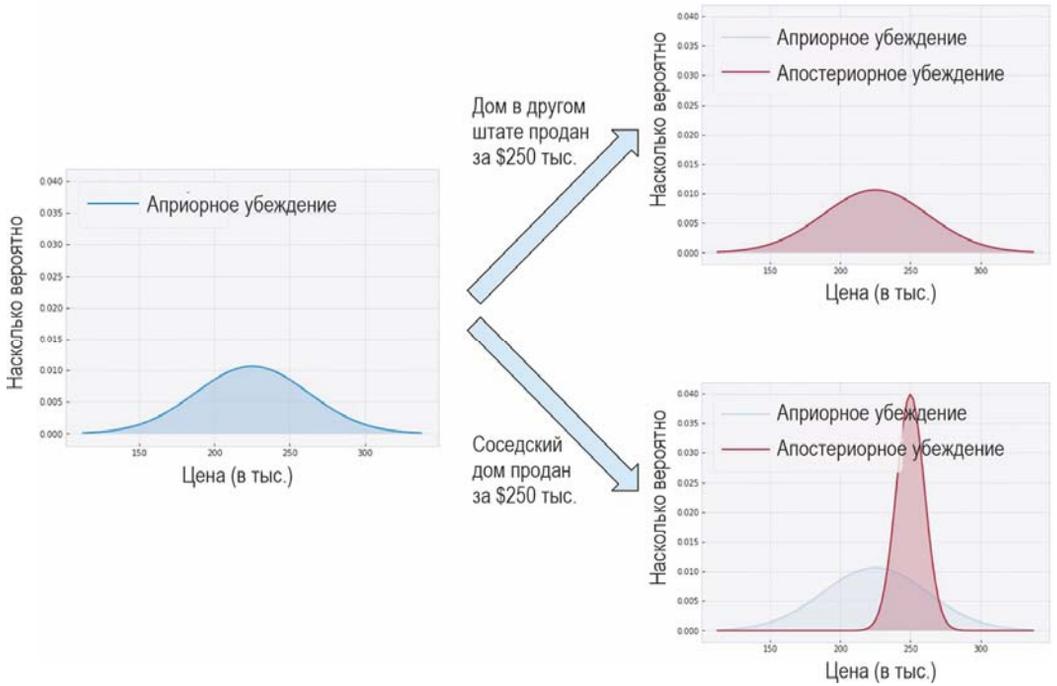


Рис. 2.3. Обновление убеждения о цене нашего дома по методу Байеса: в зависимости от того, насколько наблюдаемый дом похож на наш, апостериорное убеждение либо остается прежним, либо радикально меняется

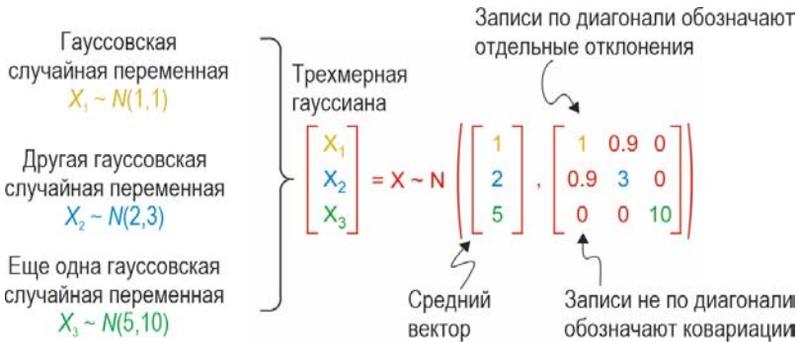


Рис. 2.4. MVN объединяет вместе несколько нормально распределенных случайных величин: пока его средний вектор объединяет средние значения, ковариационная матрица моделирует корреляции между отдельными переменными

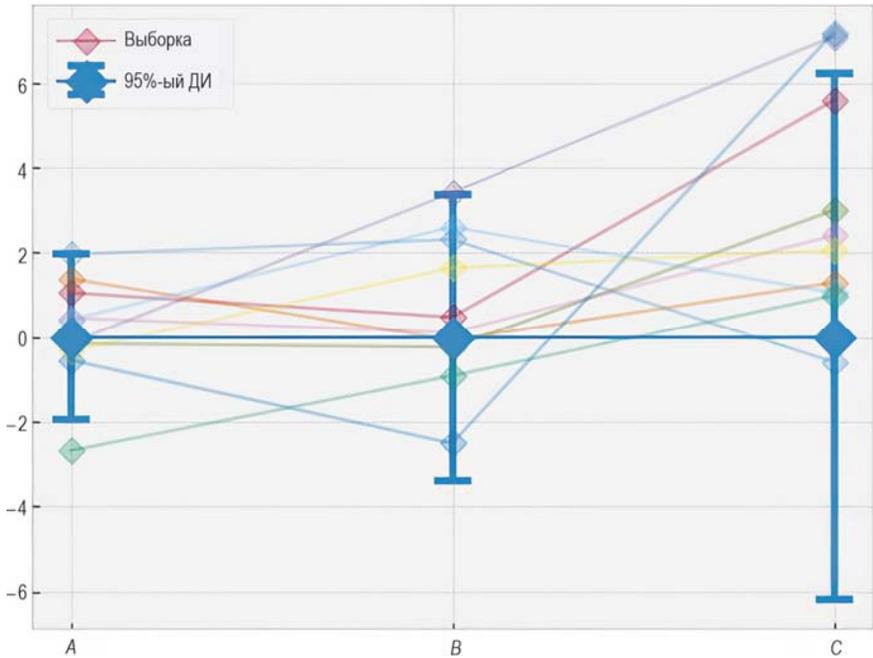


Рис. 2.5. График параллельных координат, который визуализирует MVN-распределение, нормализованное к среднему значению, для примера цен на жилье: полосы ошибок (погрешностей) обозначают 95%-ый ДИ соответствующих нормальных распределений, а бледные линии показывают выборки, взятые из многомерной гауссианы

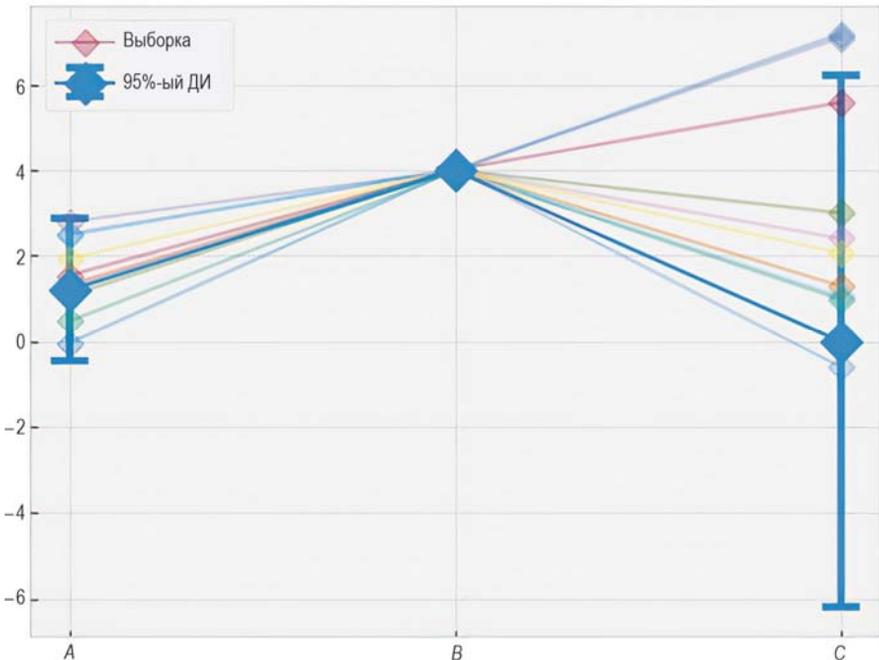


Рис. 2.6. Визуализация MVN-распределения при условии $V = 4$: распределение A обновлено; все нарисованные выборки интерполируют $V = 4$

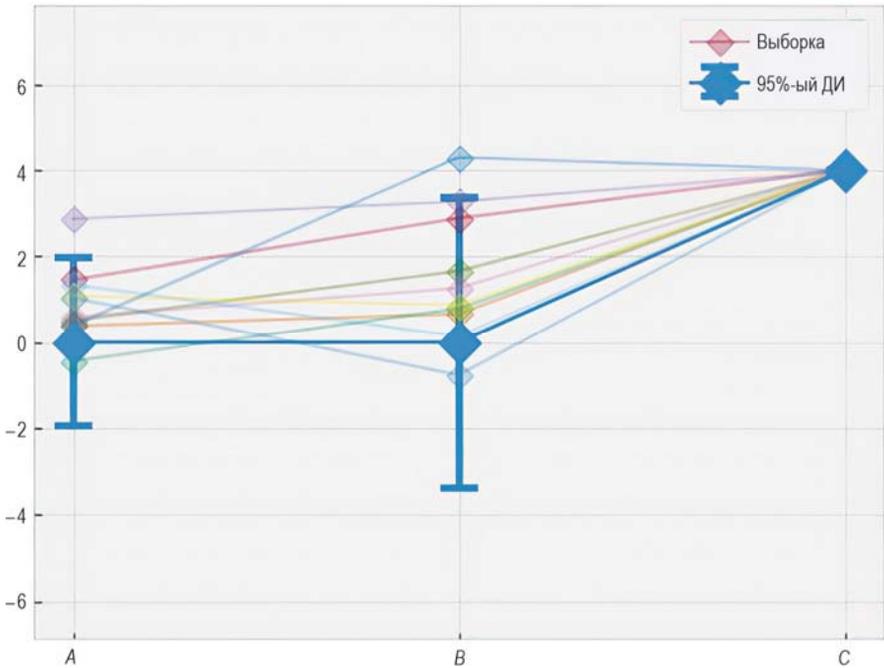


Рис. 2.7. График, визуализирующий MVN-распределение, при условии $C = 4$: никакое другое предельное распределение не меняется; все нарисованные выборки интерполируют $C = 4$

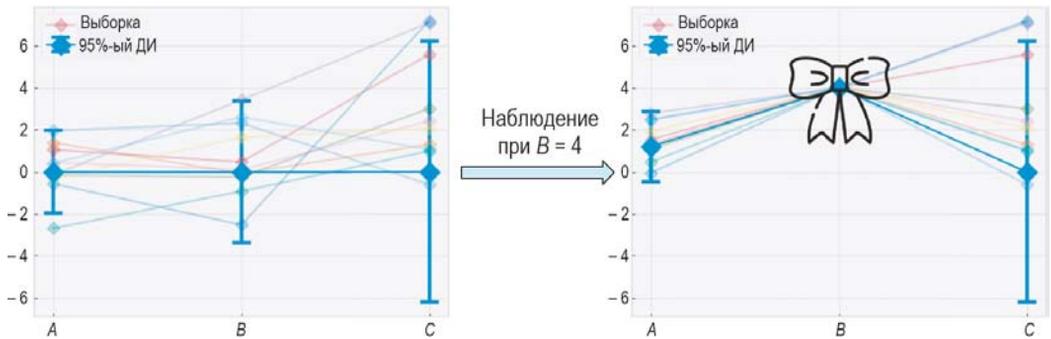


Рис. 2.8. Обусловливание гауссовой зависимости от наблюдения аналогично «завязыванию узла» вокруг этого наблюдения: все выборки из апостериорного распределения должны пройти через «узел»; в наблюдаемой точке нет никакой неопределенности

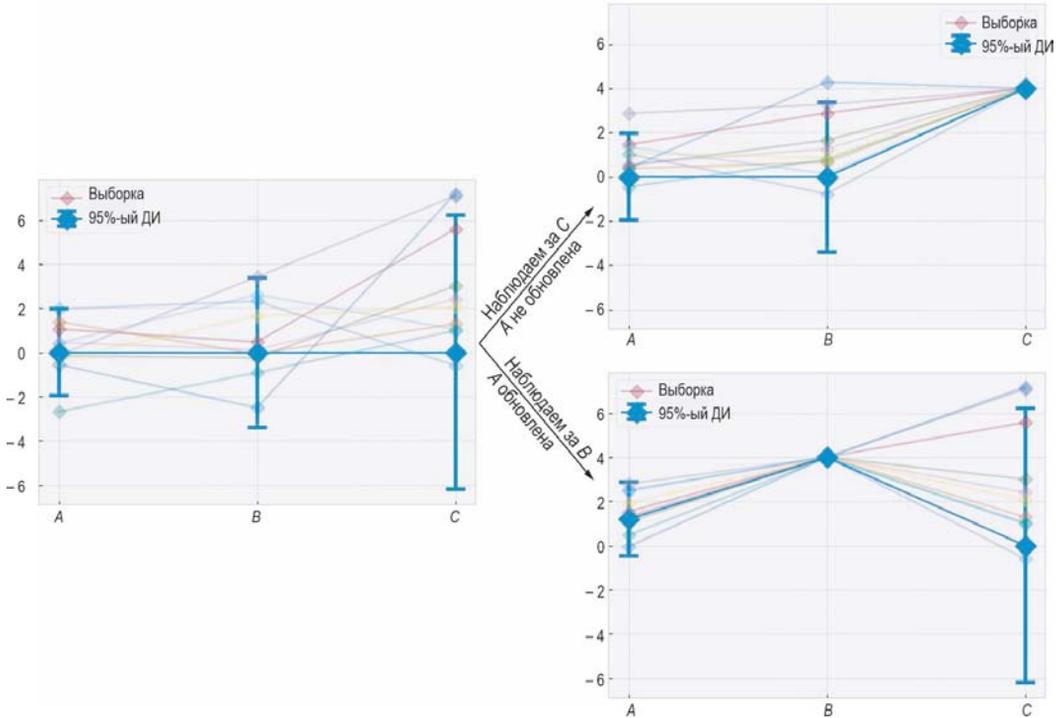


Рис. 2.9. Обновление предположения о цене дома по методу Байеса: в зависимости от того, насколько наблюдаемый дом похож на наш, апостериорное предположение о цене либо остается прежним, либо радикально обновляется

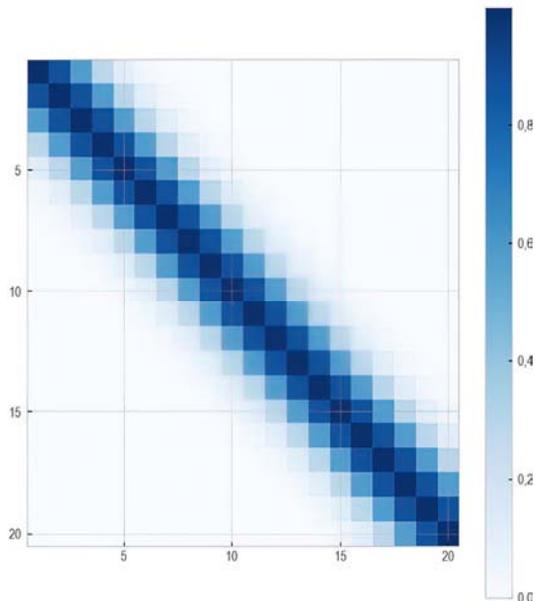


Рис. 2.10. Тепловая карта ковариационной матрицы 20-мерного распределения Гаусса: соседние переменные коррелируют сильнее, чем те, которые находятся далеко друг от друга, на что указывает более темный оттенок

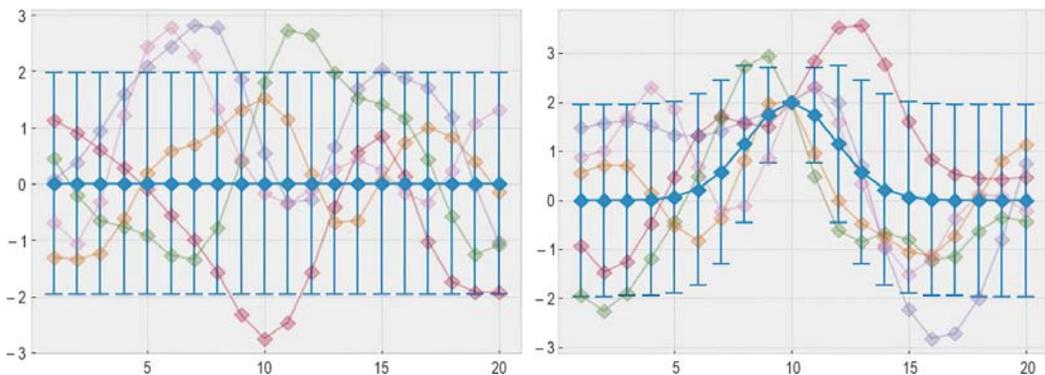


Рис. 2.11. Ошибки и выборки, взятые из априорного (слева) и апостериорного (справа) гауссовских распределений, которые обусловлены тем, что 10-я переменная имеет значение 2: неопределенность в переменных, близких к 10-й, уменьшается в апостериорном распределении, а их средние значения обновляются до близких к значению 2

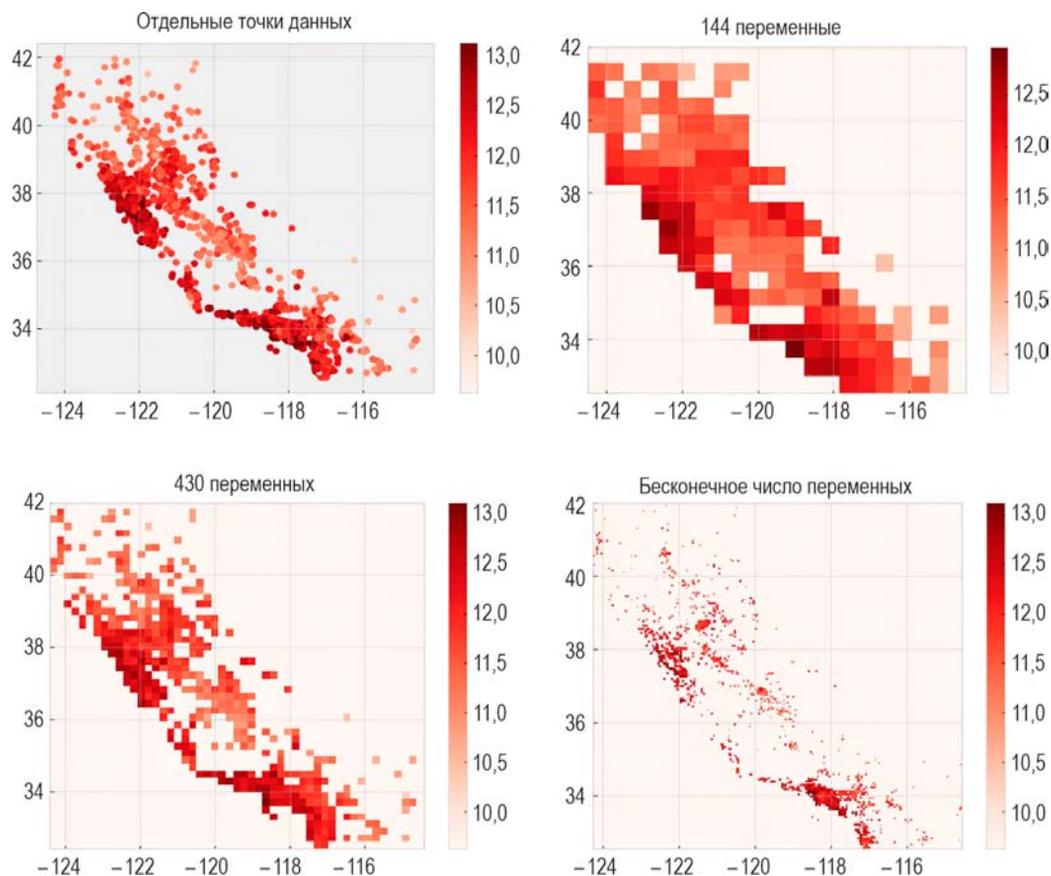


Рис. 2.12. Моделирование цен на жилье в Калифорнии с использованием различного количества переменных: чем больше переменных, тем более гладкой становится модель и тем ближе мы приближаемся к бесконечномерной гауссиане

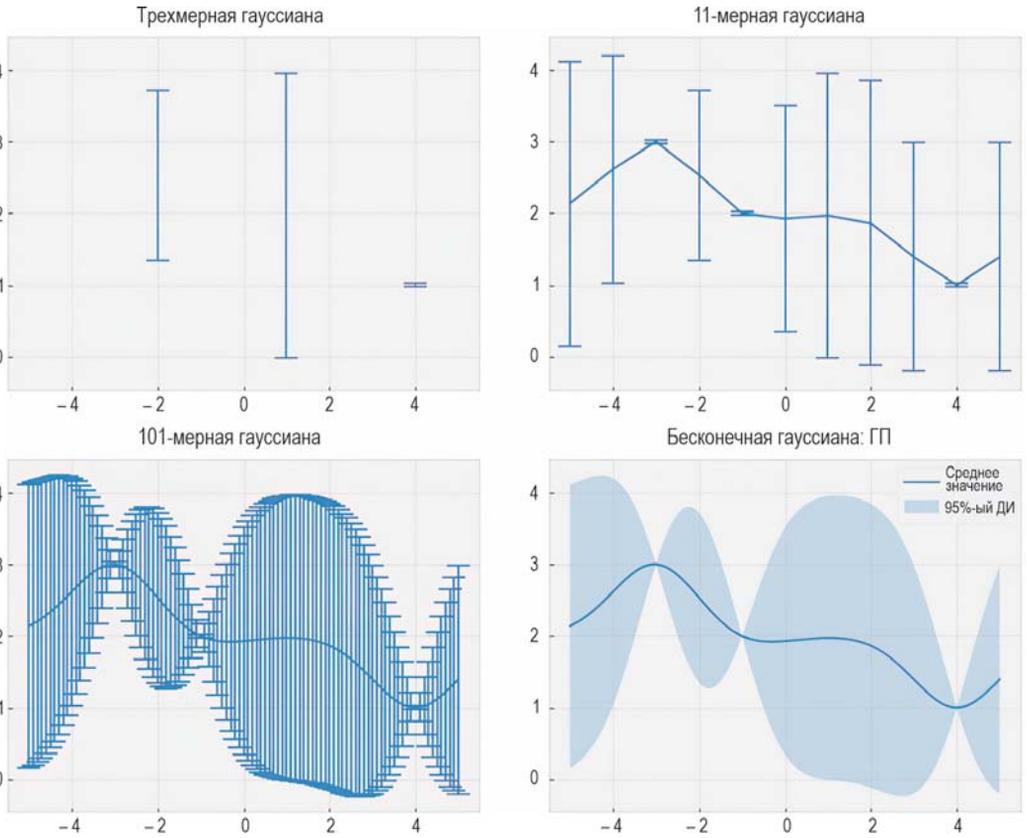


Рис. 2.13. Графики различных гауссовских распределений: любое конечное подмножество ГП является MVN-распределением; когда число переменных приближается к бесконечности, мы получаем ГП и можем делать прогнозы в любом месте области

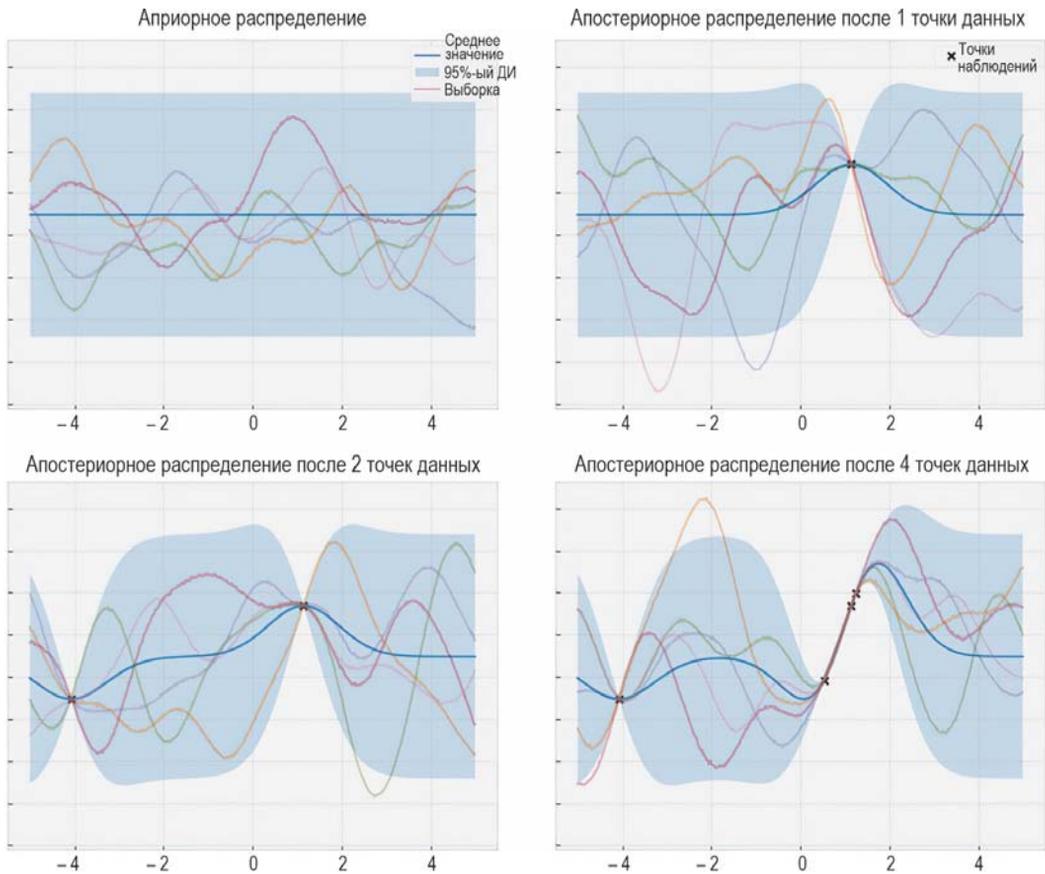


Рис. 2.14. Прогнозы ГП на основе 0, 1, 2 и 4 наблюдений

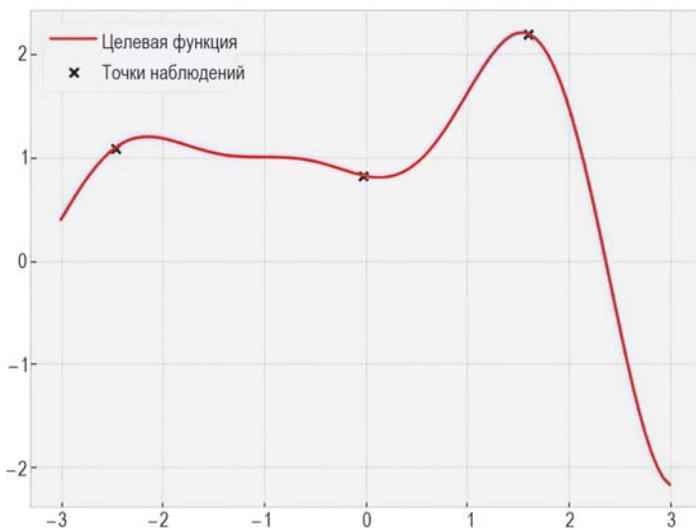


Рис. 2.15. Целевая функция, которая используется в текущем примере, показана сплошной линией; точки в наборе обучающих данных отмечены маркерами

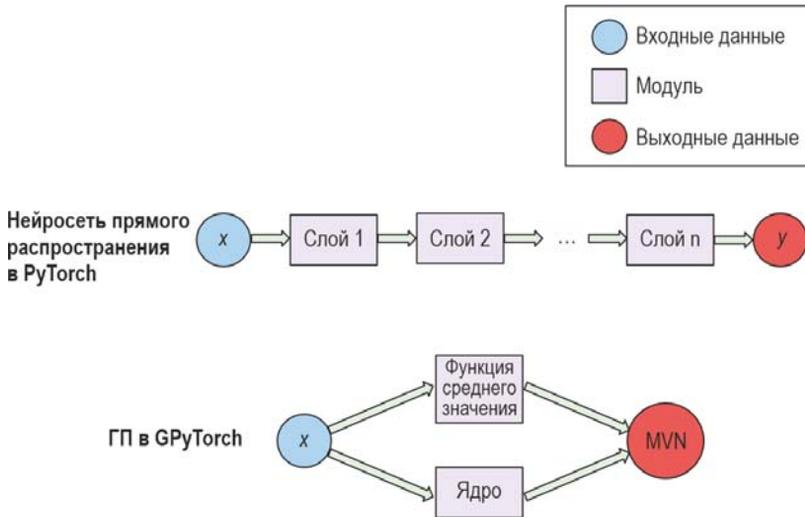


Рис. 2.16. Обработка данных в PyTorch и GPyTorch соответствующими методами `forward()`: входные данные обрабатываются различными модулями для получения окончательного результата; это либо число для нейросети прямого распространения, либо MVN-распределение для ГП

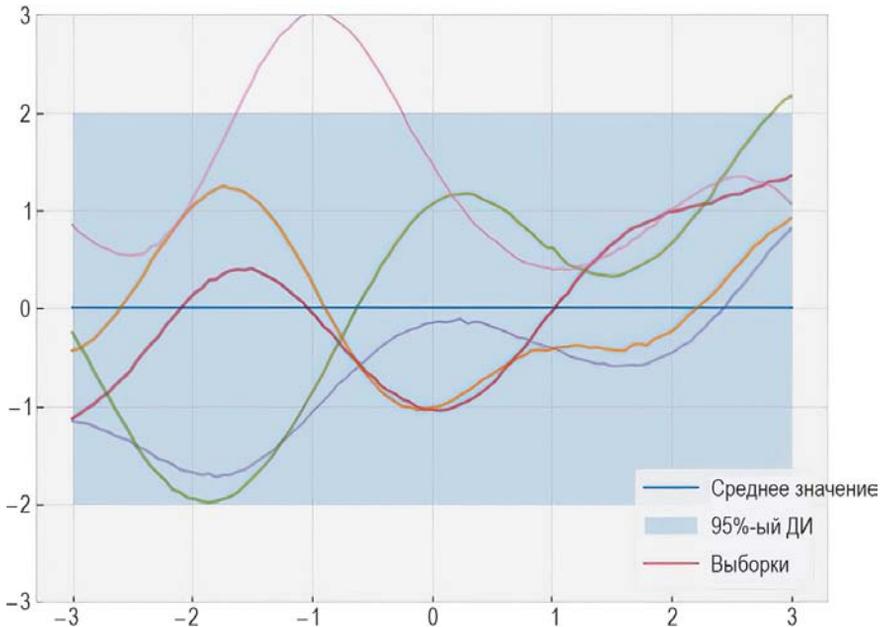


Рис. 2.17. Прогнозы, сделанные априорным ГП с нулевым средним значением и ядром RBF: несмотря на то что среднее значение и ДИ постоянны, отдельные выборки демонстрируют сложное нелинейное поведение

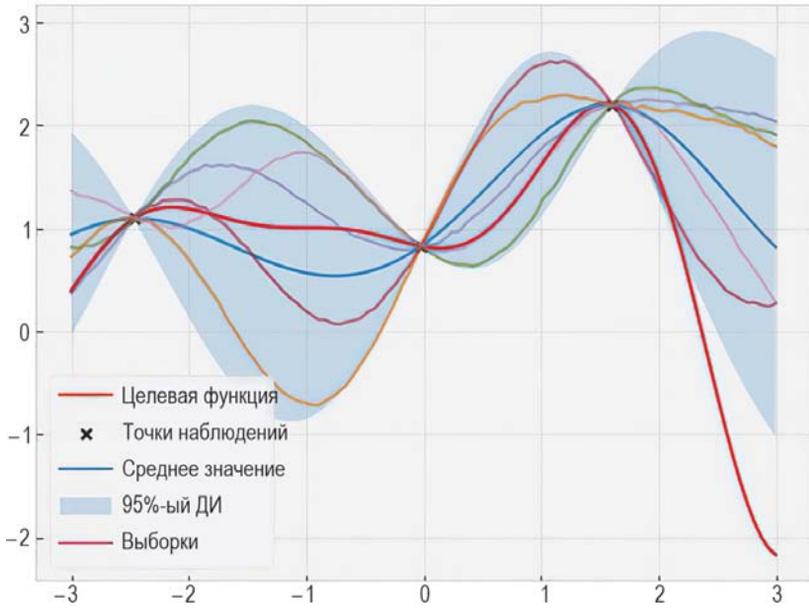


Рис. 2.18. Прогнозы ГП: функция среднего и случайные выборки плавно интерполируют точки обучающих данных, в то время как неопределенность исчезает в областях, окружающих эти точки

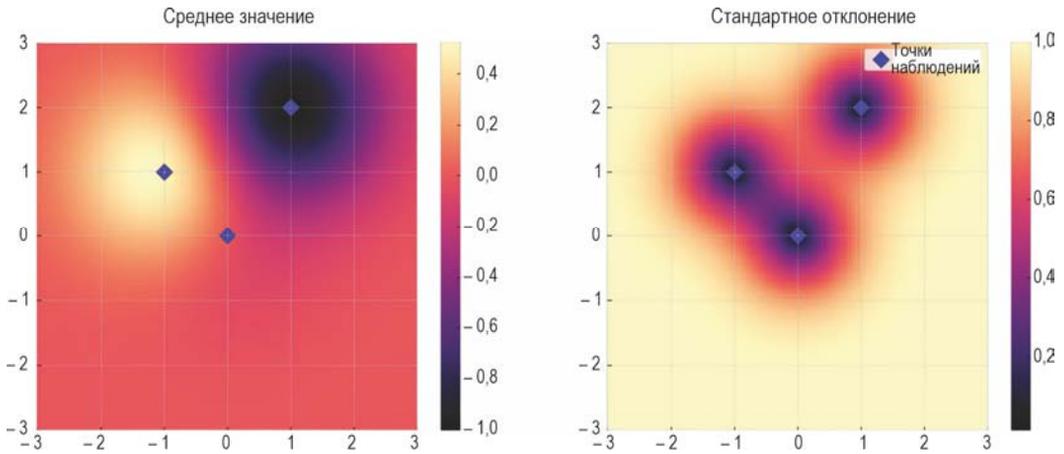


Рис. 2.19. Прогнозы двумерного ГП: средняя функция по-прежнему согласуется с данными обучения, и неопределенность снова исчезает в областях, окружающих точки данных

Глава 3



Рис. 3.1. Реакция вас и ваших друзей, когда человек правильно угадал загаданное число 100 раз: каждый из вас пришел к разным выводам в зависимости от своего изначального (априорного) убеждения

