2 Рекурсивная оценка состояния

2.1 Введение

В основе вероятностной робототехники лежит идея оценки состояния, имея данные датчиков. Оценка состояния представляет собой проблему оценки численных значений, которые напрямую измерить невозможно, но можно воспринять с помощью данных датчиков.

В большинстве приложений робототехники довольно просто определить, что же нужно делать, если бы только были известны точные значения. Например, довольно легко перемещать мобильного робота, если известны точные местоположения как самого робота, так и всех ближайших препятствий. К сожалению, напрямую измерить эти величины невозможно. Вместо этого робот вынужден полагаться на датчики, чтобы получить нужную информацию. Датчики же способны передать только частичную информацию об этих величинах, а их измерения могут искажаться шумами. Оценка состояний производится в целях восстановления состояния переменной на основе данных. Вероятностные алгоритмы оценки состояния вычисляют распределения оценок в пространстве возможных состояний. Примеры вероятностной оценки состояний были уже приведены во вводной части книги – это проблема локализации мобильного робота.

Целью этой главы является представления базовой системы понятий и математических инструментов для оценки состояния на основе данных датчиков.

• В разделе 2.2 вводятся основные концепции теории вероятности, использованные в книге

• В разделе 2.3 описывается наша формальная модель взаимодействия робота с окружающей средой, устанавливающая ключевую терминологию, которая используется в книге.

• В разделе 2.4 впервые описываются байесовские фильтры, рекурсивный алгоритм оценки состояния, образующий первооснову для практически каждого метода, представленного в книге.

• В разделе 2.5 обсуждаются трудности представлений и вычислений, которые возникают при использовании байесовских фильтров.

2.2 Основные концепции теории вероятности

Этот раздел знакомит читателя с основными обозначениями и фактами теории вероятности, которые используются в книге.

В вероятностной робототехнике все величины, такие как измерения датчиков, управление, состояния робота и окружающей среды, моделируются случайными переменными.

СЛУЧАЙНЫЕ ВЕЛИЧИНЫ

Случайные переменные способны принимать несколько значений, и делать это согласно определенным правилам теории вероятности.

Вероятностное заключение – это процесс вычисления таких правил для случайных величин, зависящих от других случайных величин и известных данных наблюдений.

Примем, что X означает случайную переменную, а x – конкретное значение, которое может принимать X. Стандартным примером случайной переменной является исход броска монеты, где X может принимать значения «орла» или «решки». Если пространство всех значений, которые может принимать X, дискретно, как в случае броска монеты, то можно записать

(2.1) p(X = x)

Обозначая вероятность того, что случайная переменная X принимает значение x. Например, для «честной» монеты эта величина характеризуется как p(X = «орел») = p(X = «решка») = 1/ 2. Сумма всех дискретных вероятностей равна единице, поэтому

x

(2.2) p(X = x) = 1

Вероятности также всегда неотрицательны, и, в силу этого, p(X = x) ≥ 0.

Для упрощения записи мы, при возможности, будем опускать точное указание случайной переменной, и использовать обычное сокращение

p(x) вместо p(X = x).

Большинство методов в этой книге имеет дело с оценкой и процессом принятия решения в непрерывных пространствах. Непрерывные пространства характеризуются тем, что случайные переменные могут принимать диапазон непрерывных значений.

Если не указано специально, будем считать, что

ПЛОТНОСТЬ ВЕРОЯТНОСТИ

Все непрерывные случайные переменные имеют функции плотности вероятности (ФПВ - PDF). Обычно функция плотности представляет собой одномерное нормальное распределение

НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ

С математическим ожиданием μ и среднеквадратичным отклонением σ2. ФПВ нормального распределения

ГАУССИАН

Выражен следующей функций Гаусса:

p(x) = … …

(2.3) exp− 21 (x −σ2μ)2

Нормальные распределения играют важную роль в книге. Часто мы будем обозначать их как

N (x; μ, σ2), что означает случайную переменную, математическое ожидание и среднеквадратичное отклонение.

Для нормального распределения (2.3) считается, что x –это скалярное значение. Однако, часто x будет представлен многомерным вектором. Нормальные распределения векторов называются *многомерными*

МНОГОМЕРНОЕ РАСПРЕДЕЛЕНИЕ

Многомерные нормальные распределения характеризуются функциями распределения вероятности следующего вида:

p(x) = …

(2.4) …

Здесь μ многомерный вектор. Σ неотрицательно определенная симметричная матрица, называемая *ковариационной*

КОВАРИАЦИОННАЯ МАТРИЦА

Верхний индекс T означает транспонирование вектора. Аргумент экспоненты в функции представляет собой вторую степень x, с параметрами квадратичной функции μ и Σ.

Читателю следует обратить внимание, что Уравнение (2.4) является строгим обобщением Уравнения (2.3). Оба уравнения эквиваленты для скалярного значения x и Σ = σ2.

Уравнения (2.3) и (2.4) – это примеры функций распределения вероятности. Точно так же, как сумма дискретного распределения вероятности равна 1, интеграл ФПВ всегда равен единице:

(2.5) p(x) dx = 1

Однако, в отличие от дискретной вероятности, значение ФПВ не ограничено сверху единицей. В ходе изложения в этой книге понятия вероятности, плотности вероятности и функции плотности вероятности будет использоваться как взаимозаменяемые. Мы будем по умолчанию считать, что все непрерывные случайные переменные измеримы, а также, что для всех непрерывных распределений существуют плотности.

СОВМЕСТНОЕ РАСПРЕДЕЛЕНИЕ

Совместное распределение двух случайных переменных X и Y выражается в виде

(2.6) p(x, y) = p(X = x and Y = y)

Это выражение описывает вероятность наступления события значения x случайной переменной X и y - для переменной Y. Если X и Y

НЕЗАВИСИМОСТЬ

независимы, то получится, что

(2.7) p(x, y) = p(x) p(y)

Часто случайные переменные содержат определенную информацию о других случайных переменных.

Допустим, уже известно, что значение Y равно y, и хотелось бы узнать вероятность того, что значение x переменной X связано с этим фактом. Такая вероятность будет равна

(2.8) p(x | y) = p(X = x | Y = y)

УСЛОВНАЯ ВЕРОЯТНОСТЬ

И называться условной вероятностью. При p(y) > 0 условная вероятность определяется как

p(x | y) = p(x, y)

(2.9) p(y)

Если X и Y независимы, то получится, что

p(x | y) = p(x) p(y)

(2.10) p(y) = p(x)

Другими словами, при независимости X и Y, значение Y ничего не скажет о значении X. Если мы заинтересованы в значении X, знание значения Y никакого преимущества не дает. Независимость, и ее генерализация, известная как условная независимость играют очень важную роль в этой книге.

Интересный факт, который является следствием определения условной вероятности

ТЕОРЕМА О ПОЛНОЙ ВЕРОЯТНОСТИ

и аксиом вероятностных мер, часто называют Теоремой о полной вероятности:

p(x) = …..

