1

Введение и мотивация

Машинное обучение — это дисциплина, посвященная разработке алгоритмов, которые автоматически извлекают ценную информацию из имеющихся данных. В данном случае акцент следует сделать на «автоматически», то есть машинное обучение связано с методологиями общего характера, которые применимы к разнообразным наборам данных (датасетам) и позволяют извлечь из них нечто осмысленное. Суть машинного обучения можно описать в виде трех концепций: «данные», «модель» и «обучение».

Поскольку МО по сути своей ориентировано на работу с данными, *данные* являются основной из его ключевых составляющих. Цель МО — проектирование методологий общего характера, позволяющих извлекать ценные закономерности из данных; в идеале для этого почти не должен требоваться опыт работы в предметной области. Например, имея большой корпус документов (скажем, книг из множества библиотек) можно, воспользовавшись методами МО, автоматически найти релевантные темы, общие для множества документов (Hoffman et al., 2010). Для достижения этой цели проектируются *модели*, как правило, связанные с процессом генерации данных, аналогичных датасету, который мы получили. Например, при выполнении регрессии модель будет описывать функцию, которая сопоставляет входные данные с реальными выходными значениями. Перефразируя Митчелла (Mitchell, 1997): «Говорят, что модель обучается на данных, если ее производительность на определенной задаче увеличивается при учете вышеупомянутых данных». Цель — найти эффективные модели, хорошо обобщающие незнакомые данные, которые могут понадобиться нам в будущем. *Обучение* можно понимать как способ автоматического выявления закономерностей и структуры в данных путем оптимизации параметров модели.

Несмотря на то что известно множество историй успеха, связанных с МО, и можно без труда найти программное обеспечение, предназначенное для проектирования и обучения многофункциональных и гибких систем такого рода,

мы считаем, что математические основы МО важны для понимания фундаментальных принципов, на основе которых строятся более сложные системы. Понимание этих принципов способствует созданию новых решений для МО, пониманию и отладке имеющихся подходов, а также изучению неотъемлемых допущений и ограничений тех методологий, с которыми мы работаем.

1.1. ПОИСК ИНТУИТИВНО ПОНЯТНЫХ ФОРМУЛИРОВОК

В машинном обучении мы постоянно сталкиваемся с ситуациями, в которых смысл концепций и слов как бы ускользает, а конкретный компонент системы МО может быть абстрагирован до математических концепций, в которые вкладывается разный смысл. Например, в контексте МО есть два понимания термина «алгоритм». В первом смысле формулировка «алгоритм машинного обучения» означает систему, делающую прогнозы на основании входных данных. Такие алгоритмы называются *предикторами*. Во втором смысле ровно та же фраза, «алгоритм машинного обучения», означает систему, которая адаптирует некоторые внутренние параметры предиктора таким образом, чтобы он хорошо работал на еще не известных данных, которые поступят в будущем. Такая адаптация будет называться *обучением* системы.

Эта книга не решает проблем, связанных с подобной двойственностью, но мы хотим заранее подчеркнуть, что в зависимости от контекста одни и те же выражения могут означать разные вещи. Тем не менее, мы стараемся во всех случаях давать достаточно ясный контекст, чтобы по возможности устранить неоднозначность.

Первая часть книги знакомит читателя с математическими концепциями и основами, необходимыми, чтобы говорить о трех основных компонентах системы машинного обучения: данных, моделях и обучении. Здесь мы кратко обрисуем эти концепции, а затем вновь обратимся к ним в главе 8, после того как будут рассмотрены необходимые математические понятия.

Хотя данные бывают не только числовыми, часто полезно трактовать их именно в числовом формате. В этой книге предполагается, что *данные* уже были требуемым образом преобразованы в числовые представления, подходящие для считывания компьютерной программой. Следовательно, мы понимаем данные как векторы. Еще один пример, иллюстрирующий ненадежность слов, заключается в том, что существует (как минимум) три разных понимания вектора: вектор как массив чисел (трактовка из информатики), вектор как стрелка, у которой есть направление и величина (трактовка из физики), и вектор как объект, поддающийся сложению и умножению (трактовка из математики).