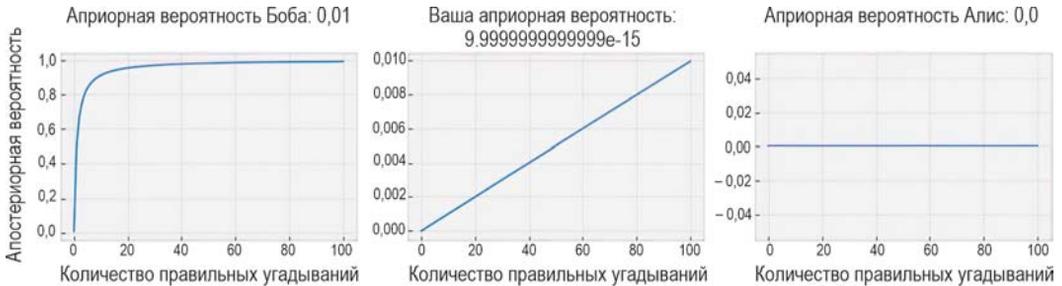


Рис. 3.2. Прогрессивная апостериорная вероятность того, что женщина на карнавале является экстрасенсом, как функция количества успешных догадок: она никогда не уменьшается, а ведет себя по-разному в зависимости от начальной априорной вероятности



Рис. 3.3. Как разные предварительные убеждения каждого человека обновляются одними и теми же данными: по сравнению с априорным показателем Боба, ваш намного ниже и увеличивается медленнее, приоритет Алис равен 0 и остается неизменным на протяжении всего процесса

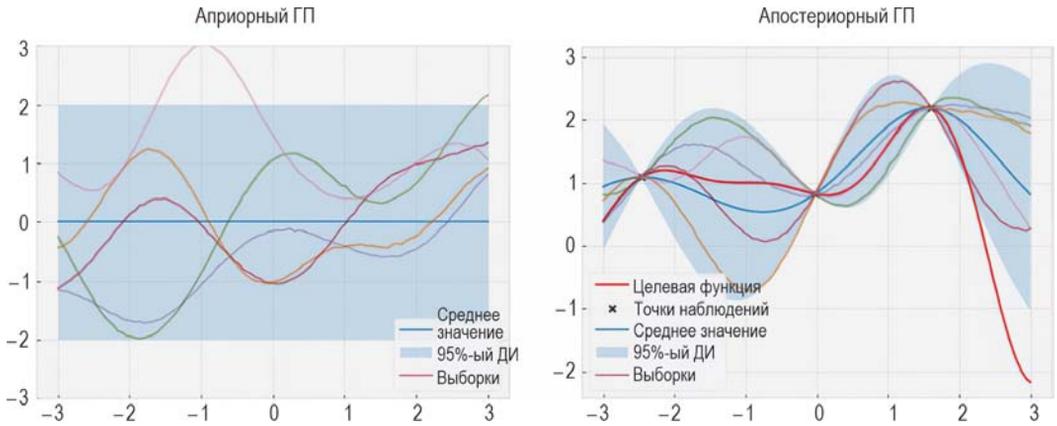


Рис. 3.4. Сравнение априорного и апостериорного ГП: априорный содержит изначальную информацию о целевой функции, а апостериорный объединяет эту информацию с фактическими наблюдениями

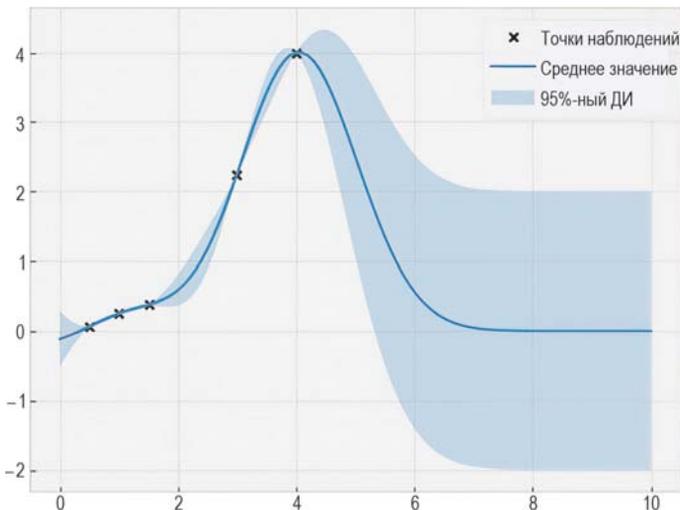
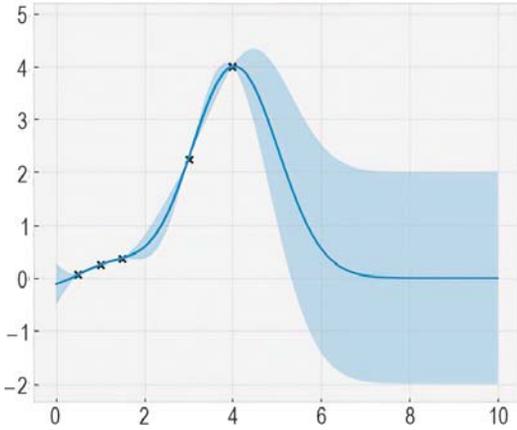


Рис. 3.5. Прогнозы ГП с функцией нулевого среднего: апостериорная функция среднего интерполирует наблюдаемые точки данных и возвращается к нулю в регионах, которые находятся далеко от наблюдений

Значение функции среднего = 0,0
Логарифмическая вероятность = -2,0545



Значение функции среднего = 2,0
Логарифмическая вероятность = -1,4465

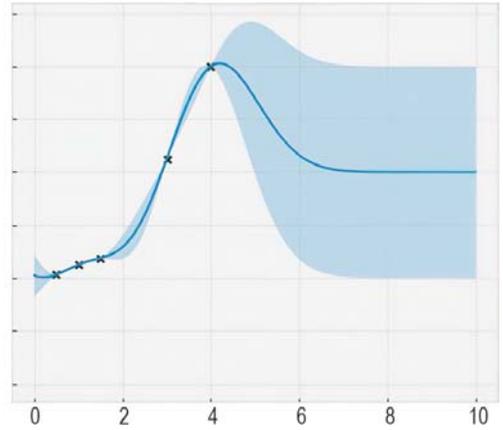


Рис. 3.6. Прогнозы ГП с учетом двух разных постоянных функций среднего: значение 2 дает более высокую вероятность, чем 0, указывая на то, что первая функция среднего лучше подходит, чем вторая

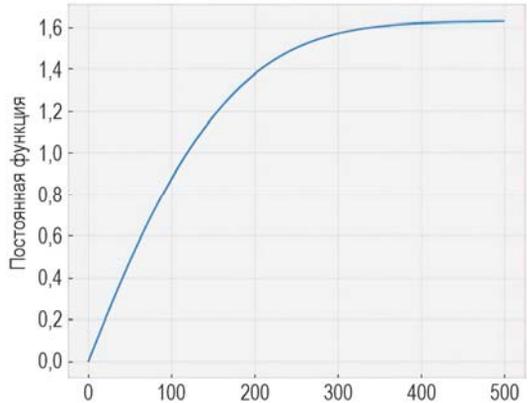
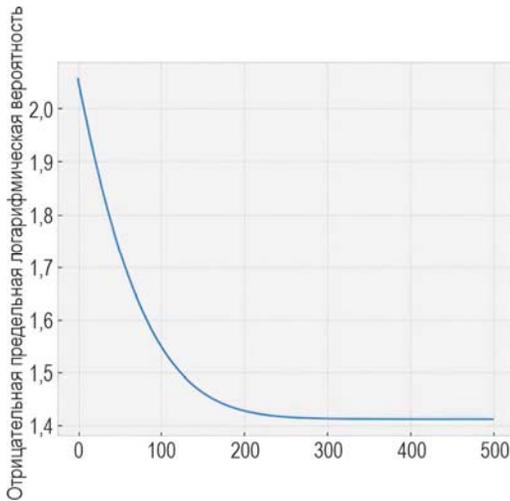


Рис. 3.7. Вероятность отрицательного логарифма (чем ниже, тем лучше) и среднее значение во время градиентного спуска: на обеих частях рисунка значения сошлись, что указывает на то, что мы достигли оптимума

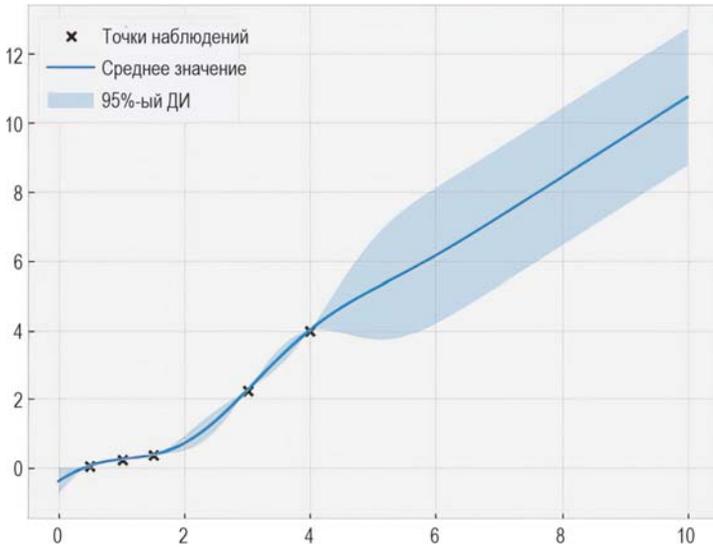


Рис. 3.8. Прогнозы ГП с линейной функцией среднего: ГП имеет тенденцию к росту, что является прямым результатом положительного наклона линейной функции среднего

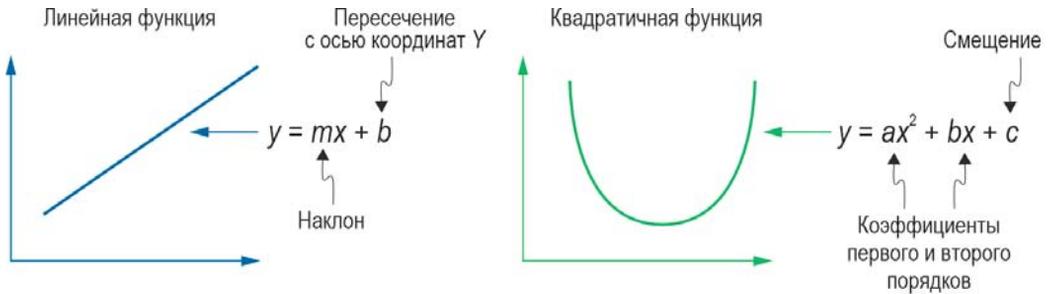


Рис. 3.9. Функциональные формы линейной и квадратичной функций: линейная имеет два параметра, а квадратичная — три; когда они используются в качестве функции среднего для ГП, соответствующие параметры являются гиперпараметрами ГП

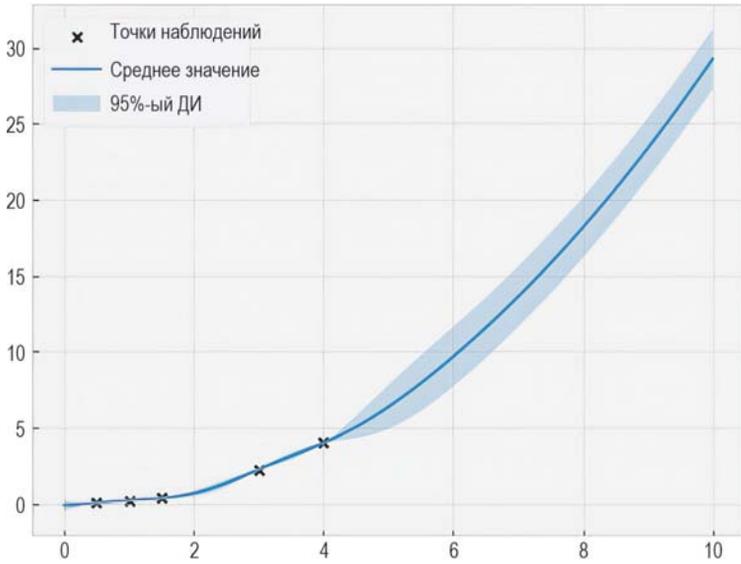


Рис. 3.10. Прогнозы ГП с функцией квадратичного среднего: чем больше жилая площадь, тем с большей скоростью будет расти цена

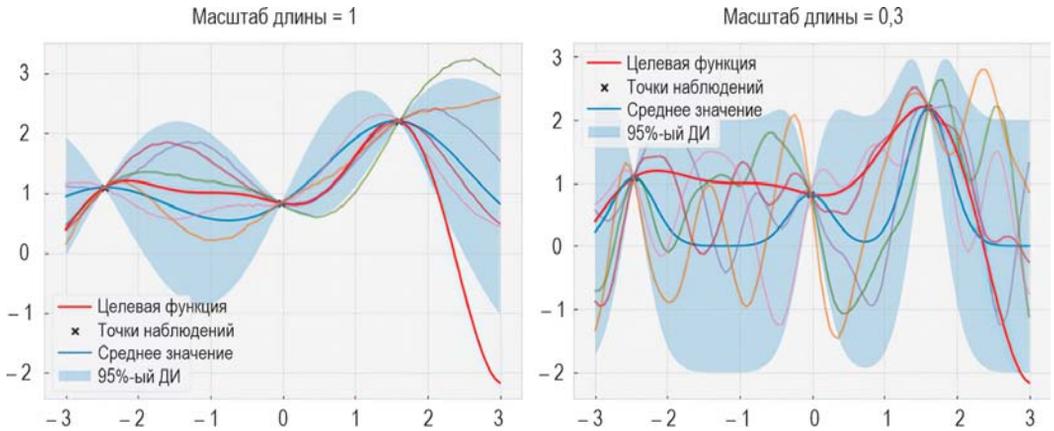


Рис. 3.11. Прогнозы ГП с масштабом длины, установленным в значения 1 (слева) и 0,3 (справа): при небольшом масштабе длины прогнозы имеют большую изменчивость, что приводит к большей неопределенности

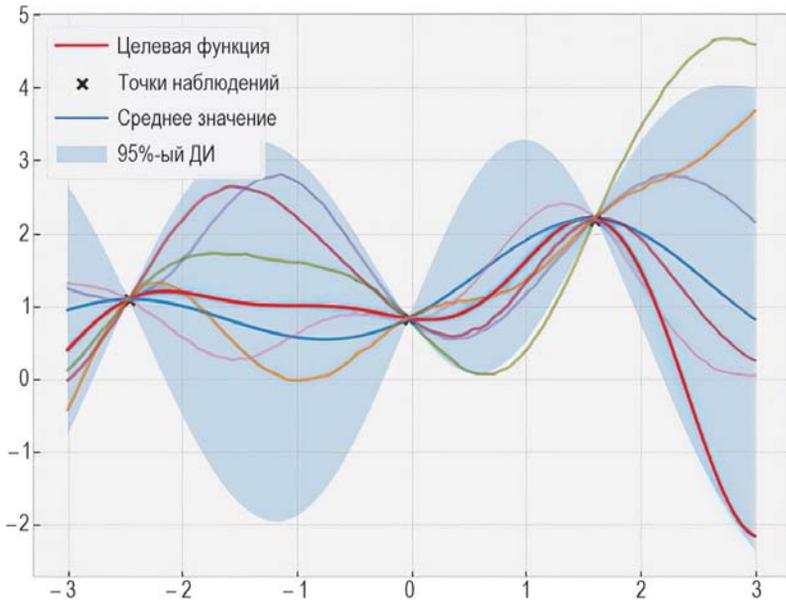


Рис. 3.12. Прогнозы ГП с масштабом вывода, равным 3: при большом масштабе ГП моделирует функцию с более широким диапазоном, что также допускает больше неопределенности в прогнозах

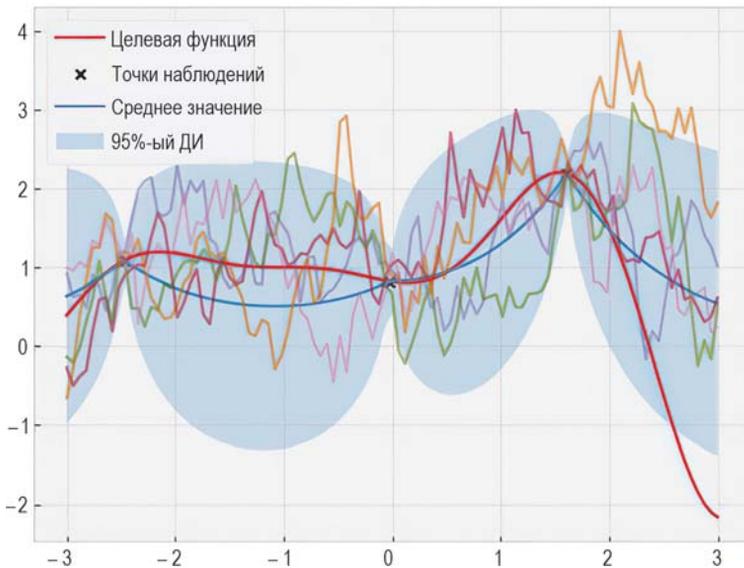


Рис. 3.13. Прогнозы ГП с помощью ядра Матерна со значением $1/2$: значение определяет убеждение, что целевая функция не дифференцируется, что соответствует очень грубым выборкам

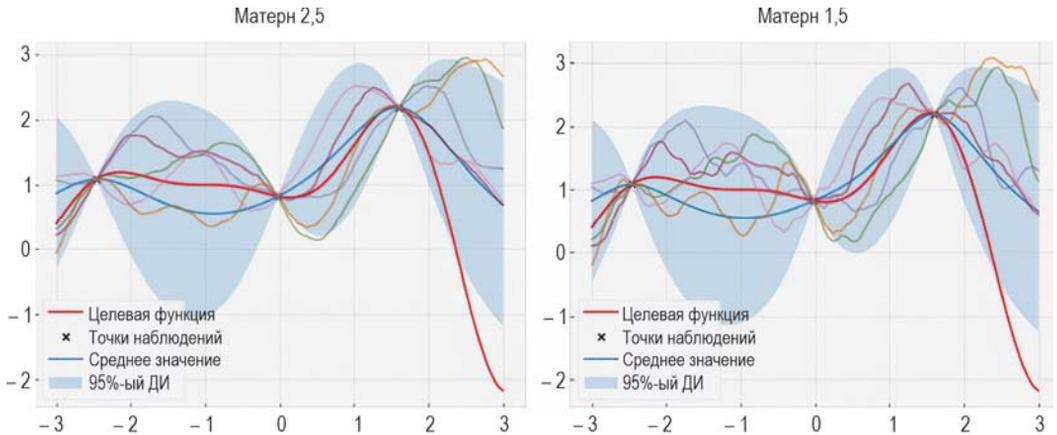


Рис. 3.14. Прогнозы ГП с ядрами Матерна 5/2 (слева) и 3/2 (справа): выборки здесь достаточно гладкие, чтобы ГП мог эффективно учиться на данных, но также достаточно неровные, чтобы реалистично моделировать процессы

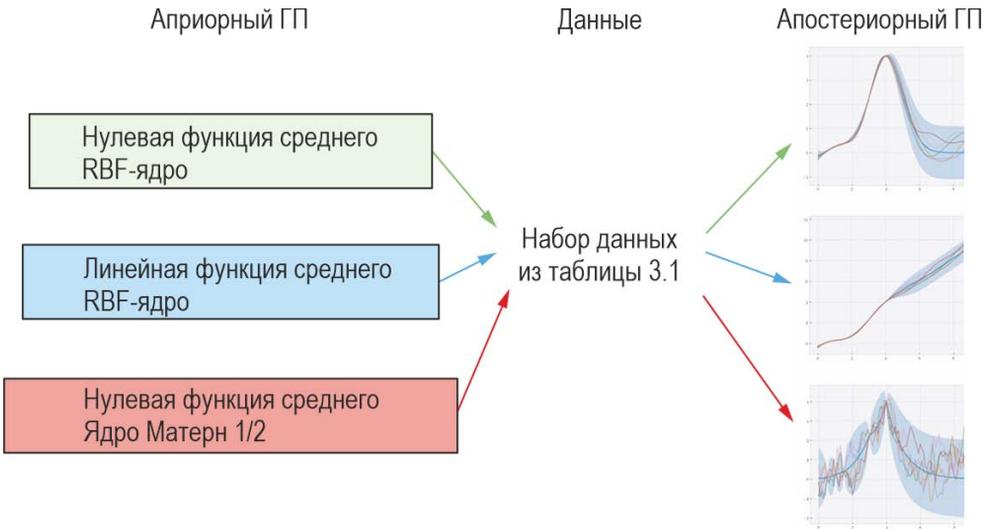


Рис. 3.15. Три разных варианта сочетания функции среднего и ядра, а также прогнозы, сделанные соответствующими апостериорными ГП при обучении на одном и том же наборе данных: каждый вариант приводит к различному поведению в прогнозировании

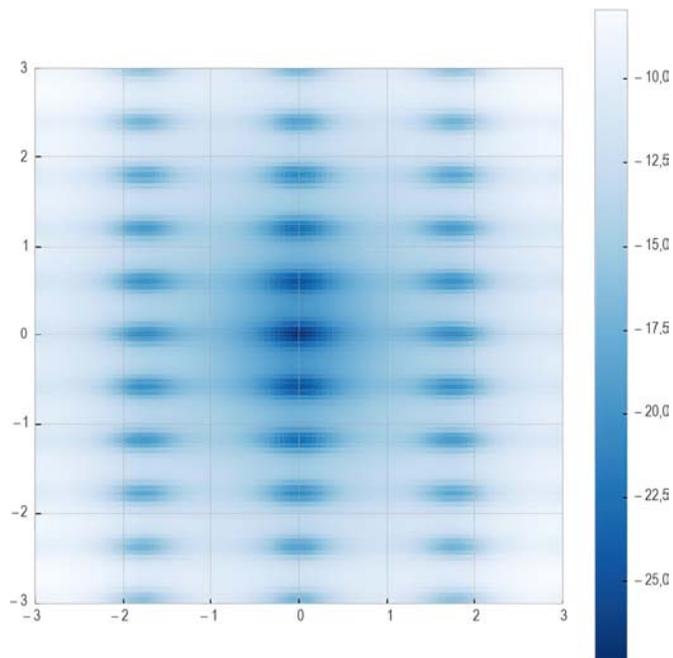


Рис. 3.16. Двумерная функция Экли в качестве целевой: ось X имеет меньшую изменчивость (она меняется меньше), чем ось Y , что требует разных масштабов длины

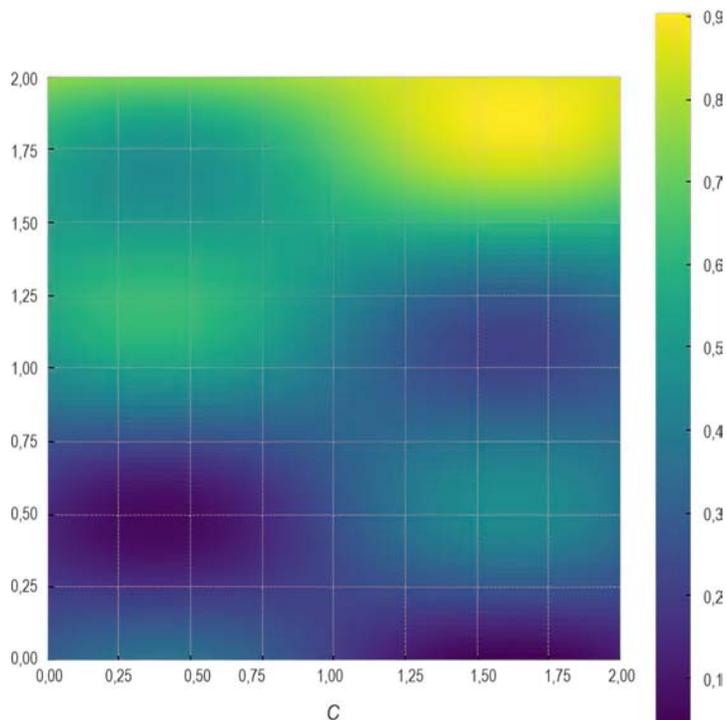
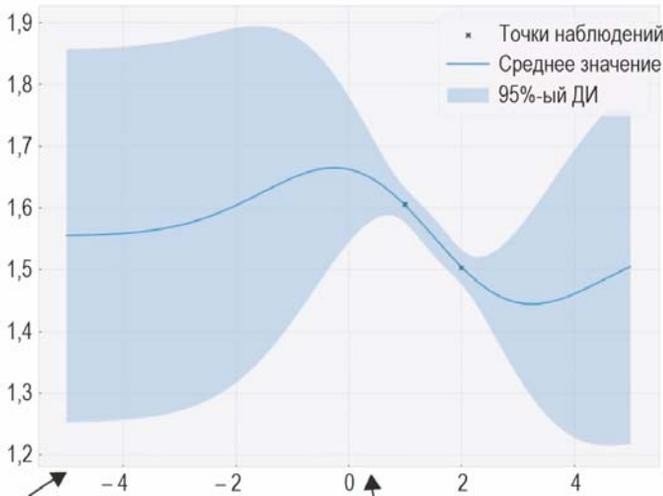


Рис. 3.17. Точность SVM-модели на тестовом наборе данных как функция штрафного параметра (c) и параметра RBF-ядра (γ): функция меняется быстрее по γ , чем по c

Глава 4



Исследование:
запросы к регионам
с высокой неопределенностью

Эксплуатация:
запросы к регионам с высоким
прогнозируемым средним значением

Рис. 4.1. Компромисс между исследованием и эксплуатацией в БО: каждая из политик должна решить, что следует делать, запрашивать регионы с высокой неопределенностью (исследование) или запрашивать регионы с высоким прогнозируемым средним значением (эксплуатация)

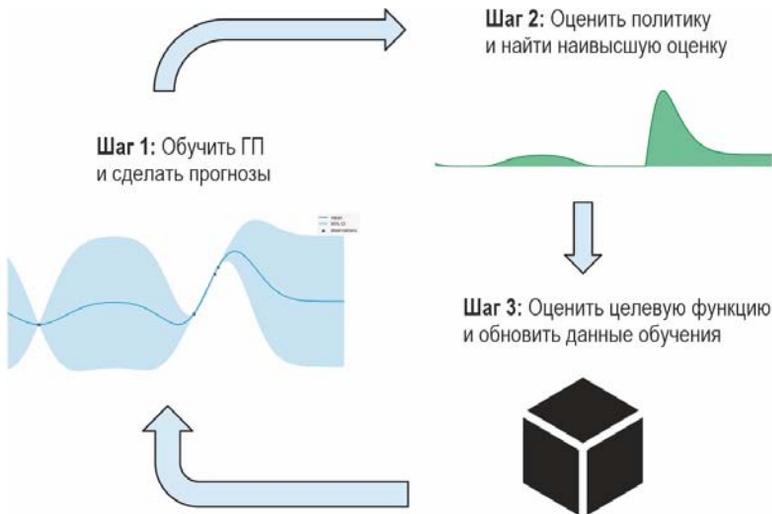


Рис. 4.2. Цикл БО, сочетающий в себе ГП для моделирования и политику для принятия решений: этот полный рабочий процесс теперь можно использовать для оптимизации функций «черного ящика»

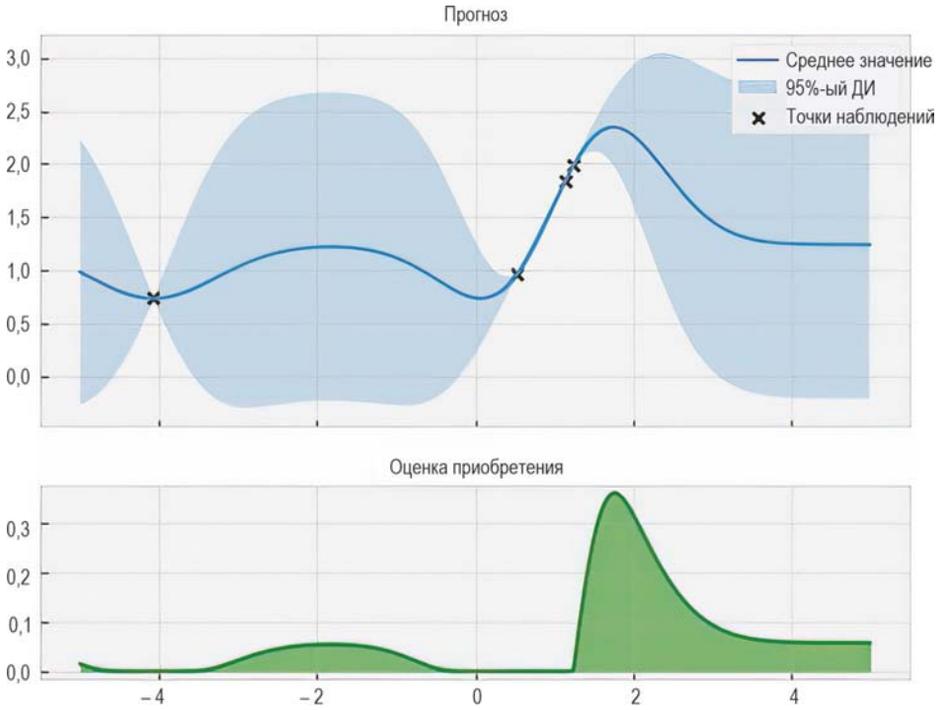
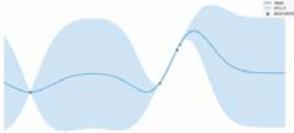


Рис. 4.3. Типичная визуализация прогресса в БО: сверху показаны прогнозы ГП и основная истинная целевая функция; снизу показаны оценки приобретения, полученные с помощью политики под названием «Ожидаемое улучшение», о которой более подробно будет рассказано в разделе 4.3

Шаг 1: Обучить ГП и сделать прогнозы



```
model, likelihood = fi_gp_model(
    train_x, train_y
)
```



Шаг 2: Оценить политику
и найти наивысшую оценку



```
policy = ...
next_x, acq_val =
    botorch.optim.optimize_acqf(...)
```



Шаг 2.5: Визуализировать
текущий прогресс

```
visualize_gp_belief_and_policy(...)
```



Шаг 3: Оценить целевую функцию
и обновить данные обучения



```
next_y = forrester_ld(next_x)
train_x = torch.cat([train_x, next_x])
train_y = torch.cat([train_y, next_y])
```

Рис. 4.4. Шаги цикла БО и соответствующий им код, который для каждого шага является модульным, что позволяет легко отслеживать весь цикл

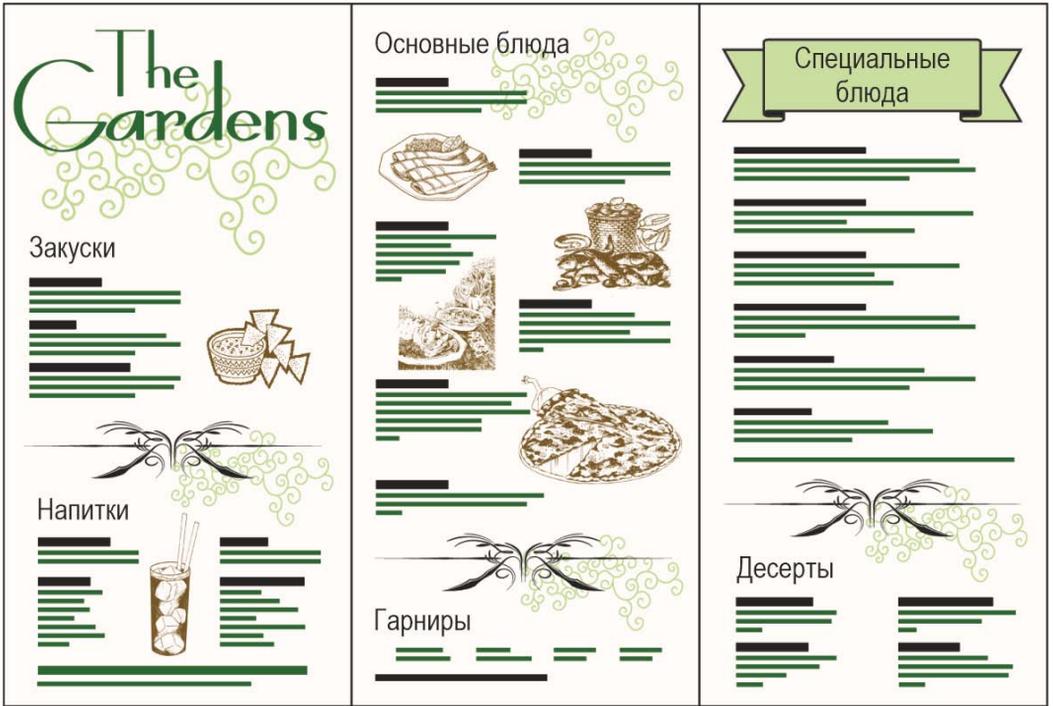


Рис. 4.5. Заказ из меню в ресторане предполагает компромисс между исследованием (попробовать что-то новое) и эксплуатацией (заказать привычное)