(2.11) p(x | y) p(y) (для дискретного случая)

(2.12) p(x) = p(x | y) p(y) dy (для непрерывного случая)

Если p(x | y) или p(y) нулевые, можно принять произведение p(x | y) и p(y) равным нулю,

вне зависимости от значения степени.

ТЕОРЕМА БАЙЕСА

Также очень важна теорема Байеса (в оригинале «правило Байеса», хотя приведена именно нотация теоремы???? – прим перев), которая определяет отношение типа p(x |

y) к его «отражению» p(y | x). Для теоремы требуется, как было указано, p(y) > 0:

p(x | y) = p(y | x) p(x)

p(y) =….

(2.13) x p(y | x) p(x) (дискретный случай)

p(x | y) = p(y | x) p(x)

p(y) =….

(2.14)

p(y | x) p(x) dx(непрерывный случай)

Теорема Байеса играет определяющую роль в вероятностной робототехнике (как и вообще в вероятностных выводах). Если x это количество, которое бы нам хотелось узнать на основании y,

АПРИОРНАЯ ВЕРОЯТНОСТЬ

То вероятность p(x) будет основана на прошлом распределении вероятности, а y

Называться данными (например, измерений датчиков). В распределении p(x) суммируются все данные, имеющиеся об X перед учетом данных y.

АПОСТЕРИОРНАЯ ВЕРОЯТНОСТЬ

Вероятность p(x | y) называется апостериорным распределением вероятности по X.

В силу (2.14), теорема Байеса дает удобный способ вычислить апостериорную вероятность p(x | y) на основе «перевернутой» условной вероятности p(y | x) и априорной вероятности p(x). Другими словами, если необходимо получить значение x на основе данных датчиков y, теорема Байеса позволяет сделать это с помощью обратной вероятности, определяющей данные

y при условии, что событие x имело место.

ПОРОЖДАЮЩАЯ МОДЕЛЬ

В робототехнике вероятность p(y | x) часто называют порождающей моделью, поскольку она, на некотором уровне абстракции, описывает, как переменные состояния X вызывают измерения датчиков Y .

Важно отметить, что знаменатель в теореме Байеса, p(y), не зависит от x. Таким образом, множитель p(y)−1 в уравнениях (2.13) и (2.14) будет одинаковым при любом значении x апостериорной вероятности p(x | y). В силу этого, p(y)−1 часто записывают как нормирующим членом в теореме Байеса и, в общем виде, записывается как η:

(2.15) p(x | y) = η p(y | x) p(x)

Преимуществом такой записи является ее лаконичность. Вместо явной записи точной формулы нормирующей константы (а она может вырасти до весьма больших размеров в некоторых математических выводах), мы просто будем писать символ нормировки η, чтобы показать, что конечный результат должен быть нормирован до единицы. В этой книге подобные нормировщики будут записываться как η(или η, η, . . . ).

Важно: Мы будем произвольно использовать одно и то же обозначение нормирующего члена η в различных равенствах даже если его реальное значение различается.

Заметим, что будет совершенно справедливым переписать любую из обсуждаемых теорем, используя произвольные случайные переменные, например, переменную Z. Например, возможно записать таким образом теорему Байеса, вводя Z = z, что даст следующее равенство:

p(x | y, z) = p(y | x, z) p(x | z)….

(2.16) p(y | z)…

пока p(y | z) > 0.

Аналогично, мы можем переписать правило для комбинирования вероятностей независимых случайных переменных (2.7) с помощью других переменных z:

(2.17) p(x, y | z) = p(x | z) p(y | z)….

УСЛОВНАЯ НЕЗАВИСИМОСТЬ

Такое отношение называется условной независимостью. Читатель может легко убедиться, что (2.17) эквивалентно

(2.18) p(x | z) = p(x | z, y)

(2.19) p(y | z) = p(y | z, x)

Условная независимость играет важную роль в вероятностной робототехнике.

Она применяется, когда переменная y не несет никакой информации о переменной x,

Если известно значение другой переменной z. Условная независимость не предполагает (абсолютной) независимости, таким образом,

(2.20) p(x, y | z) = p(x | z) p(y | z) ⇒ p(x, y) = p(x) p(y)….

Обратно также, в общем, неверно: абсолютная независимость не предполагает условной независимости:

(2.21) p(x, y) = p(x) p(y) ⇒ p(x, y | z) = p(x | z) p(y | z)….

Однако, в некоторых особых случаях, условная и абсолютная независимость могут совпадать.

В ряде вероятностных алгоритмов требуется вычислять признаки или статистики

МАТЕМАТИЧЕСКОЕ ОЖИДАНИЕ СЛУЧАЙНОЙ ПЕРЕМЕННОЙ

вероятностных распределений. Математическое ожидание случайной величины

X обозначается как

E[X] = …

(2.22) x p(x) (дискретный случай)

(2.23) E[X] = x p(x) dx (непрерывный)

Не все случайные переменные имеют конечное математическое ожидание, однако, такие величины в материале книге рассматриваться не будут.

Математическое ожидание – это линейная функция случайной переменной. В частности, можно записать

(2.24) E[aX + b] = aE[X] + b…

Для произвольных числовых значений a and b. Ковариация X получается следующим образом

(2.25) Cov[X] = ….

Ковариация означает квадрат ожидаемого отклонения математического ожидания. Как было указано ранее, математическое ожидание мультивариационного нормального распределения N (x; μ, Σ) равно μ, а его ковариация - Σ.

ЭНТРОПИЯ

Последней важной концепцией этой книги является энтропия. Энтропия вероятностного распределения задается следующим выражением:

Hp…

(2.26) (x) = …

Которое разрешается до

Hp

….

(2.27) p(x) log2 p(x) (дискретный случай)

Hp

(2.28) (x) = − p(x) log2 p(x) dx (непрерывный)

(картинка)

Концепция энтропии происходит из теории информации. Энтропия- это ожидаемое количество информации, которое несет значение x. В дискретном случае, для

− log2 p(x) это количество бит, требуемое для кодирования x оптимальной кодировкой, подразумевая, что p(x) – это вероятность наблюдения x. В данной книге понятие энтропии будет использовано в контексте сбора информации роботом, выражая, таким образом, количество информации, которую робот может получить, совершив определенные действия.

2.3 Взаимодействие робота с окружающей средой

На Рис. 2.1 показано взаимодействие робота с окружающей средой. Окружающая среда

ОКРУЖАЮЩАЯ СРЕДА РОБОТА

или «мир» - это динамическая система, обладающая внутренним состоянием.

Робот способен получать информацию об окружающем, используя датчики.

Однако, с датчиков поступает зашумленный сигнал, а, вдобавок, имеется множество явлений, которые невозможно воспринимать с помощью датчиков напрямую.