Модель, как правило, используется для описания процесса генерации данных, подобных тем, что содержатся в имеющемся датасете. Следовательно, хорошие модели также можно трактовать как упрощенные версии реального (неизвестного) процесса генерации данных, схватывающие особенности, которые важны для моделирования данных и извлечения из них скрытых закономерностей. В дальнейшем хорошая модель может применяться для прогнозирования того, что произойдет в реальных условиях, без постановки соответствующих реальных экспериментов.

Теперь мы подходим к апогею нашей темы: собственно *обучению* в рамках МО. Допустим, у нас есть датасет и подходящая модель. *Обучение* модели означает использование доступных данных для оптимизации некоторых параметров модели с учетом функции полезности, оценивающей, насколько хорошо модель прогнозирует обучающие данные. Большинство методов обучения можно понимать как попытки взобраться на вершину холма. В данной аналогии вершина холма соответствует максимуму некоторого показателя желаемой производительности. Но на практике мы заинтересованы, чтобы модель хорошо работала на тех данных, которых ей еще не показывали. Хорошая производительность модели на уже показанных ей (обучающих) данных может означать лишь то, что мы нашли хороший способ запоминания этих данных. Однако может оказаться, что модель плохо обобщает заранее не известные данные, а на практике зачастую приходится задействовать нашу модель МО в таких ситуациях, с которыми она ранее не сталкивалась.

Итак, обобщим основные концепции машинного обучения, о которых пойдет речь в этой книге:

- Мы представляем данные в виде векторов.
- Мы выбираем подходящую модель, исходя из вероятностной или оптимизационной точки зрения.
- Мы учимся на доступных данных, используя методы численной оптимизации, и стремимся добиться того, чтобы модель хорошо работала на данных, не применявшихся для ее обучения.

1.2. ДВА СПОСОБА ЧИТАТЬ ЭТУ КНИГУ

Можно рассмотреть две стратегии, помогающие понять математику в рамках машинного обучения:

- **Восходящая**: опираясь на основополагающие концепции, изучаем все более продвинутые. Такой подход зачастую предпочтителен в точных науках, например в математике. Преимущество такой стратегии в том, что читатель в любой момент может обратиться к концепциям, изученным ранее. К со-

жалению, для практика основополагающие концепции не столь интересны сами по себе, и из-за отсутствия мотивации их изучать большинство определений таких базовых концепций быстро забываются.

- **Нисходящая**: конкретизация, сведение практических потребностей к более базовым требованиям. Такой целеориентированный подход хорош тем, что читателю в любой момент ясно, зачем нужно прорабатывать конкретную концепцию, и к нужным знаниям лежит ясный путь. Недостаток такой стратегии заключается в том, что основы таких знаний могут получиться шаткими, и читателю приходится запоминать набор слов, понять которые у него нет никакой возможности.

Мы решили написать эту книгу в виде системы модулей, чтобы отделить базовые (математические) концепции от прикладных, и текст можно было читать обоими вышеупомянутыми способами. Книга разделена на две части. В части I излагаются математические основы, а в части II концепции из части I применяются для решения набора фундаментальных задач машинного обучения, показанных на рис. 1.1: это регрессия, снижение размерности, оценка плотности и классификация. Последующие главы в части I в основном базируются на предыдущих, но при необходимости можно пропустить главу, прочитать следующую, а затем вернуться к пропущенной, если потребуется. Главы в части II связаны очень слабо, и их можно читать в любом порядке. В обоих частях книги расставлено множество отсылок на предыдущие и последующие главы, они связывают математические концепции с алгоритмами машинного обучения.

Разумеется, найдутся и другие способы чтения этой книги, кроме двух вышеупомянутых. Большинству читателей подойдет комбинация восходящего и нисходящего подхода. В некоторых случаях потребуется наработать базовые математические навыки, прежде чем подступаться к сложным концепциям, но также можно выбирать темы, отталкиваясь от возможностей применения машинного обучения.

Часть I — о математике

Четыре столпа машинного обучения, рассматриваемые в этой книге (рис 1.1), требуют основательного математического базиса, и этот базис изложен в части I.

