Kapitola 1

Naivní Bayesův klasifikátor

Současná umělá inteligence používá velice často teorii pravděpodobnosti k odhadu nejistoty rozhodnutí, které stroje provádí. Z teorie pravděpodobnosti vychází v minulém desetiletí velice populární grafické modely a využívají ji i v současnosti velmi rychle se rozvíjející neuronové sítě. V této kapitole si ukážeme jeden velice jednoduchý pravděpodobnostní model, který se používá ke klasifikaci, a k jehož vysvětlení stačí pouze středoškolská matematika.

V klasickém středoškolském pojetí se pravděpodobnost používá k popisu událostí, které u kterých opředu odhadnout, jak dopadnou – například protože takový výpočet byl nepředstavitelně složitý (například u hodu kostkou). Teorie pravděpodobnosti můžeme uplatnit i v případě počítání s *nejistotou* (EN: *uncertainty*), byť se v tomto případě jedná o kvalitativně jinou veličinu. Když říkáme, že je padesátiprocentní pravděpodobnost, že na minci padne orel, myslíme tím že když hodíme mnohokrát, padne orel přibližně v polovině případů. Když ale tvrdíme, že ledovec roztaje s padesátiprocentní pravděpodobností, nemyslíme tím, že kdybychom naklonovali tisíce zeměkoulí a nechali je se nezávisle vyvíjet za stejných podmínek, na polovině z nich by ledovec roztál a na druhé polovině nikoli. Výsledek byl vždy stejný. Pravděpodobnostní modely ve strojovém učení modelují právě tuto nejistotu – na základě vstupních rysů a dostupných trénovacích dat. Model si může být jistý a zároveň se mýlit.

Často se uvádí výzkum americké armády z počátku 90. let, jehož cílem bylo automatické rozlišování snímků spojeneckých a nepřátelských tanků[1]. Jak takové fotky tanků vypadaly si můžete prohlédnout na obrázku 1.1. K rozpoznávání tehdy použili neuronovou síť a dosáhli úspěšnosti přes 90 % na nezávislé testovací sadě, kterou nepoužili při trénování. Rozhodnutí modelu vykazovala vždy velkou míru jistoty, systém přesto v praxi naprosto selhával. Později se ukázalo, že fotky nepřátelských tanků byly vyfoceny v den, kdy byla zatažená obloha, fotky spojeneckých, když bylo jasno. Neuronová síť potom využívala tuhle náhodnou pravidelnost v trénovacích datech, a když byla zatažená obloha, s jistotou odpověděla, že se jedná o nepřátelský tank.

Model, který si představíme v této kapitole je podstatně jednodušší, než neuronová síť určená k rozpoznávání tanků. Dříve, než si vysvětlíme samotný model, zopakujeme některé základní pojmy z pravděpodobnosti a statistiky, které budeme potřebovat.

[1]Kompletní technickou zprávu mohou zájemci nalézt zde: http://www.cs.colostate.edu/~vision/publications/1994_fcarson_report.pdf

Obrázek 1.1: Ukázka trénovacích dat pro rozpoznávání tanků, zdroj: Technická zpráva Fort Carson

1.1 Podmíněná pravděpodobnost

V pravděpodobnosti a statistice říkáme, že jevy A a B jsou nezávislé, když pro pravděpodobnost, že nastanou oba jevy zároveň (tzv. sdruženou pravděpodobnost) platí vztah:

$$\mathsf{P}(A, B) = \mathsf{P}(A) \cdot \mathsf{P}(B). \tag{1.1}$$

To se dá snadno ilustrovat například na hodu kostkou. Pravděpodobnost, že padne šestka je $\frac{1}{6}$. Pravděpodobnost, že padne dvakrát za sebou je podle vzorce 1.1 rovna $\frac{1}{36}$. To ověříme snadno kombinatoricky: je celkem 36 kombinací, které mohou padnout při hodu dvěma kostkami, jediná z nich jsou dvě šestky, pravděpodobnost je tedy $\frac{1}{36}$.

Další pojem, který je třeba připomenout je *podmíněná pravděpodobnost* (`EN:` *conditional probabiity*). Pravděpodobnost, že nastane jev A (třeba že na dvou hracích kostkách padne dohromady více než 7) za podmínky, že nastal jev B (třeba že na první hrací kostce už padlo 5) můžeme vyjádřit následujícím vzorcem:

$$\mathsf{P}(A|B) = \frac{\mathsf{P}(A, B)}{\mathsf{P}(B)}, \tag{1.2}$$

kde $\mathsf{P}(A, B)$ je pravděpodobnost, že oba jevy nastávají zároveň.