Как следствие, робот поддерживает некую внутреннюю гипотезу относительно состояния окружающей среды, что показано слева на схеме.

Робот также способен воздействовать на окружающую среду с помощью приводов. Эффект таких действий очень часто несет некоторую непредсказуемость.

Таким образом, каждое действие управления влияет как на состояние окружающей среды, так и на внутреннюю гипотезу робота относительно этого состояния.

Опишем это взаимодействие в формальном виде.

2.3.1 Состояние

СОСТОЯНИЕ

Окружающая среда характеризуется состоянием. В материале, представленном в этой книге, будет удобно понимать термин «состояние» как набор всех отличительных черт робота и его окружения, которые могут повлиять на будущее.

Некоторые переменные состояния изменяются со временем, например, местонахождение людей в непосредственной близости от робота. Другие остаются постоянными, такие, как расположение стен в (большинстве) зданий

Изменяющиеся состояния назовем динамическими, чтобы отличать их от статических, то есть неизменных, состояний. Состояние также включает переменные, касающиеся самого робота, такие как его положение, скорость, исправность датчиков и так далее.

В книге состояние будет обозначаться как x, несмотря на то, что конкретные переменные, включенные в x, могут различаться, в зависимости от контекста. Состояние в момент времени t будет обозначаться как xt. В книге используются следующие типичные переменные состояния:

ПОЛОЖЕНИЕ

• Положение робота, образованное его местоположением и ориентацией в пространстве относительно глобальной координатной сетки.

Жесткие мобильные роботы имеют шесть переменных состояния, три для прямоугольной системы координат и три для углов положения в пространстве (крен, тангаж, рысканье). Для жестких мобильных роботов, действующих в плоских окружающих средах, положение обычно задается тремя переменными, двумя координатами на плоскости и направлением движения (рысканье).

• При управлении роботом его положение включает переменные для конфигурации приводов. Например, они могут включать углы для вращающихся сочленений.

Каждая степень свободы робототехнического манипулятора характеризуется одномерной конфигурацией в любой момент времени, и является составляющей кинематического состояния робота. Конфигурация робота часто обозначается как кинематическое состояние.

• Скорость робота и скорости сочленений обычно обозначаются как динамическое состояние. Жесткий робот, передвигающийся в пространстве, характеризуется максимум шестью переменными скорости, по одной для каждого положения. Динамическое состояние играет очень небольшую роль в книге.

• Местоположение и признаки окружающих объектов мира также являются переменными состояния. Объектом может быть дерево, стена или даже пиксель более обширной поверхности. Характеристиками таких объектов может быть их внешний вид (цвет, текстура). В зависимости от степени детализации моделируемого состояния, окружающая среда робота имеет от нескольких десятков до сотен миллионов переменных состояния (и более). Просто представьте, сколько бит потребуется для точного описания физического мира! Для многих проблем, освещаемых в книге, местоположение объектов в окружающей среде

ОРИЕНТИР

будет неизменным. Для некоторых задач объекты могут считаться разновидностью ориентиров, которые представляют собой хорошо различимые, стационарные признаки окружающей среды, которые можно надежно распознать.

• Местоположение и скорости движущихся объектов и людей также являются потенциальными переменными состояния. Часто робот является не единственным движущимся объектом в окружающей среде. Другие движущиеся сущности имеют собственный набор из кинематического и динамического состояний.

• Присутствует множество других переменных состояния, которые могут повлиять на ориентацию робота в пространстве. Например, переменной состояния может быть факт исправности датчика, или же уровень заряда аккумулятора для робота, работающего от аккумуляторов. Список потенциальных переменных состояний бесконечен!

ПОЛНОЕ СОСТОЯНИЕ

Состояние xt назовем полным, если оно является наилучшим предиктором будущего. Другими словами, полнота означает, что знание прошлых состояний, измерений или управляющих действий не несет дополнительной информации, которая могла бы помочь более точно предсказать будущее. Важно заметить, что наше определение полноты не требует того, чтобы будущее было детерминированной функцией состояния.

Будущее может быть стохастическим, но никакая переменная, кроме xt, не может повлиять на стохастическое развитие будущих состояний, если только эта зависимость не медиируется через состояние xt.

Временные процессы, отвечающие этим условиям, общеизвестны

МАРКОВСКИЕ ЦЕПИ

как марковские цепи.

Категория полноты состояния, в основном, имеет лишь теоретическую важность. На практике невозможно определить полное состояние для любой реалистичной робототехнической системы. Полное состояние включает не только все аспекты окружающего, которые могут повлиять на будущее, но также и самого робота, содержимое памяти компьютера, образы мозга окружающих людей, и так далее. Некоторые данные довольно трудно получить и, в силу этого, практические реализации ограничены лишь небольшим набором из всех переменных состояния, таких как перечисленные выше. Такое состояние называется

НЕПОЛНОЕ СОСТОЯНИЕ

неполным.

В большинстве задач практического использования роботов состояние непрерывно и xt определено на непрерывном множестве. Хорошим примером непрерывного пространства состояний является положение робота, представляющее совокупность местоположения и ориентации относительно внешней системы координат. Иногда, состояние дискретно. Примером дискретного пространства состояний является (бинарная) переменная состояния, моделирующая факт исправности датчика. Пространства состояний, которые содержат как непрерывные, так и дискретные переменные, называются гибридными. В большинстве интересующих нас задач робототехники пространства изменяются со временем.

В данной книге время будет считаться дискретным, что означает, что все интересные события произойдут в дискретные моменты времени t = 0, 1, 2 . . .. Если робот начинает функционировать в определенный момент времени, будем считать этот момент как t = 0.

2.3.2 Взаимодействие со средой

Есть два основных типа взаимодействия между роботом и средой: робот может влиять на состояние своего окружения с помощью приводов и может собирать информацию об этом состоянии с помощью датчиков.

Оба типа взаимодействия могут происходить одномоментно, но, в целях удобства изложения, в книге они будут разделены. Взаимодействие показано на Рис. 2.1.

• Измерение параметров окружающей среды датчиками. Восприятие, это процесс, при котором робот использует датчики для получения данных о состоянии окружающей среды. Например, робот способен принимать изображение с камеры, определить дальность до препятствия или запросить данные с тактильных датчиков, чтобы получить информацию о состоянии окружающей среды. Результат такого взаимодействия в целях восприятия будем называть

ИЗМЕРЕНИЕ

измерением, хотя иногда будут использоваться такие термины, как наблюдение или представление. Обычно, информация измерений поступает с некоторой задержкой, и описывает состояние, которое было несколько мгновений назад.

• Действия управления изменяют состояние окружающего мира с помощью

УПРАВЛЯЮЩЕЕ ДЕЙСТВИЕ

приложения сил к предметам окружающей среды. Примеры управляющих действий включают движение робота и манипулирование объектами. Даже если робот не выполняет никаких действий, состояние обычно изменяется. Поэтому, для сохранения целостности изложения, будем считать, что робот всегда выполняет управляющее действие, даже если было принято решение не перемещать никаких двигателей.