Мы представляем числовые данные в векторном виде, а таблицу таких данных — как матрицу. Изучение векторов и матриц называется *линейной алгеброй,* с ней мы познакомимся в главе 2. Там же матрица описывается как совокупность векторов.

Имея два вектора, представляющих два реальных объекта, мы собираемся делать утверждения об их сходстве. Идея в том, что два схожих вектора должны давать

схожие выводы после обработки нашим алгоритмом МО (предиктором). Чтобы формализовать идею сходства векторов, необходимо ввести операции, принимающие два вектора в качестве ввода и возвращающие числовое значение, отражающее их сходство. Создание сходства и расстояний играет центральную роль в *аналитической геометрии*, рассматриваемой в главе 3.

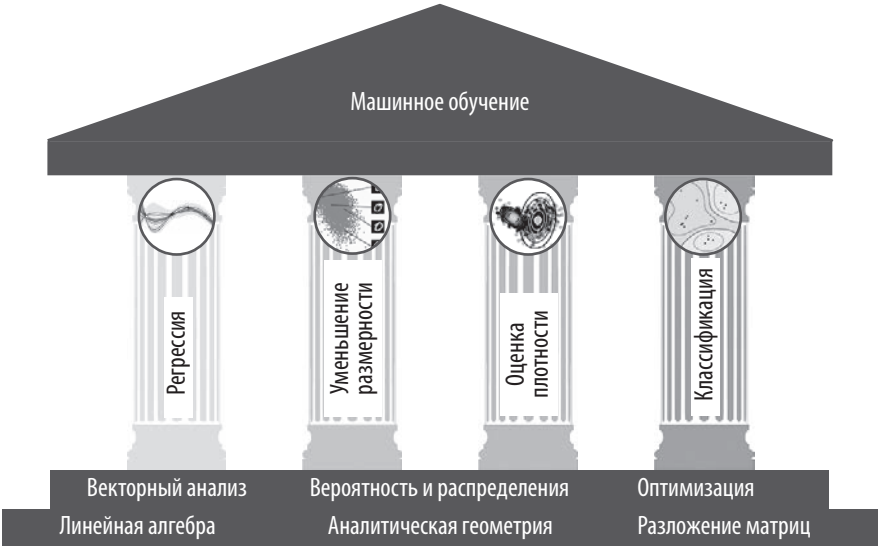

Рис. 1.1. Основания и четыре столпа машинного обучения

В главе 4 вводятся некоторые фундаментальные концепции, касающиеся матриц и их *разложения*. Некоторые операции над матрицами исключительно полезны в МО и обеспечивают интуитивно понятную интерпретацию данных, а также более эффективное обучение.

Часто данные трактуются как зашумленные наблюдения некого истинного базового сигнала. Мы надеемся, что, применив МО, сможем вычленить сигнал из шума. Для этого нам нужен язык, на котором можно было бы количественно выразить, что такое «шум». Часто нам также хотелось бы иметь предикторы, которые позволили бы выразить некоторую неопределенность, например чтобы количественно охарактеризовать степень нашей уверенности в спрогнозированном значении в конкретной точке тестовых данных. *Квантификацией* (количественной оценкой) *неопределенности* занимается теория вероятности, ей посвящена глава 6.

Для обучения моделей обычно подыскиваются параметры, доводящие до максимума некоторую меру производительности. Многие оптимизационные при-

емы требуют понимания концепции градиента, который указывает, в каком направлении искать решение. Глава 5 посвящена *векторному анализу* и подробно описывает концепцию градиентов, которыми мы затем воспользуемся в главе 7; в ней же мы поговорим об *оптимизации* для поиска максимумов и минимумов функций.

Часть II — о машинном обучении

Во второй части книги вы познакомитесь с *четырьмя столпами машинного обучения*, показанными на рис. 1.1. Мы проиллюстрируем, как математические концепции, объясненные в первой части книги, служат основаниями каждого из столпов. В широком смысле, главы упорядочены от простого к сложному.