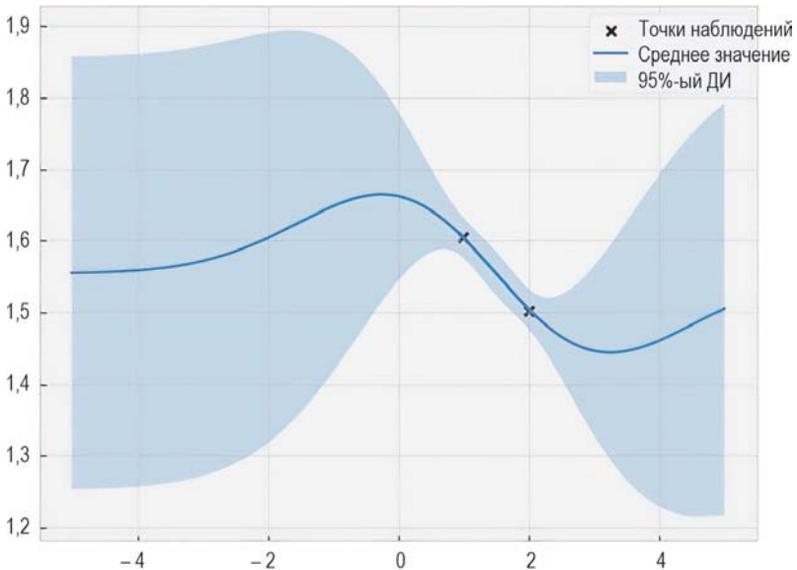


Рис. 4.6. Прогнозы ГП, обученного на двух точках данных функции Форрестера

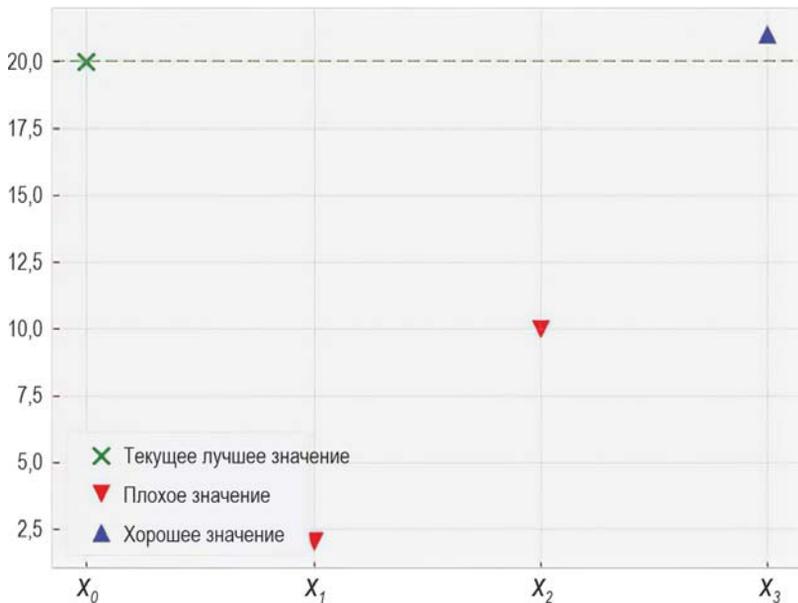


Рис. 4.7. Поиск улучшений относительно наилучшей точки обзора: несмотря на то, что точка x_2 лучше, чем x_1 , обе они являются «плохими» относительно x_0

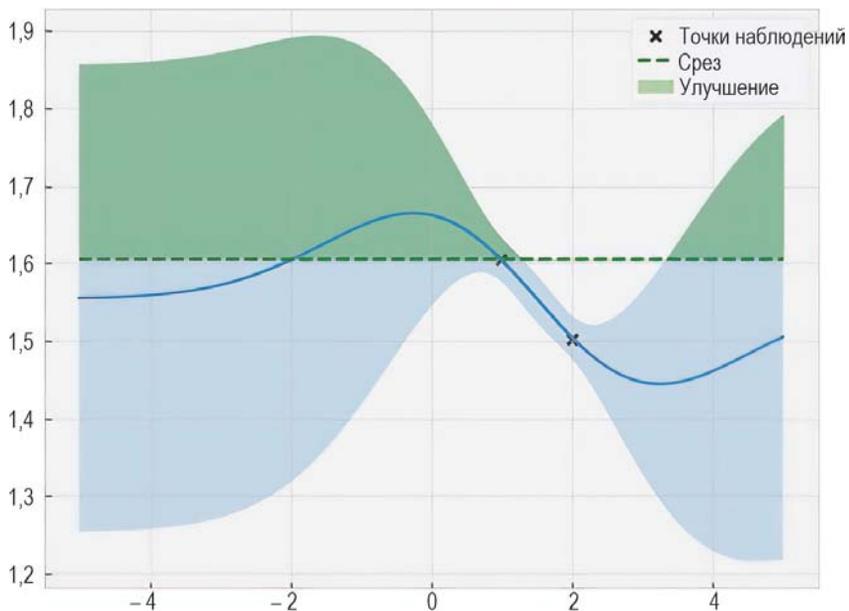


Рис. 4.8. Улучшение действующего значения с точки зрения ГП: часть прогнозов, которая соответствует улучшению относительно действующей точки, выделена более темным оттенком

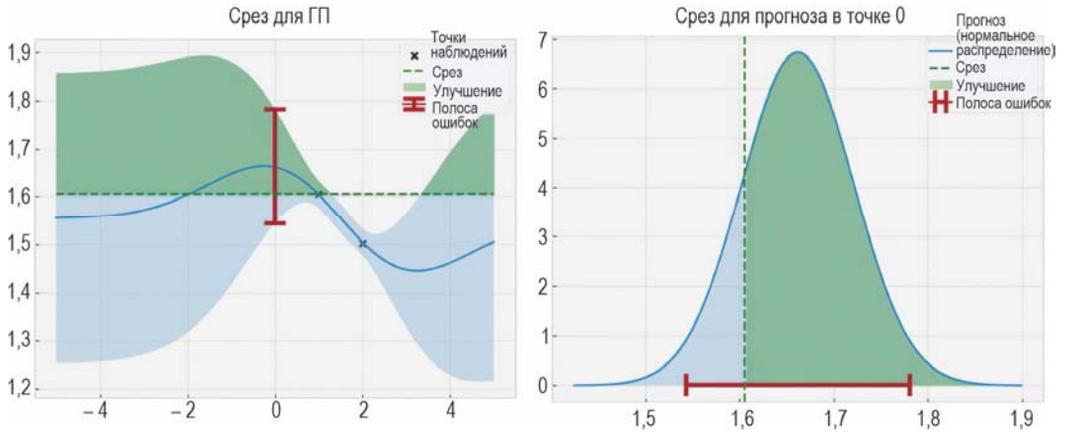


Рис. 4.9. Улучшение по сравнению с действующей точкой в значении 0 (выделено более темным оттенком): на левой части показан весь ГП, а на правой — только нормальное распределение, соответствующее прогнозу при 0 (полосы ошибок одинаковы на обеих частях); здесь улучшение по сравнению с действующей точкой следует усеченному нормальному распределению

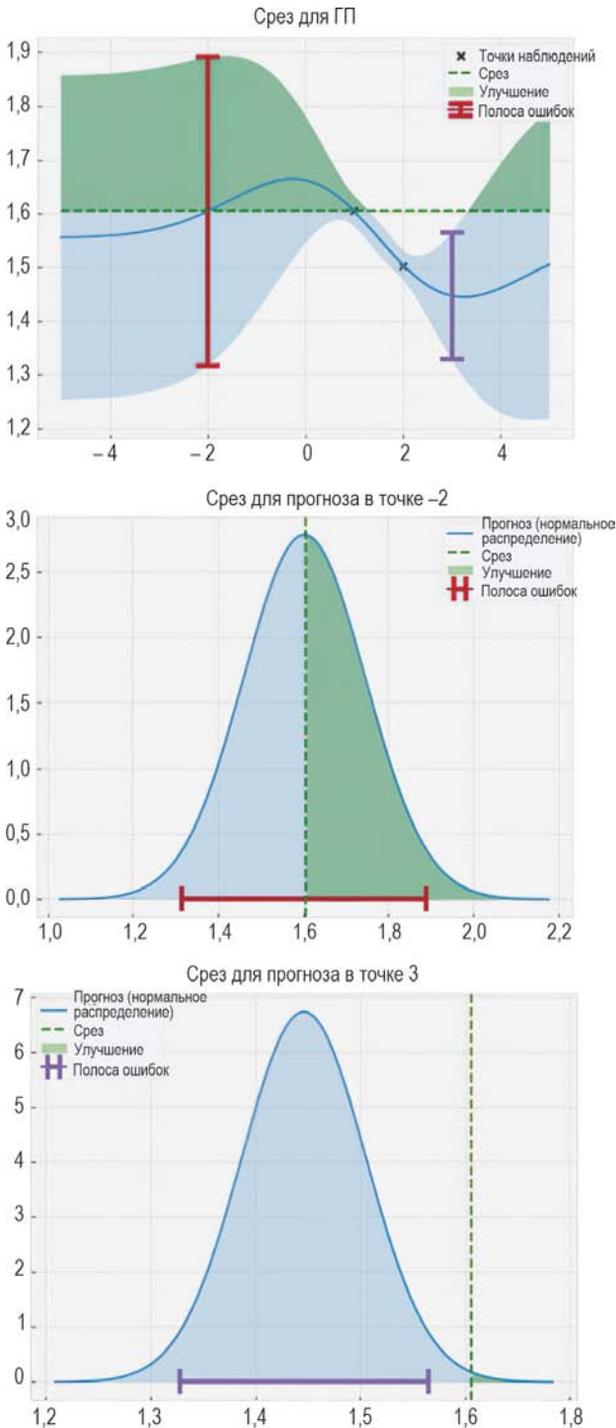


Рис. 4.10. Улучшение по сравнению с действующей точкой при значениях -2 и 3 (выделено более темным оттенком): левая часть показывает весь ГП, центральная — прогноз при -2 , а правая — прогноз при 3 ; выделенные части показывают возможные улучшения, которые зависят от нормального распределения в данной точке

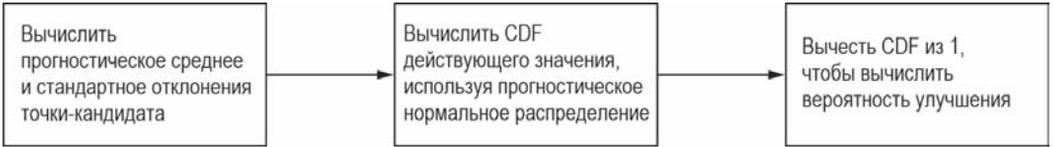


Рис. 4.11. Блок-схема расчета оценки PoI: она помогает вычислить, насколько вероятно, что любая точка-кандидат сможет улучшиться относительно действующей

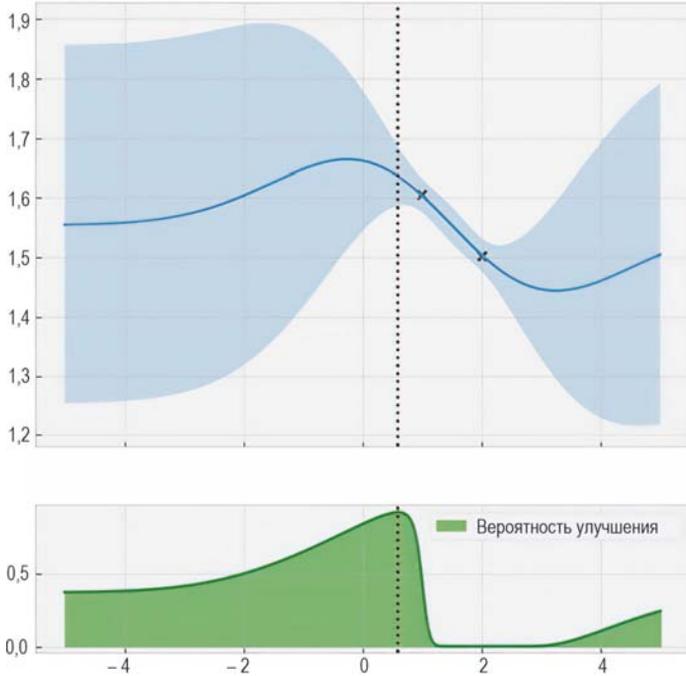


Рис. 4.12. Прогнозы ГП (вверху) и политика PoI (внизу): пунктирная линия указывает на точку, которая максимизирует оценку PoI; в этот момент мы запрашиваем целевую функцию на следующей итерации оптимизации

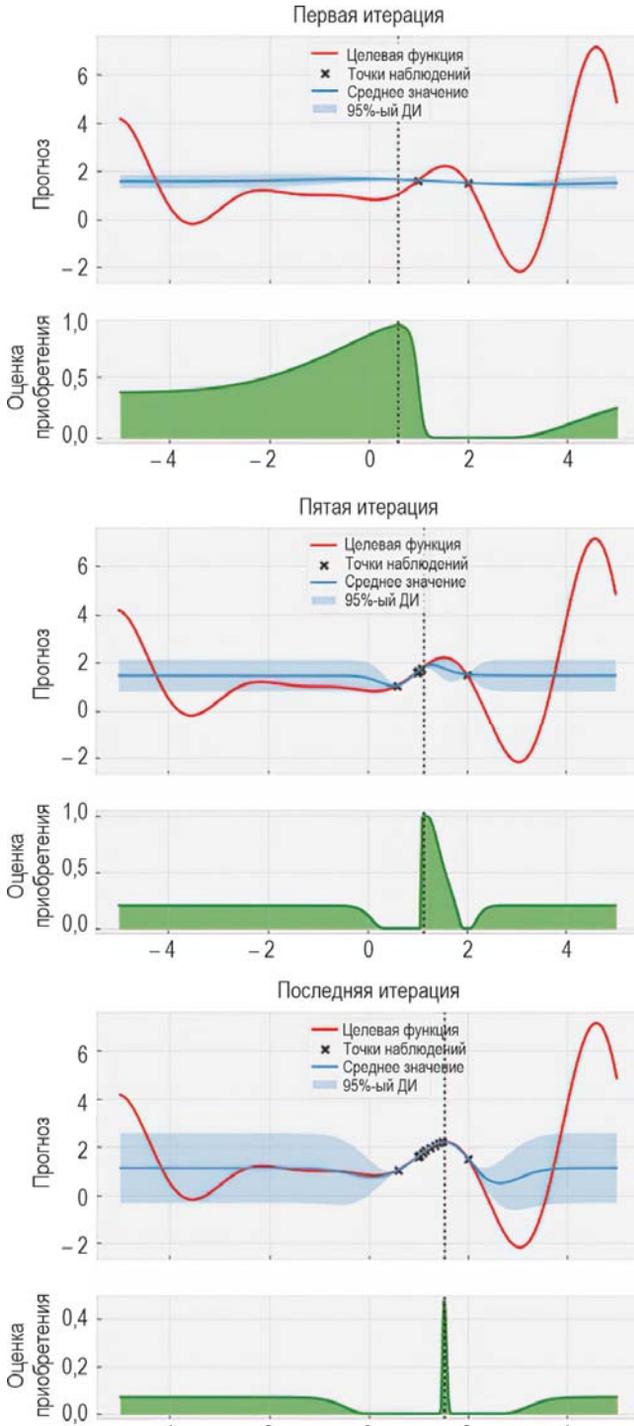


Рис. 4.13. Прогресс, достигнутый политикой PoI: поскольку политика направлена на улучшение любого масштаба, прогресс застревает на локальном оптимуме возле значения 2, и мы не можем исследовать другие области пространства поиска

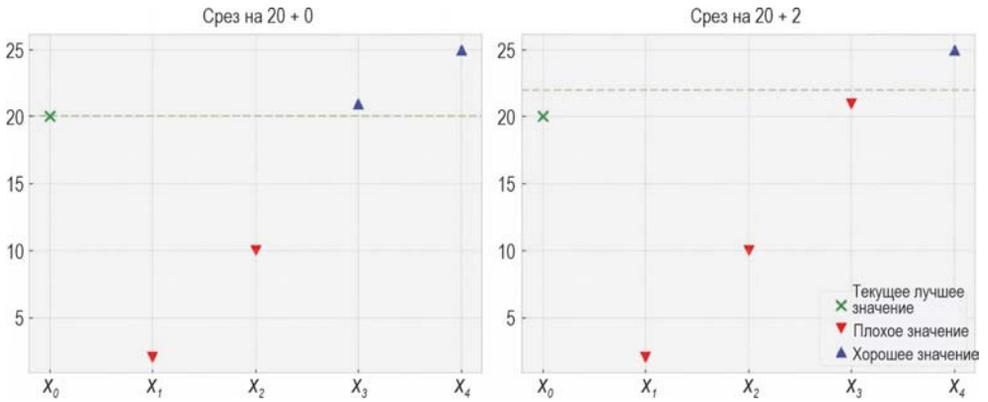


Рис. 4.14. Нахождение более строгого определения для улучшения, которое должно быть больше относительно действующей точки как минимум на $\varepsilon = 0$ (слева) и $\varepsilon = 2$ (справа): чем выше требование, тем больше политика Pol склоняется в сторону исследования (подробности в упражнении 1)

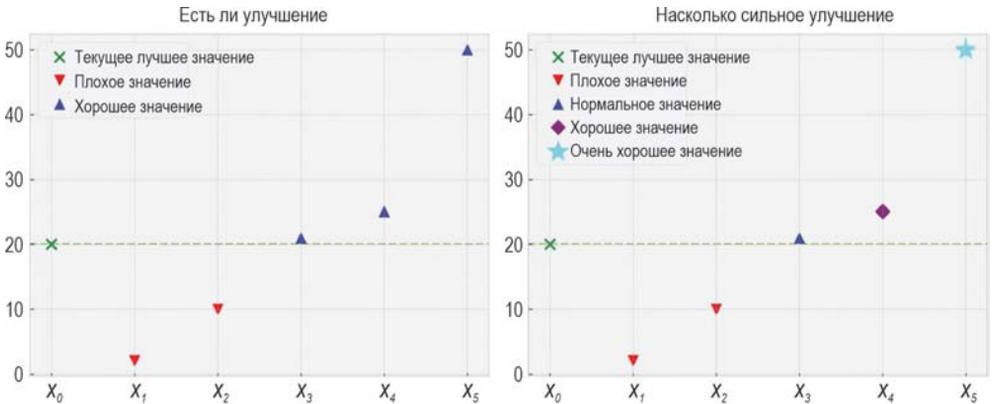


Рис. 4.15. Разница между Pol (слева) и EI (справа): первая показывает лишь наличие улучшения относительно действующего значения, а вторая учитывает, насколько сильно произойдет улучшение

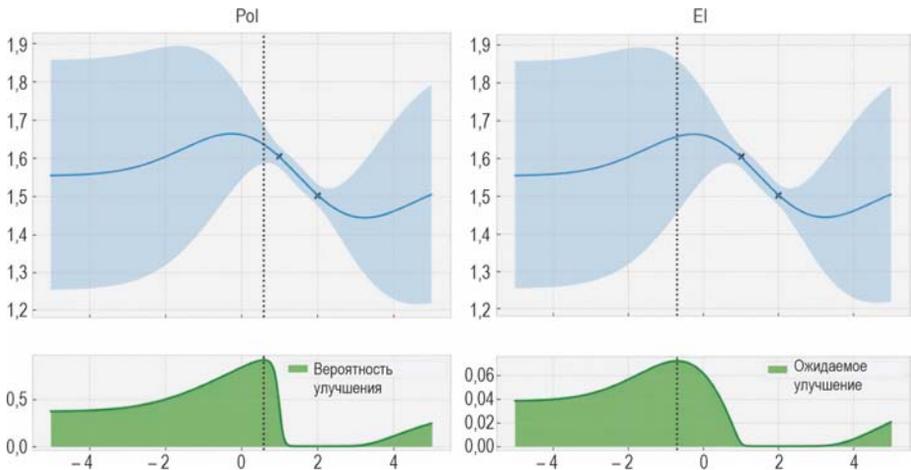


Рис. 4.16. Разница между Pol (слева) и EI (справа): EI лучше балансирует между исследованием и эксплуатацией

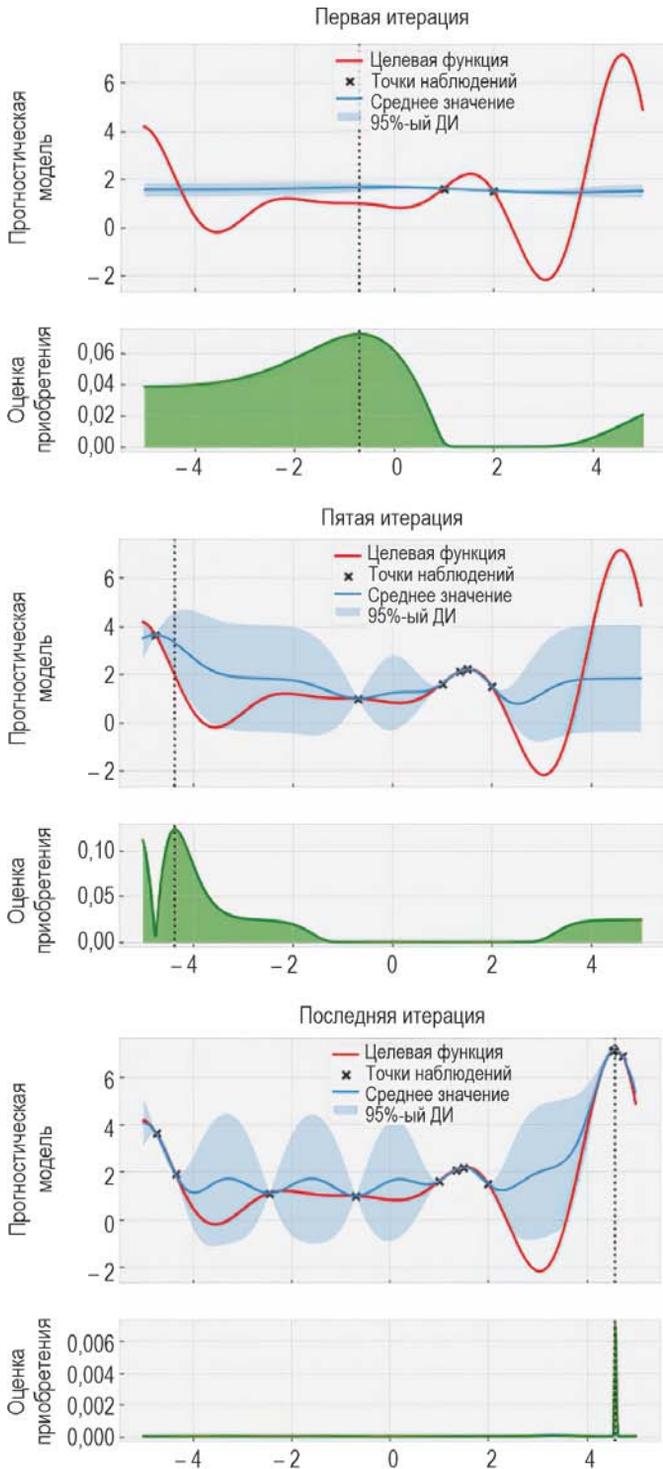


Рис. 4.17. Прогресс политики EI: она уравнивает исследование и эксплуатацию лучше, чем PoI, и в конце находит глобальный оптимум

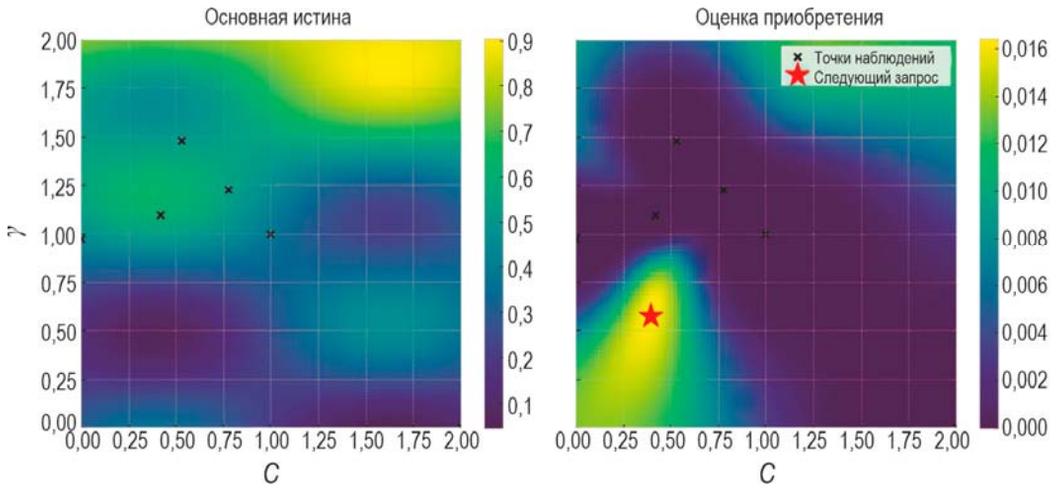


Рис. 4.18. Как должна выглядеть вспомогательная функция, которая визуализирует прогресс в БО: левая часть показывает функцию основной истины, в то время как правая часть демонстрирует оценку приобретения

Глава 5



Рис. 5.1. Игровой автомат с рычагом, который может возвращать монеты в зависимости от вероятности вознаграждения



Рис. 5.2. Ряд игровых автоматов, каждый из которых имеет свою вероятность вознаграждения



Рис. 5.3. Пример набора данных, показывающий дилемму исследования-эксплуатации: политика MAB должна выбирать между машиной с постоянно высоким уровнем успеха и машиной с неопределенной ставкой вознаграждения

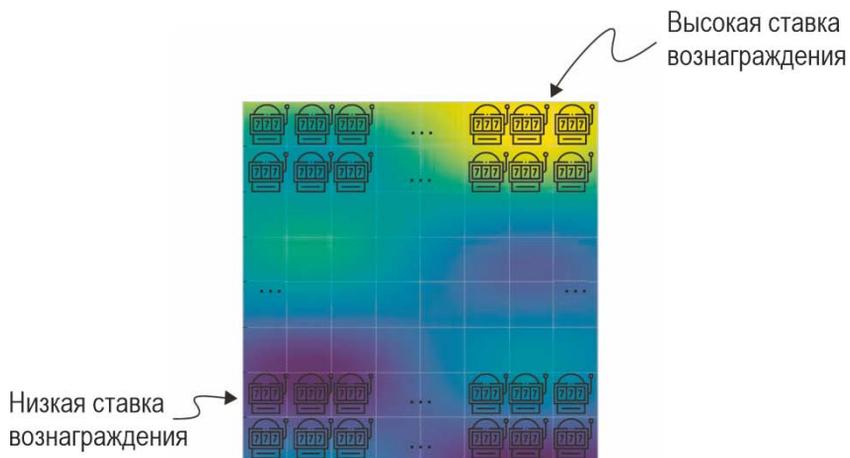


Рис. 5.4. БО аналогична MAB, но с бесконечным количеством действий: каждая бесконечно малая точка — это игровой автомат, за руку которого мы можем потянуть; кроме того, машины, расположенные близко друг к другу, коррелируют

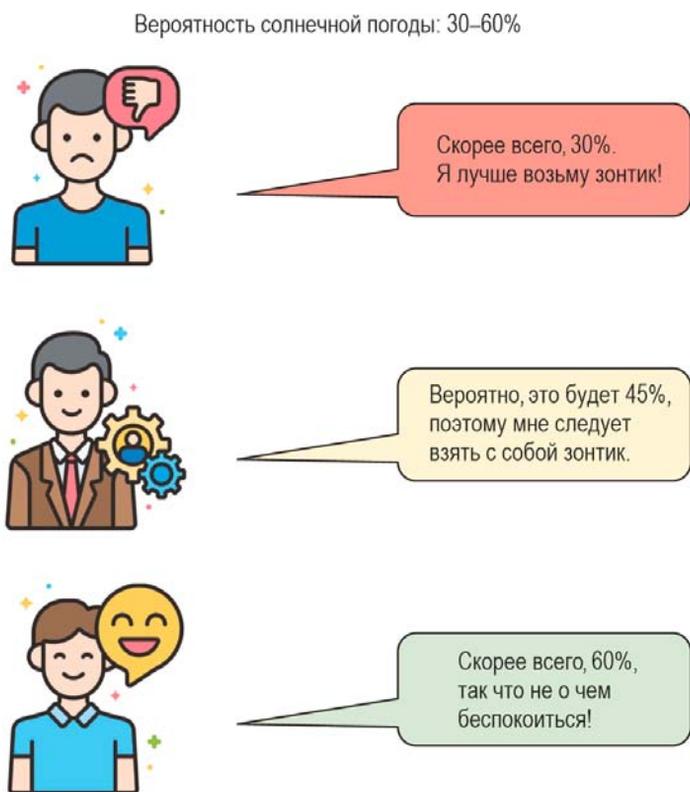


Рис. 5.5. Разные способы мышления при принятии решений: последний человек соответствует политике UCB, которая оптимистично оценивает неизвестную величину

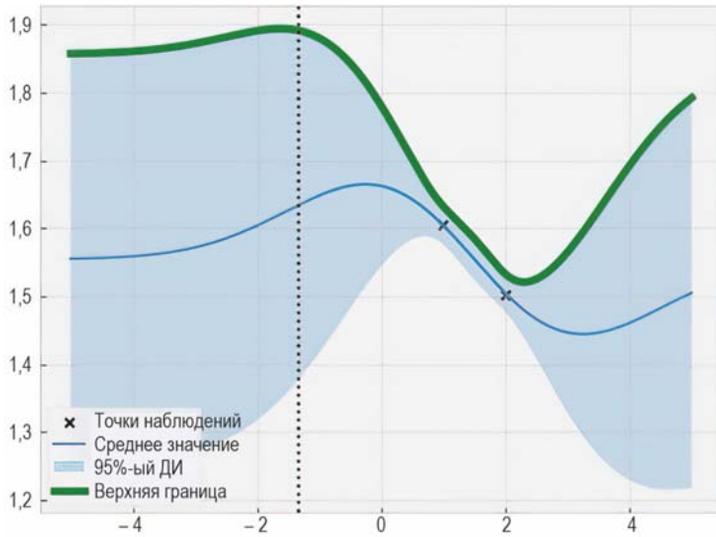


Рис. 5.6. Верхняя доверительная граница ГП, соответствующая 95%-ному ДИ, может быть использована в качестве оценки приобретения политики UCS

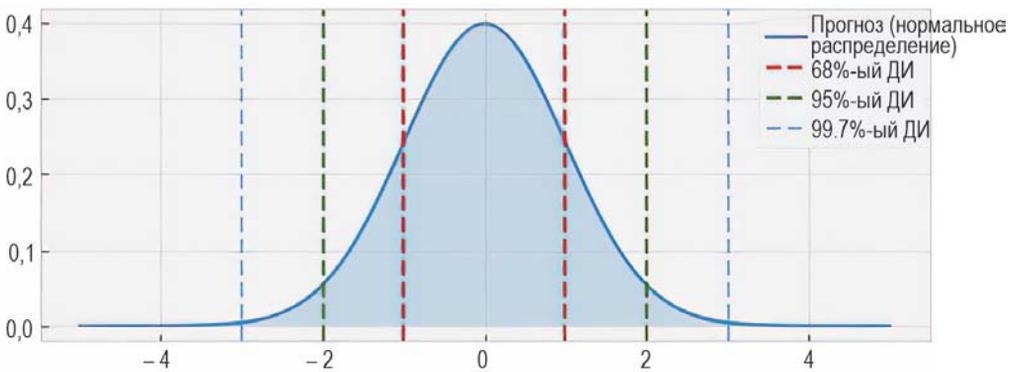


Рис. 5.7. Различные ДИ стандартных нормальных распределений: если отойти от среднего значения на одно, два и три стандартных отклонения, мы получим ДИ в 68, 95 и 99,7%; верхние границы этих интервалов используются политикой UCS

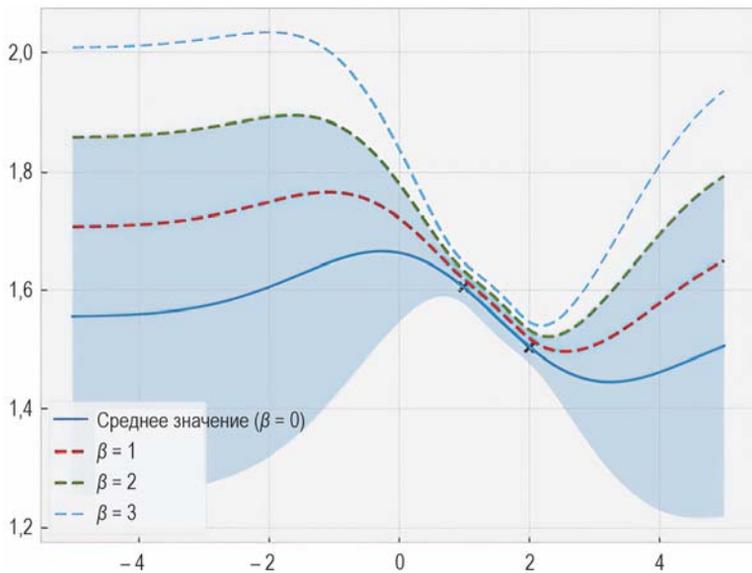


Рис. 5.8. Различные верхние границы ГП, соответствующие разным значениям ДИ и β : чем больше β , тем более исследовательской становится политика UCB

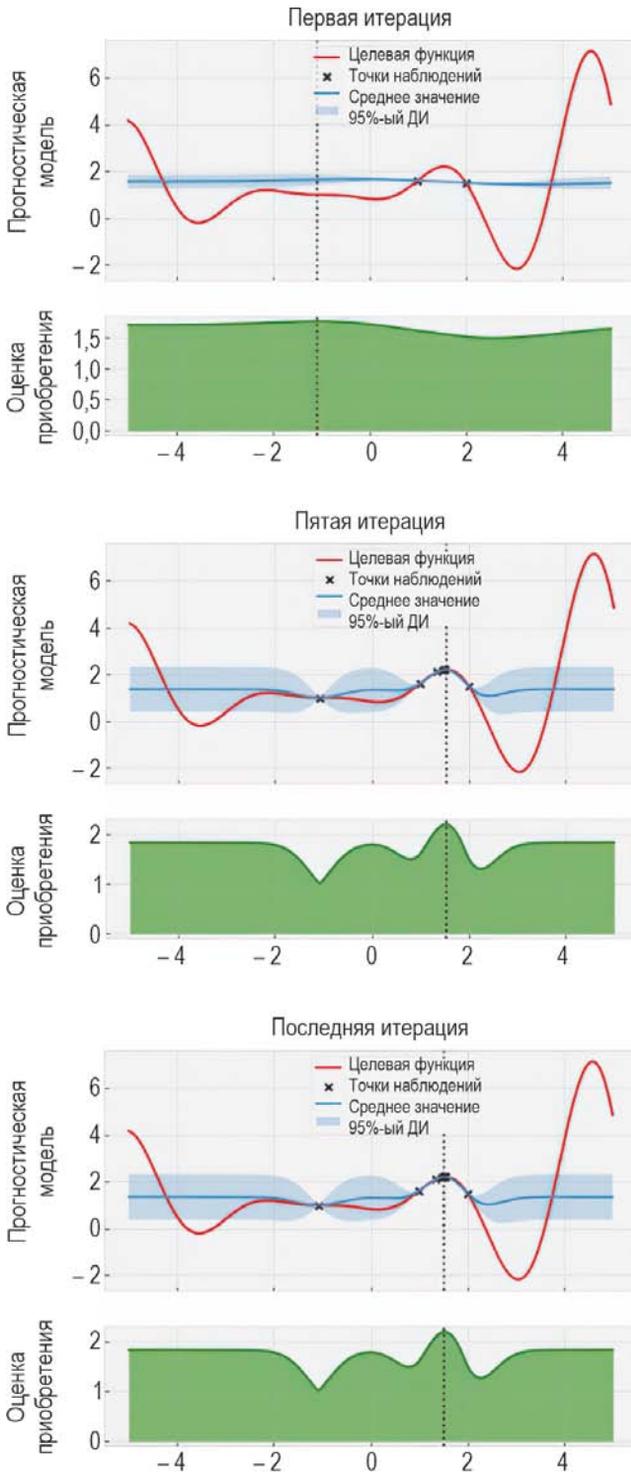


Рис. 5.9. Прогресс политики UCB с параметром $\beta = 1$: значение параметра недостаточно велико, чтобы стимулировать эксплуатацию, в результате чего прогресс застревает в локальном оптимуме

