Matematyka podstawowa

Czas zacząć przyglądać się podstawowym zasadom matematycznym, które są przydatne w pracy z nauką o danych. Słowo matematyka budzi strach w wielu sercach, ale staram się, aby było to tak przyjemne, jak to tylko możliwe. W tym rozdziale omówimy podstawy następujących tematów:

• Podstawowe symbole/terminologia

• Logarytmy/wykładniki

• Teoria mnogości

• Rachunek

• Algebra macierzowa (liniowa)

Zajmiemy się także innymi dziedzinami matematyki. Co więcej, zobaczymy, jak zastosować każdy z nich do różnych aspektów nauki o danych, a także innych przedsięwzięć naukowych. Przypomnijmy, że w poprzedniej części zidentyfikowaliśmy matematykę jako jeden z trzech kluczowych elementów nauki o danych. Tu przedstawimy koncepcje, które staną się ważne w dalszej części - patrząc na modele probabilistyczne i statystyczne - a także przyjrzymy się pojęciom, które będą przydatne w tej części. Niezależnie od tego wszystkie koncepcje zawarte w tym rozdziale należy traktować jako podstawy w dążeniu do zostania naukowcem danych.

Matematyka jako dyscyplina

Matematyka jako nauka jest jedną z najstarszych znanych form logicznego myślenia ludzkości. Od starożytnej Mezopotamii i prawdopodobnie wcześniej (3000 p.n.e.) ludzie polegali na arytmetyce i trudniejszych formach matematyki, aby odpowiedzieć na największe życiowe pytania.Dzisiaj polegamy na matematyce w większości aspektów naszego codziennego życia; tak, wiem, że to brzmi banalnie, ale mam to na myśli. Niezależnie od tego, czy podlewasz rośliny, czy karmisz psa, Twój wewnętrzny silnik matematyczny nieustannie się kręci – oblicza, ile wody roślina miała dziennie w ciągu ostatniego tygodnia i przewiduje, kiedy następnym razem Twój pies będzie głodny, biorąc pod uwagę, że jedzą teraz. Niezależnie od tego, czy świadomie używasz zasad matematyki, czy nie, koncepcje żyją głęboko w mózgach każdego. Moim zadaniem jako nauczyciela matematyki jest uświadomienie ci tego.

Podstawowe symbole i terminologia

Najpierw przyjrzyjmy się najbardziej podstawowym symbolom używanym w procesie matematycznym, a także bardziej subtelnym zapisom używanym przez naukowców zajmujących się danymi.

Wektory i macierze

Wektor jest zdefiniowany jako obiekt o wielkości i kierunku. Ta definicja jest jednak nieco skomplikowana dla naszego użytku. Dla naszego celu wektor jest po prostu jednowymiarową tablicą reprezentującą szereg liczb. Innymi słowy, wektor to lista liczb. Jest to zwykle reprezentowane za pomocą strzałki lub pogrubionej czcionki, jak pokazano:

$$\vec{x} \quad or \quad x$$

Wektory są podzielone na komponenty, które są indywidualnymi członkami wektora. Używamy notacji indeksowych do oznaczenia elementu, do którego się odnosimy, jak pokazano na ilustracji:

$$\text{If} \quad \vec{x} = \begin{pmatrix} 3 \\ 6 \\ 8 \end{pmatrix} \quad \text{then } x_1 = 3$$

W matematyce na ogół odnosimy się do pierwszego elementu jako do indeksu 1, w przeciwieństwie do informatyki, gdzie zazwyczaj odnosimy się do pierwszego elementu jako indeksu 0. Ważne jest, aby pamiętać, jakiego systemu indeksów używasz. W Pythonie możemy reprezentować tablice na wiele sposobów. Moglibyśmy po prostu użyć listy Pythona do reprezentowania poprzedniej tablicy:

```
x = [3,6,8]
```

Jednak lepiej jest użyć typu tablicy numpy do reprezentowania tablic, jak pokazano, ponieważ daje nam to znacznie większą użyteczność podczas wykonywania operacji wektorowych:

```
import numpy as np.

x = np.array([3, 6, 8])
```

Niezależnie od reprezentacji Pythona wektory dają nam prosty sposób przechowywania wielu wymiarów pojedynczego punktu danych/obserwacji. Weź pod uwagę, że mierzymy średnią ocenę satysfakcji (0-100) pracowników z trzech działów firmy jako 57 dla HR, 89 dla inżynierii i 94 dla kierownictwa. Możemy przedstawić to jako wektor za pomocą następującego wzoru:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}$$