В главе 8 мы освежим в памяти три компонента МО (данные, модели, оценка параметров), рассмотрев их с математической точки зрения. Кроме того, мы дадим некоторые рекомендации о том, как подбираются экспериментальные установки, помогающие перестраховаться от чрезмерно оптимистичной оценки систем МО. Как вы помните, наша цель — построить предиктор, хорошо работающий на ранее не известных данных.

В главе 9 мы подробно рассмотрим *линейную регрессию* и зададимся целью найти такие функции, которые сопоставляют входные данных $\boldsymbol{x} \in \mathbb{R}^D$ с соответствующими наблюдаемыми значениями функции $y \in \mathbb{R}$, которые мы сможем интерпретировать как метки соответствующих входных данных. Мы обсудим классическую подгонку модели (оценку параметров) с применением методов максимального правдоподобия и максимальной апостериорной оценки, а также байесовскую линейную регрессию, где будем исключать параметры путем интегрирования, а не оптимизировать их.

В главе 10 основное внимание уделяется *снижению размерности*, второму столпу с рис. 1.1; для этого воспользуемся анализом главных компонент. Ключевая цель снижения размерности — найти компактное представление (с малым количеством измерений) для данных, описываемых большим количеством измерений $\boldsymbol{x} \in \mathbb{R}^D$; такое компактное представление зачастую легче анализировать, нежели исходные данные. В отличие от регрессии, снижение размерности зависит только от моделирования данных — нет никаких меток, которые были бы связаны с точкой данных $\boldsymbol{x}$.

В главе 11 мы перейдем к нашему третьему столпу: *оценке плотности*. Назначение оценки плотности — найти вероятностное распределение, описывающее заданный датасет. Изучая эту тему, мы сосредоточимся на моделях гауссовых смесей и обсудим итеративную схему нахождения параметров такой модели. Как и в случае со снижением размерности, здесь нет никаких меток для точек данных $\boldsymbol{x} \in \mathbb{R}^D$. Однако мы не ищем такое представление данных, которое об-

ладало бы низкой размерностью. Нас скорее интересует плотностная модель, которая описывает эти данные.

Глава 12, завершающая книгу, содержит углубленное обсуждение четвертого столпа: *классификации*. Примерно как и при работе с регрессией (глава 9), у нас есть входные значения $\boldsymbol{x}$ и соответствующие им метки y. Однако, в отличие от регрессии, при которой метки имеют вещественные значения, метки при классификации являются целочисленными, поэтому обращаться с ними нужно особенно осторожно.

1.3. УПРАЖНЕНИЯ И ОБРАТНАЯ СВЯЗЬ

Мы приводим некоторые упражнения в части I, и для выполнения большинства из них достаточно ручки и бумаги. К главе II мы подготовили руководства по программированию (это блокноты Jupyter), которые помогут вам исследовать некоторые свойства алгоритмов машинного обучения, рассматриваемых в этой книге.

Мы высоко ценим участие издательства Cambridge University Press, активно поддерживающего нас в нашем стремлении к демократизации образования. Издательство выложило эту книгу для свободного скачивания по адресу

https://mml-book.com,

где также находятся решения упражнений, списки найденных ошибок и дополнительные материалы. Сообщать об ошибках и оставлять отзывы можно по вышеприведенной ссылке.

2

Линейная алгебра

При формализации интуитивно понятных концепций принято подбирать набор объектов (символов) и давать набор правил по обращению с этими объектами. Такая наука называется *алгеброй*. Линейная алгебра — это наука о векторах и определенных правилах операций над ними. Векторы, которые многие помнят из школьного курса, называются «геометрическими» и обычно обозначаются стрелочкой над буквой, например, $\vec{x}$ и $\vec{y}$. В этой книге речь пойдет о более обобщенном представлении векторов, и вектор будет обозначаться жирной латинской буквой, например $\boldsymbol{x}$ и $\boldsymbol{y}$.