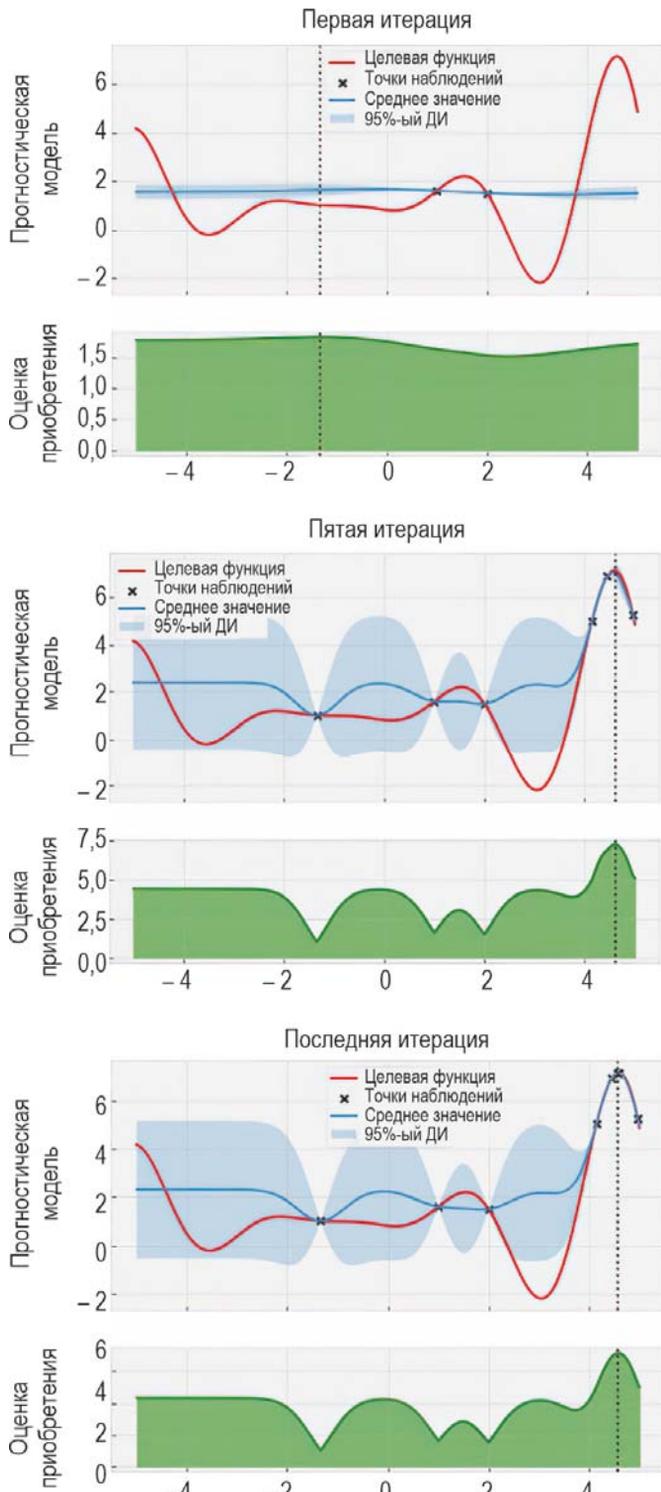


Рис. 5.10. Прогресс политики UCB с параметром $\beta = 2$: глобальный оптимум найден успешно

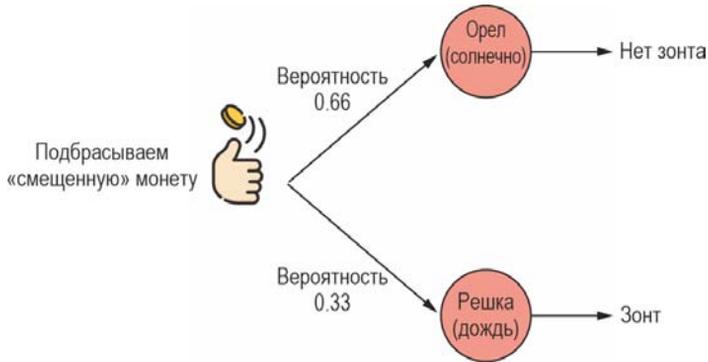


Рис. 5.11. Политика TS в виде дерева решений: мы подбрасываем монету, чтобы получить выборку распределения вероятностей солнечной погоды, и на ее основе решаем, брать ли с собой зонтик

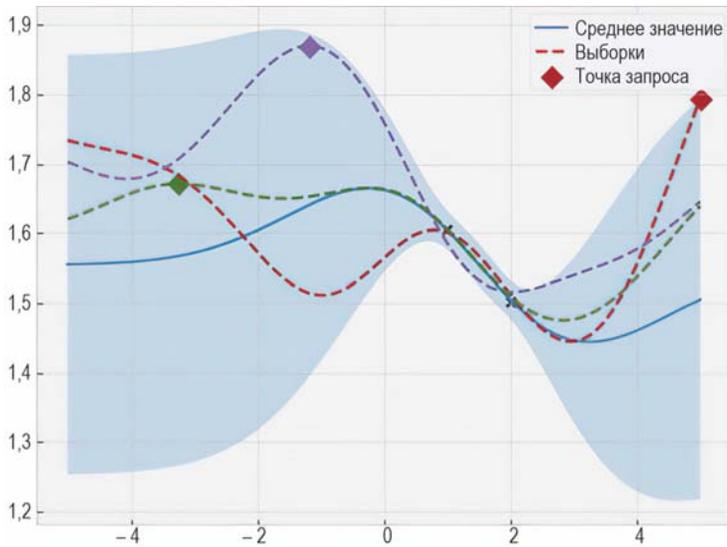
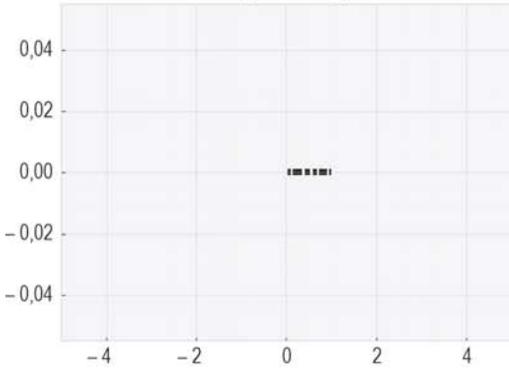
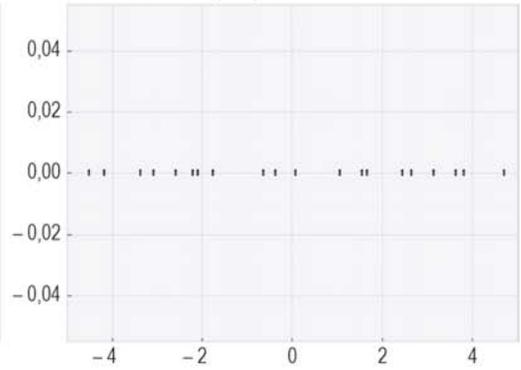


Рис. 5.12. Выборки из ГП и точки данных, максимизирующие их: какая бы выборка ни была взята, TS запросит точку, которая ее максимизирует, в качестве следующей для оценки целевой функции

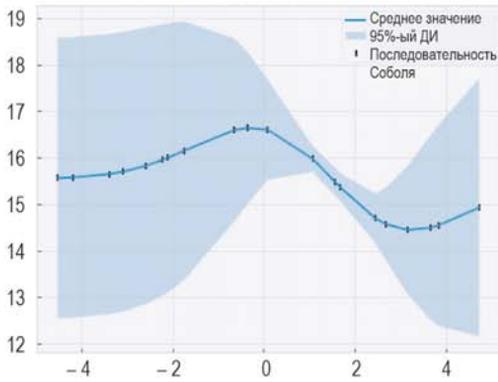
Шаг 1: Создать последовательность Соболя в единичном кубе



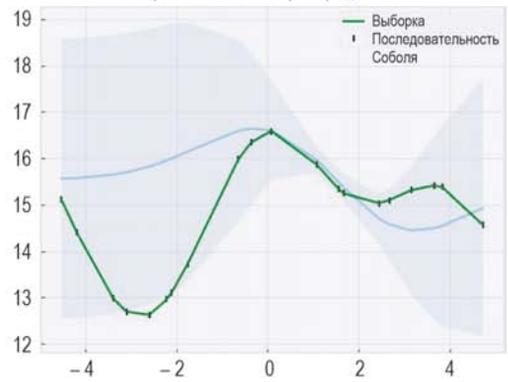
Шаг 2: Масштабировать последовательность Соболя на пространство поиска



Шаг 3: Вычислить прогнозы в точках Соболя



Шаг 4: Создать выборку из прогностического распределения



Шаг 5: Выбрать точку Соболя, которая максимизирует выборку для запроса

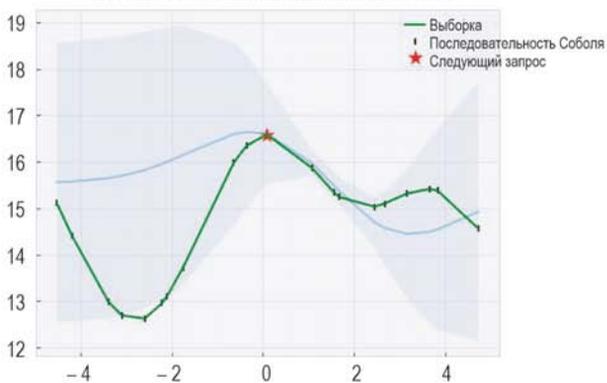


Рис. 5.13. Блок-схема реализации TS в VoTorch: мы используем последовательность Соболя для заполнения пространства поиска, извлекаем выборку из ГП и выбираем точку в последовательности, которая максимизирует выборку для оценки целевой функции

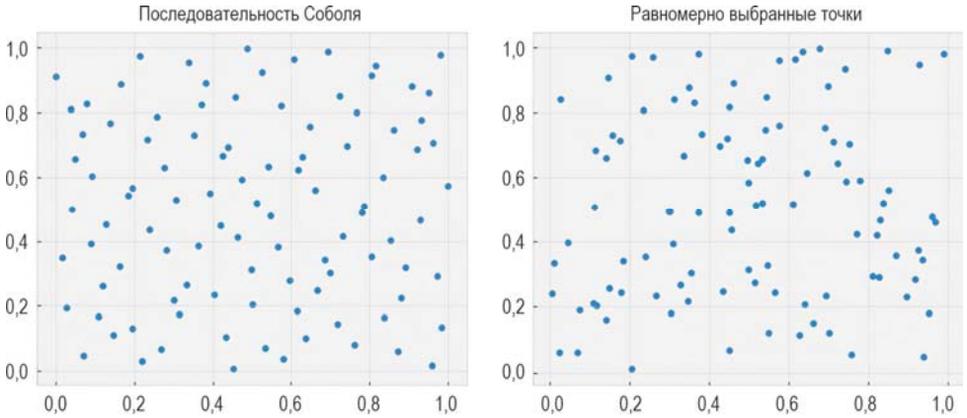


Рис. 5.14. Точки из последовательности Соболя и равномерно выбранные точки в двумерном единичном квадрате: последовательность Соболя покрывает квадрат более равномерно

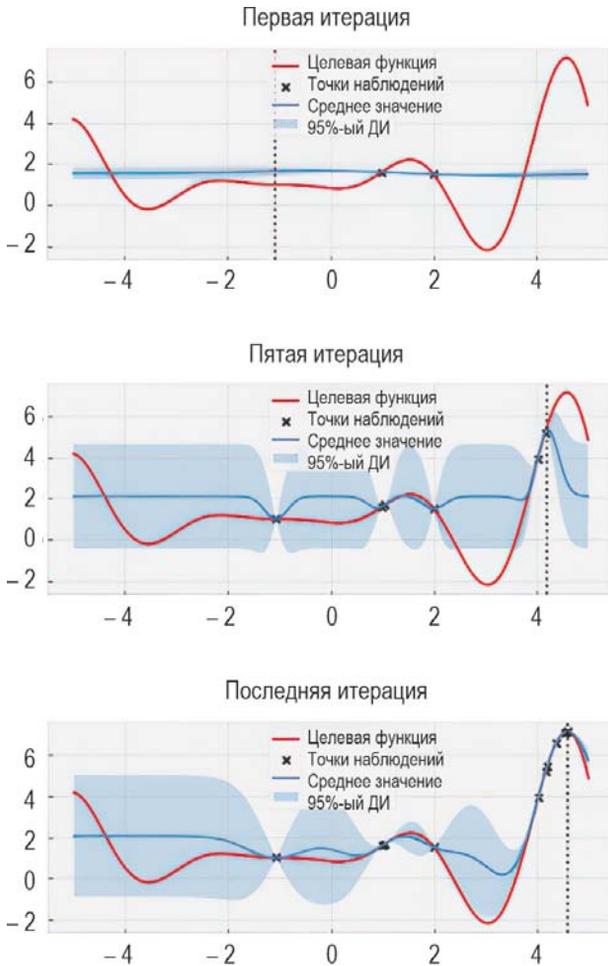


Рис. 5.15. Прогресс политики TS, которая исследует пространство поиска в течение нескольких итераций, а затем постепенно приближается к глобальному оптимуму

Глава 6

0/1

Один бит может представлять $2^1 = 2$ числа

0/1	0/1	0/1	0/1	0/1
-----	-----	-----	-----	-----

Два бита могут представлять $2^5 = 32$ числа

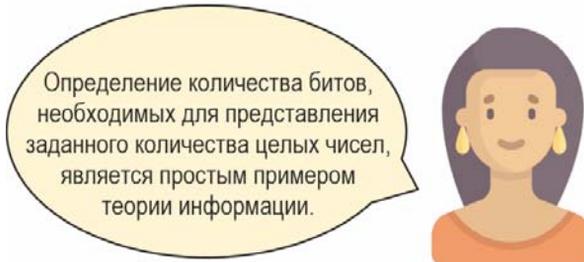


Рис. 6.1. Пример применения теории информации на практике: определение количества битов для представления определенного количества целых чисел

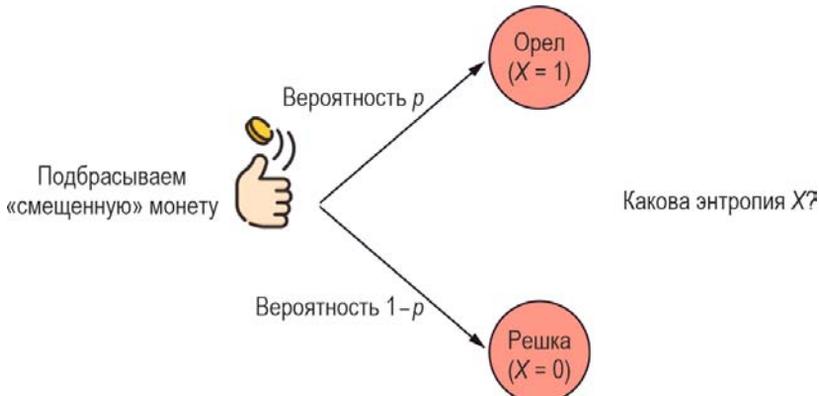


Рис. 6.2. Еще один пример применения теории информации на практике: определение вероятности выпадения «орла» или «решки» при подбрасывании монеты

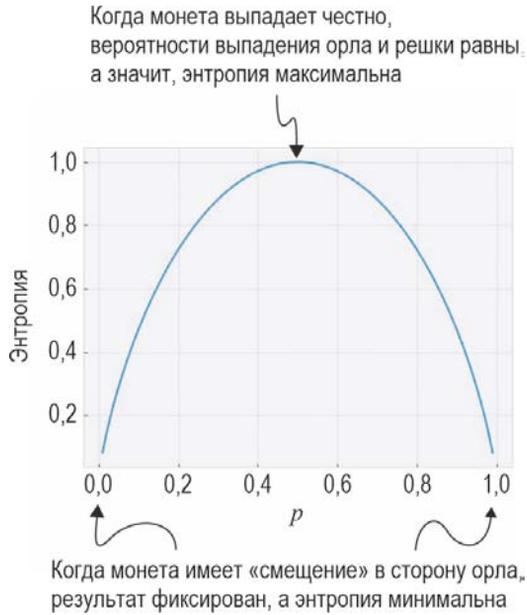


Рис. 6.3. Энтропия случайной величины Бернулли как функция вероятности успеха: энтропия максимальна (неопределённость максимальна), когда вероятность успеха равна 0,5



Рис. 6.4. План помещения в примере с поиском пульта ДУ: гостиная заштрихована равномерно, что указывает на равномерное распределение возможного нахождения пульта по всей площади



Рис. 6.5. Энтропия местоположения пульта после обыска части гостиной: если пульт найден (вверху справа), то никакой неопределенности не остается; в противном случае энтропия все равно уменьшится (внизу справа), поскольку распределение возможного местоположения пульта стало меньше



Рис. 6.6. Энтропия местоположения пульта после обыска ванной комнаты: поскольку маловероятно, что пульт будет в ванной, энтропия его апостериорного распределения не меняется

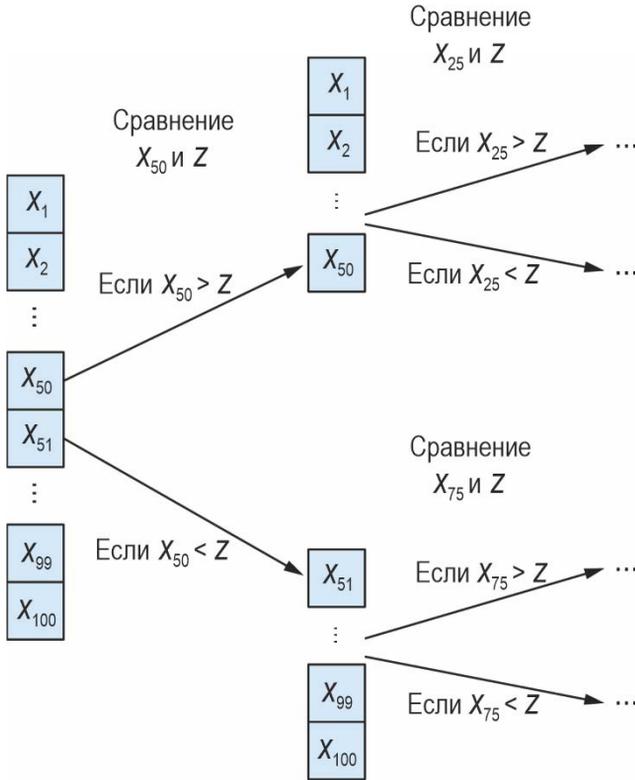


Рис. 6.7. Иллюстрация двоичного поиска в списке из 100 элементов: на каждой итерации целевое значение сравнивается со средним элементом текущего списка; в зависимости от результата из пространства поиска удаляется либо первая, либо вторая половина списка

Вероятность, что Z находится
в определенном месте

1% 1% 1%

1% 1%

1% 1% 1%



Рис. 6.8. Предварительное распределение Z в списке из 100 элементов: поскольку каждое местоположение одинаково вероятно, как и любое другое, вероятность, что данное местоположение содержит Z , составляет 1%

Вероятность, что Z находится
в определенном месте

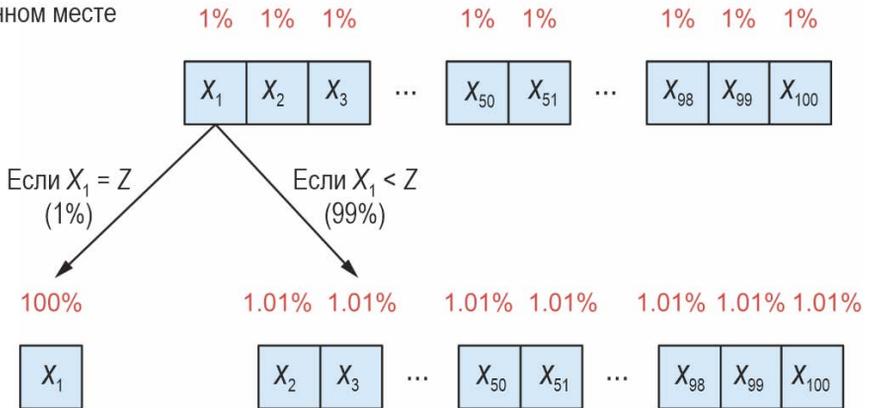


Рис. 6.9. Апостериорные распределения местоположения z в списке после проверки первого элемента: в каждом сценарии вероятность, что z находится в определенном месте, соответствующим образом обновляется

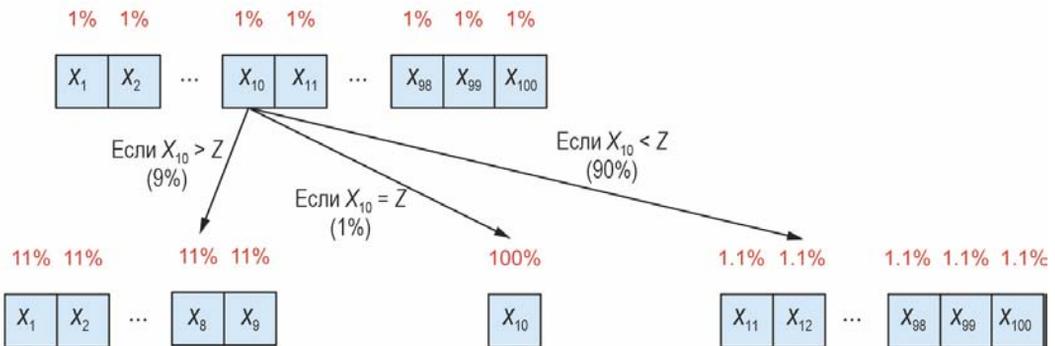


Рис. 6.10. Апостериорные распределения положения z в списке при проверке 10-го элемента: в каждом сценарии вероятность, что z находится в определенном месте, соответствующим образом обновляется

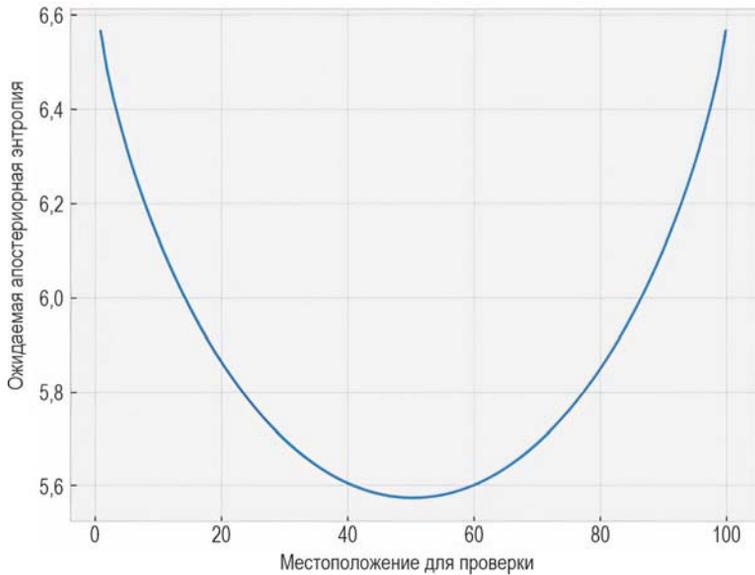


Рис. 6.11. Ожидаемая апостериорная энтропия для z в зависимости от расположения в списке для проверки: место посередине является оптимальным и минимизирует ожидаемую энтропию

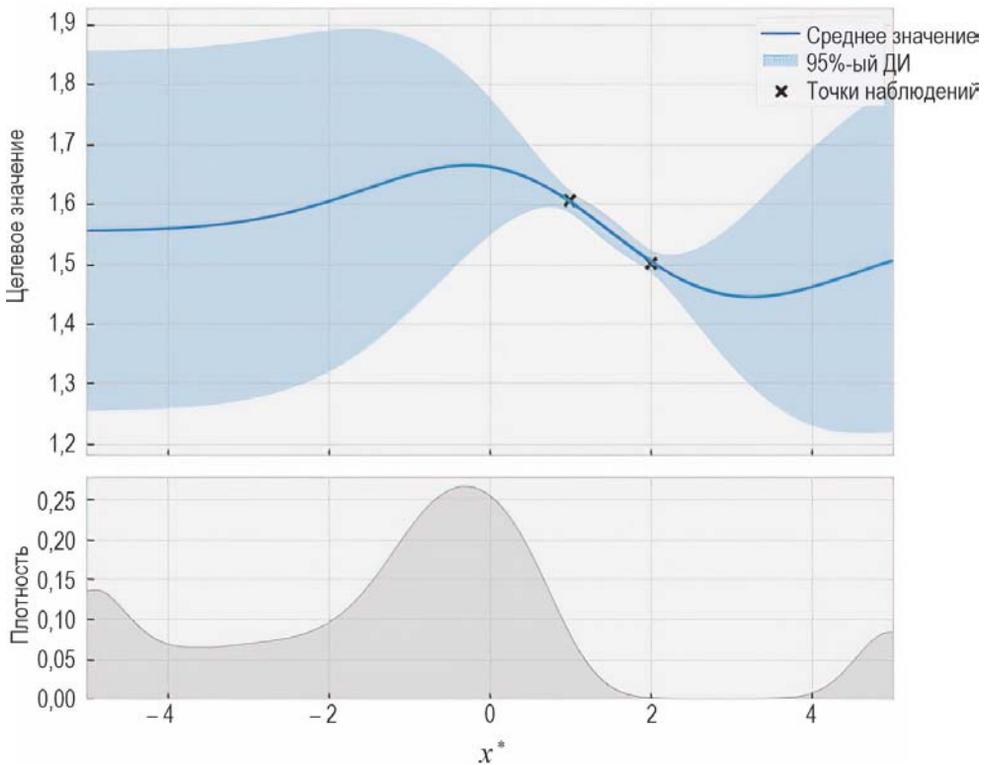


Рис. 6.12. Предположение ГП (вверху) и распределение оптимизатора функции x^* (внизу): распределение оптимизатора является негауссовым и довольно сложным, что затрудняет моделирование и принятие решений

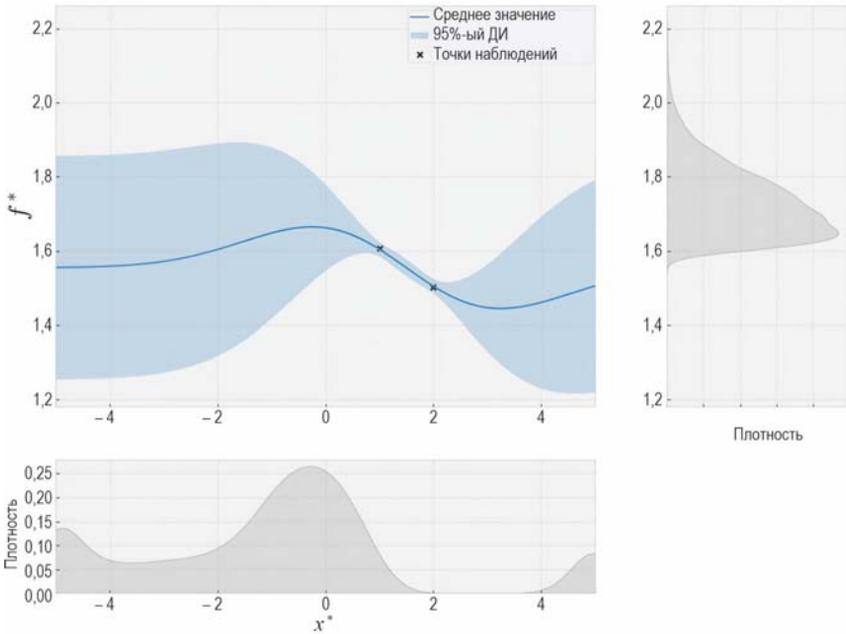


Рис. 6.13. Убеждение ГП (вверху слева), распределение оптимизатора x^* (внизу) и распределение оптимального значения f^* (справа): распределение f^* всегда одномерно и, следовательно, с ним проще работать, чем с оптимизатором

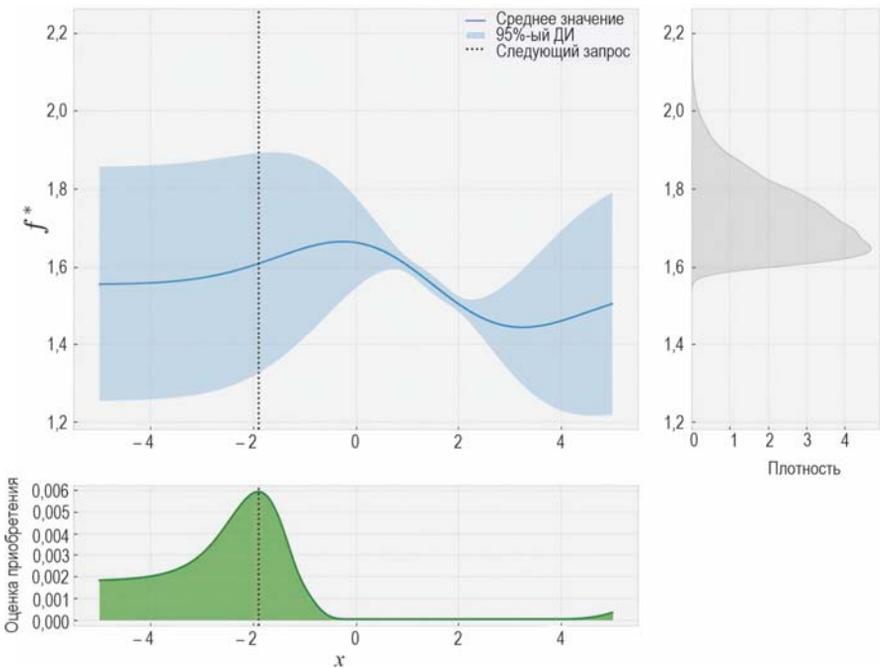


Рис. 6.14. Убеждение ГП (вверху слева), распределение оптимального значения f^* (справа) и приблизительное ожидаемое снижение энтропии, которое используется в качестве оценки приобретения (внизу): f^* всегда одномерно и, следовательно, с ним проще работать, чем с оптимизатором

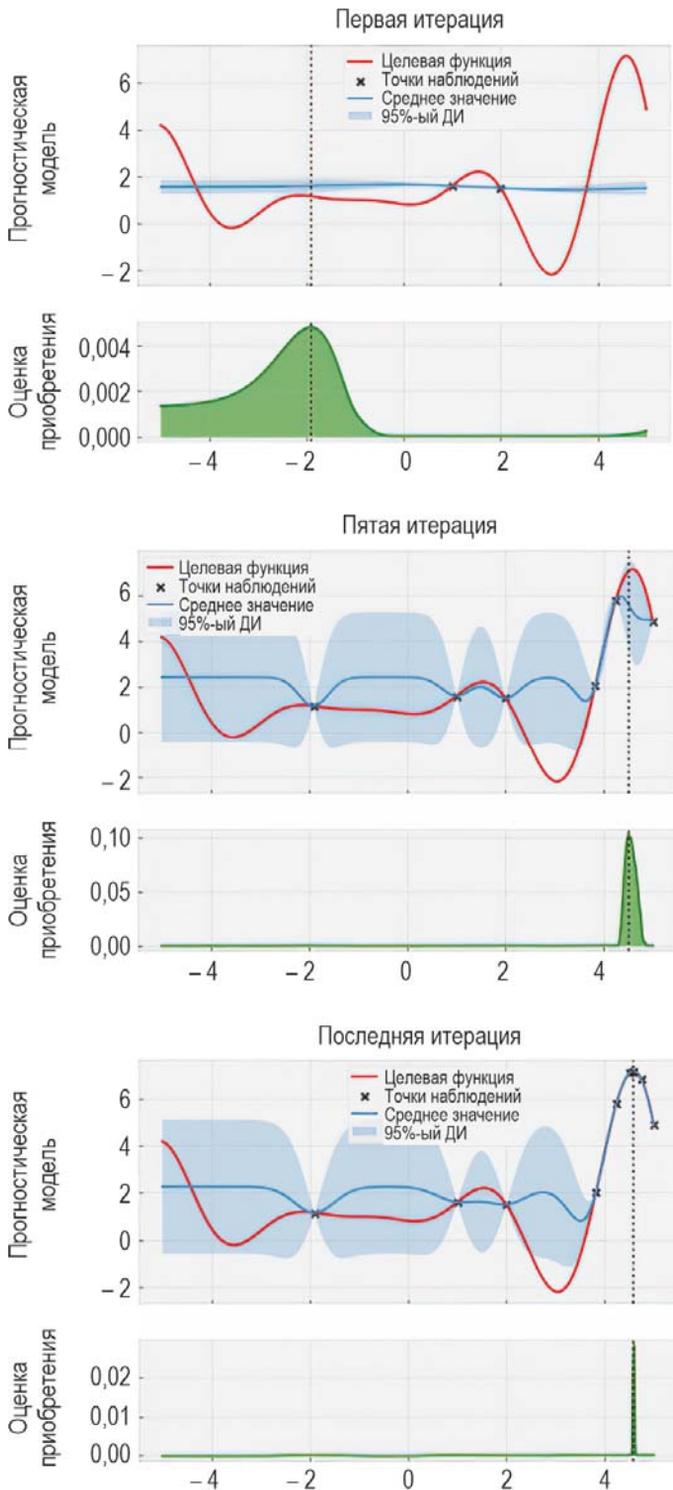


Рис. 6.15. Прогресс политики MES: она быстро находит глобальный оптимум после пяти запросов