Ten wektor zawiera trzy różne bity informacji o naszych danych. To idealne zastosowanie wektora w nauce o danych. Możesz również myśleć o wektorze jako teoretycznej generalizacji obiektu serii Pandy. Tak więc naturalnie potrzebujemy czegoś, co będzie reprezentować Dataframe. Możemy rozszerzyć nasze pojęcie tablicy, aby wyjść poza jeden wymiar i reprezentować dane w wielu wymiarach. Macierz to dwuwymiarowa reprezentacja tablic liczb. Macierze (liczba mnoga) mają dwie główne cechy, o których musimy pamiętać. Wymiar macierzy, oznaczony jako n x m (n na m), mówi nam, że macierz ma n wierszy i m kolumn. Macierze są zazwyczaj oznaczane wielką, pogrubioną literą, taką jak X. Rozważmy następujący przykład:

$$\begin{pmatrix} 3 & 4 \\ 8 & 55 \\ 5 & 9 \end{pmatrix}$$

Jest to macierz 3 x 2 (3 na 2), ponieważ ma trzy wiersze i dwie kolumny.

Jeśli macierz ma taką samą liczbę wierszy i kolumn, nazywa się ją macierzą kwadratową. Matryca jest naszym uogólnieniem Pandas Dataframe. Jest to prawdopodobnie jeden z najważniejszych obiektów matematycznych w naszym zestawie narzędzi. Służy do przechowywania uporządkowanych informacji, w naszym przypadku danych. Wracając do naszego poprzedniego przykładu, załóżmy, że mamy trzy biura w różnych lokalizacjach, każde z tymi samymi trzema działami: HR, inżynierii i zarządzania.

Moglibyśmy stworzyć trzy różne wektory, z których każdy zawiera wyniki zadowolenia z innego biura, jak pokazano:

$$x = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}, y = \begin{pmatrix} 67 \\ 87 \\ 84 \end{pmatrix}, z = \begin{pmatrix} 65 \\ 98 \\ 60 \end{pmatrix}$$

Jest to jednak nie tylko kłopotliwe, ale także nieskalowalne. A jeśli masz 100 różnych biur? Wtedy musielibyśmy mieć 100 różnych jednowymiarowych tablic do przechowywania tych informacji. W tym miejscu matryca łagodzi ten problem. Stwórzmy macierz, w której każdy wiersz reprezentuje inny dział, a każda kolumna reprezentuje inne biuro, jak pokazano:

	Office 1	Office 2	Office 3
HR	57	67	65
Engineering	89	87	98
Management	94	84	60

To jest o wiele bardziej naturalne. Teraz zdejmijmy etykiety i zostajemy z matrycą!

$$X = \begin{pmatrix} 57 & 67 & 65 \\ 89 & 87 & 98 \\ 94 & 84 & 60 \end{pmatrix}$$

Szybkie ćwiczenia

1. Jeśli dodamy czwarte biuro, czy będziemy potrzebować nowego wiersza lub kolumny?

2. Jaki byłby wymiar macierzy po dodaniu czwartego biura?

3. Jeśli wyeliminujemy dział zarządzania z pierwotnej macierzy X, jaki byłby wymiar nowej macierzy?

4. Jaki jest ogólny wzór na poznanie liczby elementów w macierzy?

Odpowiedzi

1. Kolumna.

2. 3 x 4.

3. 2 x 3.

4. n x m (n to liczba wierszy, a m to liczba kolumn).

Symbole arytmetyczne

W tej sekcji omówimy niektóre symbole związane z podstawową arytmetyką, które pojawiają się w większości, jeśli nie we wszystkich, samouczkach i książkach dotyczących nauki o danych.

Sumowanie

Wielki symbol sigma Σ jest uniwersalnym symbolem dodawania. To, co znajduje się na prawo od symbolu sigma, jest zwykle czymś iterowalnym, co oznacza, że możemy przechodzić przez to jeden po drugim (na przykład wektor). Na przykład stwórzmy reprezentację wektora:

X = [1, 2, 3, 4, 5]

Aby obliczyć sumę zawartości, możemy skorzystać z następującego wzoru:

$\Sigma X_i = 15$

W Pythonie możemy użyć następującej formuły:

sum(x) # == 15

Na przykład wzór na obliczanie średniej szeregu liczb jest dość powszechny. Jeśli mamy wektor (x) o długości n, to średnią tego wektora można obliczyć w następujący sposób:

mean = $1/n \Sigma X_i$

Oznacza to, że dodamy każdy element x, oznaczony przez x_i , a następnie pomnożymy sumę przez 1/n, inaczej zwana dzieleniem przez n, długość wektora.

