

Основания алгебраического подхода к синтезу корректных алгоритмов

Лектор — Рудаков К.В.
Наборщик — Старожилец В.М.

Оглавление

1	Лекция 1	2
	Введение	2
	Поиск решения задачи	3
2	Лекция 2	4
	Алгебра, реляционная система и алгебраическая система	4
	Первичные свойства функции	4
	О декартовом произведении множеств	5
3	Лекция 3	7
	Свободное произведение	7
	Свободная сумма	7
4	Лекция 4	9
5	Лекция 5	10
	Введение в категории	10
	Дальнейшая формализация общей задачи	11
6	Лекция 6	13
	Переход к категориям	13

Лекция 1

Введение

Данные лекции рассматривают общую задачу машинного обучения без привязки к конкретным методам и основы алгебраического подхода к синтезу корректных алгоритмов для её решения. В некотором роде они являются взглядом сверху на задачи машинного обучения и методы их решения.

В первую очередь следует сформулировать задачу машинного обучения в общем виде. По сути это задача построения алгоритма, который реализует отображение из множества начальных информации в множество конечных информации. Сразу отметим, что в курсе рассматриваются только такие отображения, для которых существует реализующий их алгоритм.

Определение. Символом \mathcal{I}_i (читается «И инициал») будем обозначать множество начальных информации, например, симптомы болезни.

Определение. Символом \mathcal{I}_f (читается «И финал») будем обозначать множество конечных информации, например, диагноз.

Таким образом, на формальном языке нам требуется найти такой алгоритм A , что он осуществляет отображение из множества начальных информации \mathcal{I}_i в множество конечных информации \mathcal{I}_f :

$$A : \mathcal{I}_i \rightarrow \mathcal{I}_f.$$

Пока что задача стоит так, что нам нужно найти некоторое произвольное отображение из одного множества в другое, реализуемое некоторым алгоритмом. При этом свойства этого отображения и алгоритма неважны. В такой постановке у нас нет каких-либо ограничений на искомый алгоритм: даже датчик случайных чисел является решением этой задачи. Поэтому вводятся дополнительные ограничения на допустимые алгоритмы. Итак,

Определение. Обозначим $\mathcal{M}^* = \{A \mid A : \mathcal{I}_i \rightarrow \mathcal{I}_f\}$ множество всех алгоритмов, реализующих отображение из \mathcal{I}_i в \mathcal{I}_f .

Определение. Обозначим I_{str} структурную информацию, содержащую условия и требования, накладываемые на A .

Определение. Обозначим $\mathcal{M}(I_{str}) \subset \mathcal{M}^*$ некоторое подмножество \mathcal{M}^* , удовлетворяющее I_{str} .

Теперь у нас есть дополнительная информация I_{str} , позволяющий накладывать дополнительные ограничения на нашу задачу. Введём определения допустимого отображения и корректного алгоритма.

Определение. Любое отображение из множества $\mathcal{M}(I_{str})$ называется допустимым.

Определение. Задача Z заключается в построении алгоритма, реализующего допустимое отображение.

Определение. Любой алгоритм реализующий любое допустимое отображение называется корректным.

В такой формулировке необходимым и достаточным условием разрешимости задачи Z является выполнение выражения:

$$\mathfrak{M}(I_{str}) \neq \emptyset,$$

а условием единственности решения — выполнение равенства:

$$|\mathfrak{M}(I_{str})| = 1.$$

Заметим также, что в данной формулировке корректный алгоритм — это алгоритм, не допускающий ни одной ошибки, а множество $\mathfrak{M}(I_{str})$ — множество алгоритмов не допускающих ошибок. Однако можно поставить условия несколько мягче, и дать алгоритмам возможность ошибаться.

Поиск решения задачи

Пусть $\mathfrak{M}(\pi)$ — некоторое параметрическое семейство отображений. После того как мы выбрали некоторое семейство отображений $\mathfrak{M}(\pi)$, попытаемся попасть в $\mathfrak{M}(I_{str})$, взяв в $\mathfrak{M}(\pi)$ какое-нибудь отображение за начальное. Это возможно, если данные семейства пересекаются:

$$\mathfrak{M}(\pi) \cap \mathfrak{M}(I_{str}) \neq \emptyset.$$

Но, с одной стороны, чем *сложнее* наше семейство, тем выше вероятность, что оно пересекается с семейством $\mathfrak{M}(I_{str})$, однако, с другой стороны, достижение этого пересечения может быть *затратно*, если $\mathfrak{M}(\pi)$ сложное. Также всегда остаётся вероятность, что множество $\mathfrak{M}(\pi)$ с $\mathfrak{M}(I_{str})$ не пересекается. Для поиска компромиссного решения используют идею *расширения множества*.