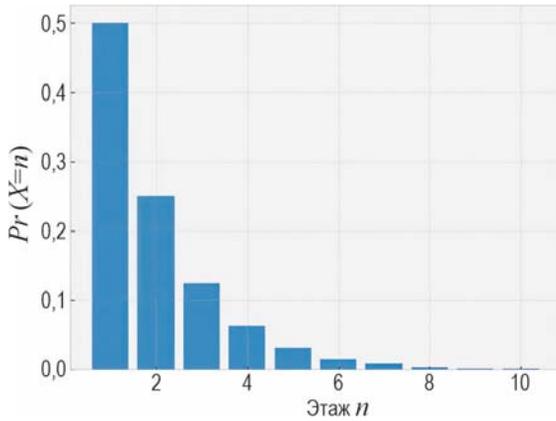


Рис. 6.16. Вероятность, что X равен числу от 1 до 10 (вероятность, что этаж является самым высоким, при падении с которого телефон не сломается)

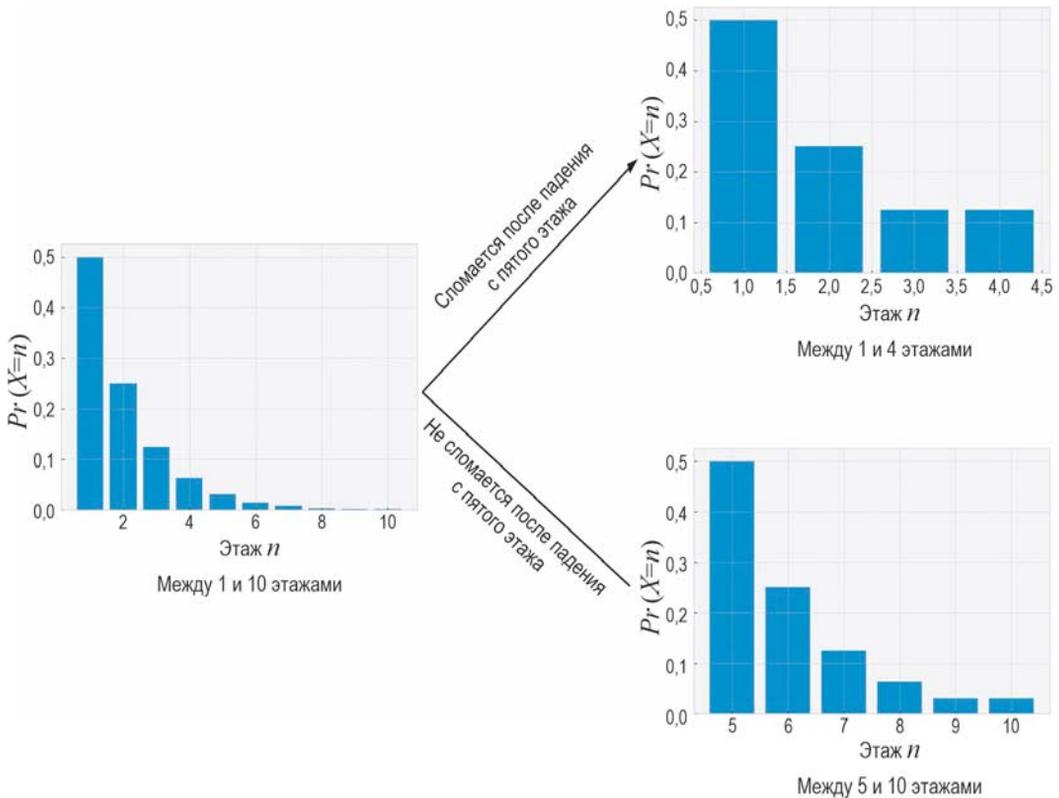


Рис. 6.17. Апостериорные распределения вероятностей для X в двух разных сценариях, когда телефон упал с пятого этажа: каждое апостериорное распределение по-прежнему является экспоненциальным

Глава 7

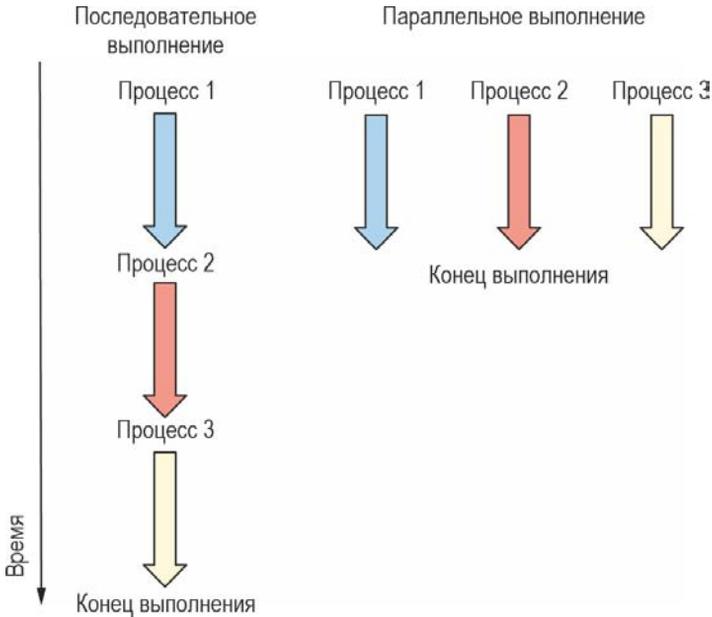


Рис. 7.1. Иллюстрация преимуществ параллелизма: три процесса выполняются либо последовательно (слева), либо параллельно (справа); при параллельной работе эти процессы занимают лишь одну треть от общего времени, необходимого при последовательном запуске

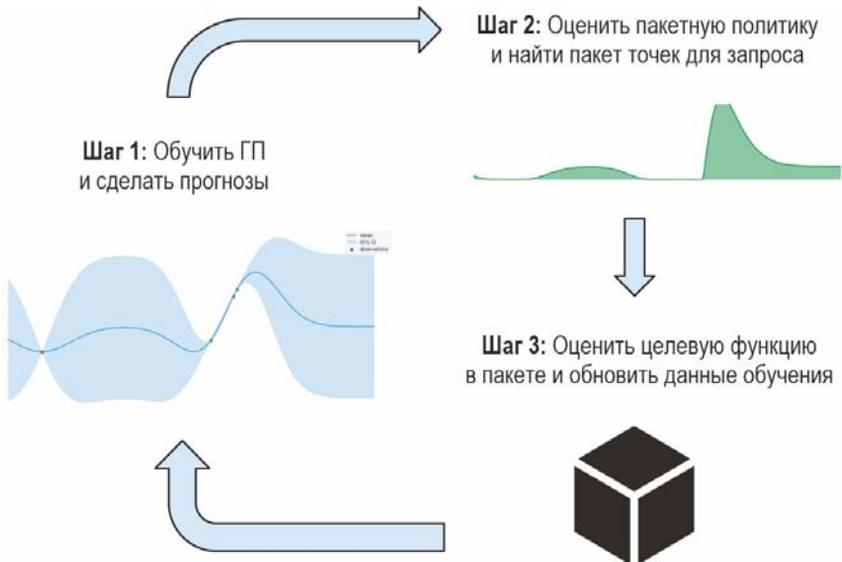


Рис. 7.2. Пакетный цикл БО: по сравнению с последовательным, пакетный вариант требует идентификации нескольких точек запроса на шаге 2 и одновременно оценивает целевую функцию в этих точках на шаге 3

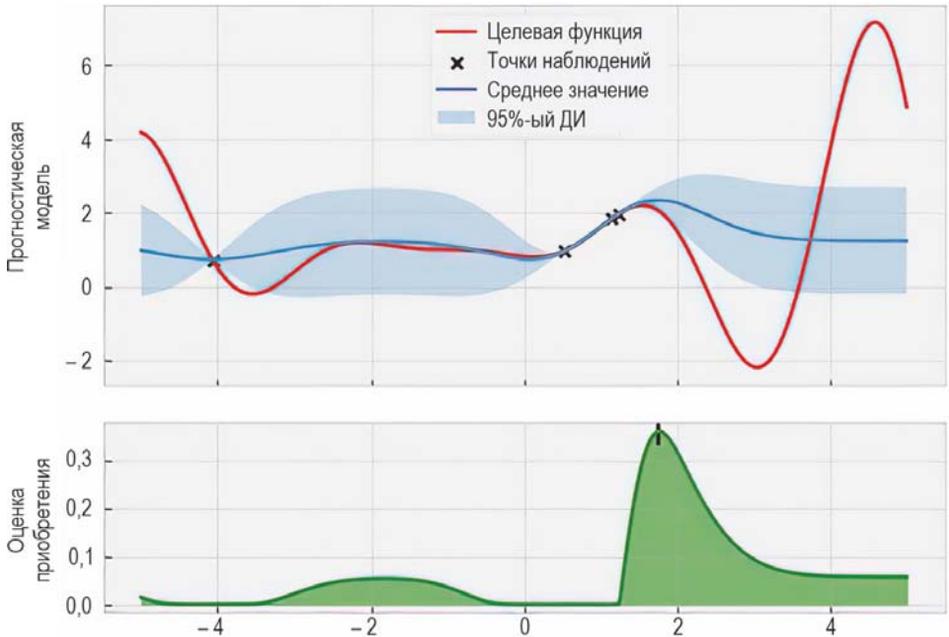


Рис. 7.3. Пример БО: на верхней части показаны прогнозы ГП и основная истинная целевая функция, а на нижней — оценки приобретения, полученные EI; вертикальная отметка на нижней кривой в области 1,75 указывает на точку, которая будет запрошена следующей

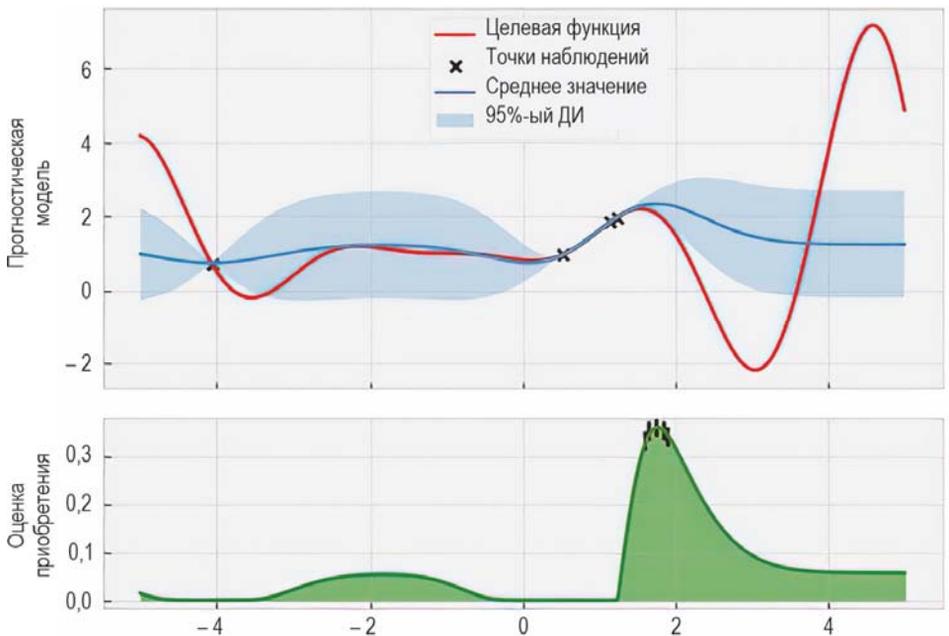
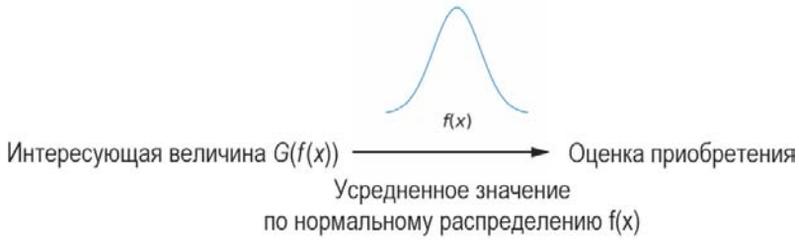


Рис. 7.4. Запросы, сделанные в пакетном режиме: мы просто выбираем точки с наивысшими оценками приобретения (обозначены вертикальными отметками на нижней кривой); эти запросы расположены близко друг к другу и менее полезны, чем если бы они были разбросаны

Последовательные политики



Пакетные политики



Рис. 7.5. Распространение математической формулировки политики на пакетный вариант: в обоих случаях используется среднее значение интересующей величины; мы берем максимальное значение по точкам в пакете, а затем находим среднее значение, представляющее полезность всего пакета

Выбор олимпийской сборной



Пакетная БО

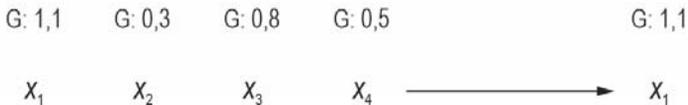


Рис. 7.6. Пакетная эвристика БО выбирает лучший элемент с наибольшим значением G , чтобы представить весь пакет (внизу): эта стратегия аналогична отбору команд на Олимпийских играх, где остаются только лучшие спортсмены, которые будут представлять страну

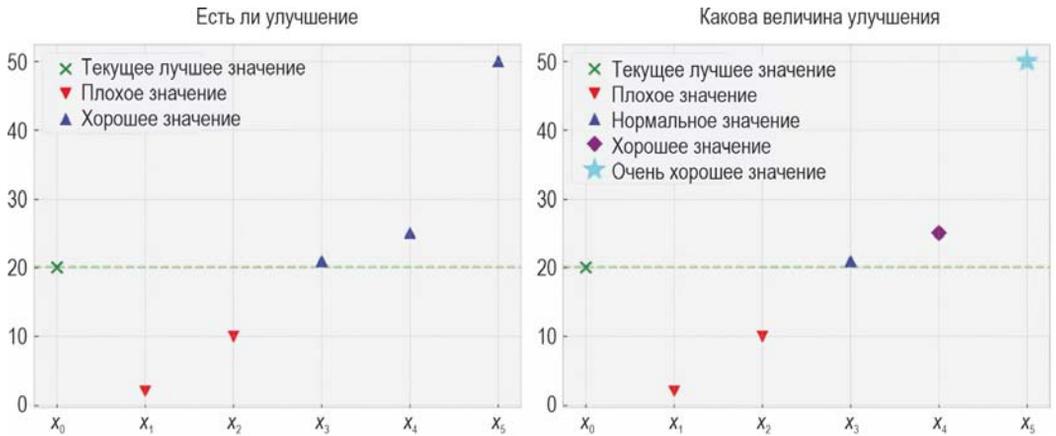


Рис. 7.7. Разница между PoI (слева) и EI (справа): первая политика учитывает только, есть ли улучшения по сравнению с действующей точкой или нет, а вторая — какова величина этих улучшений

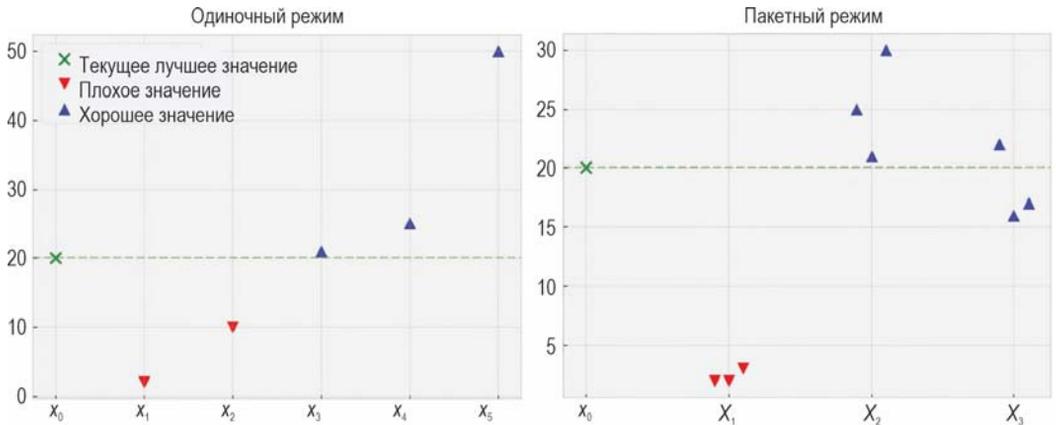
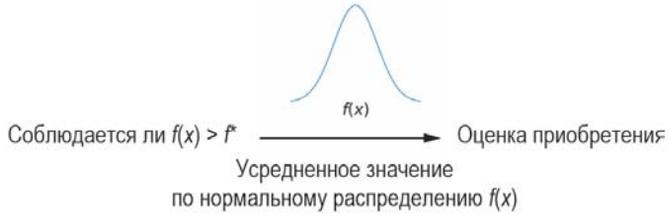


Рис. 7.8. Иллюстрация, приводит ли один запрос (слева) или пакет запросов (справа) к улучшению по сравнению с действующим значением: в пакетной настройке учитывается только максимальное значение в каждом пакете, чтобы определить, есть ли улучшение

Последовательные политики



Пакетные политики

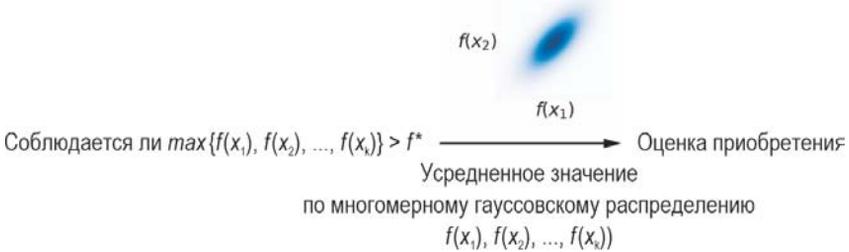
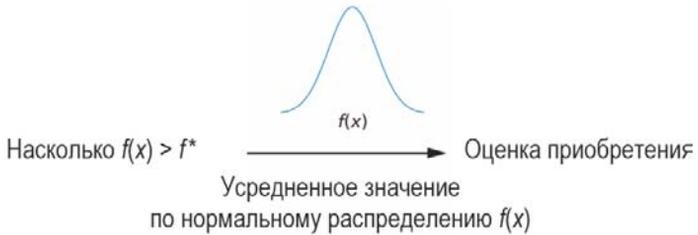


Рис. 7.9. Расширение политики POI для пакетного режима: в последовательном случае (вверху) мы рассматриваем, улучшается ли следующий запрос по сравнению с действующей точкой; в пакетном варианте (внизу) мы рассуждаем, улучшается ли максимальное значение по точкам из набора по сравнению с действующей точкой

Последовательные политики



Пакетные политики



Рис. 7.10. Расширение политики EI для пакетного режима: в последовательном случае (вверху) используется среднее значение того, насколько следующий запрос улучшается по сравнению с действующим; в пакетной настройке (внизу) мы берем среднее значение того, насколько максимальное значение по всем точкам в пакете улучшается по сравнению с действующей точкой

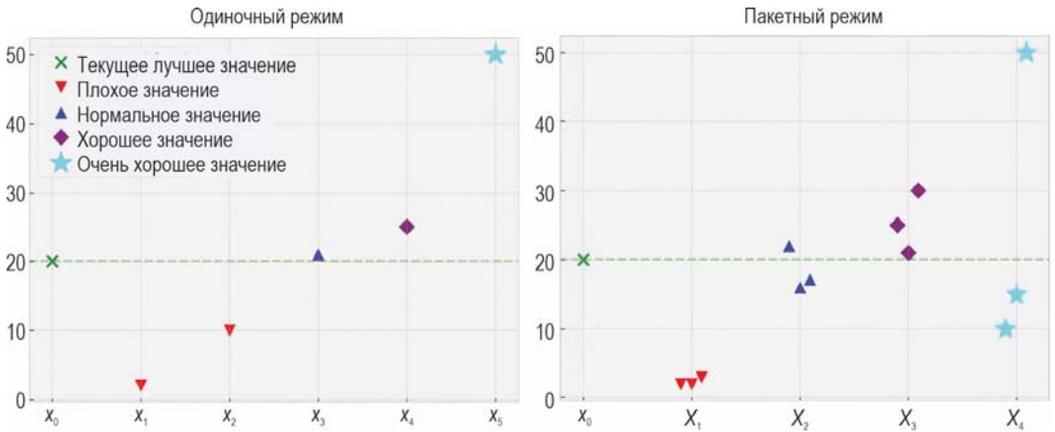


Рис. 7.11. Иллюстрация, приводит ли запрос (слева) или пакет запросов (справа) к улучшению по сравнению с действующим значением: в пакетном варианте (справа) мы учитываем только максимальное значение в каждом пакете, чтобы определить, есть ли улучшение

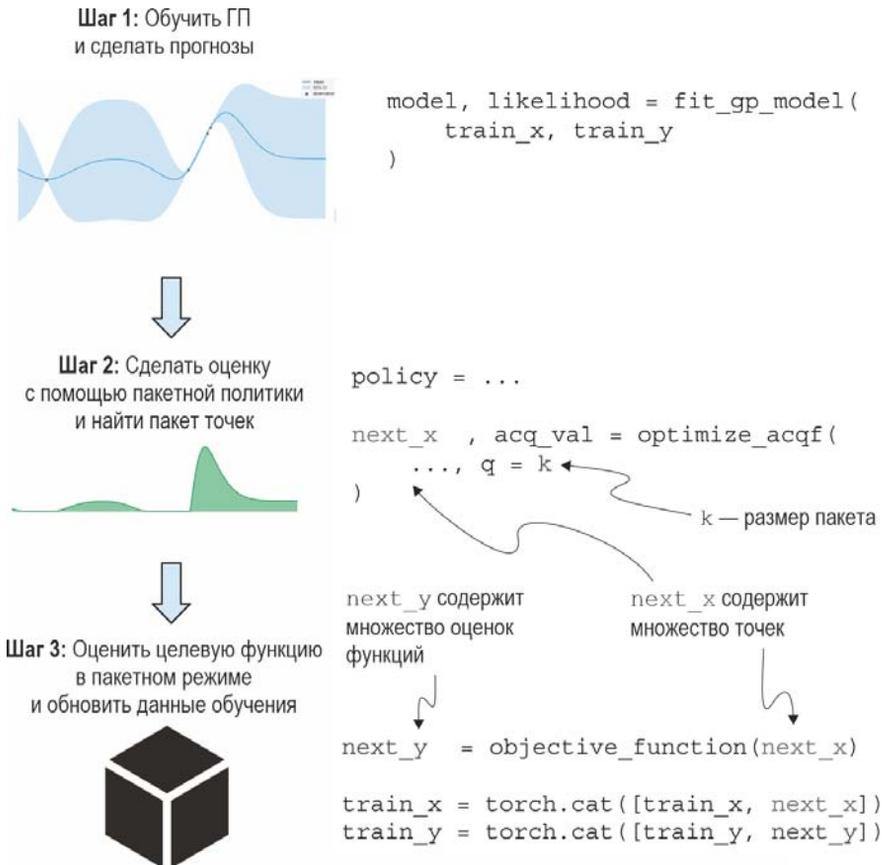


Рис. 7.12. Шаги пакетного цикла БО и соответствующий код: по сравнению с последовательным режимом изменения в коде минимальны

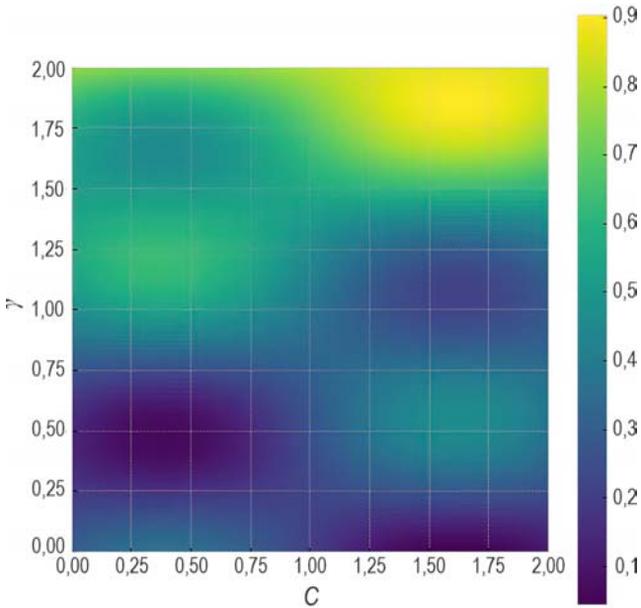


Рис. 7.13. Точность модели SVM на наборе обучающих данных как функция штрафного параметра C и параметра γ ядра RBF: эту целевую функцию мы стремимся оптимизировать в данной главе

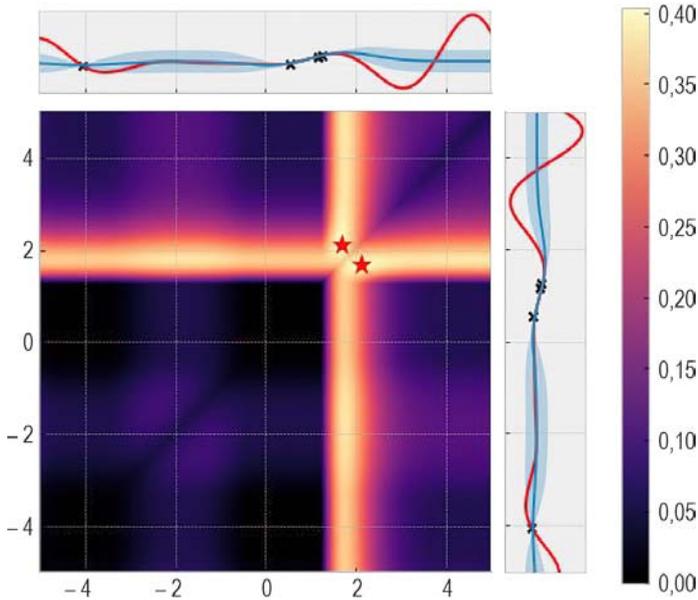


Рис. 7.14. Тепловая карта, показывающая оценки приобретения пакетной политики EI для пакетов размером 2 с одномерной целевой функцией: верхняя и правая части рисунка показывают наблюдаемые данные и текущее предположение ГП о целевой функции по осям тепловой карты; две оптимальные пары запросов, обозначенные звездочками, содержат значения 1,68 и 2,12, которые находятся относительно далеко друг от друга

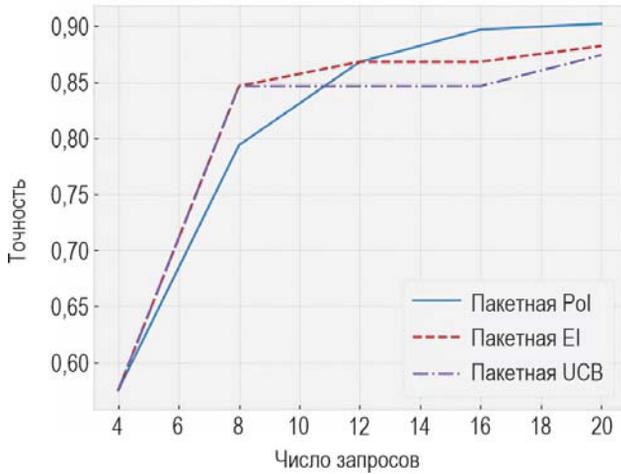


Рис. 7.15. Прогресс, достигнутый различными пакетными политиками БО для настройки гиперпараметров: прогресс выполняется пакетами по четыре, что соответствует используемому размеру

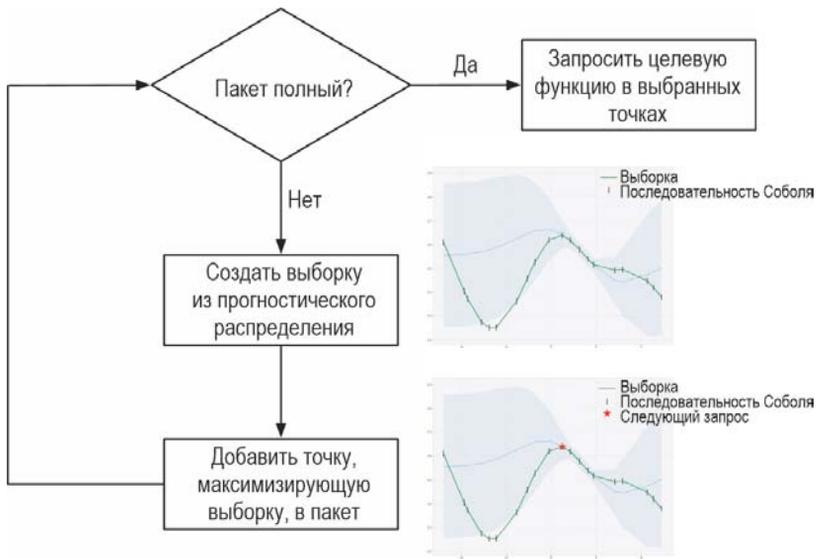


Рис. 7.16. Блок-схема реализации пакетной версии TS: мы делаем выборки из ГП и добавляем точку, которая максимизирует выборку, к текущему пакету, пока он не заполнится

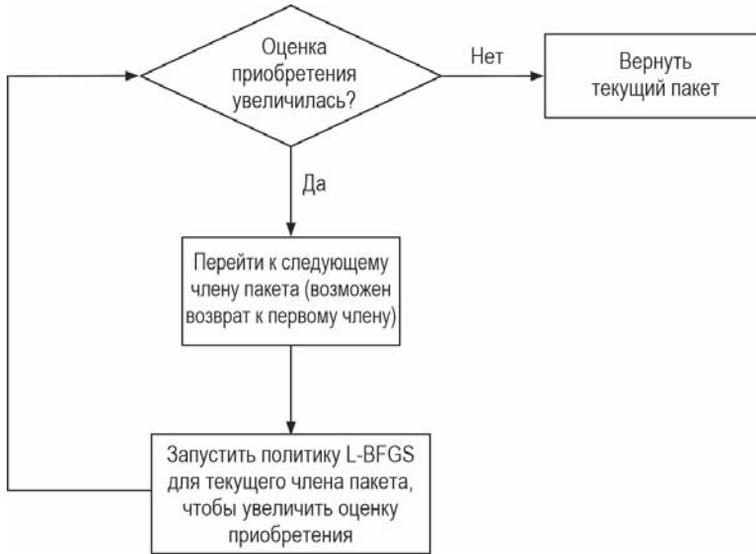


Рис. 7.17. Блок-схема циклической оптимизации, используемая при поиске пакета, который максимизирует информацию о наибольшем целевом значении в пакетной политике MES: процедура является циклической, поскольку мы последовательно уточняем каждый член пакета, пока не достигнем хорошей оценки приобретения

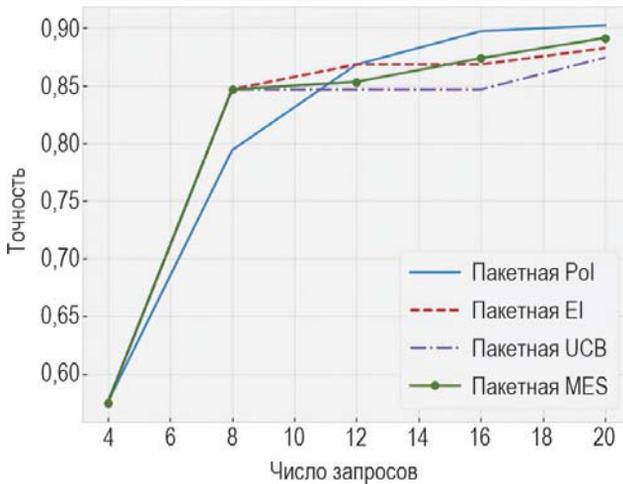


Рис. 7.18. Прогресс, достигнутый различными пакетными политиками в задаче настройки гиперпараметров (включая MES)