Proporcjonalny

Mały symbol alfa α reprezentuje wartości, które są do siebie proporcjonalne. Oznacza to, że wraz ze zmianą jednej wartości zmienia się druga. Kierunek, w którym przesuwają się wartości, zależy od tego, jak są one proporcjonalne. Wartości mogą się zmieniać bezpośrednio lub pośrednio. Jeśli wartości zmieniają się bezpośrednio, obie poruszają się w tym samym kierunku (w miarę wzrostu jednego, drugie też). Jeśli różnią się pośrednio, poruszają się w przeciwnych kierunkach (jeśli jeden idzie w dół, drugi idzie w górę). Rozważ następujące przykłady:

• Sprzedaż firmy zależy bezpośrednio od liczby klientów. Można to zapisać jako Sales α Customer.

• Ceny gazu zmieniają się (zwykle) pośrednio w zależności od dostępności ropy, co oznacza, że w miarę zmniejszania się dostępności ropy (jest jej coraz mniej), ceny gazu będą rosły. Można to oznaczyć jako Gas α Oil Availability.

Później zobaczymy bardzo ważną formułę zwaną formułą Bayesa, która zawiera symbol wariacji.

Iloczyn skalarny

Iloczyn skalarny jest operatorem takim jak dodawanie i mnożenie. Służy do łączenia dwóch wektorów, jak pokazano:

$$\begin{pmatrix} 3 \\ 7 \end{pmatrix} \cdot \begin{pmatrix} 9 \\ 5 \end{pmatrix} = 3*9 + 7*5 = 62$$

Więc co to oznacza? Załóżmy, że mamy wektor, który reprezentuje sentyment klienta do trzech gatunków filmów - komediowego, romantycznego i akcji.

Używając iloczynu skalarnego, zauważ, że odpowiedzią jest pojedyncza liczba, zwana skalarem.

Weź pod uwagę, że w skali od 1 do 5 klient uwielbia komedie, nienawidzi filmów romantycznych i nie ma nic przeciwko filmom akcji. Możemy to przedstawić w następujący sposób:

$$\begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}$$

Tutaj:

• 5 oznacza miłość do komedii,

• 1 to nienawiść do romantyków

• 3 to obojętność działania

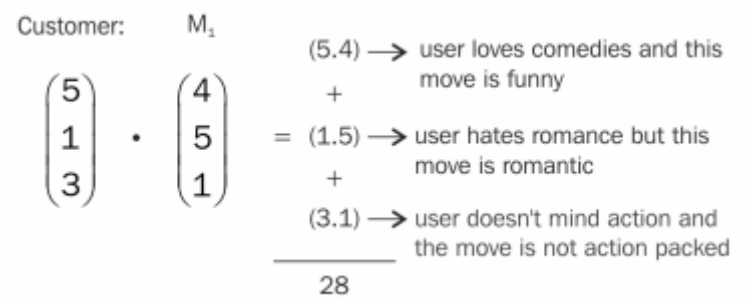
Załóżmy teraz, że mamy dwa nowe filmy, z których jeden to komedia romantyczna, a drugi to zabawny film akcji. Filmy miałyby swój własny wektor cech, jak pokazano:

$$m_1 = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix} \text{ and } m_2 = \begin{pmatrix} 5 \\ 1 \\ 5 \end{pmatrix}$$

Tutaj m₁ to nasza komedia romantyczna, a m₂ to nasz zabawny film akcji. W celu dokonania rekomendacji zastosujemy iloczyn skalarny pomiędzy preferencjami klienta dla każdego filmu. Wyższa wartość wygra, a zatem zostanie polecona użytkownikowi. Obliczmy wynik rekomendacji dla każdego filmu. Dla filmu 1 chcemy obliczyć:

$$\begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix} . \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix}$$

Możemy myśleć o tym problemie w następujący sposób:



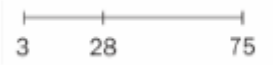
Uzyskujemy odpowiedź 28, ale co oznacza ta liczba? Na jaką skalę to jest? Cóż, najlepszy wynik, jaki można uzyskać, to gdy wszystkie wartości wynoszą 5, co daje następujący wynik:

$$\begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} . \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} = 5^2 + 5^2 + 5^2 = 75$$

Najniższy możliwy wynik jest wtedy, gdy wszystkie wartości wynoszą 1, jak pokazano:

$$\begin{pmatrix}1\\1\\1\end{pmatrix}.\begin{pmatrix}1\\1\\1\end{pmatrix}=1^2+1^2+1^2=3$$

Musimy więc pomyśleć o 28 w skali od 3-75. Aby to zrobić, wyobraź sobie linię liczbową od 3 do 75 i gdzie będzie na niej 28. Jest to zilustrowane w następujący sposób:



Nie tak daleko. Spróbujmy do filmu 2:

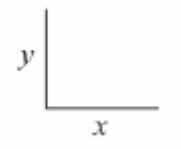
$$\begin{pmatrix}5\\1\\3\end{pmatrix}.\begin{pmatrix}5\\1\\5\end{pmatrix}=(5*5)+(1*1)+(3*5)=41$$