На практике, робот непрерывно и одновременно выполняет управляющие действия и измерения.

Теоретически, робот может сохранять записи всех прошлых измерений и управляющих действий. Условимся называть такую коллекцию данными (вне зависимости от того, были ли они сохранены). Поскольку выполняется два типа взаимодействий с окружающей средой, робот имеет доступ к двум разным потокам данных.

• Данные измерений параметров окружающей среды предоставляют информацию о текущем состоянии окружения. Примерами данных измерений служат изображение с камеры, определение расстояния и так далее. В большинстве разделов книги мы будем просто игнорировать малые эффекты запаздывания (например, большинство лазерных датчиков сканируют окружающее пространство последовательно на очень высоких скоростях, но мы просто будем считать, что есть измерение, относящееся к конкретному моменту во времени). Данные измерений в момент времени t будет обозначаться, как zt.

По большей части, в течение изложения в этой книге, будем просто считать, что робот выполняет в точности одно измерение в один момент времени. Это допущение, в основном, для удобства обозначений, поскольку почти все алгоритмы книги могут быть легко расширены для роботов, которые получают различное число измерений в течение одного такта времени.

Запись

(2.29)

Обозначает набор всех переменных, полученных от момента времени t1 до t2, при этом

t1 ≤ t2.

• Данные управления содержат информацию об изменении состояния в окружающей среде. В мобильной робототехнике типичным примером данных управления является скорость робота. Установка скорости в значение 10 см в секунду на пять секунд предполагает, что после выполнения команды на движение робот переместится в положение, приблизительно на 50 см вперед от начального. Таким образом, управление передает информацию об изменении состояния.

ОДОМЕТР

Дополнительным источником данных управления являются одометры – датчики, которые измеряют количество оборотов колес робота и, таким образом, передают информацию об изменении состояния. Хотя, строго говоря, одометры – это датчики, условимся считать одометрию данными управления, поскольку измеряется эффект управляющего действия.

Данные управления будут обозначаться как ut. Переменная ut будет всегда связана с изменением состояния в интервале времени (t − 1; t]. Как и ранее, обозначим последовательности данных управления через ut1:t2, for t1 ≤ t2:

(2.30) ut1:t2 = ut1, ut1+1, ut1+2, . . . , ut2

Поскольку среда может изменяться даже если робот не выполняет никакого определенного действия управления, сам факт изменения времени, технически говоря, содержит информацию об управлении. В силу этого, условимся, что каждый такт времени t происходит ровно одно действие управления, и добавим в список разрешенных действие «не делать ничего».

Разница между измерением и управлением критически важна, поскольку в материале, который будет излагаться, оба типа данных играют фундаментально различные роли. Восприятие окружающей среды предоставляет информацию о состоянии окружающей среды, таким образом, увеличивая осведомленность робота. Движение, с другой стороны, часто ведет к потере информации из-за наличия собственных шумов приводов и стохастичности окружающей среды. Такое разделение ни в коей мере не предполагает, что восприятие и действия разделены во времени. Скорее, восприятие и управление осуществляются одновременно и наше разделение преследует лишь цели удобства изложения.

2.3.3 Вероятностные генеративные правила

Эволюция состояний и измерений управляется правилами теории вероятности.

В общем случае, состояние xt стохастически возникает из состояния xt−1. В силу этого, имеет смысл обозначить вероятностное распределение, из которого было сгенерировано xt.

На первый взгляд, появление состояния xt может быть обусловлено всеми предыдущими состояниями, измерениями и действиями управления. Следовательно, вероятностные правила, характеризующие эволюцию состояния, могут быть изложены в виде вероятностного распределения следующего вида:

: p(xt | x0:t−1, z1:t−1, u1:t). Обратите внимание на отсутствие какой-либо причины считать, что робот сначала совершает действие управления u1, а затем – производит измерение z1.

Важным является следующее рассуждение: Если состояние x полное, значит оно в достаточной степени суммирует все, произошедшее на предыдущих тактах времени. В частности, xt−1 является достаточной статистической величиной для всех предыдущих действий управления и измерений вплоть до текущего момента времени,

, u1:t−1 и z1:t−1, соответственно. Из всех переменных, приведенных выше, только управление ut имеет значение, если известно состояние xt−1.

В терминах теории вероятности, это рассуждение можно выразить следующим равенством:

(2.31) p(xt | x0:t−1, z1:t−1, u1:t) = p(xt | xt−1, ut)

Выраженное этим равенством свойство является примером условной независимости. Утверждается, что некоторые переменные независимы от других, если известны значения третьей группы переменных, называемых условными переменными. Условная независимость будет широко использована в книге, поскольку она – главная причина вычислительной разрешимости большинства приведенных алгоритмов.

Может возникнуть желание смоделировать процесс на основании того, какие измерения были сгенерированы. И снова, если xt полная, наблюдается важная условная независимость:

(2.32) p(zt | x0:t, z1:t−1, u1:t) = p(zt | xt)

Другими словами, переменной состояния xt достаточно для предсказания (потенциально зашумленного) измерения zt. Знание любых других переменных, таких, как прошлые измерения, управление или даже прошлых состояний, роли не играет, при условии, если xt полная.

(картинка 2.2)

Рис. 2.2 Динамическая байесовская сеть, характеризующая эволюцию управления, состояний и измерений.

Это обсуждение раскрывает смысл двух результирующих условных вероятностей: p(xt | xt−1, ut) и p(zt | xt). Вероятность p(xt | xt−1, ut)

ВЕРОЯТНОСТЬ

ПЕРЕХОДА СОСТОЯНИЙ

Это вероятность перехода состояний. Она определяет, изменение состояния среды со временем в виде функции от управления роботом ut. Окружающие среды робота имеют стохастический характер, что отражено фактом того, что p(xt | xt−1, ut) представляет собой вероятностное распределение, а не детерминированную функцию. В некоторых случах распределение вероятности перехода состояний не зависит от временного индекса t, и в этом случае мы можем написать p(x[1] | u, x), где x[1] последующее, а x – предыдущее состояние.

Вероятность измерения

Вероятность p(zt | xt) называется вероятностью измерения. Она также может не зависеть от показателя времени t, и, в этом случае, записываться, как p(z |x). Вероятность измерения определяет вероятностное соотношение, согласно которому измерения

z генерируются из окружающей среды с состоянием x. Верно считать измерения зашумленными проекциями состояния.