Pro náš jednoduchý příklad s hracími kostkami je jasné, že aby byl součet větší než 7, musí na druhé kostce padnout cokoli, kromě jedničky nebo dvojky. Pokud se jedná o férovou kostku, vidíme pravděpodobnost, že součet bude alespoň 7 je $\frac{2}{6} = \frac{1}{3}$.

Pokud počítáme podle rovnice 1.2, dosadíme do čitatele pravděpodobnosti, že na první kostce padlo 5 a zároveň je součet větší než 7. To nastává tehdy, když na druhé padne alespoň 3. Jsou tedy 4 možnosti z celkově 36 kombinací, co může na kostkách padnout. Do jmenovatele dosadíme pravděpodobnost, že na první kostce padlo 5, to je $\frac{1}{6}$. Po dosazení získáme stejnou $\frac{1}{3}$ jako v případě přímočarého řešení.

Z této dvou definic můžeme odvodit Bayesovu větu. Definici zformulujeme pro podmíněnou pravděpodobnost $\mathsf{P}(B|A)$ a rovnici vynásobíme jmenovatelem na pravé straně:

$$\mathsf{P}(A, B) = \mathsf{P}(B|A) \cdot \mathsf{P}(A)$$

Tento vztah potom dosadíme rovnou do definice 1.2, čímž dostáváme znění Bayesovy věty:

$$\mathsf{P}(A|B) = \frac{\mathsf{P}(B|A) \cdot \mathsf{P}(A)}{\mathsf{P}(B)} \tag{1.3}$$

Zatím jsme o pravděpodobnosti jako o nástroji pro popis něčeho, co se chová nepředvidatelně a co lze mnohokrát opakovat. Nyní si ukážeme jednoduchý příklad, kdy můžeme použít Bayesovu věty pro odhad nejistoty.

Častým příkladem, na kterém se vysvětlují různé pravděpodobnostní modely, je tahání kuliček z různých nádob (přinejmenším už od 19. století). Nedává moc praktický smysl něco takového počítat. Úvaha, kterou na tomto příkladu provedeme, se nám ale bude hodit později. Mějme dvě krabičky a v každé deset barevných kuliček. V krabičce a je 6 modrých a 4 zelené kuličky, v krabičce b je jedna modrá 9 zelených. Vybereme jednu z krabiček náhodně a vytáhneme jednu kuličku. Dokud nevíme, jakou má kulička barvu, nemáme žádný důvod preferovat některou z krabiček. Jakmile zjistíme, že kulička má zelenou barvu, začne nám být jasné, že je pravděpodobnější, že pochází z krabičky b. Míru naší jistoty můžeme odhadnout pomocí Bayesovy věty. Zajímá nás pravděpodobnost, že jsme vybrali krabičku b, za podmínky, že kulička je zelená:

$$\mathsf{P}(b|z) = \frac{\mathsf{P}(z|b)\mathsf{P}(b)}{\mathsf{P}(z)}.$$

Pravděpodobnost, že z krabičky b vytáhneme zelenou kuličku ($\mathsf{P}(z|b)$), je $\frac{9}{10}$; pravděpodobnost, že zvolíme krabičku b je $\frac{1}{2}$. Zbývá nám odhadnout jmenovatele – tedy pravděpodobnost, že bude vytažena zelená kulička.

Zajímá nás, jaká je pravděpodobnost, že byla tažena zelená kulička, jedno z jaké krabičky. Protože kuličku nelze vytáhnout napůl z jedné a napůl z druhé krabičky, jedná se o logickou disjunkci a můžeme pravděpodobnost vyjádřit jako součet pravděpodobnosti událostí, že byla zelená kulička tažena z každé krabičky:

$$\mathsf{P}(z) = \mathsf{P}(z, a) + \mathsf{P}(z, b)$$

Sdružené pravděpodobnosti $\mathsf{P}(z, a)$ a $\mathsf{P}(z, b)$ stále neznáme, ale můžeme je opět rozepsat podle definice podmíněné pravděpodobnosti. Dostáváme

$$\mathsf{P}(z) = \mathsf{P}(z|a)\mathsf{P}(a) + \mathsf{P}(z|b)\mathsf{P}(b) = \frac{4}{10} \cdot \frac{1}{2} + \frac{9}{10} \cdot \frac{1}{2} = \frac{13}{10} \cdot \frac{1}{2}$$

Po dosazení dostaneme pravděpodobnost toho, že kulička pocházela z krabičky b rovnu $\frac{9}{13} \approx 70$ %. To je v souladu s úvahou selským rozumem, že když je kulička zelená, nejspíš pochází z krabičky b, když je jich tam více.

V toto příkladě získaná pravděpodobnost není odhad četnosti při opakování experimentu do nekonečna. Událost vytažení kuličky se jednou pro vždy odehrála. Jediné, co můžeme počítat, jak moc jsme si jisti, že byla vytažena z jedné nebo druhé krabičky.