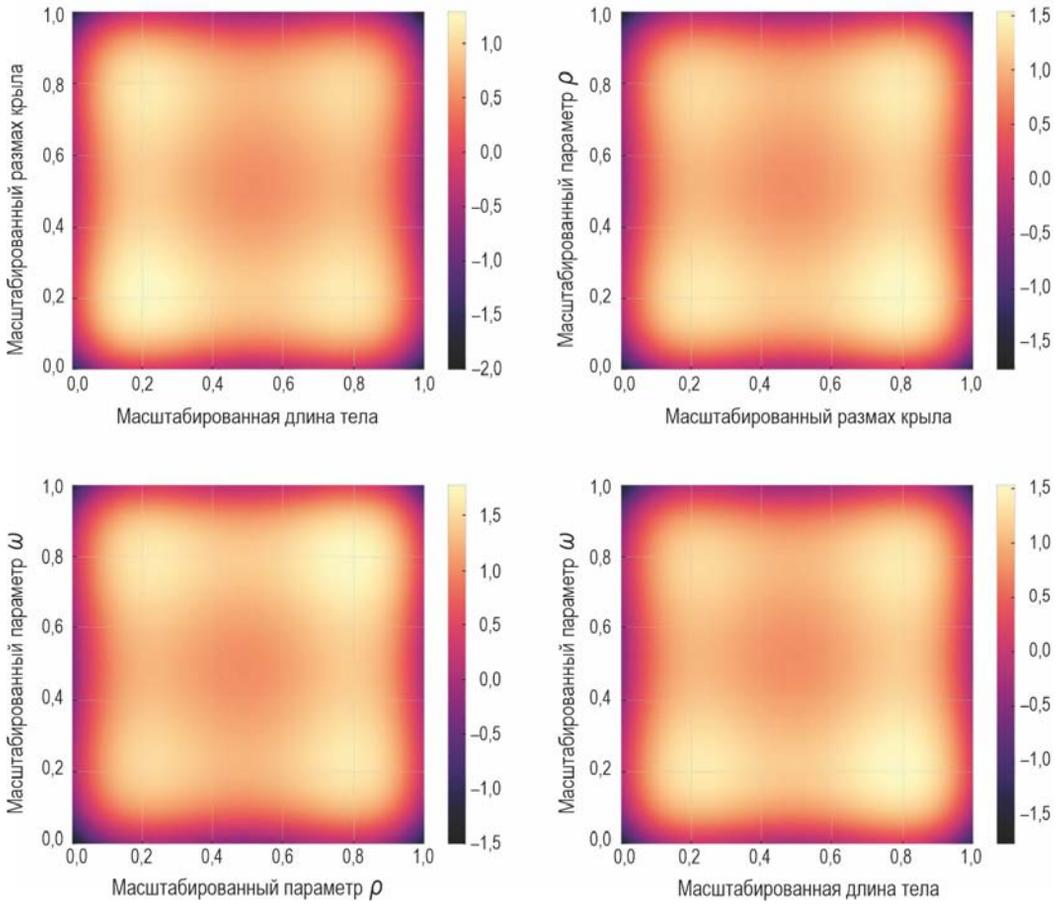
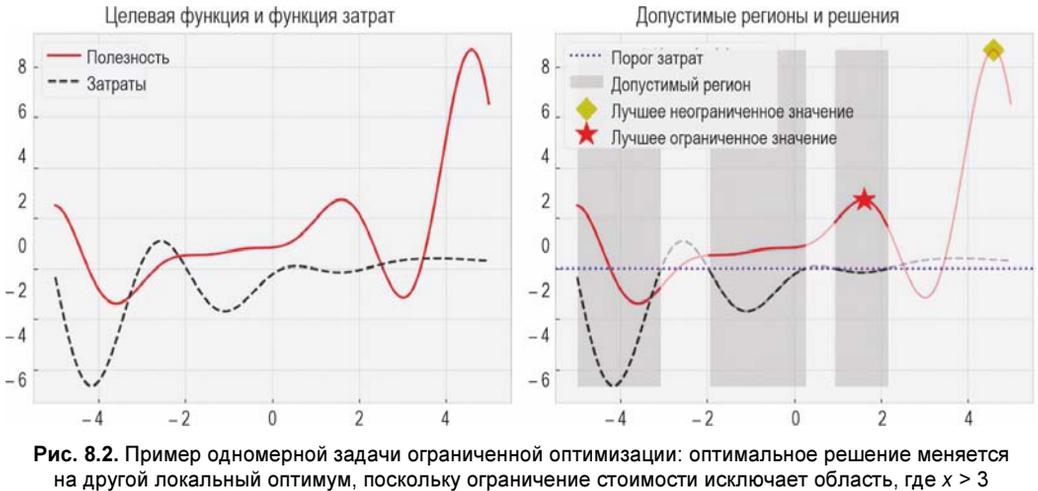
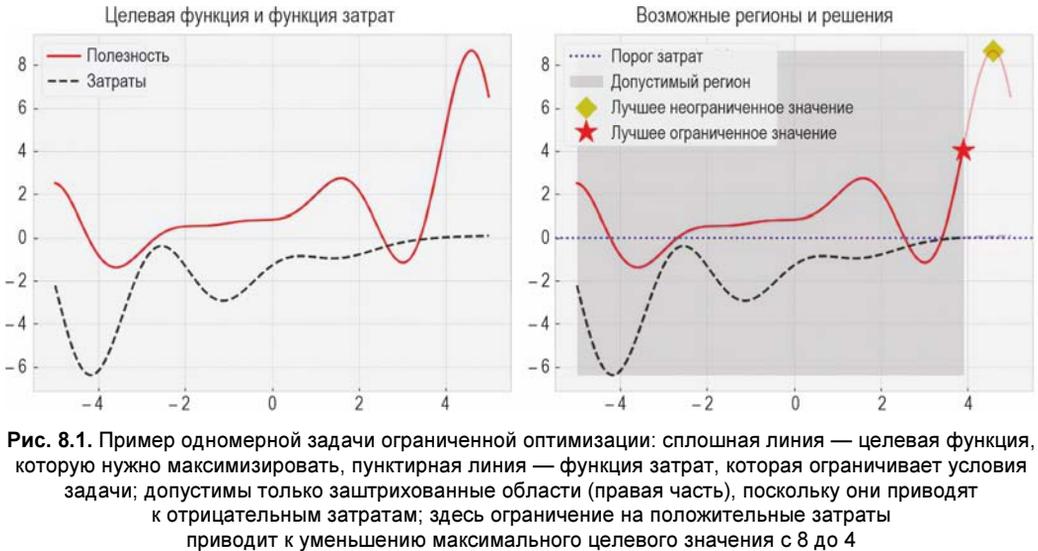


Рис. 7.19. Целевая функция моделируемой задачи по оптимизации конструкции самолета в различных двумерных подпространствах (парах настраиваемых параметров), которые показаны в виде осевых меток: яркие пятна указывают на высокие искомые значения, которые являются нашими целями оптимизации

Глава 8



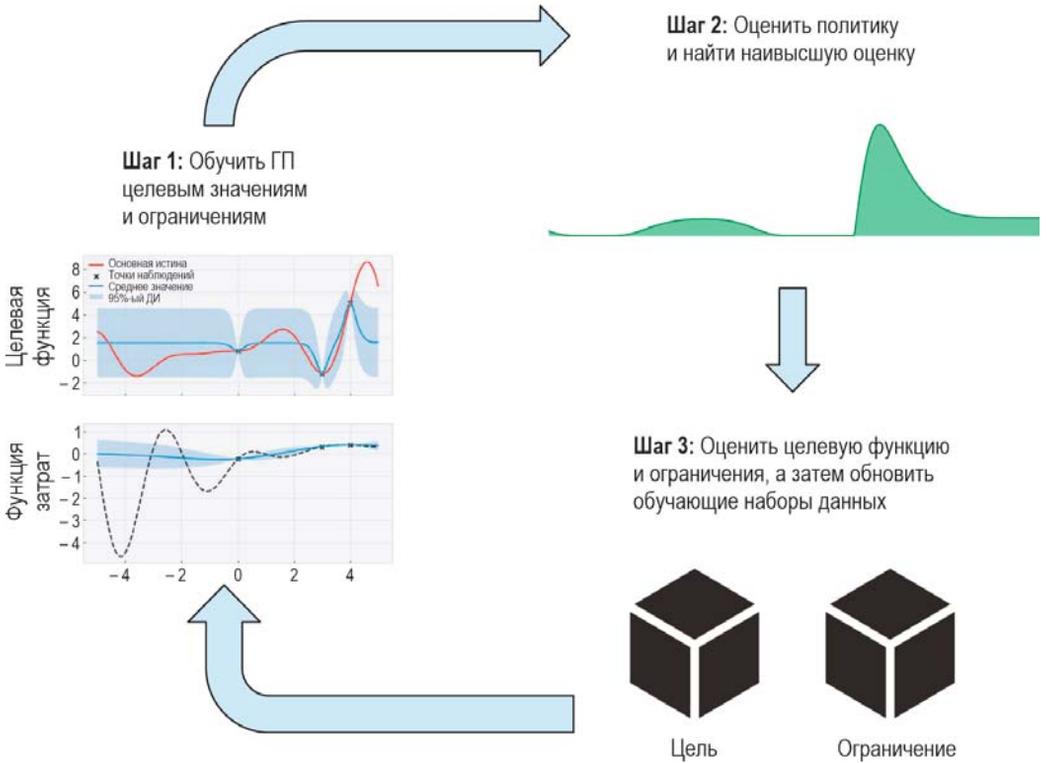


Рис. 8.3. Цикл БО с ограничениями: все функции (целевая и ограничительные) имеют свой ГП; политика рекомендует на следующем шаге запросить как целевую функцию, так и функции, определяющие ограничения

Оценка приобретения для ограниченной EI = Оценка приобретения EI × Вероятность допустимости

- Отдает предпочтение точкам с высокими целевыми значениями
- Рассчитывается на основе ГП, моделирующего целевую функцию
- Отдает предпочтение допустимым точкам
- Рассчитывается на основе ГП, моделирующего ограничения

Рис. 8.4. Формула для оценки приобретения ограниченной EI, которая выражается как произведение оценки приобретения для обычной EI и вероятности допустимости; политика направлена на оптимизацию целевого значения и удовлетворение ограничений одновременно

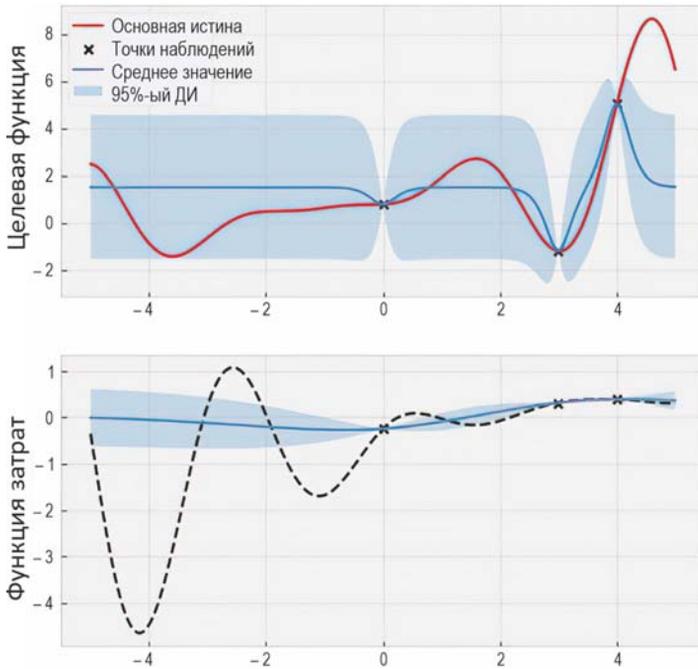


Рис. 8.5. Прогнозы относительно целевой и затратной функций, сделанные соответствующими ГП: каждый ГП позволяет рассуждать о форме каждой из функций вероятностным образом

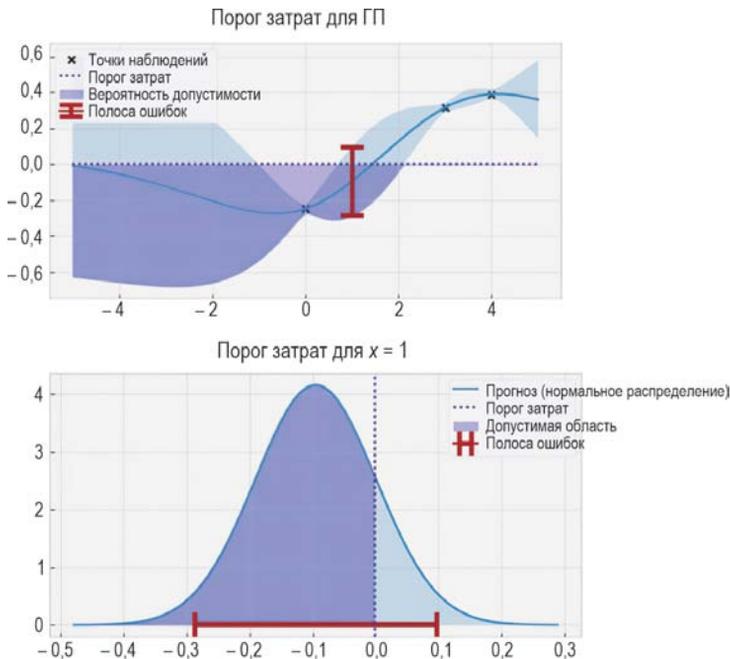


Рис. 8.6. Вероятность, что $x = 1$ является допустимой точкой, выделена более темным оттенком: на левой части показан весь ГП, а на правой — только нормальное распределение, соответствующее прогнозу при $x = 1$ (шкалы ошибок одинаковы на обеих частях); здесь допустимость соответствует усеченному нормальному распределению

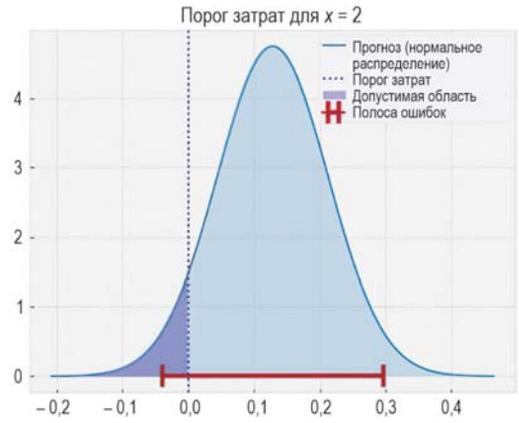
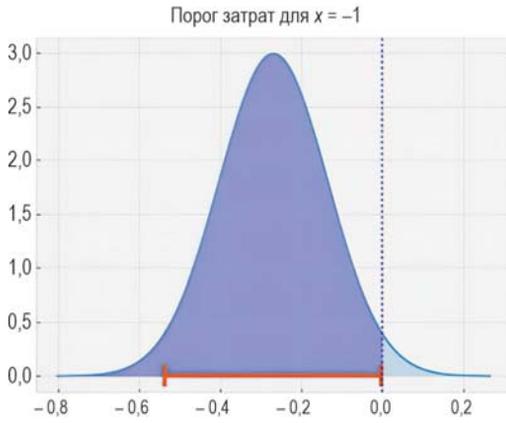
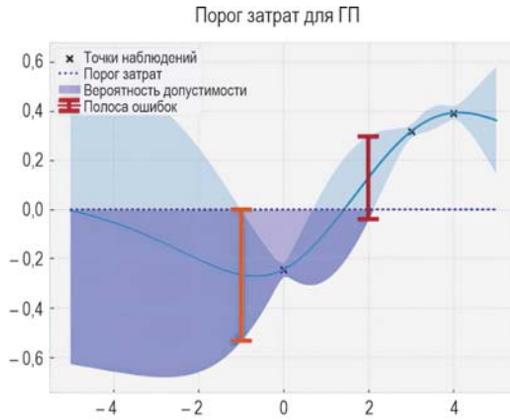


Рис. 8.7. Вероятность допустимости при $x = -1$ и $x = 2$ выделена более темным оттенком: на левой части показан весь ГП, на центральной — прогноз при $x = -1$, а на правой — прогноз при $x = 2$; выделенные части показывают вероятность допустимости, которая зависит от нормального распределения в заданной точке

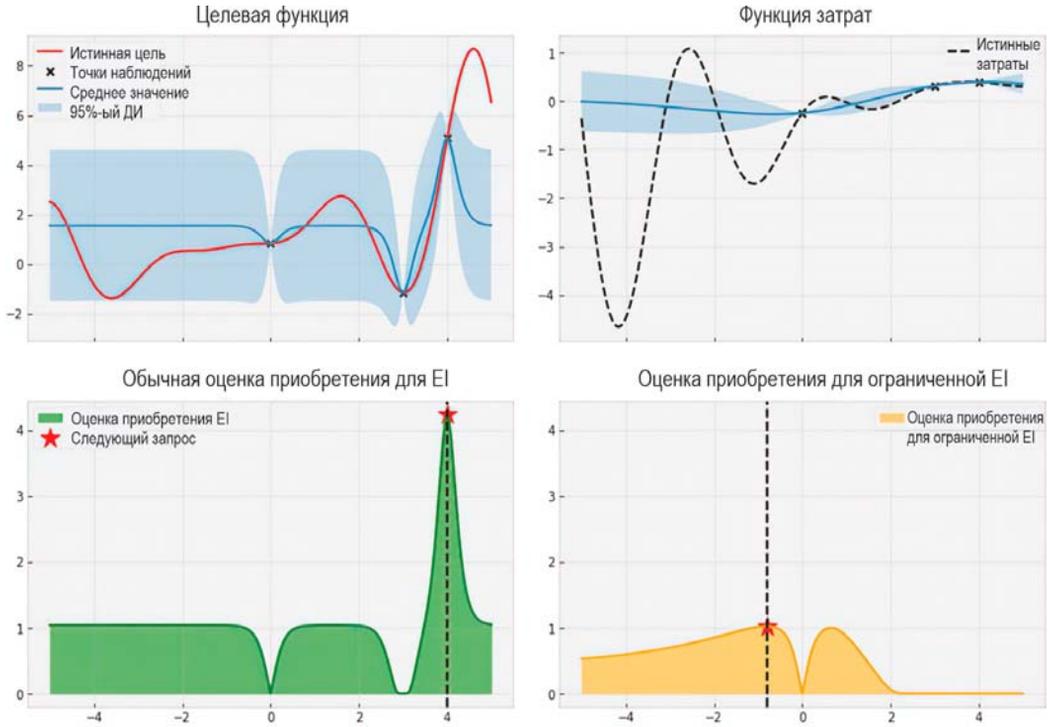


Рис. 8.8. Оценка приобретения обычной EI (внизу слева), оценка приобретения ограниченной EI (внизу справа), текущие прогнозы о целевой функции (вверху слева) и предположения о затратной функции (вверху справа): зная о требовании по затратам, ограниченная EI может избежать недопустимой области и порекомендовать для запроса совершенно другую точку

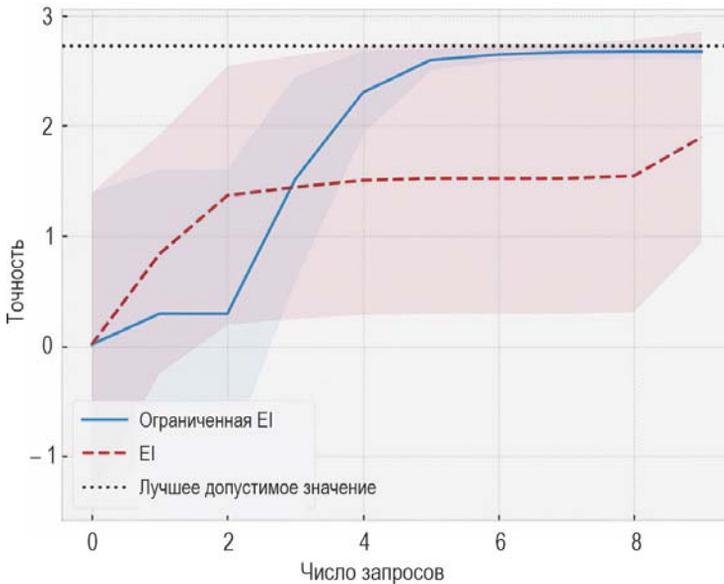


Рис. 8.9. Прогресс в одномерной задаче оптимизации ограниченной политики EI: по сравнению с обычной EI ограниченный вариант в среднем находит лучшее допустимое решение

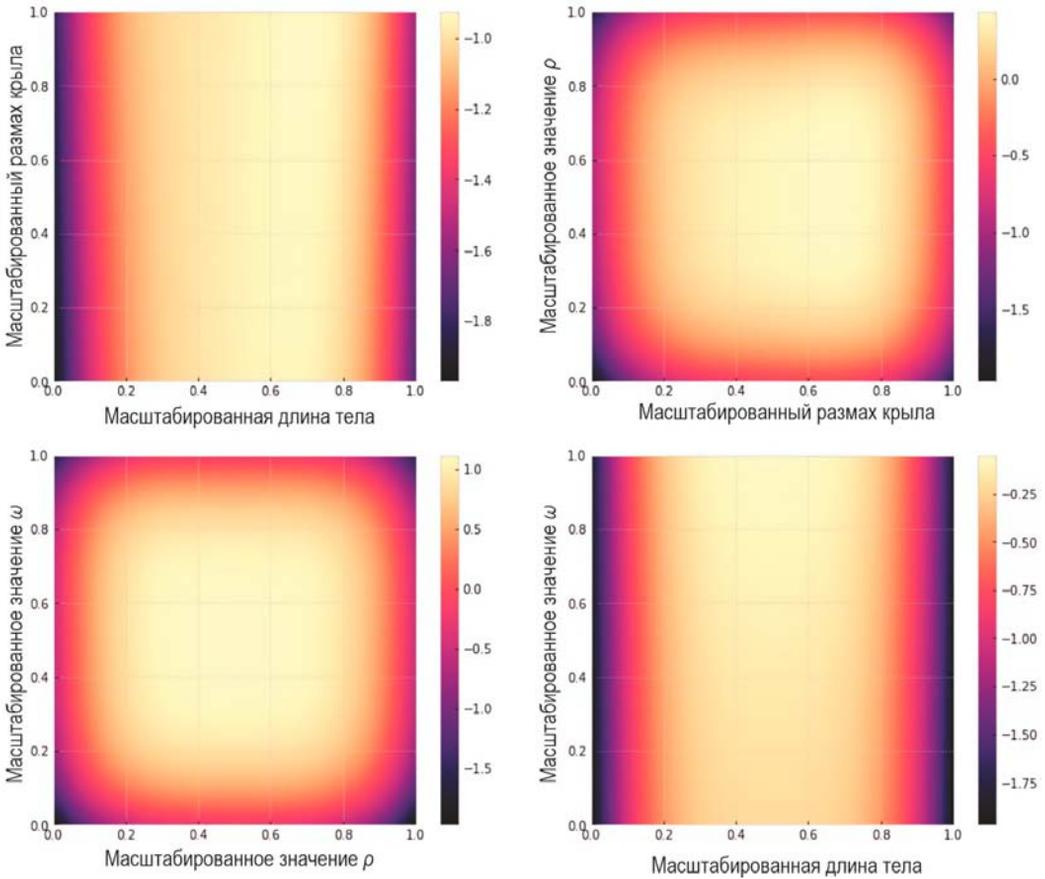


Рис. 8.10. Функция затрат моделируемой задачи оптимизации для конструкции самолета в различных двумерных подпространствах, которые соответствуют парам настраиваемых параметров, показанных в виде меток осей

Глава 9

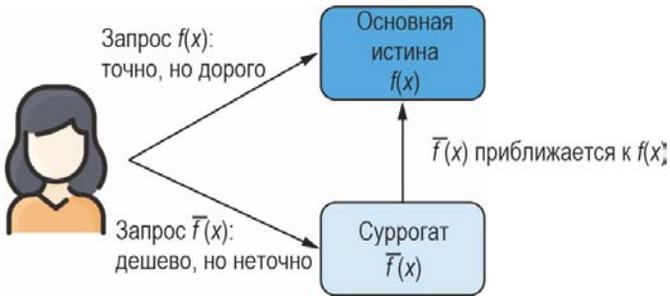


Рис. 9.1. Модель разноточного принятия решений, где необходимо сбалансировать запрос истинной целевой функции $f(x)$ для получения точной информации и запрос недорогого суррогата $\bar{f}(x)$

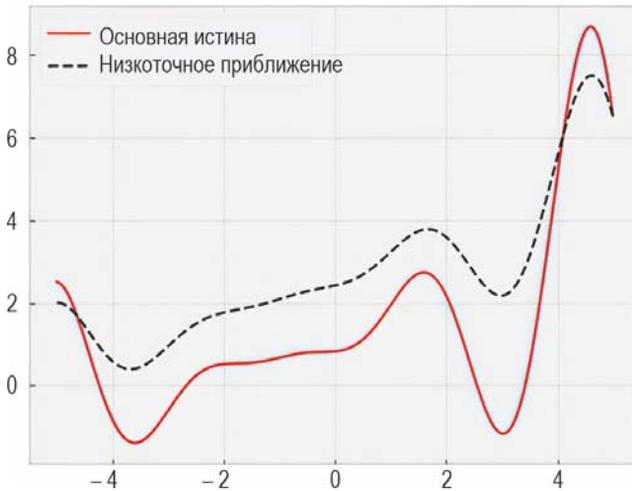


Рис. 9.2. Функция Форрестера (сплошная линия) и ее низкоточное приближение (пунктирная линия): несмотря на то что низкоточное приближение не совсем соответствует истине, оно дает достаточно информации о цели, поскольку обе функции имеют примерно одинаковую форму

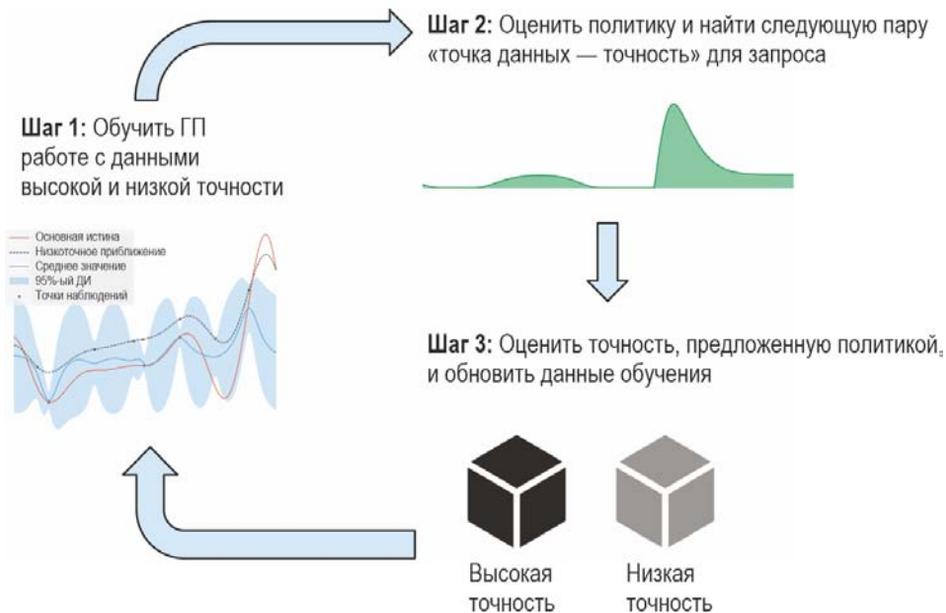


Рис. 9.3. Разноточный цикл БО: ГП обучается на данных как из целевой функции высокой точности, так и из низкоточного приближения, а политика решает, где и что в них запрашивать на каждой итерации цикла



Рис. 9.4. Шкала корреляции от 0 до 1 между низкоточным приближением и истинной целью: чем выше значение, тем больше информации дает приближение

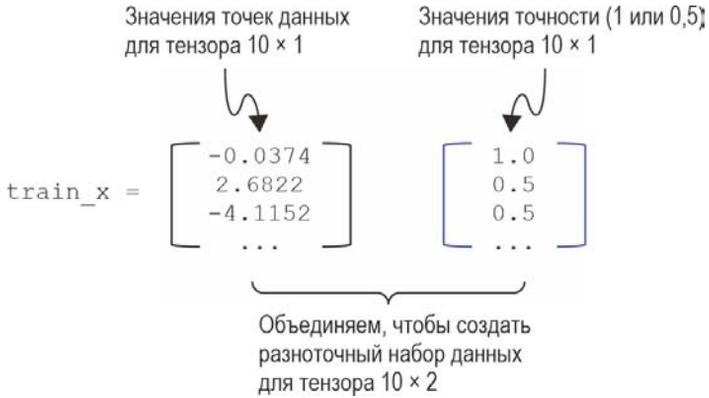


Рис. 9.5. Форматирование объектов в наборе разноточных данных: каждая точка связана с соответствующей точностью; значения точностей сохраняются в дополнительном столбце обучающего набора

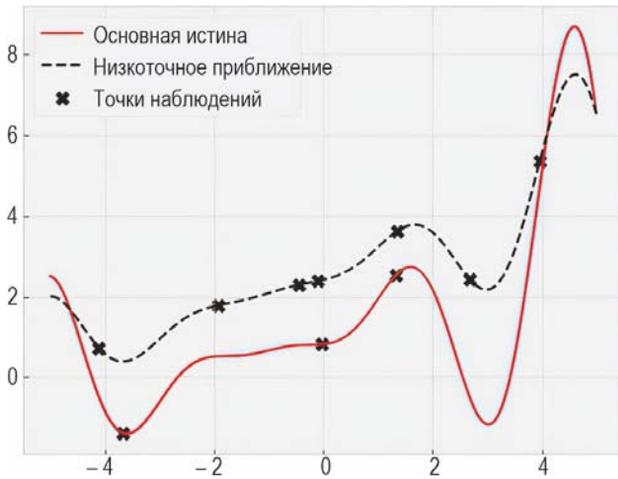
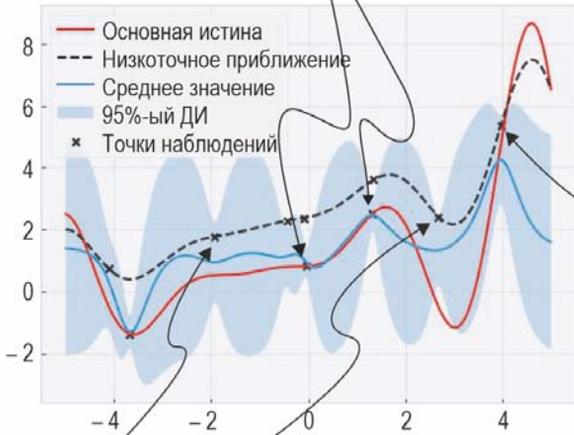


Рис. 9.6. Случайный разноточный набор обучающих данных на основе функции Форрестера и ее низкоточного приближения: он содержит три наблюдения с высокой точностью и семь наблюдений с низкой точностью

1. Среднее значение проходит через наблюдения с высокой точностью

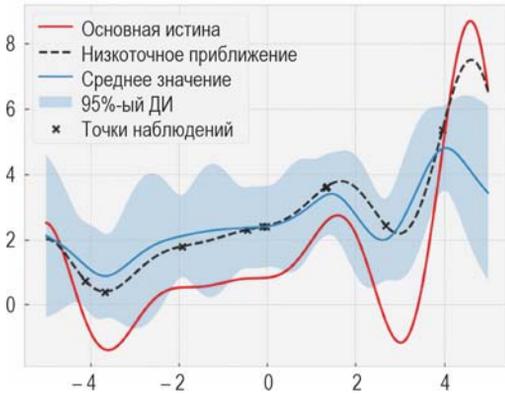


2. Прогнозы ГП отражают важные направления

3. Существует низкая неопределенность в отношении наблюдений с низкой точностью

Рис. 9.7. Предсказания разноточного ГП относительно целевой функции (основная истина): средние прогнозы соответствующим образом учитывают высокоточные наблюдения, но в отношении низкоточных наблюдений неопределенность по-прежнему снижается

Значение корреляции = 0,5



Значение корреляции = 0,9

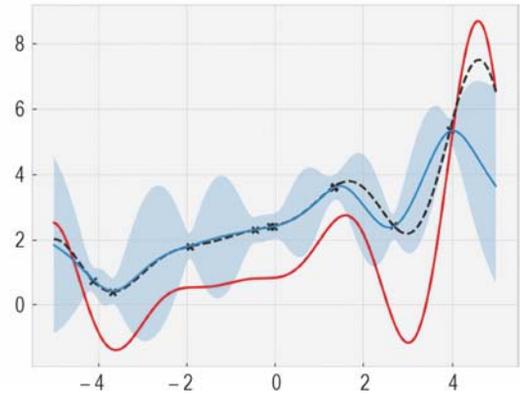


Рис. 9.8. Прогнозы ГП относительно целевой функции (основная истина), основанные только на наблюдениях с низкой точностью: слева показан результат, когда значение корреляции равно 0,5; справа — когда значение корреляции равно 0,9, что демонстрирует меньшую неопределенность

Стоимость *низкоточного* запроса = **Фиксированная стоимость** + **Вес** × *Значение низкоточной корреляции*

Стоимость *высокоточного* запроса = **Фиксированная стоимость** + **Вес** × *Значение высокоточной корреляции*

Настраиваемые параметры

Рис. 9.9. Линейная модель затрат для разноточной оптимизации: стоимость запроса точности линейно зависит от корреляции между этой точностью и основной истиной $f(x)$

$$\text{Оценка приобретения ROI} = \frac{\text{Получение информации}}{\text{Стоимость запроса}}$$

↑ Рассчитывается политикой MES

↑ Рассчитывается моделью затрат

Рис. 9.10. Формула оценки приобретения для рентабельности инвестиций в разноточной оптимизации: она количественно определяет объем информации, которую предоставляет запрос об оптимуме целевой функции для каждой единицы затрат

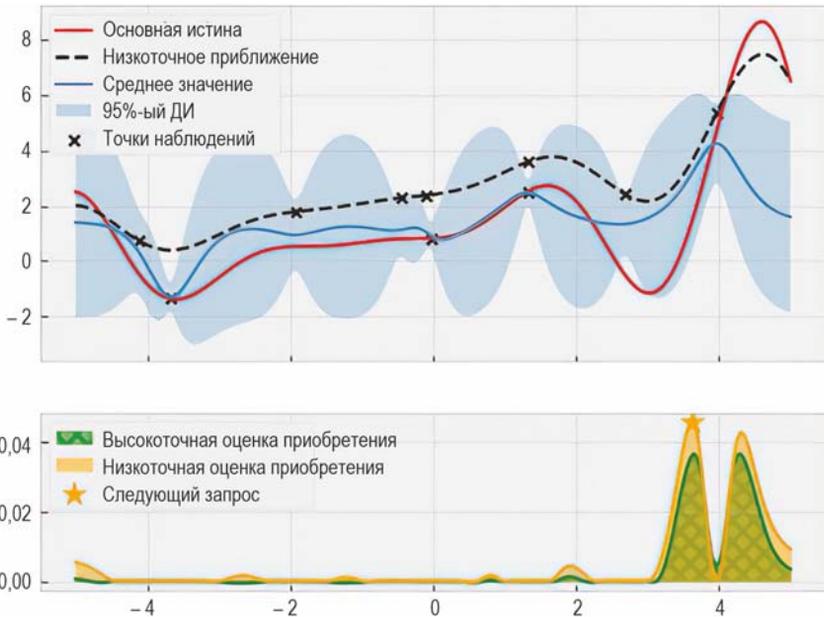


Рис. 9.11. Текущее предположение ГП о целевой функции (вверху) и оценках приобретения, рассчитанных с помощью разноточной политики MES (внизу): в данном примере низкоточные запросы предпочтительнее высокоточных из-за их низкой стоимости

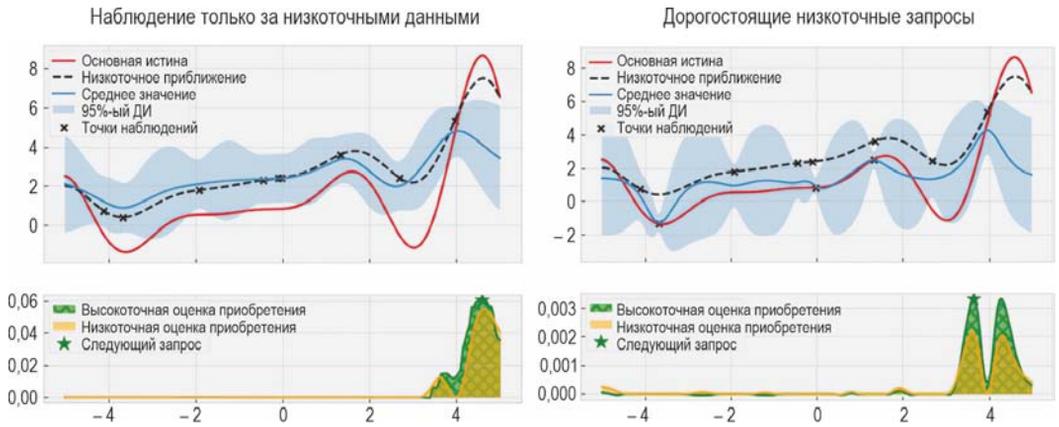


Рис. 9.12. Сценарий, когда высокоточные запросы предпочтительнее низкоточных: слева обучающий набор содержит только низкоточные наблюдения; справа — запросы с низкой точностью почти так же дороги, как высокоточные