Вместе вероятность перехода состояний и вероятность измерения описывают динамическую стохастическую систему, состоящую из робота и его окружения. На Рис. 2.2 показаны развитие состояний и измерений, определенных этими вероятностями. Состояние в момент времени t стохастически зависимо от состояния в момент времени t − 1 и управления ut. Измерение zt стохастически зависит от состояния в момент времени t. Такая временная генеративная модель также известна как скрытая марковская модель

**(hidden Markov model -**HMM) **или динамическая байесовская сеть** **(**dynam**ic Bayes network -**DBN).

2.3.4 Распределения оценок

ОЦЕНКА

Другой ключевой концепцией вероятностной робототехники является оценка. Оценка отражает внутреннюю осведомленность робота о состоянии окружающей среды. Мы уже говорили, что состояние невозможно измерить напрямую. Например, положение робота может быть (xt = 14.12, 12.7, 45◦) в некоторой глобальной системе координат, но своего положения он, обычно, узнать не способен, поскольку положение невозможно измерить напрямую (даже с помощью GPS!). Вместо этого, робот должен сделать заключение о своем положении на основании данных. Таким образом, необходимо отделять настоящее состояние от внутренней оценки этого состояния. Синонимами оценки в литературе являются

термины

ИНФОРМАЦИОННОЕ СОСТОЯНИЕ

«знание о состоянии» (state of knowledge) и «информационное состояние» (information state) (не нужно путать с информационным вектором и информационной матрицей, которые будут обсуждаться ниже).

В вероятностной робототехнике оценки рассматриваются через распределения условной вероятности. Распределение оценки назначает вероятность (или значение плотности) для каждой возможной гипотезы относительно реального состояния. Распределения оценок являются апостериорными вероятностями переменных состояния, вычисленными на основании доступных данных. Условимся обозначать оценку переменной состояния xt как bel(xt), в качестве сокращения для апостериорной вероятности

(2.33) bel(xt) = p(xt | z1:t, u1:t)

Апостериорная вероятность - это вероятностное распределение по состоянию xt в момент времени t, вычисленная на основании всех прошлых измерений z1:t и всех прошлых действий управления u1:t.

Читатель может заметить, что мы приняли, что оценка производится после получения измерения zt. Время от времени более полезным будет вычислить апостериорную вероятность сразу после выполнения управляющего действия ut, но до включения в расчет zt.

Такая апостериорная оценка будет иметь следующий вид:

(2.34) bel(xt) = p(xt | z1:t−1, u1:t)

ПРОГНОЗ

Это вероятностное распределение в контексте вероятностной фильтрации часто обозначается как «прогноз». Этот термин отражает факт того, что bel(xt) «прогнозирует» состояние в момент времени t, основываясь на апостериорных предыдущих состояниях и до принятия в расчет измерения в момент времени t. Вычисление bel(xt) из bel(xt) называется коррекцией или обновления измерения

2.4 Байесовские фильтры

2.4.1 Алгоритм байесовского фильтра

ФИЛЬТР БАЙЕСА

Самый общий алгоритм для вычисления оценок – это алгоритм байесовского фильтра. Этот алгоритм вычисляет распределение оценок bel на основе данных измерения и управления. Рассмотренный в начале базовый алгоритм, будет, проиллюстрирован числовым примером. После этого мы выведем его математическим путем на основании сделанных допущений.

В Таблице 2.1 в псевдо-алгоритмической форме приведен основной байесовский фильтр. Байесовский фильтр рекурсивный, поэтому оценка bel(xt) в момент времени t вычислена

1: Algorithm Bayes\_filter(bel(xt−1), ut, zt):

2: for all xt do

3:bel(xt) = p(xt | ut, xt−1) bel(xt−1) dxt−1

4: bel(xt) = η p(zt | xt) bel(xt)

5: endfor

6: return bel(xt)

Таблица 2.1 Общий алгоритм байесовской фильтрации.

на основании ценки bel(xt−1) для момента времени t − 1. На вход поступает оценка bel в момент времени t − 1, наряду с последним значением управления ut и самым последним измерением zt. Выходном алгоритма является значение оценки bel(xt) в момент времени t. В Таблице 2.1 приведена только одна итерация

ОБНОВЛЕНИЕ ДЛЯ БАЙЕСОВСКОГО ФИЛЬТРА

алгоритма байесовского фильтра: правило обновления. Обновление применяется рекурсивно, для того, чтобы вычислить оценку bel(xt) из предварительно вычисленной оценки bel(xt−1).

Алгоритм байесовского фильтра последовательно проходит два важных шага. В строке 3, обрабатывается управление ut. Это происходит через оценку состояния по xt на основе предыдущей оценки по состоянию xt−1 и управляющему воздействию ut. В частности, оценка

bel(xt) того, что робот перейдет в состояние xt, получается интегрированием (суммированием) произведения двух распределений: априорного, назначенного для xt−1, и вероятности того, что управляющее воздействие ut вызовет переход от xt−1 к xt. Читатель может увидеть схожесть этого такта обновления с Равенством (2.12). Как было сказано ранее, этот такт обновления называется обновлением управления или прогнозом.

Второй такт работы байесовского фильтра называется обновлением измерения. В строке

4, алгоритм байесовского фильтра умножает оценку bel(xt) на вероятность того, что будет обнаружено наблюдение zt. Он выполняет это действие для каждого гипотетического апостериорного состояния xt. Как станет очевидным далее при выводе основных равенств алгоритма фильтрации, результирующее произведение, вообще-то, вероятностью не является. Оно может не интегрироваться до 1. В силу этого, результат необходимо нормализовать с помощью нормализующего члена η. Это дает итоговую оценку bel(xt),

значение которой и возвращается в строке 6 алгоритма.

Для рекурсивного вычисления апостериорной оценки, алгоритму требуется начальная оценка bel(x0) в момент времени t = 0 в качестве граничного условия. Если известно точное значение x0, bel(x0) следует инициализировать сосредоточенное распределение масс, при котором вся вероятность сконцентрирована на верном значении x0, и

(рисунок 2.3)

Рис. 2.3 Мобильный робот определяет состояние двери.

равна нулю во всех остальных точках. Если же значение x0 совершенно неизвестно, bel(x0) можно инициализировать равномерным распределением в окрестности x0 (или применимым распределением Дирехле). Частичное знание о значении x0 возможно выразить с помощью неравномерного распределения, хотя эти два случая полной осведомленности и полного незнания являются наиболее часто встречающимися на практике.

В приведенной форме алгоритм байесовского фильтра может быть применен только для очень простых оценочных задач. В частности, необходимо или гарантировать закрытую форму интегрирования в строке 3 и умножения в строке 4, или ограничиться конечными пространствами состояний, чтобы интеграл в строке 3 свелся к (конечной) сумме.