To więcej niż 28! Umieszczając tę liczbę na tej samej osi czasu, co poprzednio, możemy również wizualnie zaobserwować, że jest to znacznie lepszy wynik, jak pokazano:

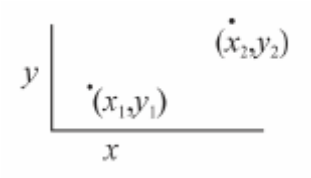
Tak więc, pomiędzy filmem 1 a filmem 2, zdecydowanie polecimy film 2 naszemu użytkownikowi. Tak właśnie działa większość silników przewidywania filmów. Tworzą profil klienta, który jest reprezentowany jako wektor. Następnie biorą wektorową reprezentację każdego filmu, który mają do zaoferowania, łączą je z profilem klienta (być może z iloczynem skalarnym) i stamtąd wydają rekomendacje. Oczywiście większość firm musi to robić na znacznie większą skalę, czyli tam, gdzie określona dziedzina matematyki, zwana algebrą liniową, może być bardzo przydatna.

Wykresy

Bez wątpienia w swoim życiu spotkałeś już dziesiątki, jeśli nie setki wykresów. Chciałbym porozmawiać głównie o konwencjach w odniesieniu do wykresów i notacji.



To jest podstawowy wykres kartezjański (współrzędne x i y). Notacja x i y jest bardzo standardowa, ale czasami nie wyjaśnia całego obrazu. Czasami nazywamy zmienną x jako zmienną niezależną, a y jako zmienną zależną. Dzieje się tak, ponieważ kiedy piszemy funkcje, mówimy o nich jako o tym, że y jest funkcją x, co oznacza, że wartość y zależy od wartości x. To właśnie próbuje pokazać wykres. Załóżmy, że na wykresie mamy dwa punkty, jak pokazano:



Odnosimy się do punktów jako (x_1,y_1) i (x_2,y_2) .

Nachylenie między tymi dwoma punktami definiuje się w następujący sposób:

$$slope = m = \frac{y_2 - y_1}{x_2 - x_1}$$

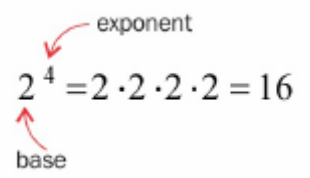
Zapewne widziałeś już tę formułę, ale warto o niej wspomnieć, gdyby nie jej znaczenie. Nachylenie określa szybkość zmian między dwoma punktami. Tempo zmian może być bardzo ważne w nauce o danych, szczególnie w obszarach związanych z równaniami różniczkowymi i rachunkiem różniczkowym. Tempo zmian to sposób na przedstawienie, jak zmienne poruszają się razem i w jakim stopniu. Weź pod uwagę, że modelujemy temperaturę Twojej kawy w zależności od czasu, w którym siedziała. Być może tempo zmian jest następujące:

$$-\frac{2\,degrees\,F}{1\,minute}$$

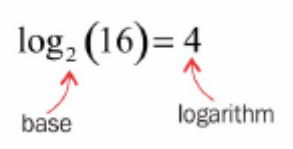
Ta szybkość zmian mówi nam, że z każdą minutą temperatura naszej kawy spada o dwa stopnie Fahrenheita. W dalszej części tej książki przyjrzymy się algorytmowi uczenia maszynowego, zwanemu regresją liniową. W regresji liniowej zajmujemy się szybkościami zmian między zmiennymi, ponieważ pozwalają one wykorzystać tę zależność do celów predykcyjnych. Pomyśl o płaszczyźnie kartezjańskiej jako o nieskończonej płaszczyźnie wektorów z dwoma elementami. Kiedy ludzie odnoszą się do wyższych wymiarów, takich jak 3D lub 4D, mają na myśli po prostu nieskończoną przestrzeń, w której znajdują się wektory z większą liczbą elementów. Przestrzeń 3D zawiera wektory o długości trzy, podczas gdy przestrzeń 7D zawiera wektory z siedmioma elementami.

Logarytmy/wykładniki

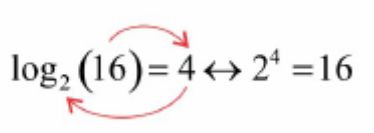
Wykładnik mówi Ci, ile razy musisz pomnożyć przez siebie liczbę, jak pokazano na ilustracji:



Logarytm to liczba odpowiadająca na pytanie: „jaki wykładnik przenosi mnie z podstawy do tej innej liczby?” Można to opisać w następujący sposób:



Jeśli te dwie koncepcje wydają się podobne, to masz rację! Wykładniki i logarytmy są ściśle powiązane. W rzeczywistości słowa wykładnik i logarytm oznaczają to samo! Logarytm jest wykładnikiem. Poprzednie dwa równania to w rzeczywistości dwie wersje tego samego. Podstawowym założeniem jest to, że 2 razy 2 razy 2 razy 2 daje 16. Poniżej znajduje się obraz, w jaki sposób możemy użyć obu wersji, aby powiedzieć to samo. Zwróć uwagę, jak używam strzałek, aby przejść z formuły logarytmicznej do formuły wykładniczej:



Rozważ następujące przykłady:

• $\log_3 81 = 4$ ponieważ $3^4 = 81$

• $\log_5 125 = 3$ ponieważ $5^3 = 125$

Zwróć uwagę na coś interesującego, jeśli przepiszemy pierwsze równanie tak:

$\log_3 81 = 4$

Następnie zastępujemy 81 równoważnym stwierdzeniem 34, w następujący sposób:

$\log_3 3^4 = 4$

Możemy zauważyć coś interesującego: trójki wydają się znosić. W rzeczywistości jest to bardzo ważne, gdy mamy do czynienia z liczbami trudniejszymi do pracy niż trójki i czwórki. W przypadku wzrostu najważniejsze są wykładniki i logarytmy. Najczęściej, jeśli jakaś wielkość rośnie (lub maleje), wykładnik/logarytm może pomóc w modelowaniu tego zachowania. Na przykład liczba e wynosi około 2,718 i ma wiele praktycznych zastosowań. Bardzo powszechnym zastosowaniem jest obliczanie wzrostu dla pieniędzy. Załóżmy, że masz 5000 USD zdeponowane w banku z ciągle naliczanymi odsetkami w wysokości 3%, wtedy możemy użyć następującego wzoru do modelowania wzrostu naszego depozytu:

$A = Pe^{rt}$

Gdzie:

• A oznacza ostateczną kwotę

• P oznacza inwestycję główną (5000)

• e oznacza stałą (2.718)

• r oznacza tempo wzrostu (0,03)

• t oznacza czas (w latach)

Jesteśmy ciekawi, kiedy nasza inwestycja się podwoi? Jak długo musiałbym mieć pieniądze w tej inwestycji, aby osiągnąć 100% wzrost? Zasadniczo:

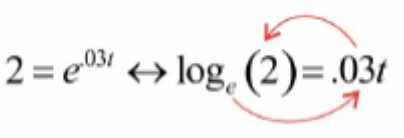
$10\,000 = 5000e^{0.3t}$

Czy formuła, którą chcemy rozwiązać:

$10\,000 = 5000e^{0.3t}$

$2 = e^{0.3t}$ (dzielone przez 5000 po obu stronach)

W tym momencie mamy zmienną w wykładniku, którą chcemy rozwiązać. Kiedy tak się stanie, możemy użyć notacji logarytmicznej, aby to rozgryźć!



To pozostawia nas z $\log_e(2) = .03t$

Logarytm liczby o podstawie e nazywamy logarytmem naturalnym. Przepisujemy logarytm w następujący sposób:

$\ln(2) = .03t$

Używając kalkulatora (lub Pythona), stwierdzamy, że ln(2) = 0,69.

$0.69 = 03.t$

$t = 2,31$

Oznacza to, że podwojenie naszych pieniędzy zajęłoby 2,31 roku.

Teoria mnogości

Teoria mnogości obejmuje operacje matematyczne na ustalonym poziomie. Czasami uważa się ją za podstawową, podstawową grupę twierdzeń, która rządzi resztą matematyki. Do naszych celów wykorzystujemy teorię mnogości, aby manipulować grupami elementów. Zestaw to zbiór odrębnych przedmiotów. Otóż to! Zestaw można traktować jako listę w Pythonie, ale bez powtarzających się obiektów. W rzeczywistości w Pythonie istnieje nawet zestaw obiektów:

```
s = set()

s = set([1, 2, 2, 3, 2, 1, 2, 2, 3, 2])

# usunie duplikaty z listy
```

s == {1, 2, 3}

Zauważ, że w Pythonie nawiasy klamrowe - {, } - mogą oznaczać zbiór lub słownik. Pamiętaj, że słownik w Pythonie to zestaw par klucz-wartość, na przykład:

dict = {"dog": "human's best friend", "cat": "destroyer of world"}

dict["dog"]# == "human's best friend"

len(dict["cat"]) # == 18

but if we try to create a pair with the same key as an existing key

dict["dog"] = "Arf"

dict

{"dog": "Arf", "cat": "destroyer of world"}

It will override the previous value

dictionaries cannot have two values for one key.