Определение. Пусть f — некоторая операция над множеством \mathfrak{M}^* . Тогда $f(\mathfrak{M}(\pi))$ будем называть расширением множества $\mathfrak{M}(\pi)$.

Таким образом, мы хотим расширить некоторое «простое» множество до пересечения с $\mathfrak{M}(I_{str})$. Однако, не любая функция f нам подходит, так как «простое» множество может расширяться до слишком «сложного» множества. Важно, что f мы выбираем сами, поэтому можем выбрать его так, чтобы искать нужный алгоритм было не слишком сложно.

Лекция 2

Алгебра, реляционная система и алгебраическая система

Данная лекция посвящена вопросам терминологии, используемой в данном курсе. Поэтому тут будет очень много определений (ещё больше, чем в предыдущей). Для начала определим понятия алгебры, реляционной системы и алгебраической системы.

Определение. *Сигнатура* — набор характеристик, однозначно идентифицирующий объект.

Определение. *Отношение* — математическая структура, которая формально определяет свойства различных объектов и их взаимосвязи. В нашем курсе будет обозначаться буквой R .

Определение. *Алгеброй* называется структура

$$\left(\begin{array}{cccccc} A & Op_1 & Op_2 & \dots & Op_k \\ & n_1 & n_2 & \dots & n_k \end{array} \right)$$

где A — множество, Op_i — операции на этом множестве, n_i — сигнатуры.

Определение. *Реляционной системой* называется структура

$$\left(\begin{array}{cccccc} A & R_1 & R_2 & \dots & R_k \\ & n_1 & n_2 & \dots & n_k \end{array} \right)$$

где A — множество, R_i — отношения на этом множестве, n_i — сигнатуры.

Определение. *Алгебраической системой* называется структура

$$\left(\begin{array}{cccccccccc} A & R_1 & R_2 & \dots & R_k & Op_1 & Op_2 & \dots & Op_l \\ & n_{1,1} & n_{1,2} & \dots & n_{1,k} & n_{2,1} & n_{2,2} & \dots & n_{2,l} \end{array} \right)$$

где A — множество, R_i — отношения на этом множестве, Op_i — операции на этом множестве, n_i — сигнатуры.

Первичные свойства функции

Теперь поговорим о функциях. Первичные свойства функций — это инъективность, суръективность и биективность. Все остальные свойства требуют задать некоторую структуру на тех множествах, на которых они действуют (например, метрику). Пусть функция f действует из A в B . То есть:

$$\begin{cases} f : A \rightarrow B \\ f(A) \subseteq B \end{cases}$$

Определим также понятие отношения эквивалентности на множестве A :

Определение. Отношением эквивалентности π_f на множестве A называется бинарное отношение, которое обладает свойствами транзитивности, симметричности и рефлексивности.

Данное определение приводит нас к определению фактор-множества A_{π_f} и ядерной эквивалентности отображения f .

Определение. Фактор-множество A_{π_f} — это множество всех классов эквивалентности заданного множества A , по заданному отношению π_f .

Определение. Ядерная эквивалентность отображения f :

$$(a_1 \equiv a_2) \equiv (f(a_1) = f(a_2)),$$

где эквивалентность понимается в смысле π_f . Следует понимать, что мы выбираем π_f так, чтобы это свойство было выполнено. Это выполнено не для любого отношения эквивалентности.

Саня, тут надо как нибудь переписать... непонятно что откуда идет. Я так понимаю что мы π_f выбираем по f , но чёрт его знает.

Таким образом, мы можем, например, показать, что любое отображение из A в B раскладывается в суперпозицию суръекции, инъекции и биекции. Данный факт легко понять с помощью рисунка 2.1:

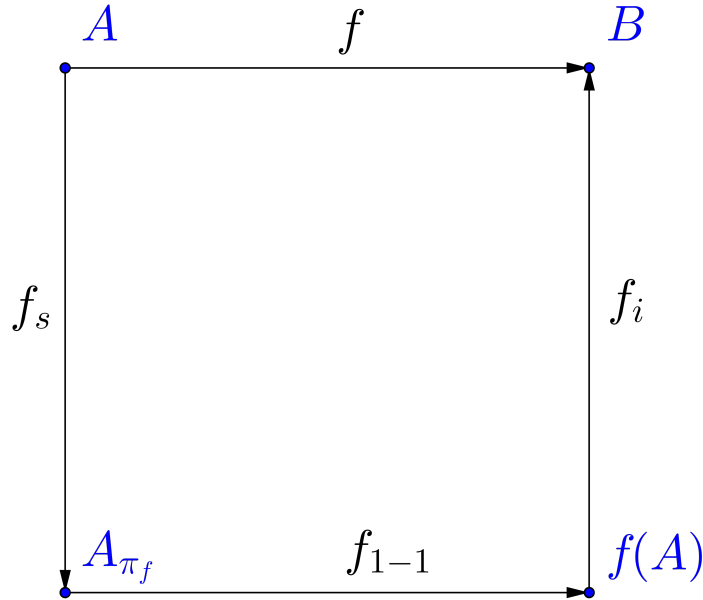


Рис. 2.1: Иллюстрация представления отображения f как суперпозиции суръекции, инъекции и биекции. f_s - суръективное отображение, f_{1-1} - биекция, f_i - инъекция

На данном рисунке изображены четыре множества: A , B , A_{π_f} — фактормножество, и $f(A)$. Множество A суръективно отображается в свое фактормножество A_{π_f} , из-за ядерной эквивалентности отображения f , A_{π_f} биективно отображается в $f(A)$. В свою очередь $f(A)$ инъективно вкладывается в B как его подмножество.

О декартовом произведении множеств

Поговорим о декартовом произведении множеств и том, как его можно представить через другие операции с множествами. Итак, пусть есть множество индексов $\mathfrak{A} = \{\alpha\}$ и соответствующий этому множеству индексов набор множеств $\{A_\alpha \mid \alpha \in \mathfrak{A}\}$. Чему тогда равно

произведение $\prod_{\alpha \in \mathfrak{A}} A_\alpha$? По крайней мере оно не нулевое так как любое декартово произведение произвольного семейства непустых множеств в непустом количестве непусто (об этом свидетельствует теорема выбора). Оказывается, что

$$\prod_{\alpha \in \mathfrak{A}} A_\alpha = \{f \mid f : \mathfrak{A} \rightarrow \bigcup_{\alpha \in \mathfrak{A}} A_\alpha, \forall \alpha \in \mathfrak{A} : f(\alpha) \in A_\alpha\}$$

Например, пусть

$$A_\alpha = A_{(x,y)} = \{(x', y') \mid \rho((x, y), (x', y')) \leq 1\}$$

Тогда,

$$\prod_{\alpha \in \mathfrak{A}} A_\alpha = \{f \mid f : R^2 \rightarrow R^2, \forall (x, y) : \rho((x, y), f((x, y))) \leq 1\}$$

Лекция 3

Свободное произведение

Саня, Я вообще не уверен, что тут все правильно. Проверь меня, если сам понял хоть что-то в этой теме:). Надо ещё как то обосновать представление прямой суммы как специфического объединения. Я не понял почему оно так представляется. Очень может быть что я пропустил тут в определениях какое нибудь "для любого альфа мне самому не совсем понятно

Прежде чем вводить свободное произведение, вспомним о понятии гомоморфизма, играющего важную роль в свободном произведении.

Определение. Гомоморфизм - это отображение алгебраической системы A , сохраняющее основные операции и основные соотношения.

Определение (Свободное произведение). Пусть имеется $\{A_\alpha \mid \alpha \in \mathfrak{A}\}$ - некоторое множество алгебраических систем. Тогда, алгебраическая система B , порождённая системами A_α , так, что гомоморфизм $\varphi_\alpha : C \rightarrow A_\alpha$, где C - произвольная алгебраическая система, продолжается до гомоморфизма $f : C \rightarrow B$ называется свободным произведением и обозначается $\prod_{\alpha \in \mathfrak{A}}^* A_\alpha$.

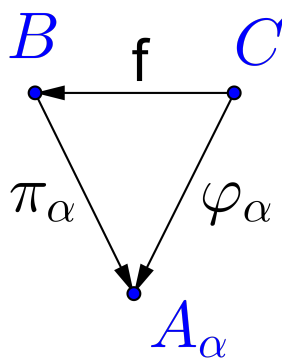


Рис. 3.1: Иллюстрация к понятию свободного произведения

На рисунке 3.1 видно, как работает это определение. Мы берем произвольное C и его морфизм в произвольный A_α . Тогда, имея морфизм $\pi_\alpha : B \rightarrow A_\alpha$ мы автоматически задаём единственный морфизм из C в B . Вот в чём суть. Важное замечание состоит в том, что, возможно, лектор ошибся, и перепутал свободное произведение с свободной суммой, которая будет определена в дальнейшем.