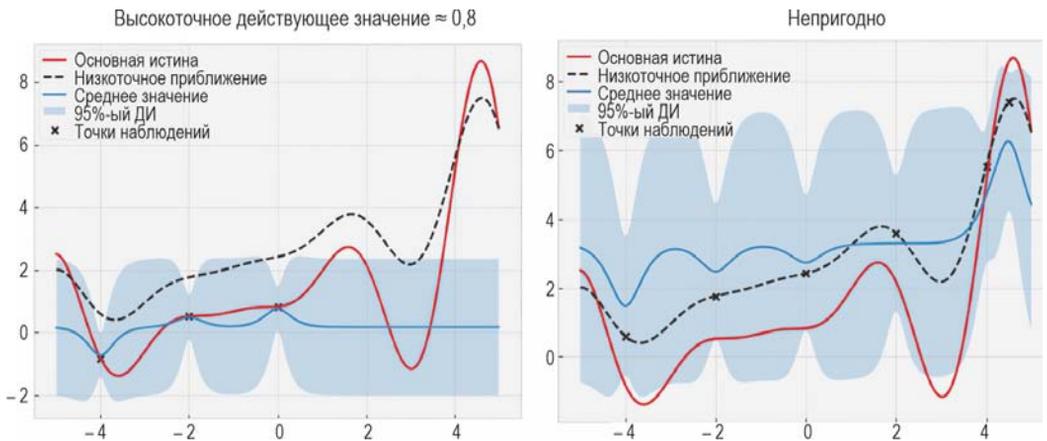


Рис. 9.13. Измерение производительности с помощью высокоточного действующего значения неприменимо при разноточной оптимизации: слева высокоточное действующее значение составляет примерно 0,8, а оптимум цели еще не обнаружен; справа мы близки к обнаружению оптимума, но нет высокоточного запроса, с помощью которого можно было бы записать действующее значение

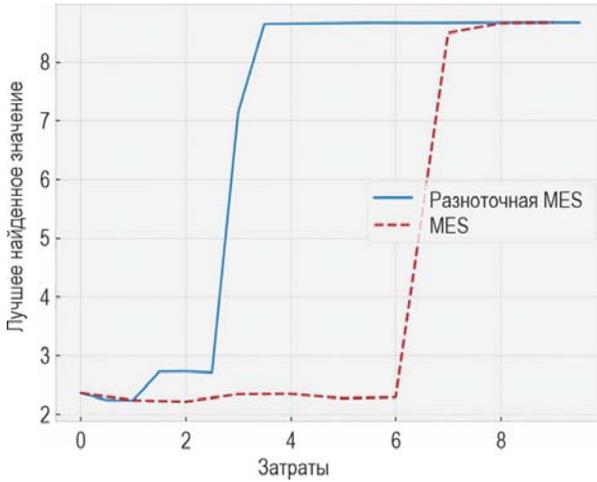


Рис. 9.14. Целевое значение максимизатора апостериорного среднего как функция затрат, потраченных двумя политиками: разноточная политика MES значительно превосходит обычную

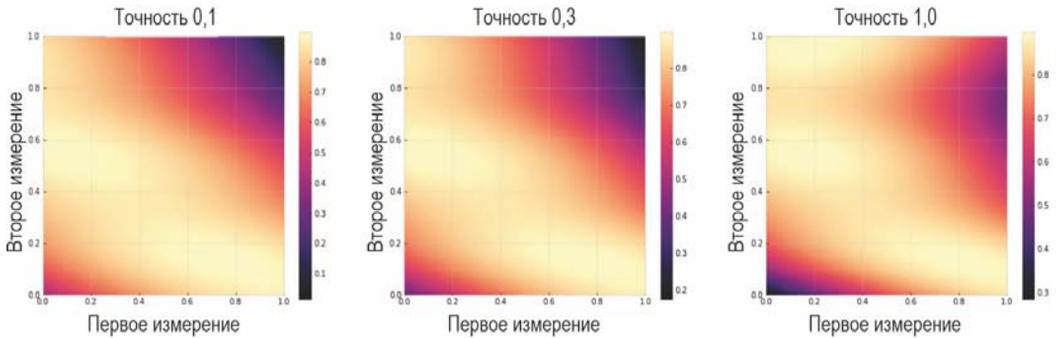


Рис. 9.15. Целевая функция Бранина (справа) и два приближения низкой точности

Глава 10



Рис. 10.1. Примеры выявления предпочтений пользователя в рекомендациях продуктов: слева предлагается оценить рекомендуемый продукт по шкале; справа предлагается выбрать продукт, который ему больше нравится; второй вариант помогает лучше выявить предпочтения пользователя



Рис. 10.2. Различия между числовыми рейтингами и парными сравнениями по информативности и сложности получения: каждый метод выявления предпочтений имеет свои преимущества и недостатки

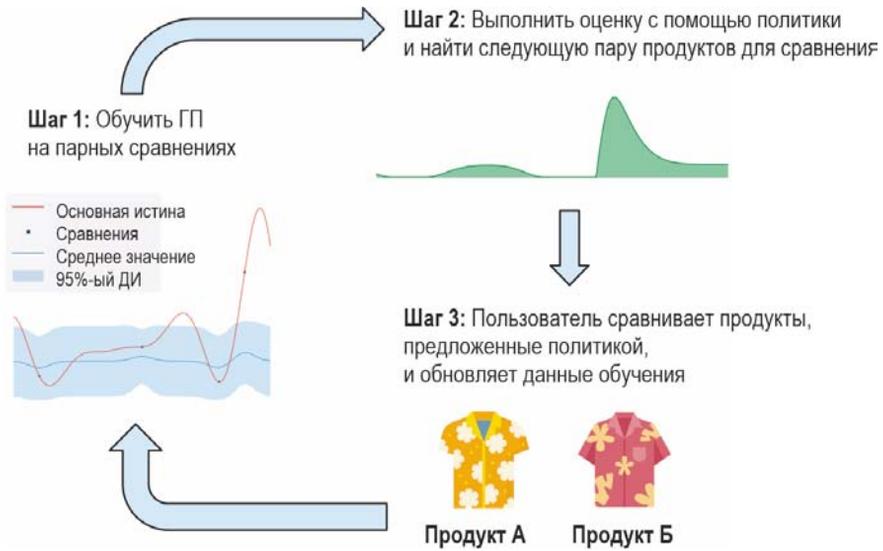


Рис. 10.3. Цикл БО с парными сравнениями для оптимизации предпочтений: ГП обучается на данных парного сравнения, а политика БО решает, какую пару точек данных следует попросить пользователя сравнить

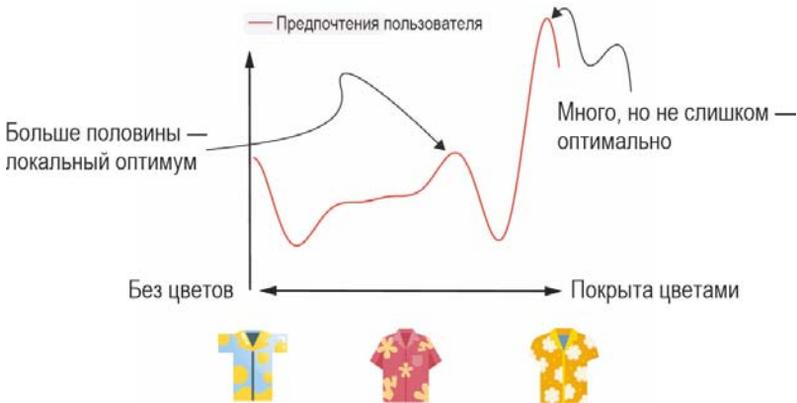


Рис. 10.4. Поиск рубашки с оптимальным количеством цветов в задаче рекомендации товара: пространство поиска одномерно, поскольку поиск выполняется только среди количества цветов на рубашке; рубашка, покрытая цветами больше чем наполовину, является локальным оптимумом, а рубашка, покрытая почти полностью, максимизирует предпочтения пользователя

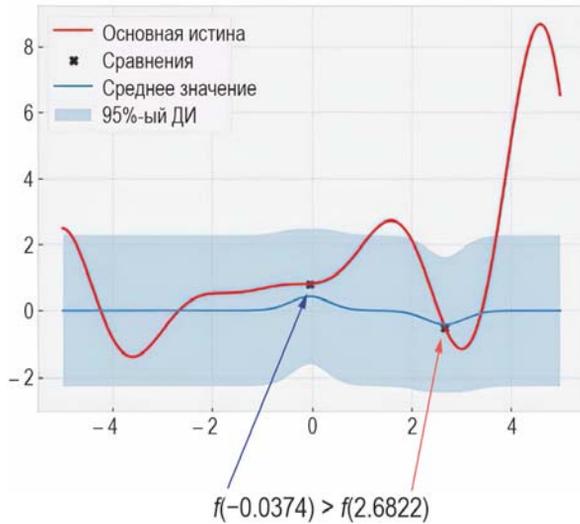


Рис. 10.5. Прогнозы ГП, обученные на парном сравнении $f(-0,0374) > f(2,6822)$: апостериорное среднее отражает результат данного сравнения, в то время как апостериорное стандартное отклонение вокруг двух точек данных немного уменьшилось по сравнению с априорным

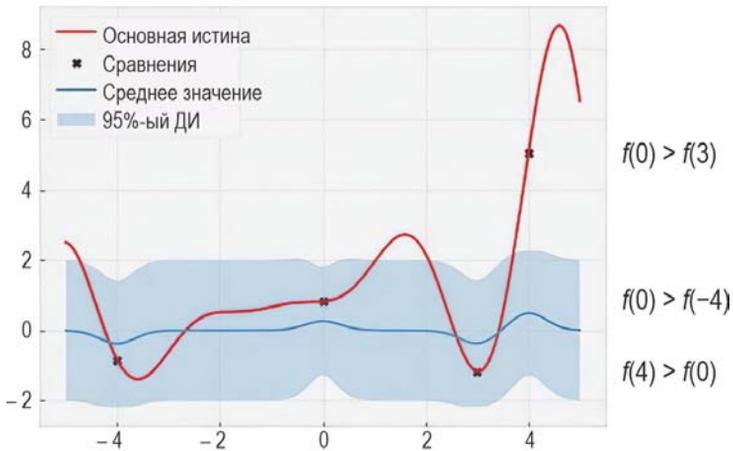


Рис. 10.6. Прогнозы ГП, обученного на парных сравнениях, показаны справа: апостериорное среднее отражает результат сравнения, в то время как апостериорное стандартное отклонение вокруг точек данных в обучающем наборе уменьшилось по сравнению с априорным

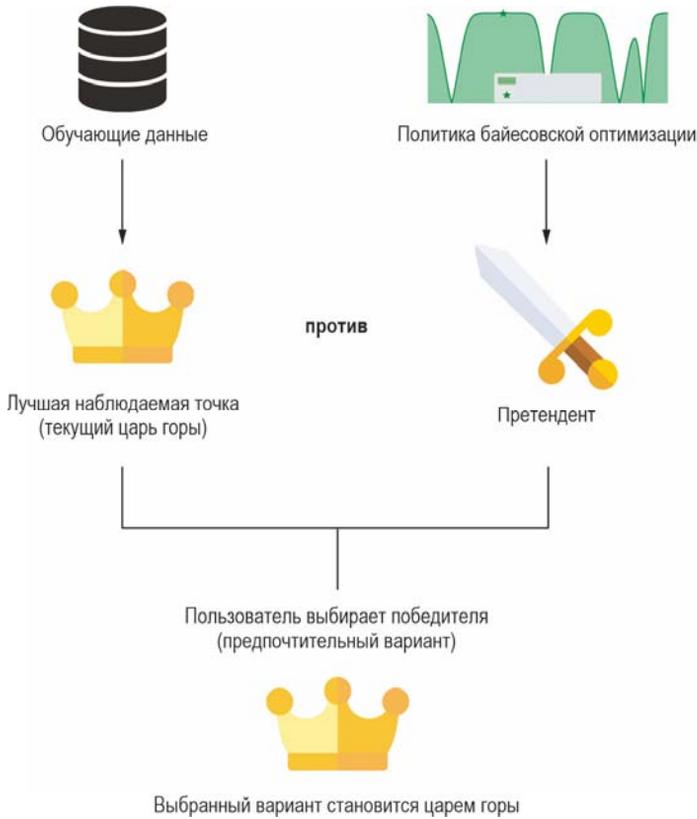


Рис. 10.7. Иллюстрация стратегии «Царь горы» в байесовской оптимизации предпочтений: лучший результат, найденный к текущему моменту, сравнивается с многообещающим кандидатом, определенным политикой

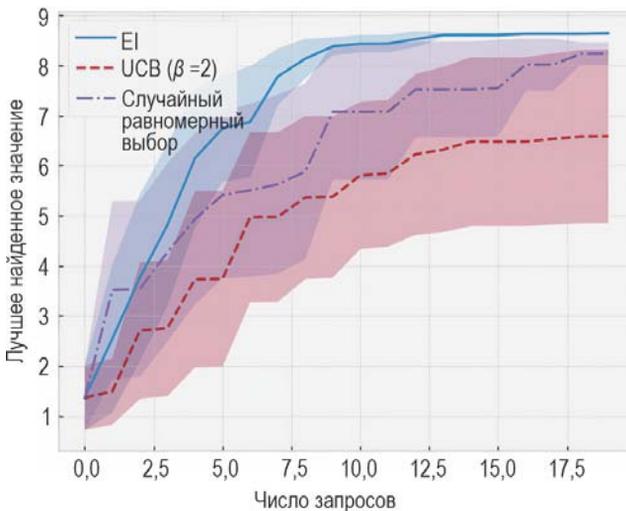


Рис. 10.8. Производительность оптимизации различных политик, совокупная по 10 экспериментам: EI работает лучше всего, последовательно находя глобальный оптимум; UCSB не удалось превзойти стратегию случайного равномерного поиска претендента

Глава 11



Рис. 11.1. Карикатура, иллюстрирующая баланс, которого мы должны достичь при многоцелевой оптимизации

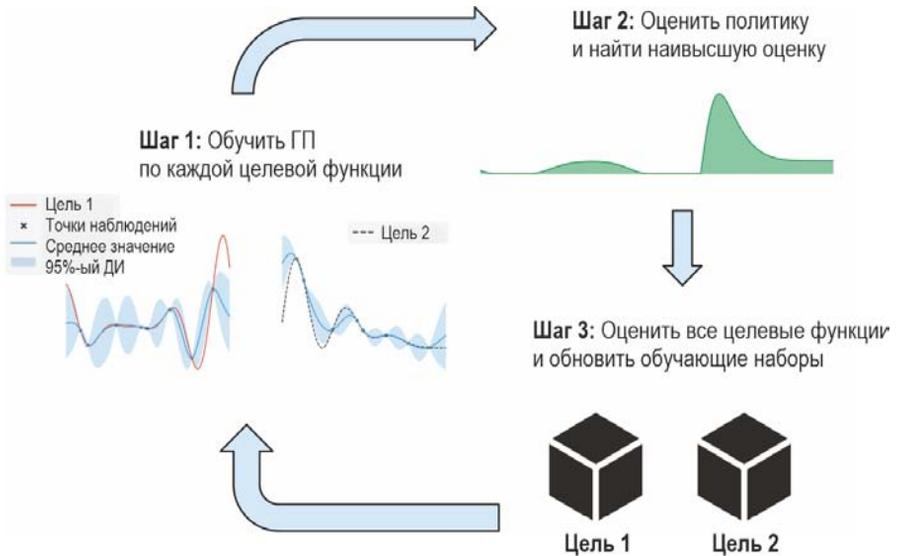


Рис. 11.2. Многоцелевой цикл БО с двумя функциями: ГП обучается на данных каждой целевой функции, а политика решает, по какой точке данных оценивать целевые функции в следующий раз

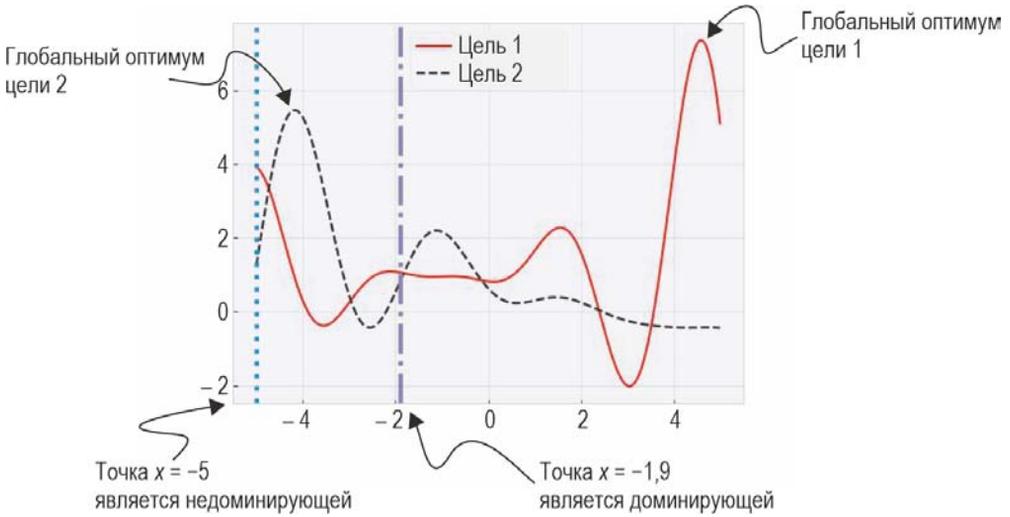


Рис. 11.3. Визуализация двух целевых функций в текущей задаче многоцелевой оптимизации: точки данных, которые максимизируют цели, отличаются друг от друга, поэтому существует компромисс при их оптимизации

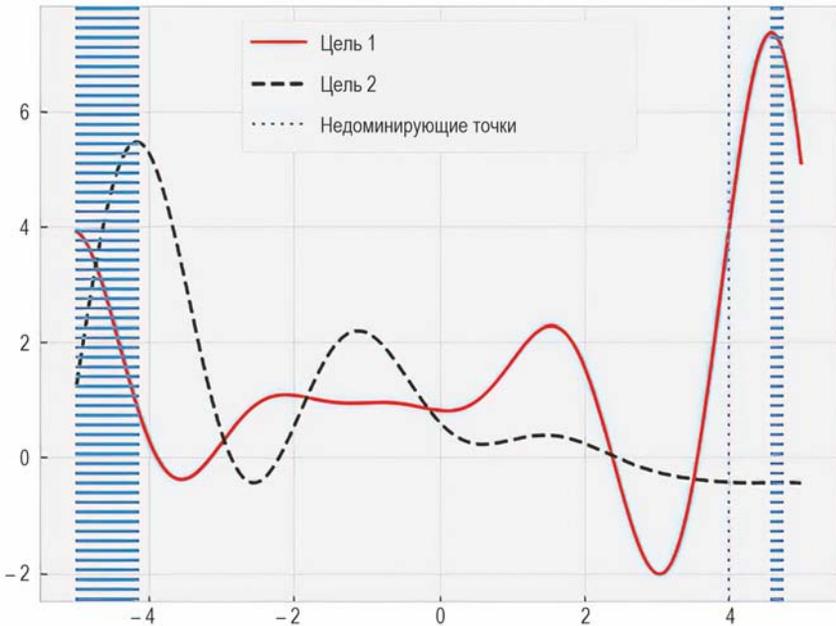


Рис. 11.4. Две целевые функции и недоминирующие точки: в задаче многоцелевой оптимизации существует бесконечно много недоминирующих точек

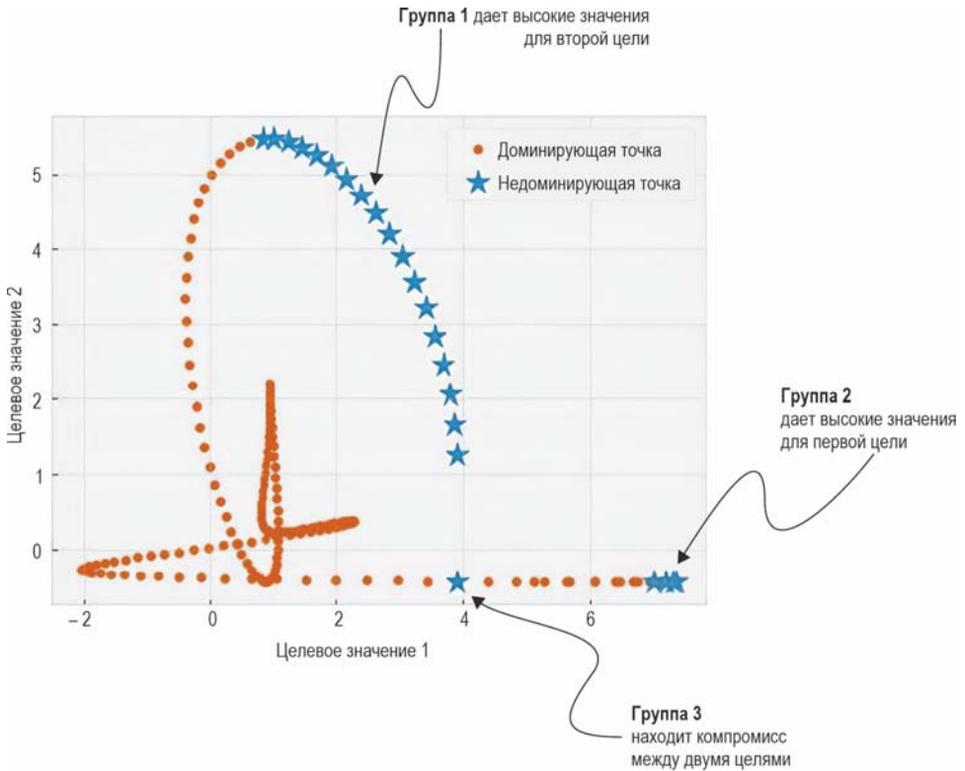


Рис. 11.5. График, основанный на значениях точек данных для двух целевых функций: доминирующие точки обозначены «точками», недоминирующие — «звездочками»; три группы недоминирующих точек соответствуют тем точкам, которые обсуждались на рис. 11.4

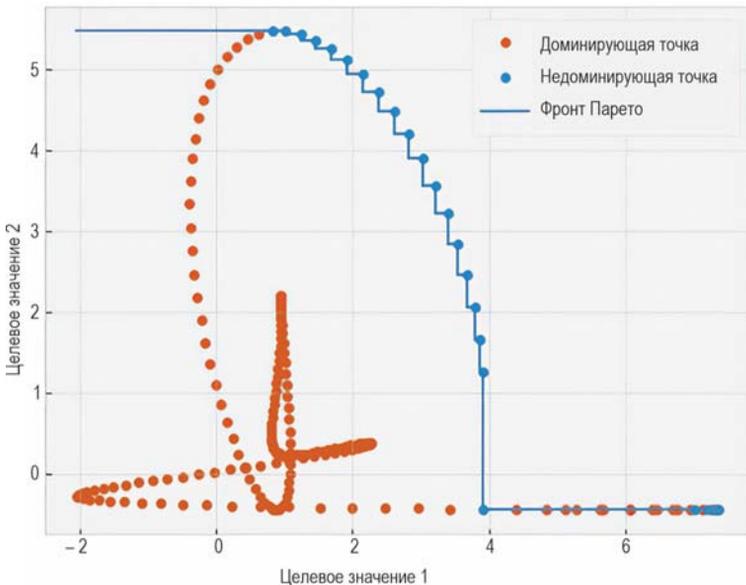


Рис. 11.6. Фронт Парето, проходящий через недоминирующие точки: за пределами фронта нет ни одной точки данных (выше и правее)

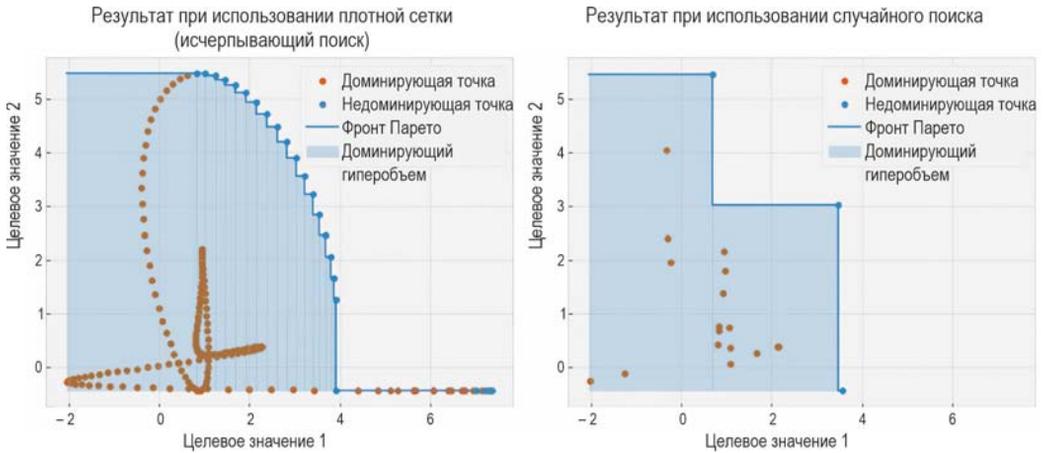


Рис. 11.7. Доминирующий гиперобъем плотной сетки (слева), который эквивалентен исчерпывающему поиску, и случайный поиск (справа) с помощью 20 точек данных, выбранных случайным образом: первый набор имеет больший доминирующий объем и, следовательно, лучше справляется с многоцелевой оптимизацией, чем второй

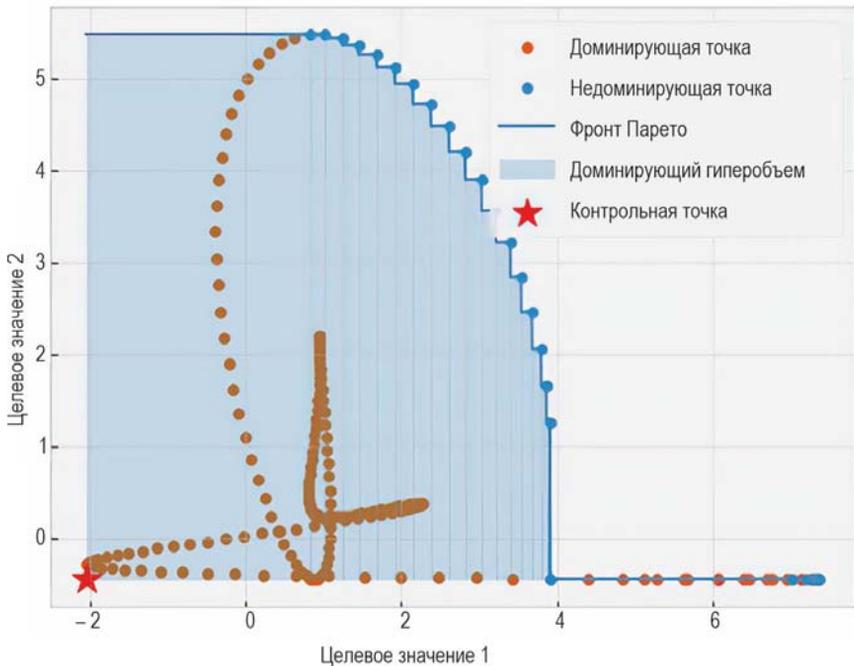


Рис. 11.8. Контрольная точка (ориентир) в задаче многоцелевой оптимизации, которая устанавливает нижнюю границу доминирующего пространства: гиперобъем вычисляется как объем области между контрольной точкой и фронтом Парето

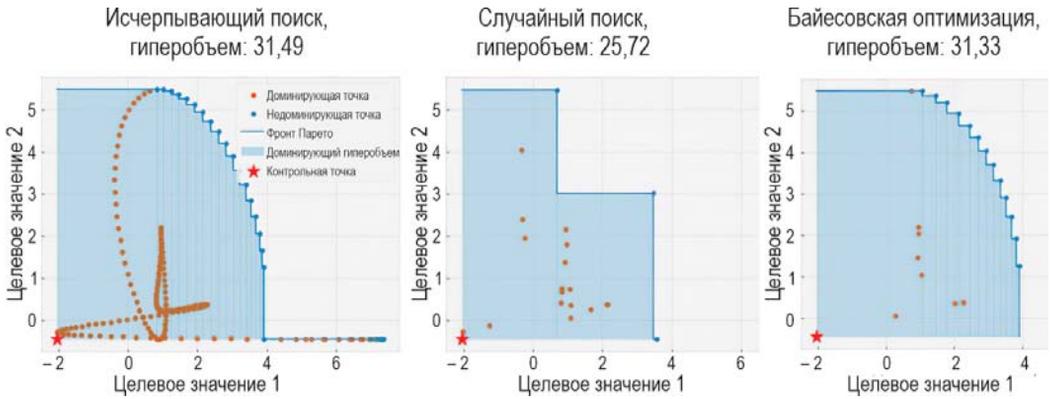


Рис. 11.9. Результаты многоцелевой оптимизации различных стратегий поиска и соответствующих гиперобъемов: БО обеспечивает почти такой же гиперобъем, как и исчерпывающий поиск, но со значительно меньшим количеством запросов

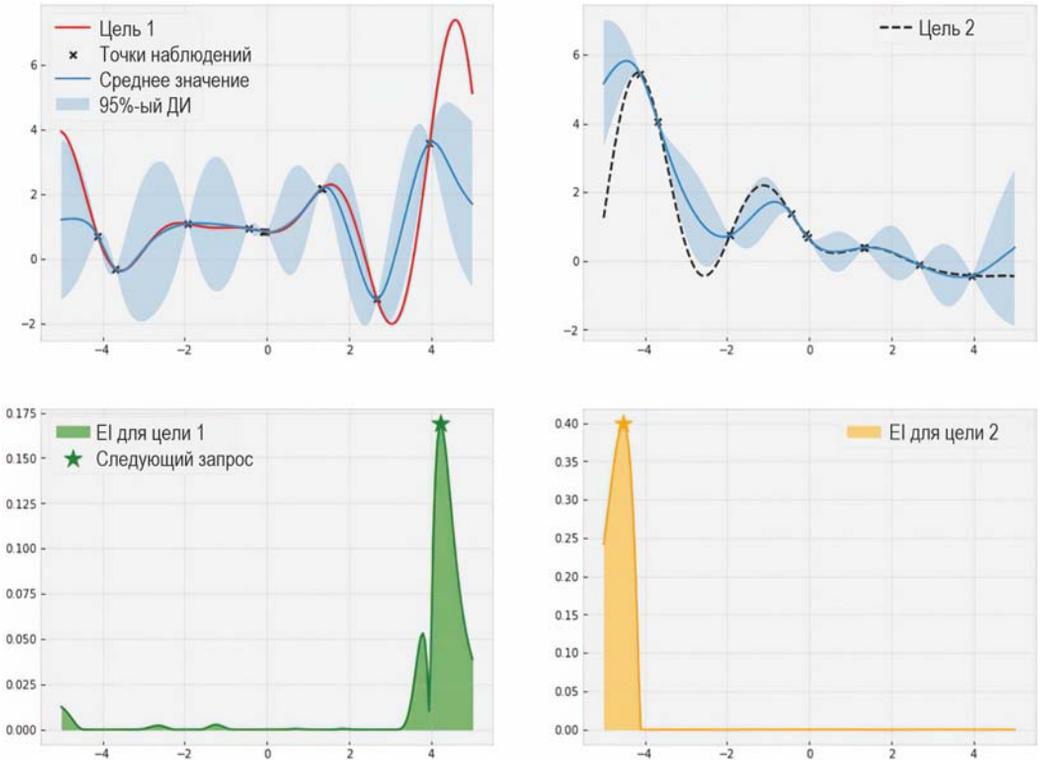


Рис. 11.10. Текущее предположение ГП о двух целевых функциях (вверху) и их соответствующих оценках приобретения (внизу): каждая политика EI направлена на оптимизацию собственной целевой функции и фокусируется на своих регионах

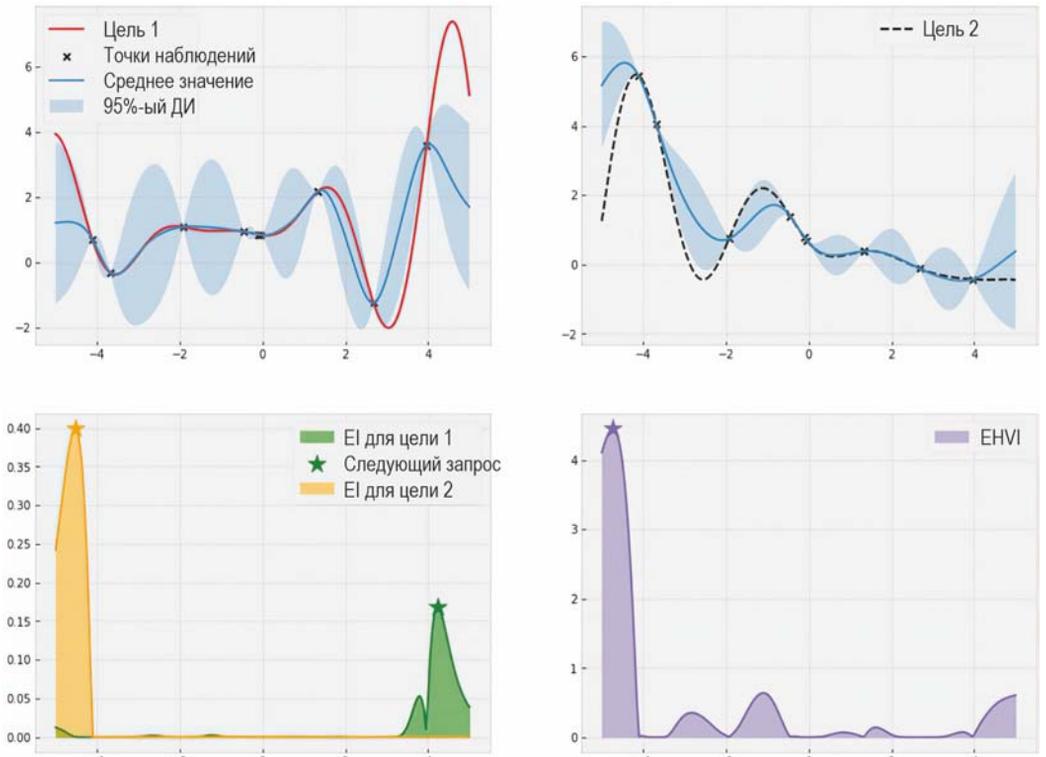


Рис. 11.11. Текущее предположение ГП о каждой из двух целевых функций (вверху), соответствующие оценки приобретения политики EI (внизу слева) и оценки приобретения политики EHVI (внизу справа): EHVI уравнивает обе цели, присваивая высокие оценки нескольким регионам, которые расширяют фронт Парето