В принципе, векторы — это объекты особого рода, которые можно складывать друг с другом и умножать на скаляры, чтобы получить новый объект того же рода. С точки зрения абстрактной математики, любой объект, обладающий двумя этими свойствами, может считаться вектором. Вот несколько примеров таких векторных объектов:

1. Геометрические векторы. Векторы такого рода изучаются в старших классах, в курсах математики и физики. Геометрические векторы — см. рис. 2.1(a) — это направленные отрезки, которые можно чертить (как минимум в двух измерениях). Два геометрических вектора $\vec{\boldsymbol{x}}$, $\vec{\boldsymbol{y}}$ можно сложить, так что их суммой $\vec{\boldsymbol{x}} + \vec{\boldsymbol{y}} = \vec{\boldsymbol{z}}$ будет третий геометрический вектор. Более того, умножение на скаляр, $\lambda\vec{\boldsymbol{x}}$, $\lambda \in \mathbb{R}$, также даст геометрический вектор. Фактически это исходный вектор, умноженный на λ. Следовательно, геометрические векторы — это примеры воплощения концепции вектора, с которой мы познакомились выше. Интерпретируя векторы как геометрические сущности, мы можем интуитивно судить об их направлении и величине, а также рассуждать о математических операциях над ними.
2. Многочлены — это тоже векторы; см. рис. 2.1(b): два многочлена можно сложить друг с другом, получив в результате третий многочлен; также их можно умножать на скаляр $\lambda \in \mathbb{R}$, и в результате тоже получится многочлен.

Следовательно, многочлены — это образцы векторов (пусть и довольно необычные). Обратите внимание на то, что многочлены очень отличаются от геометрических векторов. Тогда как геометрический вектор — это конкретный «рисунок», многочлен — это абстрактная концепция. Однако и те, и другие являются векторами в вышеизложенном смысле.

3. Аудиосигналы — это векторы. Аудиосигнал можно представить как последовательность чисел. Также можно складывать аудиосигналы друг с другом, и их суммой будет новый аудиосигнал. Если умножить аудиосигнал, также получится аудиосигнал. Следовательно, аудиосигналы также являются своеобразными векторами.
4. Элементы $\mathbb{R}^n$ (кортежи из n вещественных чисел) — это векторы. $\mathbb{R}^n$ более абстрактны, чем многочлены, и именно этой концепции уделяется особое внимание в данной книге. Так,

$$\boldsymbol{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^3 \tag{2.1}$$

— это пример тройки чисел. Покомпонентное сложение двух векторов $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^n$ дает еще один вектор: $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{c} \in \mathbb{R}^n$. Более того, при умножении $\boldsymbol{a} \in \mathbb{R}^n$ на $\lambda \in \mathbb{R}$ получается умноженный вектор $\lambda\boldsymbol{a} \in \mathbb{R}^n$. Рассматривать векторы как элементы $\mathbb{R}^n$ удобно еще и потому, что в таком представлении они условно соответствуют массивам вещественных чисел с точки зрения компьютера[1]. Во многих языках программирования поддерживаются операции над массивами, благодаря чему удобно реализовывать алгоритмы, связанные с выполнением операций над векторами.

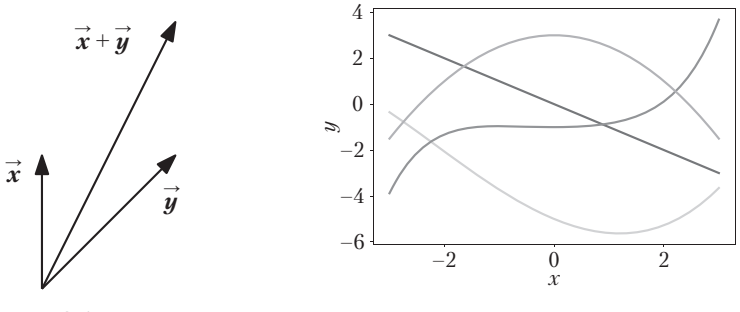

Рис. 2.1. Векторы разных типов. Векторы порой удивительны, и к ним относятся как (*a*) геометрические векторы, так и (*b*) многочлены

[1] Тщательно проверьте, на самом ли деле операции над массивами тождественны операциям над векторами, если реализовать их на компьютере.