2.4.2 Пример

Наша иллюстрация алгоритма байесовской фильтрации основана на сценарии на Рис. 2.3, на котором показан робот, который оценивает состояние двери с помощью камеры. Чтобы упростить проблему, давайте предположим, что дверь может быть в одном из двух возможных состояний, открыта или закрыта, и только робот может изменить ее состояние. Допустим также, что робот изначально не знает о текущем состоянии двери. Вместо этого, назначается равная априорная вероятность для двух возможных состояний двери:

bel(X0 = открыта) = 0.5

bel(X0 = закрыта) = 0.5

Также допустим, что датчики робота подвержены зашумлению. Шум характеризуется следующими условными вероятностями:

p(Zt = sense\_open | Xt = is\_open) = 0.6

p(Zt = sense\_closed | Xt = is\_open) = 0.4

и

p(Zt = sense\_open | Xt = is\_closed) = 0.2

p(Zt = sense\_closed | Xt = is\_closed) = 0.8

Эти вероятности говорят нам, что датчики робота довольно надежно распознают закрытую дверь, и в этом случае вероятность ошибки составляет 0,2. Однако, когда дверь открыта, имеется вероятность 0,4 ошибочного измерения.

И, наконец, давайте допустим, что робот с помощью манипулятора толкает дверь, чтобы открыть ее. Если дверь уже была открыта, она остается открытой. Если же она была закрыта, робот имеет шанс 0,8, что после воздействия она откроется:

p(Xt = is\_open | Ut = push, Xt\_1 = is\_open) = 1

p(Xt = is\_closed | Ut = push, Xt\_1 = is\_open) = 0

p(Xt = is\_open | Ut = push, Xt\_1 = is\_closed) = 0.8

p(Xt = is\_closed | Ut = push, Xt\_1 = is\_closed) = 0.2

Робот также может принять решение не использовать манипулятор, в этом случае состояние окружающего мира не меняется. Этот случай описывается следующими вероятностями:

p(Xt = is\_open | Ut = do\_nothing, Xt\_1 = is\_open) = 1

p(Xt = is\_closed | Ut = do\_nothing, Xt\_1 = is\_open) = 0

p(Xt = is\_open | Ut = do\_nothing, Xt\_1 = is\_closed) = 0

p(Xt = is\_closed | Ut = do\_nothing, Xt\_1 = is\_closed) = 1

Допустим, в момент времени t = 1, робот не выполняет никаких управляющих действий, но обнаруживает открытую дверь. В результате апостериорная оценка вычисляется байесовским фильтром, используя в качестве входных значений априорную оценку bel(X0), управляющее воздействие u1 = do\_nothing, и измерение sense\_open. Поскольку пространство состояний конечно, интеграл в строке 3 превращается в конечную сумму:

bel(x1)

= p(x1 | u1, x0) bel(x0) dx0

= [1]

x0

p(x1 | u1, x0) bel(x0)

= p(x1 | U1 = do\_nothing, X0 = is\_open) bel(X0 = is\_open)

+ p(x1 | U1 = do\_nothing, X0 = is\_closed) bel(X0 = is\_closed)

Теперь можно заменить два возможных значения переменной состояния X1. Для гипотезы X1 = is\_open, получаем

bel(X1 = is\_open)

= p(X1 = is\_open | U1 = do\_nothing, X0 = is\_open)

bel(X0 = is\_open)

+ p(X1 = is\_open | U1 = do\_nothing, X0 = is\_closed)

bel(X0 = is\_closed)

= 1 · 0.5 + 0 · 0.5 = 0.5

Аналогично, для X1 = is\_closed получается

bel(X1 = is\_closed)

= p(X1 = is\_closed | U1 = do\_nothing, X0 = is\_open)

bel(X0 = is\_open)

+ p(X1 = is\_closed | U1 = do\_nothing, X0 = is\_closed)

bel(X0 = is\_closed)

= 0 · 0.5 + 1 · 0.5 = 0.5

Факт равенства оценки bel(x1) априорной оценке bel(x0) не должен удивлять, поскольку действие do\_nothing не влияет на состояние окружающего мира, а сам окружающим мир в нашем примере не изменяется со временем.

Однако, когда в расчет принимается измерение, оценка изменяется. В строке 4 алгоритма байесовского фильтра

bel(x1) = η p(Z1 = sense\_open | x1) bel(x1)

Для двух возможных случаеыв X1 = is\_open и X1 = is\_closed, получаем

bel(X1 = is\_open)

= η p(Z1 = sense\_open | X1 = is\_open) bel(X1 = is\_open)

= η 0.6 · 0.5 = η 0.3

и

bel(X1 = is\_closed)

= η p(Z1 = sense\_open | X1 = is\_closed) bel(X1 = is\_closed)

= η 0.2 · 0.5 = η 0.1

Нормализующий член η теперь легко вычисляем:

η = (0.3 + 0.1)−1 = 2.5

Отсюда, получаем:

bel(X1 = is\_open) = 0.75

bel(X1 = is\_closed) = 0.25

Эти вычисления очень просто повторяются на следующем такте времени. Как читатель может убедиться, для u2 = push и z2 = sense\_open, получаем

bel(X2 = is\_open) = 1 · 0.75 + 0.8 · 0.25 = 0.95

bel(X2 = is\_closed) = 0 · 0.75 + 0.2 · 0.25 = 0.05

и

bel(X2 = is\_open) = η 0.6 · 0.95 ≈ 0.983

bel(X2 = is\_closed) = η 0.2 · 0.05 ≈ 0.017

К этому моменту оценка робота указывает на то, что, с вероятностью 0,983, дверь открыта.

На первый взгляд, эта вероятность может показаться достаточно большой, чтобы просто принять эту гипотезу в качестве состояния окружающего мира и действовать соответствующим образом. Однако, такой подход стать чрезмерно дорогим. Если ошибка перепутать закрытую дверь с открытой имеет цену (например, робот врежется в дверь), принятие во внимание обеих гипотез будет очень важным, какой бы невероятной одна из них не была. Просто представьте, каково будет лететь на самолете, автопилот которого может избежать катастрофы с вероятностью 0,983!

2.4.3 Математический вывод для байесовского фильтра

Продемонстрируем правильность алгоритма байесовской фильтрации с помощью индукции. Чтобы это сделать, необходимо показать, что он верно рассчитывает апостериорное распределение p(xt | z1:t, u1:t) из соответствующего апостериорного же, но взятого на один такт ранее p(xt−1 | z1:t−1, u1:t−1). Правильность доказывается с помощью индукции при условии верной инициализации априорной оценки bel(x0) в момент времени t = 0.

Наш вывод требует, чтобы состояние xt было полным, как определено в Главе 2.3.1, а управляющие воздействия – выбирались случайным образом. Первый этап нашего вывода включает применение формулы Байеса (2.16) к целевой апостериорной вероятности:

p(xt | z1:t, u1:t) = p(zt | xt, z1:t−1, u1:t) p(xt | z1:t−1, u1:t)

(2.35) p(zt | z1:t−1, u1:t)

= η p(zt | xt, z1:t−1, u1:t) p(xt | z1:t−1, u1:t)

Теперь используем допущение о том, что состояние полное. В Главе 2.3.1,

Мы определили xt как полное, если никакие переменные до xt не могут повлиять на стохастическую эволюцию будущих состояний. В частности, если мы (гипотетически),

знаем состояние xt, и заинтересованы в предсказании измерения zt, никакие измерения или действия управления не предоставят дополнительной информации. В математических терминах это можно выразить следующей условной независимостью:

(2.36) p(zt | xt, z1:t−1, u1:t) = p(zt | xt)

Это утверждение – еще один пример условной независимости. Оно позволяет упростить (2.35) следующим образом:

(2.37) p(xt | z1:t, u1:t) = η p(zt | xt) p(xt | z1:t−1, u1:t)

и получить

(2.38) bel(xt) = η p(zt | xt) bel(xt)

Это равенство используется в строке 4 алгоритма байесовского фильтра в Таблице 2.1.