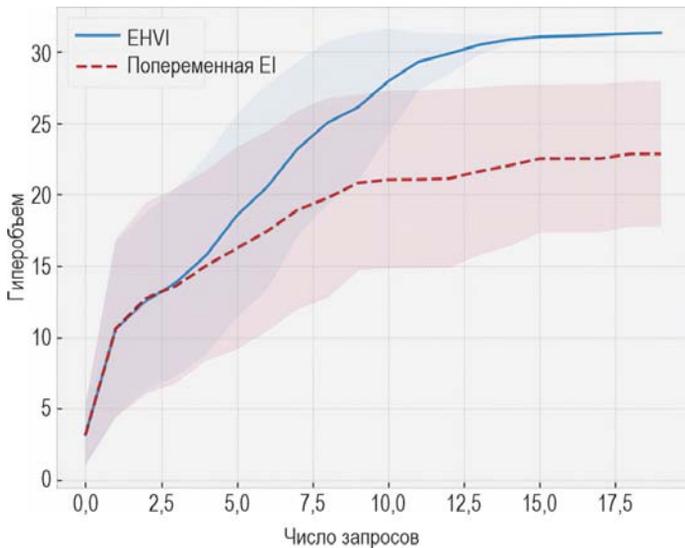


Рис. 11.12. Средний гиперобъем и полосы ошибок в зависимости от количества запросов, выполненных двумя политиками: EHVI неизменно превосходит альтернативную политику EI

Глава 12



Рис. 12.1. Прогнозы обычного ГП и VGP: VGP дает почти такие же прогнозы, как и ГП, но требует значительно меньше времени на обучение



Рис. 12.2. Блок-схема вспомогательной функции, визуализирующая прогнозы ГП: функция также показывает стимулирующие точки, если передано условие, что модель является VGP

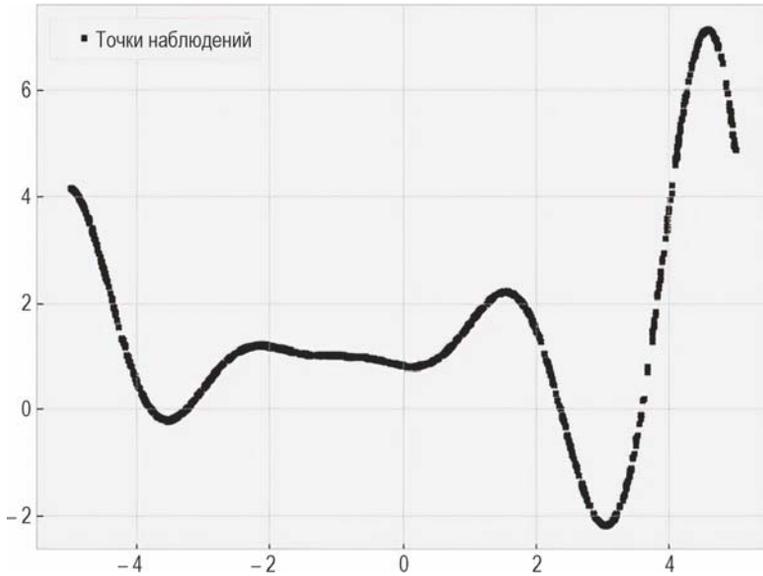


Рис. 12.3. Набор обучающих данных, содержащий 1000 точек: обучение обычного ГП на этом наборе занимает значительное время



Рис. 12.4. Прогнозы обычного ГП: они хорошо соответствуют обучающим данным, но обучение занимает много времени

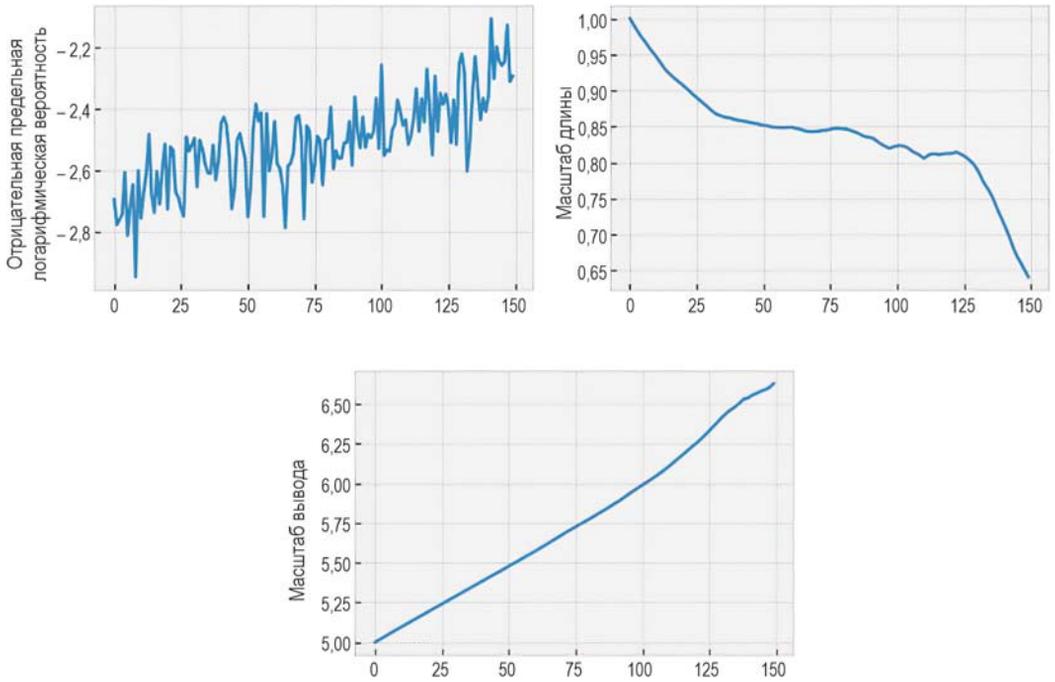


Рис. 12.5. Прогрессирующие потери обычного ГП во время градиентного спуска: из-за численной нестабильности кривая потерь является неровной и не может эффективно минимизироваться



Рис. 12.6. Выполнение градиентного спуска с численно нестабильным расчетом потерь похоже на спуск с горы с завязанными глазами

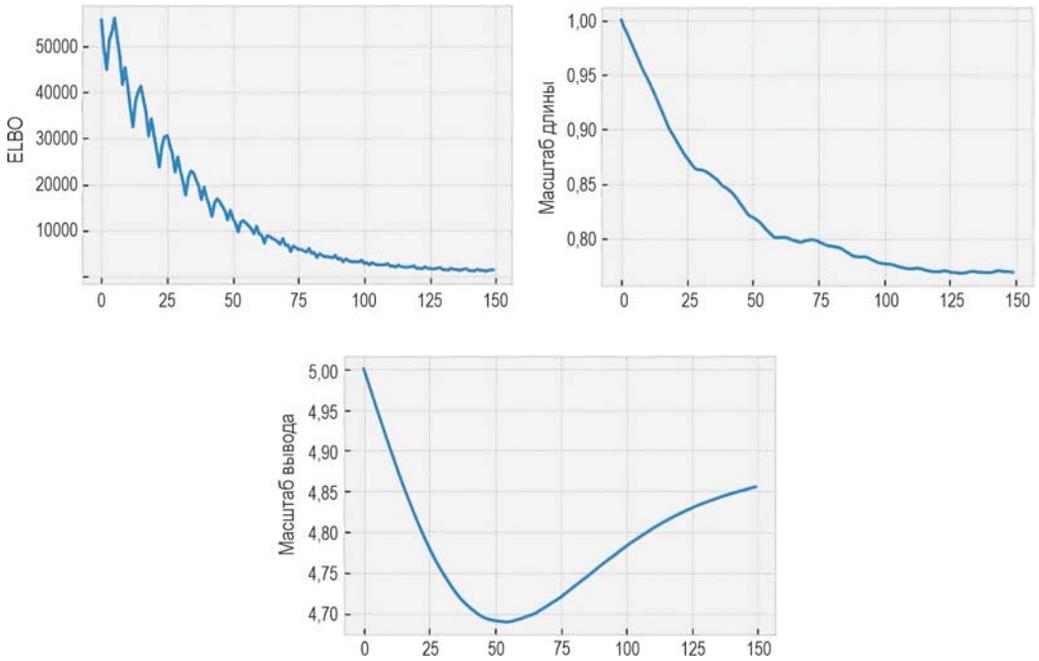


Рис. 12.10. Прогресс уменьшения потерь и соответствующие масштабы длины и вывода VGP во время мини-пакетного градиентного спуска

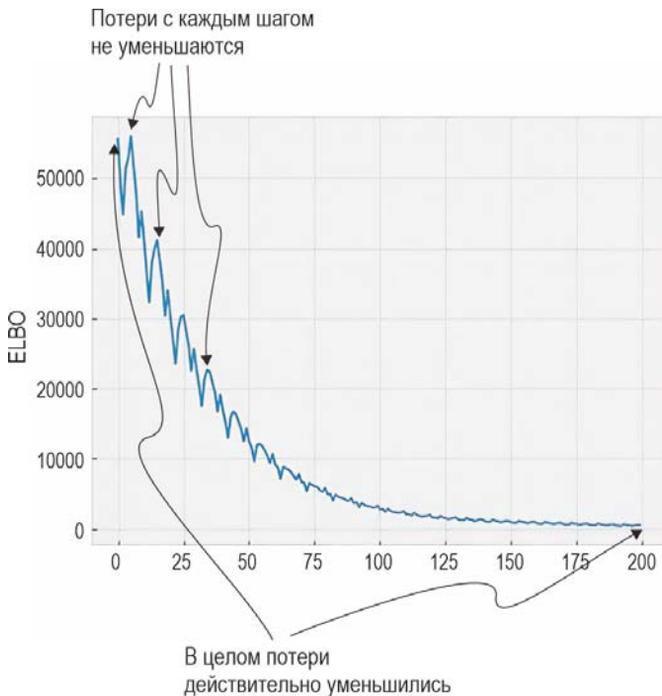


Рис. 12.11. Прогресс уменьшения потерь VGP во время мини-пакетного градиентного спуска: несмотря на некоторое непостоянство, потери эффективно минимизируются

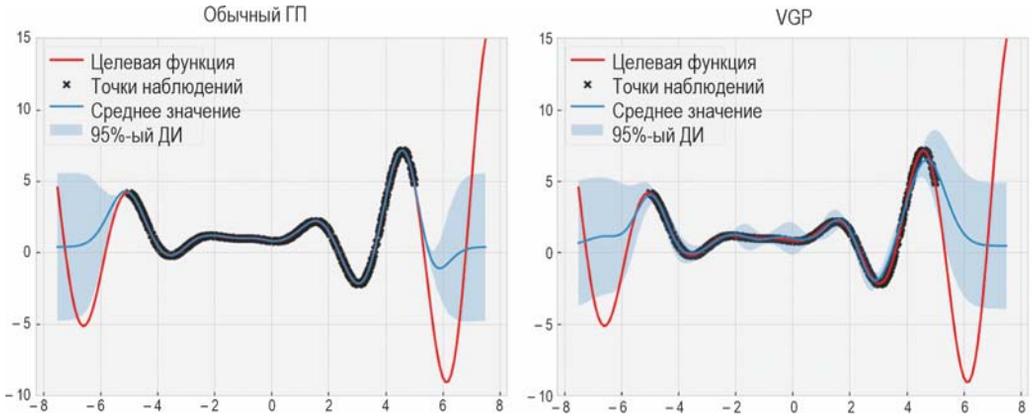


Рис. 12.12. Прогнозы обычного ГП и VGP примерно совпадают

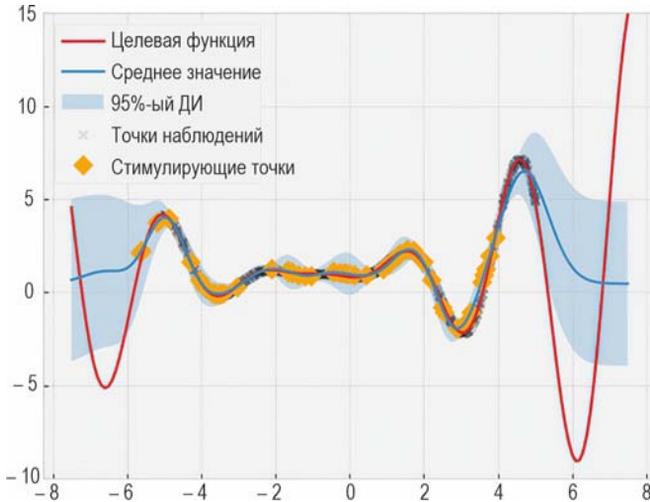


Рис. 12.13. Стимулирующие точки VGP расположены так, что представляют все данные и отражают наиболее важные тенденции

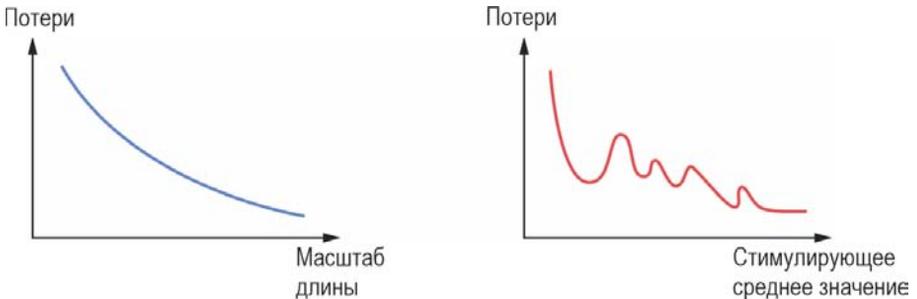


Рис. 12.14. Пример, как потери могут вести себя по-разному по отношению к обычному и вариационному параметрам: это порождает необходимость учета геометрии потерь



Рис. 12.15. Демонстрация обычного, мини-пакетного и естественного градиентных спусков в «долине потерь», где центр долины дает наименьшие потери: учитывая геометрию функции потерь, естественный градиентный спуск достигает минимума потерь быстрее, чем мини-пакетный вариант

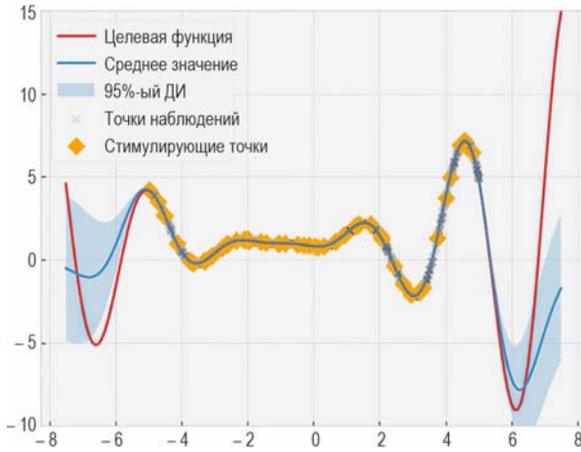


Рис. 12.16. Прогнозы VGP и стимулирующие точки, обученные с помощью естественного градиентного спуска: качество прогнозов превосходное

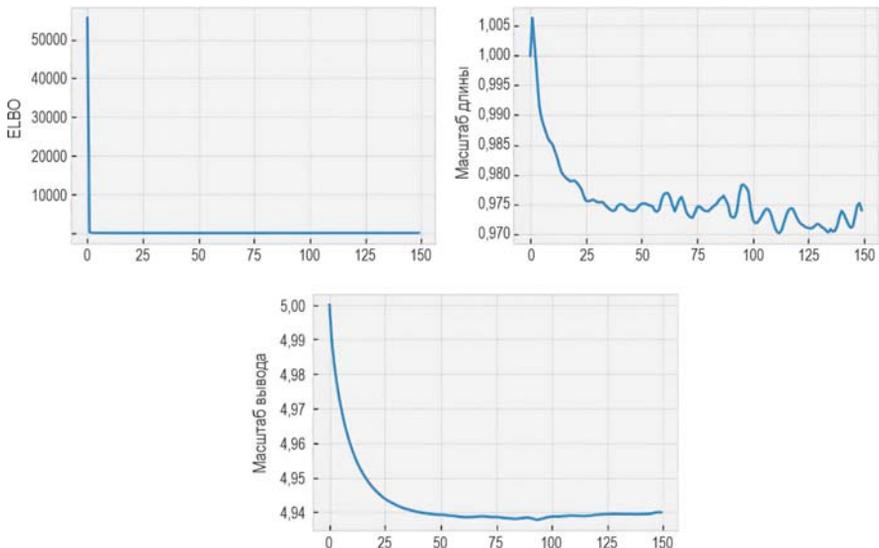


Рис. 12.17. Прогресс потерь VGP во время естественного градиентного спуска: потери были эффективно минимизированы после нескольких итераций

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-117.08	32.70	37	2176	418.0	1301	375	2.8750	98900
1	-117.91	34.11	20	3158	684.0	2396	713	3.5250	153000
2	-117.10	32.75	11	2393	726.0	1905	711	1.3448	91300
3	-117.22	32.74	52	1260	202.0	555	209	7.2758	345200
4	-121.99	37.29	32	2930	481.0	1336	481	6.4631	344100
...
4995	-118.41	34.19	42	779	145.0	450	148	3.9792	193800
4996	-122.96	38.42	50	2530	524.0	940	361	2.9375	122900
4997	-118.00	33.77	24	1324	267.0	687	264	3.4327	192800
4998	-122.81	38.54	12	2289	611.0	919	540	1.1553	139300
4999	-118.08	33.84	25	3696	953.0	2827	860	3.3438	153300

Рис. 12.18. Набор данных о ценах на жилье, показанный в виде фрейма данных Pandas: это тренировочный набор для данного упражнения

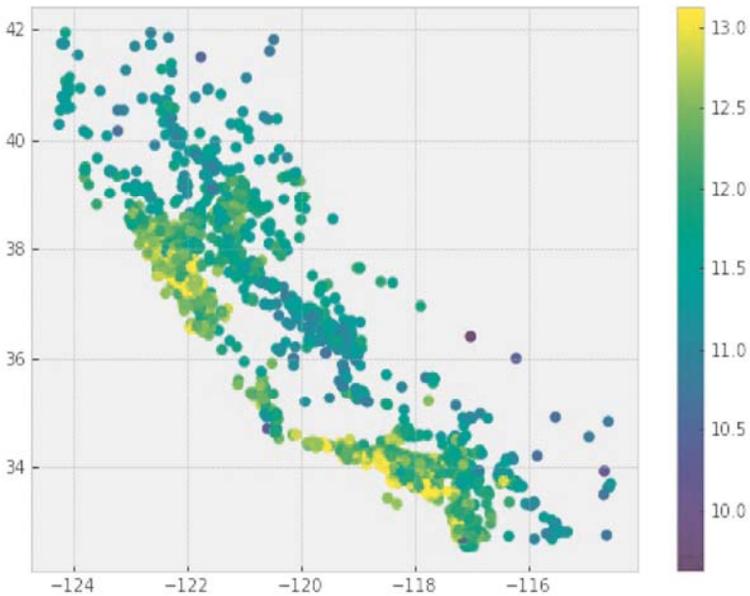


Рис. 12.19. Набор данных о ценах на жилье показан в виде рассеяния точек

Глава 13



Рис. 13.1. Ковариация между домами, вычисленная с помощью неподходящего ядра: поскольку ядро учитывает только цвет входной двери, оно не создает необходимых соответствий между домами

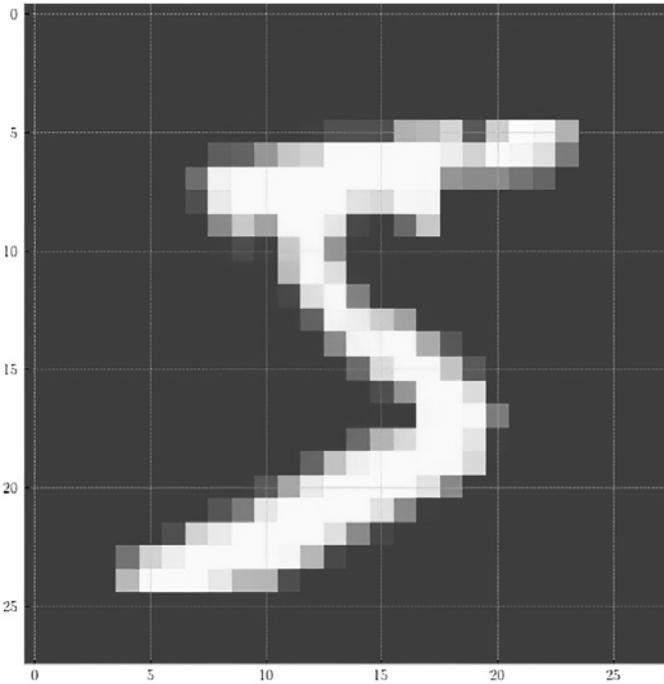
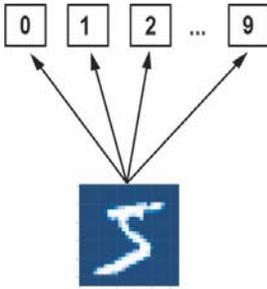
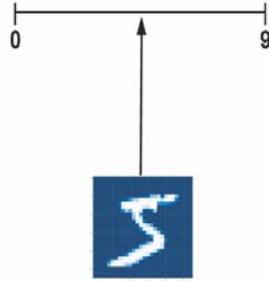


Рис. 13.2. Точка из набора данных MNIST: она выглядит как изображение размером 28×28 и представлена в виде тензора PyTorch

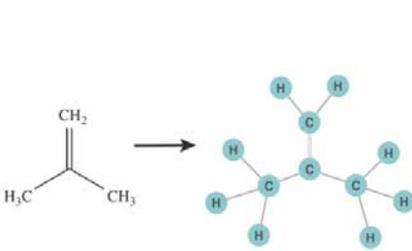


Классификация:
каждый прогноз —
это один из классов
(одна из «коробочек»)

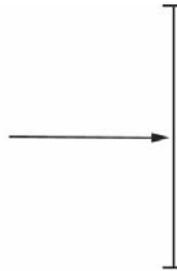


Регрессия:
каждый прогноз —
это число в диапазоне

Рис. 13.3. Классификация и регрессия в контексте данных MNIST: каждый прогноз в классификации соответствует одному из классов, каждый прогноз в регрессии представляет собой число внутри непрерывного диапазона



Каждая точка (входные данные)
представляет собой молекулу в виде графа



Каждый прогноз (выходные данные)
представляет собой действительное число
внутри диапазона

Рис. 13.4. Открытие лекарств как пример задачи структурированной регрессии: каждое соединение представлено в виде графа, и мы стремимся предсказать эффективность соединения при лечении какого-либо заболевания по шкале от 0 до 10

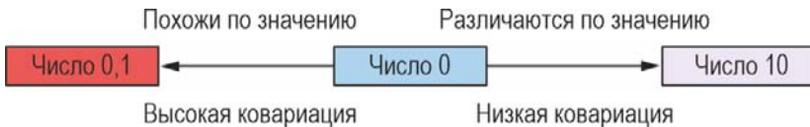


Рис. 13.5. Ковариация между различными числами: когда разница между двумя числами мала, ковариация увеличивается; когда разница велика, ковариация уменьшается

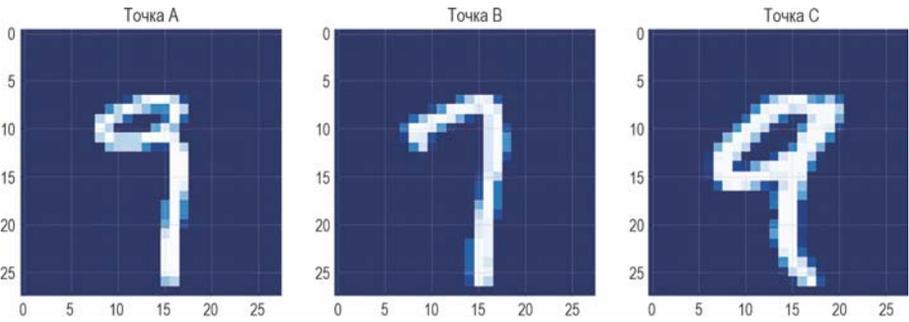


Рис. 13.6. Три конкретных значения из набора данных MNIST: первая и вторая точки имеют ненулевую ковариацию, несмотря на разные метки; первая и третья точки имеют нулевую ковариацию, несмотря на то, что имеют одинаковую метку

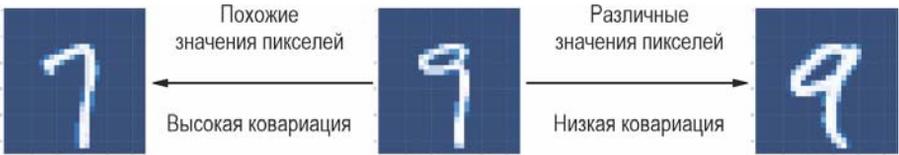


Рис. 13.7. Ковариация между рукописными цифрами, вычисленная ядром RBF: поскольку ядро рассматривает только значения пикселей, оно не делает соответствующих ковариаций

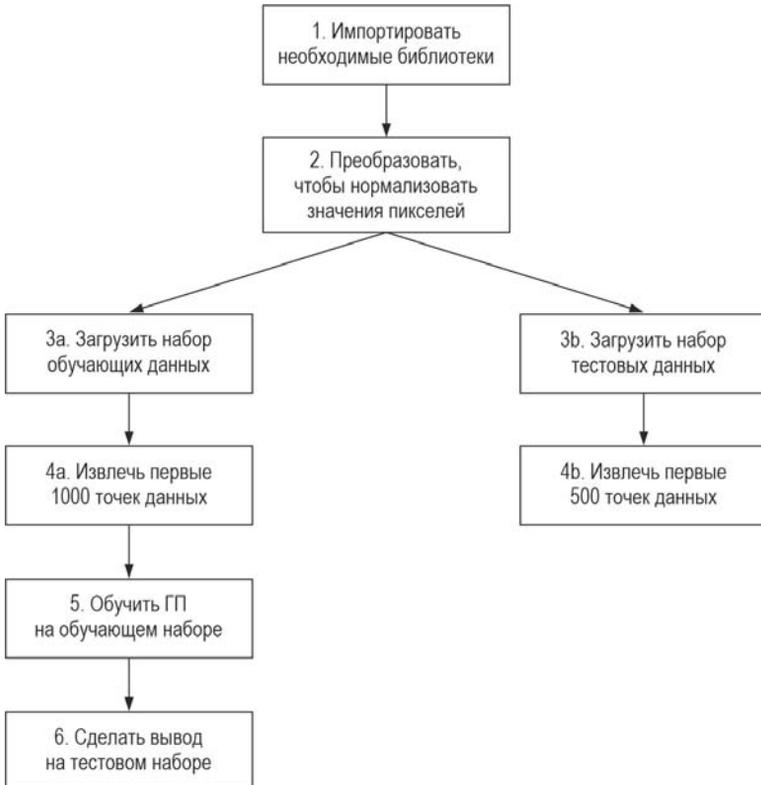


Рис. 13.8. Блок-схема обучения ГП на данных MNIST: сначала мы извлекаем 1000 точек данных, чтобы составить обучающий набор, затем еще 500 точек в качестве тестового набора

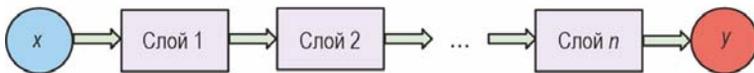


Рис. 13.9. Нейронная сеть — это совокупность многоуровневых вычислений, которая может хорошо моделировать сложные функции

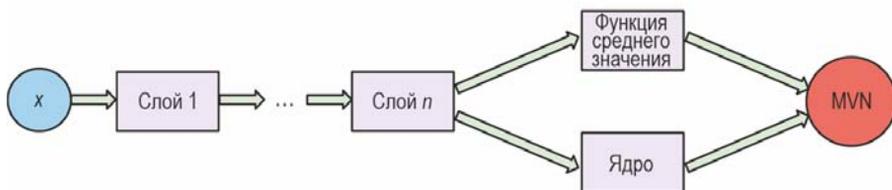


Рис. 13.10. Объединение нейросети и ГП: нейросеть сначала обрабатывает входные структурированные данные x , а затем передает их в функцию среднего и ядро

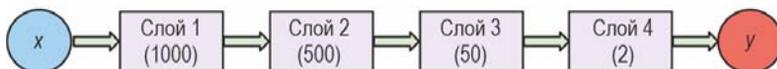


Рис. 13.11. Архитектура реализуемой нейросети: она имеет четыре слоя и создает массив размером 2 для каждой точки входных данных

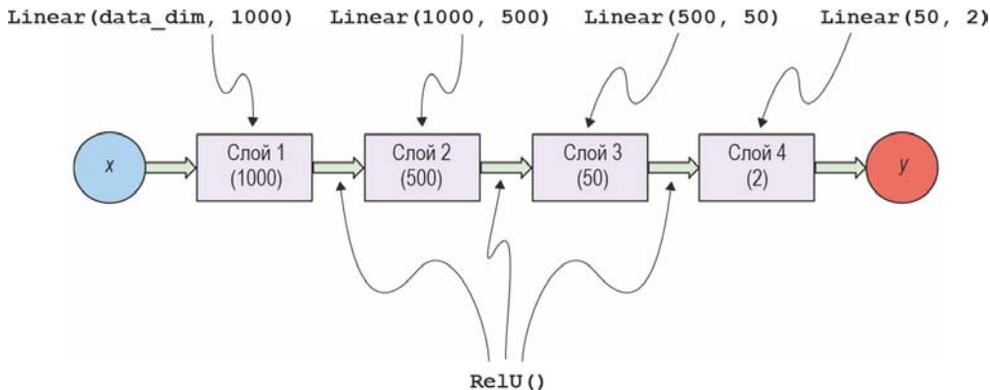


Рис. 13.12. Архитектура реализуемой нейросети и соответствующий код PyTorch: каждый уровень реализуется с помощью `torch.nn.Linear`, а каждая функция активации — с помощью `torch.nn.ReLU`

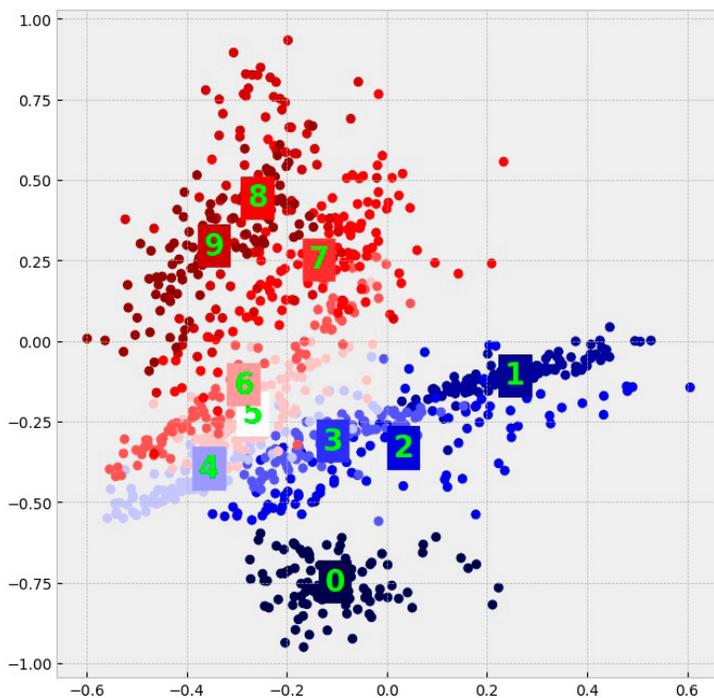


Рис. 13.13. Характеристики, извлеченные из набора данных MNIST с помощью нейросети: точки данных с одной и той же меткой не просто группируются вместе, на графике существует целый градиент меток; при движении снизу вверх значение метки увеличивается

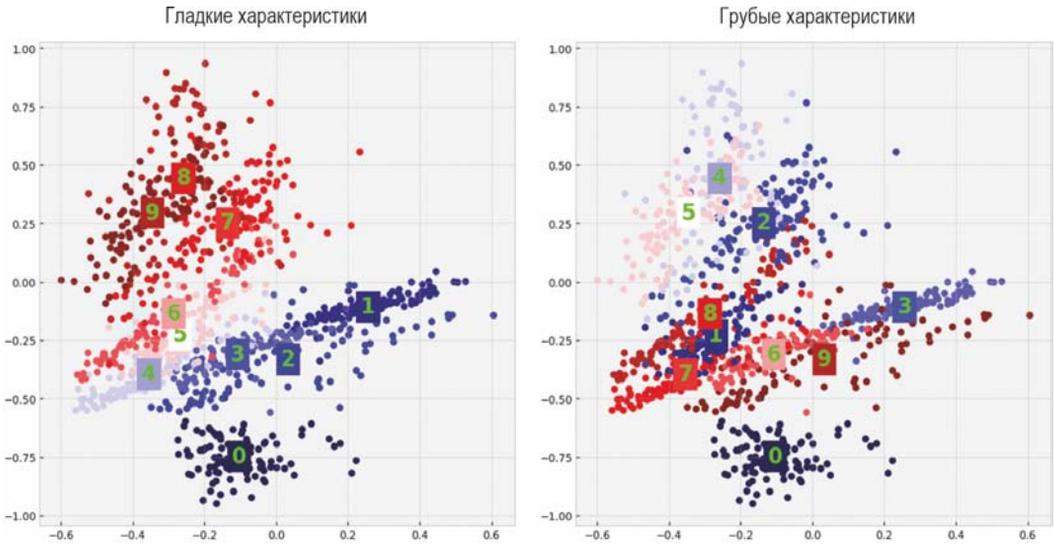


Рис. 13.14. Сравнение сглаженных извлеченных характеристик на рис. 13.13 (слева) и случайной замены меток, делающей объекты грубыми (справа): гладкие черты легче изучить с помощью ГП, чем грубые