В линейной алгебре особое внимание уделяется сходству между двумя этими векторными концепциями. Такие векторы можно складывать друг с другом и умножать на скаляры. Мы сосредоточимся преимущественно на векторах из $\mathbb{R}^n$, так как большинство алгоритмов линейной алгебры формулируются в $\mathbb{R}^n$. В главе 8 будет показано, что данные часто трактуются как векторы в $\mathbb{R}^n$. В этой книге мы сосредоточимся на изучении конечномерных векторных пространств, где существует однозначное соответствие между вектором любого рода и $\mathbb{R}^n$. Когда это будет удобно, мы будем использовать аналогии из области геометрических векторов, рассуждая об алгоритмах, основанных на массивах.

Одной из важнейших идей в математике является замыкание. Вот вопрос: каково множество всех результатов, которые могут быть получены при выполнении предлагаемых мной операций? В случае векторов формулировка такова: какое множество векторов можно получить, взяв за основу небольшой набор векторов, а затем складывая и нормируя их? В результате получится векторное пространство (раздел 2.4). На концепции векторного пространства и его свойствах во многом базируется машинное обучение. Концепции, введенные в этой главе, обобщены на рис. 2.2.

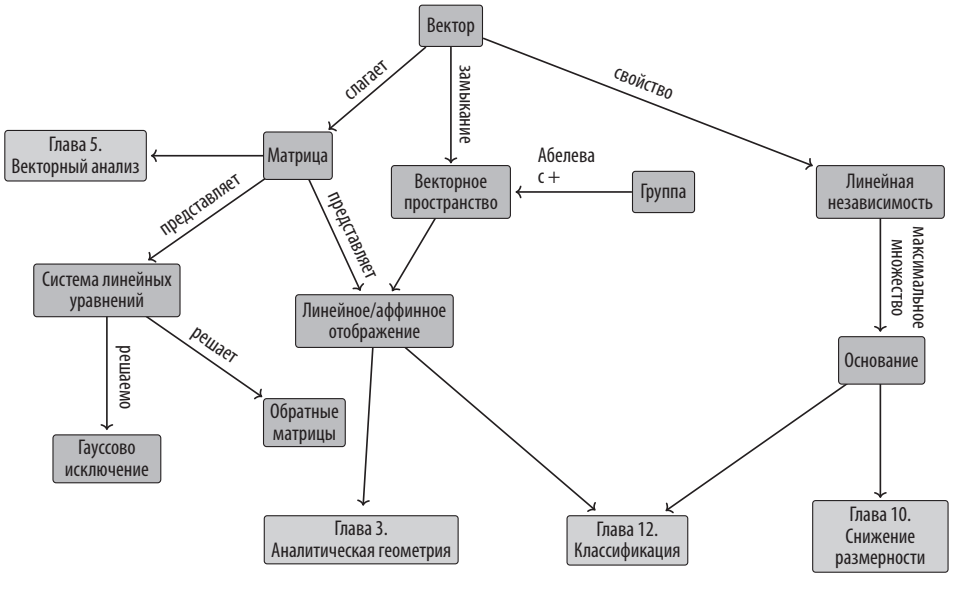

Рис. 2.2. Ассоциативная карта концепций, вводимых в этой главе, с указанием, в каких еще частях книги они фигурируют

Эта глава большей частью основана на конспектах и книгах Drumm and Weil (2001), Gilbert Strang (2003), Hogben (2013), Liesen and Mehrmann (2015), а так-

же на серии Павла Гринфельда (Pavel Grinfeld) «Линейная алгебра» (http://tinyurl.com/nahclwm). Другие отличные ресурсы — это курс Гилберта Стрэнга по линейной алгебре (MIT) (http://tinyurl.com/29p5q8j) и серия «Линейная алгебра» с 3Blue1Brown (https://tinyurl.com/h5g4kps).