Свободная сумма

Данное понятие очень похоже по сути на понятие свободной суммы. Вы сами увидите, что конструкции практически идентичны.

Определение (Свободная сумма). Пусть имеется $\{A_\alpha \mid \alpha \in \mathfrak{A}\}$ - некоторое множество алгебраических систем. Тогда, алгебраическая система B , порождённая системами A_α , так, что гомоморфизм $g_\alpha : A_\alpha \rightarrow C$, где C - произвольная алгебраическая система, продолжается до гомоморфизма $g : B \rightarrow C$ называется свободной суммой и обозначается $\sum_{\alpha \in \mathfrak{A}}^* A_\alpha$.

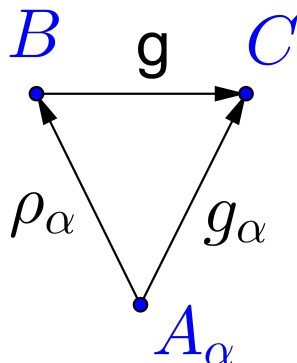


Рис. 3.2: Иллюстрация к понятию свободной суммы

На рисунке 3.2 видно, как работает это определение. Его принцип очень схож с принципом свободного произведения. И на самом деле - свободная сумма двойственна свободному произведению. На всякий случай напомним понятие двойственности из теории категорий:

Определение. Двойственность в теории категорий — соотношение между свойствами категории C и так называемыми двойственными свойствами двойственной категории C^* . Взяв утверждение, касающееся категории C и поменяв местами образ и прообраз каждого морфизма, так же как и порядок применения морфизмов, получим двойственное утверждение, касающееся категории C^* . Принцип двойственности состоит в том, что истинные утверждения после такой операции переходят в истинные, а ложные в ложные.

Чем же является прямая сумма в более привычных терминах теории групп? На самом деле если множества A_α не пересекаются то это просто объединение. Если же они пересекаются, то это объединение, которое ставит в соответствие $N = \sum_{\alpha \in \mathfrak{A}} |A_\alpha|$ элементное множество. Например, пусть $A_\alpha^* = \{(a, \alpha) \mid a \in A_\alpha\}$. Тогда $B = \bigcup A_\alpha^*$.

Лекция 4

Кажется, тут меня не было

Лекция 5

Введение в категории

Определение. *Класс* - это объект который не может быть элементом множества или другого класса, в остальном свойства класса и множества совпадают.

Определение. Будем обозначать $\mathfrak{C}_{ql}(\mathfrak{U})$ - пространство всех матриц над произвольным неоднородным множеством \mathfrak{U} . Заметим, что если \mathfrak{U} произвольные множества, то $\mathfrak{C}_{ql}(\mathfrak{U})$ это класс.

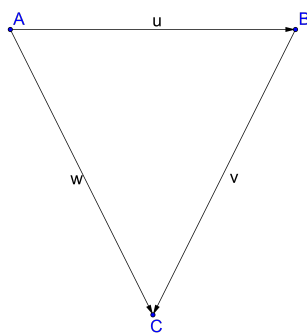
Определение (Категория). Это очень важное определение. Будем обозначать Ψ категорию. Тогда,

- $Ob\Psi$ - класс
- $\forall (A, B) \in Ob\Psi \rightarrow Hom_{\Psi}(A, B)$ - множество.
То есть любой паре элементов из $Ob\Psi$ ставится в соответствие множество отображений из A в B .

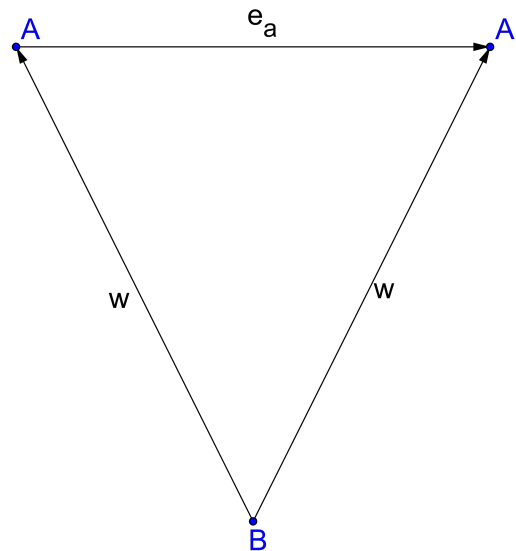
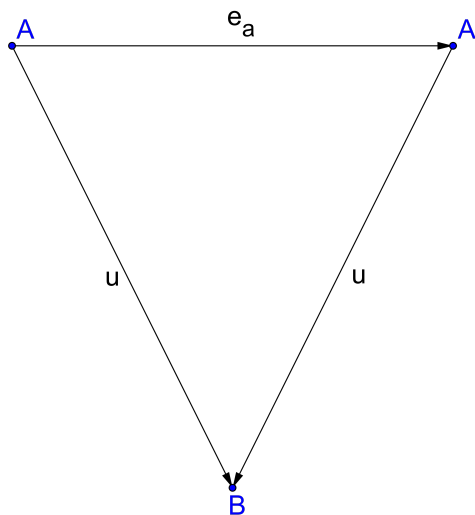
Таким образом категория это множество объектов и операций над ними.