Dzielą tę notację, ponieważ mają tę samą cechę, że zestawy nie mogą zawierać zduplikowanych elementów, podobnie jak słowniki nie mogą mieć zduplikowanych kluczy. Wielkość zestawu to liczba elementów w zestawie i jest reprezentowana w następujący sposób:

|A| = wielkość A

s # == {1,2,3}

len(s) == 3 # magnitude of s

Pojęcie zbioru pustego istnieje i jest oznaczane przez znak . Mówi się, że ten pusty zestaw ma wartość 0. Jeśli chcemy zaznaczyć, że element znajduje się w zestawie, używamy notacji epsilon, jak pokazano:

$2 \in \{1,2,3\}$

Oznacza to, że element 2 istnieje w zbiorze 1, 2 i 3. Jeśli jeden zbiór jest całkowicie wewnątrz innego zbioru, mówimy, że jest podzbiorem jego większego odpowiednika.

A ={1,5,6} , B= {1,5,6,7,8}

$A \sqsubseteq B$

Tak więc A jest podzbiorem B, a B nazywa się nadzbiorem A. Jeśli A jest podzbiorem B, ale A nie jest równe B (co oznacza, że w B jest co najmniej jeden element, który nie znajduje się w A), to A nazywa się właściwym podzbiorem B.

Rozważ następujące przykłady:

• Zbiór liczb parzystych jest podzbiorem wszystkich liczb całkowitych

• Każdy zbiór jest sam w sobie podzbiorem, ale nie właściwym podzbiorem

• Zestaw wszystkich tweetów to nadzbiór angielskich tweetów

W nauce o danych używamy zestawów (i list) do reprezentowania listy obiektów i często do uogólniania zachowań konsumentów. Często sprowadza się klienta do zestawu cech. Weź pod uwagę, że jesteśmy firmą marketingową, która próbuje przewidzieć, gdzie dana osoba chce kupować ubrania. Dostajemy zestaw marek odzieżowych, które użytkownik odwiedził wcześniej, a naszym celem jest przewidzenie nowego sklepu, który również by mu się spodobał. Załóżmy, że określony użytkownik robił wcześniej zakupy w następujących sklepach:

user1 = {"Target","Banana Republic","Old Navy"}

note that we use {} notation to create a set

compare that to using [] to make a list

Tak więc użytkownik 1 wcześniej robił zakupy w Target, Banana Republic i Old Navy. Spójrzmy również na innego użytkownika, zwanego user2, jak pokazano:

user2 = {"Banana Republic","Gap","Kohl&priem;s"}

Załóżmy, że zastanawiamy się, jak podobni są ci użytkownicy. Mając ograniczone informacje, którymi dysponujemy, jednym ze sposobów na zdefiniowanie podobieństwa jest sprawdzenie, w ilu sklepach oboje robią zakupy. Nazywa się to skrzyżowaniem. Przecięcie dwóch zbiorów to zbiór, którego elementy występują w obu zbiorach. Jest oznaczony symbolem ∩, jak pokazano:

user1 ∩ user2 = { Banana Republic}

|user1 ∩ user2| = 1

Skrzyżowanie tych dwóch użytkowników to tylko jeden sklep. Więc od razu to nie wydaje się wspaniałe. Jednak każdy użytkownik ma w swoim zestawie tylko trzy elementy, więc posiadanie 1/3 nie wydaje się takie złe. Załóżmy, że jesteśmy ciekawi, ile sklepów jest reprezentowanych między nimi dwoma; nazywa się to związkiem. Połączenie dwóch zestawów to zestaw, którego elementy pojawiają się w każdym zestawie. Jest oznaczony symbolem ∪, jak pokazano:

user1 ∪; user2 = {Banana Republic, Target, Old Navy, Gap, Kohl′s}

|user1 ∪ user2| = 5

Patrząc na podobieństwo między user1 i user2 powinniśmy użyć kombinacji sumy i przecięcia ich zbiorów. user1 i user2 mają jeden wspólny element z pięciu różnych elementów między sobą. Możemy więc zdefiniować podobieństwo między dwoma użytkownikami w następujący sposób:

|user1 ∪user2| / |user1 ∪; user2| = 1/5 =.2

W rzeczywistości ma to swoją nazwę w teorii mnogości. Nazywa się to miarą jaccarda. Ogólnie rzecz biorąc, dla zestawów A i B miara jaccarda (podobieństwo jaccarda) między tymi dwoma zestawami jest zdefiniowana w następujący sposób:

$JS(A,B) = |A \cap B| / |A \cup B|$

Można go również zdefiniować jako wielkość przecięcia dwóch zbiorów podzieloną przez wielkość sumy tych dwóch zbiorów. Daje nam to sposób na ilościowe określenie podobieństw między elementami reprezentowanymi za pomocą zestawów. Intuicyjnie, miara jaccarda jest liczbą z zakresu od 0 do 1, tak że gdy liczba jest bliższa 0, ludzie są bardziej do siebie podobni, a gdy miara jest bliższa 1, ludzie są uważani za podobnych do siebie. Jeśli myślimy o definicji, to faktycznie ma to sens. Jeszcze raz spójrz na miarę:

JS = Liczba wspólnych sklepów / Unikalna liczba sklepów, które lubią łączyć

Tutaj licznik reprezentuje liczbę sklepów, które użytkownicy mają wspólnego (w tym sensie, że lubią tam robić zakupy), podczas gdy mianownik reprezentuje unikalną liczbę sklepów, które lubią razem. Możemy to przedstawić w Pythonie za pomocą prostego kodu, jak pokazano:

```
user1 = {"Target","Banana Republic","Old Navy"}

user2 = {"Banana Republic","Gap","Kohl's"}

def jaccard(user1, user2):

stores_in_common = len(user1 & user2)

stores_all_together = len(user1 | user2)

return stores / float(stores_all_together)

# I cast stores_all_together as a float to return a decimal answer instead of python's default integer division

# so

jaccard(user1, user2) == # 0.2 or 1/5
```

Teoria mnogości staje się bardzo rozpowszechniona, gdy wkraczamy w świat prawdopodobieństwa, a także gdy mamy do czynienia z danymi wielowymiarowymi. Użyjemy zbiorów do reprezentowania zachodzących wydarzeń w świecie rzeczywistym, a prawdopodobieństwo stanie się teorią mnogości z dodanym słownictwem.

Algebra liniowa

Pamiętasz silnik rekomendacji filmów, który widzieliśmy wcześniej? Co by było, gdybyśmy mieli 10 000 filmów do polecenia i musielibyśmy wybrać tylko 10 do przekazania użytkownikowi? Musielibyśmy umieścić iloczyn skalarny między profilem użytkownika a każdym z 10 000 filmów. Algebra liniowa dostarcza narzędzi, które znacznie usprawniają te obliczenia. Jest to dziedzina matematyki zajmująca się matematyką macierzy i wektorów. Ma na celu rozbicie tych obiektów i zrekonstruowanie ich w celu zapewnienia praktycznych zastosowań. Przyjrzyjmy się kilku zasadom algebry liniowej, zanim przejdziemy dalej.

Mnożenie macierzy

Podobnie jak liczby, możemy wielokrotne macierze razem. Mnożenie macierzy jest w istocie masową metodą pobierania kilku iloczynów skalarnych na raz. Spróbujmy na przykład pomnożyć następujące macierze:

$$\begin{pmatrix} 1 & 5 \\ 5 & 8 \\ 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix}$$

Kilka rzeczy:

• W przeciwieństwie do liczb, mnożenie nie jest przemienne, co oznacza, że kolejność mnożenia macierzy ma ogromne znaczenie.

• Aby pomnożyć macierze, ich wymiary muszą się zgadzać. Oznacza to, że pierwsza macierz musi mieć taką samą liczbę kolumn, jak druga macierz ma wiersze.

Aby to zapamiętać, wypisz wymiary matryc. W tym przypadku mamy macierz 3x2 razy 2x2. Możesz mieć wiele macierzy razem, jeśli druga liczba w pierwszej parze wymiarów jest taka sama jak pierwsza liczba w drugiej parze wymiarów.

$$3 \times 2 \cdot 2 \times 2$$

Otrzymana macierz zawsze będzie miała wymiary równe liczbom zewnętrznym w parach wymiarów (te, których nie zakreśliłeś w drugim punkcie). W takim przypadku otrzymana macierz będzie miała wymiar 3 x 2.

Jak pomnożyć macierze

Aby pomnożyć macierze, istnieje całkiem prosta procedura. Zasadniczo wykonujemy kilka produktów skalarnych. Przypomnij sobie nasz wcześniejszy przykładowy problem, który wyglądał następująco:

$$\begin{pmatrix} 1 & 5 \\ 5 & 8 \\ 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix}$$

Wiemy, że nasza wynikowa macierz będzie miała wymiar 3 x 2. Więc wiemy, że będzie wyglądać mniej więcej tak:

$$\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{pmatrix}$$

Zauważ, że każdy element macierzy jest indeksowany przy użyciu podwójnego indeksu. Pierwsza liczba reprezentuje wiersz, a druga - kolumnę. Tak więc element jest elementem m_3 w trzecim rzędzie, drugiej kolumnie. Każdy element jest wynikiem iloczynu skalarnego między wierszami i kolumnami macierzy oryginalnych. Element m_{xy} jest wynikiem iloczynu skalarnego x-tego wiersza pierwszej macierzy i y-tej kolumny drugiej macierzy. Rozwiążmy kilka:

$$m_{11} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 13$$

$$m_{12} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 5 \end{pmatrix} = 29$$