Линейная алгебра играет важную роль в машинном обучении и в математике в целом. Концепции, включенные в эту главу, далее раскрываются в главе 3 с захватом геометрии. В главе 5 мы поговорим о векторном анализе, где важны хорошо усвоенные знания об операциях над матрицами. В главе 10 мы будем пользоваться проекциями (с ними вы познакомитесь в разделе 3.8) для снижения размерности с применением анализа главных компонент. В главе 9 мы поговорим о линейной регрессии, где линейная алгебра играет центральную роль в решении задач наименьших квадратов.

2.1. СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

Системы линейных уравнений играют центральную роль в линейной алгебре. Многие задачи можно сформулировать в виде систем линейных уравнений, а линейная алгебра предоставляет инструментарий для их решения.

Пример 2.1

Компания производит линейку продуктов $N_1, ..., N_n$, для которых требуются ресурсы $R_1, ..., R_m$. На производство единицы продукта N_j требуется a_{ij} единиц ресурса R_i, где $i = 1, ..., m$, а $j = 1, ..., n$.

Цель — подобрать оптимальный производственный план, то есть спланировать, сколько единиц x_j продукта N_j должно быть произведено, если доступно всего b_i единиц ресурса R_i и, в идеале, неизрасходованных ресурсов не остается.

Если мы произведем $x_1, ..., x_n$ единиц соответствующего продукта, то нам понадобится всего

$$a_{i1}x_1 + \cdots + a_{in}x_n \tag{2.2}$$

единиц ресурса R_i. Следовательно, оптимальный производственный план $(x_1, ..., x_n) \in \mathbb{R}^n$ должен удовлетворять следующей системе уравнений:

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m, \end{aligned} \tag{2.3}$$

где $a_{ij} \in \mathbb{R}$ и $b_i \in \mathbb{R}^n$.

Уравнение (2.3) — это обобщенная форма *системы линейных уравнений*, а $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ — это *неизвестные* данной системы. Каждый n-кортеж $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathbb{R}^n$, удовлетворяющий (2.3), является *решением* системы линейных уравнений.

Пример 2.2

Система линейных уравнений

$$\begin{array}{rcrcrcll} x_1 & + & x_2 & + & x_3 & = & 3 & (1) \\ x_1 & - & x_2 & + & 2x_3 & = & 2 & (2) \\ 2x_1 & & & + & 3x_3 & = & 1 & (3) \end{array} \tag{2.4}$$

не имеет решения. Сложив два первых уравнения, имеем $2x_1 + 3x_3 = 5$, что противоречит третьему уравнению (3).

Рассмотрим систему линейных уравнений

$$\begin{array}{rcrcrcll} x_1 & + & x_2 & + & x_3 & = & 3 & (1) \\ x_1 & - & x_2 & + & 2x_3 & = & 2 & (2) \\ & & x_2 & + & 3x_3 & = & 2 & (3). \end{array} \tag{2.5}$$

Из первого и третьего уравнения следует, что $x_1 = 1$. Из (1) + (2) имеем, что $2x_1 + 3x_3 = 5$, то есть $x_3 = 1$. Затем из (3) получаем, что $x_2 = 1$. Следовательно, (1, 1, 1) — это единственно возможное, *единственное решение* (чтобы убедиться, что (1, 1, 1) является решением, подставьте его в уравнение).

В качестве третьего примера рассмотрим:

$$\begin{array}{rcrcrcll} x_1 & + & x_2 & + & x_3 & = & 3 & (1) \\ x_1 & - & x_2 & + & 2x_3 & = & 2 & (2) \\ 2x_1 & & & + & 3x_3 & = & 5 & (3). \end{array} \tag{2.6}$$

Поскольку (1) + (2) = (3), третье уравнение можно опустить (оно избыточно). Из (1) и (2) следует, что $2x_1 = 5 - 3x_3$ и $2x_2 = 1 + x_3$. Мы определяем $x_3 = a \in \mathbb{R}^n$ как свободную переменную, такую что любая тройка

$$\left(\frac{5}{2} - \frac{3}{2}a,\ \frac{1}{2} + \frac{1}{2}a,\ a\right), a \in \mathbb{R} \tag{2.7}$$

является решением системы линейных уравнений, то есть мы получаем множество, в котором содержится *бесконечное количество* решений.