Рассмотрим свойства категории Ψ .

1. Пусть есть 3 элемента $A, B, C \in \Psi$, причём Тогда $w = v \circ u$. Таким образом определе-



ны суперпозиции для морфизмов. Более того, $(u \circ v) \circ w = u \circ (v \circ w)$ для любых u, v, w для которых такая суперпозиция имеет смысл. А также существует единственный элемент e_A , такой, что



2. Ψ' подкатегория Ψ если:

- $Ob\Psi'$ - подкласс $Ob\Psi$
- $\forall A, B \in Ob\Psi' : Hom_{\Psi'} \subseteq Hom_{\Psi}(A, B)$

Причём, если во втором условии стоит равенство, то Ψ' - полная подкатегория категории Ψ .

Дальнейшая формализация общей задачи

Для начала введём несколько понятий, тут будут использоваться следующие сокращения: $ob = object$, $cl = class$, $i = initial$, $f = final$. Приступим:

- $\Upsilon = \{S\}$
- $\mathfrak{D}_{ob} : \Upsilon \rightarrow \mathfrak{J}_{ob}$
- K_1, \dots, K_l ; $K_j \subseteq \Upsilon$, $j = \overline{1, l}$. Мы рассматриваем задачу классификации и это - наши классы. Они являются элементами одного множества но сами могут быть различны.
- $\mathfrak{D}_{cl} : 2^\Upsilon \rightarrow \mathfrak{J}_{cl}$

Наша задача: $A : \mathfrak{J}_i \rightarrow \mathfrak{J}_f$. Её решение может быть формализовано в различных вариантах:

1. $\mathfrak{J}_i = \mathfrak{J}_{ob} \times \mathfrak{J}_{cl}$. В такой постановке мы игнорируем тот факт, что классов у нас 1. Неявно предполагаем, что классы однородны.
2. $\mathfrak{J}_i = \mathfrak{J}_{ob} \times \mathfrak{J}_{cl}^l$. Этот вариант лишен проблемы первого пункта и классы наши различны.
3. $\mathfrak{J}_i = \mathfrak{J}_{ob}^q \times \mathfrak{J}_{cl}^l$. Ещё более сложный вариант в котором участвует количество объектов - q . Таким образом S_1, \dots, S_q - объекты, $(S_1, \dots, S_q) \in \Upsilon^q$ и окончательно $\mathfrak{D}(S_1), \dots, \mathfrak{D}(S_q) \in \mathfrak{J}_{ob}^q$.

4. $\mathfrak{J}_i = \mathfrak{C}_{ql}(\mathfrak{J})$, $\mathfrak{J}_f = \mathfrak{C}_{ql}(\tilde{\mathfrak{J}})$, где \mathfrak{C}_{ql} - пространство $q \times l$ матриц над чем либо. Данный вариант наиболее общий и его мы и будем в дальнейшем рассматривать считая \mathfrak{J} , $\tilde{\mathfrak{J}}$ неоднородными множествами (иначе множество матриц тривиально). Может показаться, что пункт 4 и 3 дают одинаковую задачу, но это не так. **Саня, Я не помню контрпример. Если у тебя есть - впиши**

В новых терминах решение задачи Z формулируется таким образом: Пусть $\hat{I} \in \mathfrak{C}_{ql}(\mathfrak{J})$ и $\hat{\hat{I}} \in \mathfrak{C}_{ql}(\tilde{\mathfrak{J}})$. Мы ищем такие отображения A , что они являются допустимыми и $A(\hat{I}) = \hat{\hat{I}}$

Определение. Алгоритм называется корректным на прецедентах, если $A(\hat{I}) = \hat{\hat{I}}$

Саня, я тут не особо хорошо записывал. Проверь. Как то странно вышло.

Лекция 6

Переход к категориям