Далее, расширим значение bel(xt), используя (2.12):

(2.39) bel(xt) = p(xt | z1:t−1, u1:t)

= p(xt | xt−1, z1:t−1, u1:t) p(xt−1 | z1:t−1, u1:t) dxt−1

И снова мы используем допущение о том, что состояние – полное. Это предполагает, что, если мы знаем xt−1, прошлые измерения и действия управления не дают дополнительной информации относительно состояния xt. Это дает

(2.40) p(xt | xt−1, z1:t−1, u1:t) = p(xt | xt−1, ut)

Здесь сохраняется переменная ut, поскольку она не относилась к состоянию xt−1.

Фактически, читатель может быстро убедиться, что p(xt | xt−1, ut) =p(xt | xt−1).

Наконец, заметим, что переменную управляющего воздействия в случае случайно выбранных управляющих воздействий ut можно опустить из набора условных переменных в p(xt−1 | z1:t−1, u1:t).

Это дает рекурсивно обновляемое равенство

(2.41) bel(xt) = p(xt | xt−1, ut) p(xt−1 | z1:t−1, u1:t−1) dxt−1

Как читатель может легко убедиться, это выражение используется в строке 3 алгоритма байесовской фильтрации, приведенного в Таблице 2.1.

Подводя итог, байесовский алгоритм фильтра вычисляет апостериорную вероятность по состоянию xt, при условии наличия данных измерений и управления вплоть до момента времени t. Вывод подразумевает, что окружающий мир задан согласно марковской модели, а значит, состояние полное. Любая практическая реализация этого алгоритма требует трех вероятностных распределений: первоначальной оценки p(x0), вероятности измерения p(zt | xt), и вероятности изменения состояния p(xt | ut, xt−1).

**Мы еще не определили эти плотности для реальных робототехнических систем. Однако, скоро это сделаем: Глава 5 полностью посвящена p(xt | ut, xt−1), а Глава 6 - p(zt | xt). Нам также понадобится выражение для оценки bel(xt), которая будет обсуждаться в Главах 3 и 4.**

2.4.4 Марковское свойство

МАРКОВСКОЕ СВОЙСТВО

Настало время сказать о марковском свойстве или свойстве полного состояния, поскольку оно играет столь фундаментальную роль в материале, представленном в книге.

Марковское свойство постулирует независимость прошлых и будущих данных, если известно текущее состояние xt. Чтобы увидеть, насколько важно это свойство, вернемся к нашему примеру локализации мобильного робота. В задаче локализации мобильного робота xt – это расположение робота, а байесовские фильтры используются для оценки расположения относительно фиксированной карты. Следующие факторы могут оказывать систематическое воздействие на показания датчика и, таким образом, вызывать нарушения марковского свойства:

• Не смоделированная динамика окружающей среды не включена в xt (например, движущиеся люди и эффекты их действия на измерения датчиков в нашем примере локализации),

• неточности вероятностных моделей p(zt | xt) and p(xt | ut, xt−1) (например, ошибки карты для локализации робота,

• ошибки аппроксимации при использовании приблизительных представлений функций оценки (например, сетки или гауссовы функции, которые будут обсуждаться ниже), и

• программные переменные в управляющем программном обеспечении робота, которые влияют на несколько управляющих действий (например, переменная «расположение цели» обычно влияет на всю последовательность управляющих команд).

В принципе, многие из этих переменных могут быть включены в представления состояния.

Однако, представления неполного состояния часто предпочтительны более полным из-за уменьшения вычислительной сложности алгоритма байесовского фильтра. На практике байесовские фильтры показали удивительную устойчивость к таким нарушениям.

Обычно рекомендуется соблюдать внимательность при определении состояния xt, поскольку не смоделированные переменные состояния могут вызвать непредсказуемые, почти случайные эффекты.

Представление и вычисление

В вероятностной робототехнике байесовские фильтры применяются несколькими разными способами. Как мы увидим в двух следующих главах, существует достаточно много разнообразных методов и алгоритмов на основе байесовского фильтра. Каждый из этих методов основан на различных допущениях относительно вероятностей измерений и перехода состояний, а также первоначальной оценки. Эти допущения затем порождают различные типы апостериорных распределений, а алгоритмы для их вычисления имеют различные вычислительные характеристики. В общем и целом, точные методы для вычислений оценок существуют только для очень узких задач. В общем случае, оценки предстоит аппроксимировать. Способ этой аппроксимации во многом, определяет сложность алгоритма. Нахождение подходящего способа аппроксимации обычно весьма сложная проблема, в которой не существует единого универсального ответа для всех проблем робототехники. При выборе аппроксимации приходится выбирать между рядом показателей:

1. Вычислительная эффективность. Некоторые способы аппроксимации, такие как линейные гауссовские методы, которые будут обсуждаться ниже, дают возможность вычисления оценки за время, кратное размерности пространства состояний. Другие могут потребовать времени, экспоненциально зависящего от размерности. Методы многочастичной фильтрации имеют переменную характеристику затрат времени, позволяя пожертвовать точностью ради вычислительной эффективности.

2. Точность аппроксимации. Некоторые виды приближения могут аппроксимировать широкий набор распределений с большей точностью. Например, линейное гауссовское приближение ограничено одномодальными распределениями (точно??-прим перев), а частотные представления могут аппроксимировать мультимодальные распределения, хотя и с ограниченной точностью. Многочастичные представления способны аппроксимировать широкий диапазон распределений, но количество частиц, необходимых для получения требуемой точности, может быть велико.

3. Простота использования. Трудность применения вероятностных алгоритмов зависит от целого ряда факторов, таких, как форма записи вероятности измерения p(zt | xt) и вероятности перехода состояния p(xt |ut, xt−1). Многочастичные представления часто имеют удивительно простые для сложных нелинейных систем реализации, что является одной из причин их сегодняшней популярности.

В следующих двух главах будут представлены конкретные алгоритмы реализации, которые довольно сильно различаются по описанным выше критериям.

2.6 Выводы

В этом разделе была представлена общая идея байесовских фильтров в робототехнике в целях оценки состояния окружающей среды и робота.