Idąc dalej, ostatecznie otrzymamy wynikową macierz w następujący sposób:

$$\begin{pmatrix} 13 & 29 \\ 31 & 60 \\ 37 & 68 \end{pmatrix}$$

Tak trzymać! Wróćmy do przykładu rekomendacji filmu. Przypomnij sobie preferencje użytkownika dotyczące gatunku filmu, takie jak komedia, romans i akcja, które są zilustrowane w następujący sposób:

$$U = user\ prefs = \begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}$$

Załóżmy teraz, że mamy 10 000 filmów, wszystkie z oceną w tych trzech kategoriach. Aby dokonać rekomendacji, musimy wziąć iloczyn skalarny wektora preferencji dla każdego z 10 000 filmów. Aby to przedstawić, możemy użyć mnożenia macierzy. Zamiast wypisywać je wszystkie, wyraźmy to za pomocą notacji macierzowej. Mamy już U, zdefiniowane tutaj jako wektor preferencji użytkownika (można go również traktować jako macierz 3x1) i potrzebujemy również macierzy filmowej:

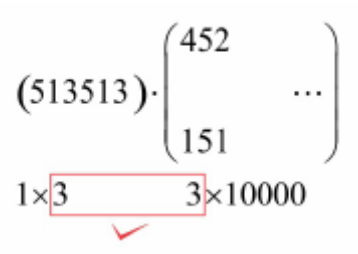
M = filmy = matryca 3 x 10 000 wymiarów

Więc teraz mamy dwie macierze, jedna to 3 x 1, a druga to 3 x 10 000. Nie możemy pomnożyć tych macierzy, ponieważ wymiary się nie sprawdzają. Będziemy musieli trochę zmienić U. Możemy wziąć

transpozycję macierzy (zamieniając wszystkie wiersze w kolumny, a kolumny w wiersze). To zmieni wymiary wokół:

U^T = transpozycja U = (513)

Więc teraz mamy dwie macierze, które można przez siebie pomnożyć. Aby zwizualizować, jak to wygląda:



Otrzymana macierz będzie macierzą 1 x 1000 (wektor) z 10 000 przewidywań dla każdego pojedynczego filmu. Wypróbujmy to w Pythonie!

```
# create user preferences

user_pref = np.array([5, 1, 3])

# create a random movie matrix of 10,000 movies

movies = np.random.randint(5,size=(3,1000))+1

# Note that the randint will make random integers from 0-4

# so I added a 1 at the end to increase the scale from 1-5
```

Używamy funkcji numpy array do tworzenia naszych macierzy. Będziemy mieli zarówno macierz user_pref, jak i macierz filmów do reprezentowania naszych danych. Aby sprawdzić nasze wymiary, możemy użyć zmiennej kształtu numpy, jak pokazano:

```
print user_pref.shape # (1, 3)

print movies.shape # (3, 1000)
```

To się sprawdza. Na koniec użyjmy metody mnożenia macierzy numpy (zwanej kropką), aby wykonać operację, jak zilustrowano:

```
# np.dot does both dot products and matrix multiplication

np.dot(user_pref, movies)
```

Wynikiem jest tablica liczb całkowitych, które reprezentują rekomendacje każdego filmu. Aby to szybko rozszerzyć, uruchommy kod, który przewiduje w ponad 10 000 filmów, jak pokazano:

```
import time
```

```
for num_movies in (10000, 100000, 1000000, 10000000, 100000000):

movies = np.random.randint(5,size=(3, movies))+1

now = time.time()

np.dot(user_pref, movies)

print (time.time() - now), "seconds to run", movies, "movies"
```

0.000160932540894 seconds to run 10000 movies

0.00121188163757 seconds to run 100000 movies

0.0105860233307 seconds to run 1000000 movies

0.096577167511 seconds to run 10000000 movies

4.16197991371 seconds to run 100000000 movies

Przejrzenie 100 000 000 filmów za pomocą mnożenia macierzy zajęło tylko nieco ponad 4 sekundy.

Podsumowanie

W tej części przyjrzeliśmy się kilku podstawowym zasadom matematycznym, które staną się bardzo ważne w miarę postępów. Pomiędzy logarytmami/wykładnikami, algebrą macierzową i proporcjonalnością matematyka wyraźnie odgrywa dużą rolę nie tylko w analizie danych, ale w wielu aspektach naszego życia. W kolejnych częściach zajmiemy się znacznie głębiej dwoma dużymi obszarami matematyki: prawdopodobieństwem i statystyką. Naszym celem będzie zdefiniowanie i zinterpretowanie najmniejszych i największych twierdzeń w tych dwóch gigantycznych dziedzinach matematyki. To w kilku następnych rozdziałach wszystko zacznie się układać. Do tej pory w tej książce przyjrzeliśmy się przykładom matematycznym, wskazówkom dotyczącym eksploracji danych oraz podstawowym wglądowi w typy danych. Czas zacząć łączyć wszystkie te koncepcje razem.