• Взаимодействие робота с окружающей средой моделируется в виде связанной динамической системы, в которой робот управляет окружающей средой путем выбора управляющих действий и воспринимать окружающую среду с помощью датчиков.

• В вероятностной робототехнике динамика робота и окружающей среды описывается в виде двух вероятностных закономерностей: распределения перехода между состояниями и распределение измерения. Распределение перехода между состояниями показывает, каким образом состояние меняется со временем, возможно, в результате управляющих действий. Распределение измерений показывает, каким образом измерения управляются состояниями. Оба правила имеют вероятностную природу и принимают в расчет собственную неопределенность в оценке состояния и восприятия.

• оценка робота – это апостериорное распределение по состоянию в окружающей среде (включая состояние робота) на основании всех прошлых измерений и управляющих действий.

Байесовский фильтр представляет собой основной алгоритм для вычисления оценки в робототехнике. Байесовский фильтр рекурсивен: оценка в момент времени t вычисляется на основе оценок в момент времени t – 1.

• Байесовский фильтр основан на марковском свойстве, согласно которому состояние является полной суммой прошлого. Это допущение предполагает, что оценки достаточно для отображения прошлой истории робота. В робототехнике марковское свойство обычно только приближение. Далее мы определим условия, при которых оно нарушается.

• Поскольку байесовский фильтр не является практическим алгоритмом и не может быть реализован на цифровом компьютере, вероятностные алгоритмы используют управляемые приближения. Такие приближения могут быть оценены согласно различным критериям, таким как точность, эффективность и простота реализации. В следующих двух главах обсуждаются два популярных семейства рекурсивных методов оценки состояния на основе байесовского фильтра.

2.7 Библиографические сведения

Базовый материал по статистике, размещенный в этой главе, освещается в большинстве базовых учебников по статистике и теории вероятности. Некоторые ранние классические тексты ДеГрута ( DeGroot, 1975), Сабраманиана (Subrahmaniam, 1979) и Торпа (Thorp, 1966) дают очень доступный вводной курс в материал. Более серьезный материал можно найти в работах Феллера, Каселла и Бергера, Таннера (Feller 1968; Casella and Berger 1990; Tanner 1996), а также Девро и Дуда с соавторами (Devroye et al. 1996; Duda et al. 2000). Парадигма взаимодействия робота с окружающей средой общепринята в робототехнике. Она обсуждается с точки зрения искусственного интеллекта Расселом и Норвигом (Russell and Norvig, 2002).

2.8 Упражнения

1. Робот использует датчик расстояния, который способен определять расстояния от 0 м до 3 м. Для простоты, допустим, что реальные значения расстояния равномерно распределены в этом интервале. К сожалению, датчик может быть неисправен. При неисправности датчика он постоянно выдает значения менее 1 м, вне зависимости от реального расстояния до объекта в конусе измерения робота. Известно, что априорная вероятность неисправности датчика p = 0.01.

Допустим, робот запрашивает датчик N и каждый раз значение измерения менее 1 м. Какова апостериорная вероятность неисправности датчика для N = 1, 2, . . . , 10. Сформулируйте соответствующую вероятностную модель.

2. Допустим, мы живем в месте, где погода может быть только солнечной, облачной или дождливой. Функция изменения погоды представляет собой цепь Маркова со следующей таблицей переходов:

Завтра будет. . .

солнечно облачно дождливо

солнечно .8 .2 0

сегодня. . . солнечно .4 .4 .2

дождливо .2 .6 .2

(a) Допустим, День 1 - солнечный. Какова вероятность следующей последовательности дней: День 2 = облачный, День 3 = облачный, День 4 = дождливый?

(b) Написать симулятор, который случайным образом генерирует последовательности «погоды» из функции перехода состояний.

(c) Используйте симулятор для определения стационарного распределения данной марковской цепи. Стационарное распределение измеряет вероятность того, что случайный день окажется солнечным, облачным или дождливым.

(d) Можете ли Вы определить закрытую форму решения для вычисления стационарного распределения на основании матрицы перехода состояний, приведенной выше.

(e) Какова энтропия стационарного распределения?

(f) Используя формулу Байеса, вычислить таблицу вероятности для вчерашней погоды по известной погоде на сегодня. (Разрешено представлять вероятности в числовом виде, а также использовать результаты предыдущих вопросов этого упражнения.)

(g) Допустим, мы добавили к модели времена года. Функция перехода состояний выше применима только для Лета, а для Зимы, Весны и Осени используются другие. Нарушит ли это марковское свойство этого процесса? Объясните ответ.

3. Допустим, мы не можем напрямую измерять погоду, и вынуждены полагаться на датчик.

Проблема состоит в том, что датчик зашумлен. Его показания управляются следующей моделью измерений:

Показания датчика. . .

солнечно облачно дождливо

солнечно .6 .4 0

текущая погода. . . облачно .3 .7 0

дождливо 0 0 1

(a) Допустим, День 1 солнечный (это известно точно), а в последующие четыре дня датчик показал облачно, облачно, дождливо, солнечно. Какова вероятность, что День 5 действительно солнечный, как показывает датчик?

(b) И снова, допустим, День 1. В течение дней 2-4 датчик показывает солнечно, солнечно, дождливо. Для каждого из дней со второго по четвертый, какая погода будет наиболее вероятна? Ответить на вопрос двумя способами: основываясь только на данных, доступных на нужный день, и используя перспективу, когда данные будущих дней также доступны.

(c) В той же самой ситуации (День 1 солнечный, измерения для

Дней 2, 3, и 4 солнечный, солнечный, дождливый). Какая последовательность погоды для дней со второго по четвертый наиболее вероятна? Какова вероятность для наиболее вероятной последовательности?

4. В этом упражнении применим теорему Байеса к гауссовским распределениям. Допустим, имеется мобильный робот, который существует на длинной прямой дороге. Местоположение x будет просто положение на дороге. Допустим, первоначально, мы считаем, что находимся на отметке xinit = 1, 000 м, но знаем, что оценка неточная. Основываясь на неопределенности, смоделируем первоначальную оценку гауссовым распределением со среднеквадратичным отклонением σ2init = 900 м2.

Чтобы уточнить свое расположение, запросим данные с GPS приемника. Запрос GPS

показывает, что местоположение zGPS = 1,100 м. Известно, что GPS приемник имеет среднеквадратичную ошибку σ2init = 100м2.

(a) Написать функции плотности вероятности для априорной вероятности p(x) и измерений p(z | x).

(b) Используя теорему Байеса, определить апостериорную вероятность p(x | z). Можно ли доказать, что функция гауссова?

(c) Насколько правдоподобно было априорное измерение xGPS = 1,100м и какова вероятность ошибки GPS приемника?

Подсказка: Это упражнение на использование квадратичными выражениями.

5. Вывести Равенства (2.18) и (2.19) из (2.17) и законов теории вероятности, описанных в тексте.

6. Доказать Равенство (2.25). Какие выводы можно сделать из этого выражения?