1.2 Model

Když klasifikaci používáme pravděpodobností modely, odhadujeme pravděpodobnost (jistotu) $\mathsf{P}(y|x)$, tedy pravděpodobnost přiřazení značky y, je-li dán vektor rysů x. V základní verzi, kterou si ukážeme pracuje Naivní Bayes pouze s kategoriálními proměnnými.

Pravděpodobnost klasifikace si rozepíšeme pomocí Bayesovy věty:

$$\mathsf{P}(y|x) = \frac{\mathsf{P}(x|y)\mathsf{P}(y)}{\mathsf{P}x}. \tag{1.4}$$

rys	možné hodnoty
věk	10–19, 20–29, 30–39, 40–49, …, 90-99
menopauza	předmenstruační věk, věk do 40 let, věk nad 40 let
velikost primárního nádoru	0–4 mm, 5–9 mm, 10–14 mm, …, 55–59 mm
počet mízních uzlin s nádorovými buňkami	0-2, 3-5, 6-8, 9-11, 12-14, …, 36–39
ložisko zapouzdřeno	ano, ne
stupeň maligity	1, 2, 3
prs	levý, pravý
kvadrant umístění nádoru	levý horní, levý spodní, pravý horní, pravý dolní, střední
ozařování	ano, ne

Výraz ve jmenovateli nezávisí na značce y. Protože chceme klasifikovat, nemusí nás zajímat konkrétní odhad nejistoty, zajímá nás pouze, jaká značka má nejvyšší pravděpodobnost – nejvyšší hodnotu čitatele. Pokud by nás zajímal i jmenovatel, můžeme ho spočítat obdobně jako v našem příkladu s krabičkami.

Naše odhadnutá klasifikace $\hat{y}$ je taková značka y, pro kterou je hodnota čitatele nejvyšší. Formálně píšeme

$$\hat{y} = \operatorname*{argmax}_y \mathsf{P}(x|y)\mathsf{P}(y). \tag{1.5}$$

Zápis $\operatorname{argmax}_y f(y)$ zde znamená y, pro nějž je hodnota výrazu $f(y)$ nejvyšší.

Odhadování pravděpodobností v předchozí rovnici si vysvětlíme na příkladu další úlohy, která bývá uváděna v literatuře o strojovém učení. Úlohou je odhadování možnosti recidivy karcinomu prsu. K této úloze jsou k dispozici anonymizované informace o 277 pacientkách z roku 1988. Rysy, které budeme používat k rozhodnutí jsou uvedeny v tabulce **??**.

Sada rysů i počet pacientek jsou velmi omezené. Každého určitě napadne, že existují i další faktory, které hrají roli – například genetická zátěž, zda kouří, míra stresu, které je vystavena atd. Navíc data byla sbírána v 80. letech v Lublani. Lze tedy očekávat, že skupina pacientek byla etnicky homogenní a v socialistické se Jugoslávii bez velkých ekonomických rozdílů, se pacientky se příliš nelišily svým životním stylem. Model tyto informace nemá k dispozici a rozhoduje vždy stejně.

Pravděpodobnost $\mathsf{P}(y)$ v rovnici **??** je pravděpodobnosti že dojde k recidivě nádoru bez ohledu na to, jaké hodnoty jsme u pacientky naměřili. Tuto pravděpodobnost můžeme jednoduše odhadnout jako poměr pacientek, u kterých došlo k recidivě ku celkovému počtu pacientek. Stejně spočítáme nepodmíněnou pravděpodobnost, že k recidivě nedojde.

Situace je komplikovanější při odhadu pravděpodobnosti $\mathsf{P}(x|y)$ – tedy pravděpodobnosti naměřených hodnot pacientky za předpokladu (za podmínky), že dojde k recidivě nádoru. Triviální způsob, jak pravděpodobnost odhadnout, by bylo spočítat, jak často byly mezi pacientkami ty, které vykazovaly přesně tuto kombinaci rysů. Možných kombinací rysů je ale obvykle řádově více, než mám k dispozici trénovacích datech žádná pacientka s naprosto

stejnou kombinací rysů nevyskytuje. To je problém, který musí řešit všechny klasifikační algoritmy, které jsem si představili. Metoda nejbližších sousedů (kapitola **??**) řeší tento problém hledáním podobných instancí. Perceptronový algoritmus (kapitola **??**) problém jej obchází tak, že vůbec neuvažuje o společném výskytu rysů a jednotlivým rysům přiřazuje různé váhy.

Obdobný přístup volí i naivní bayesovský klasifikátor, který předpokládá takzvanou *podmíněnou nezávislost* (EN: *conditional independence*) rysů. Tedy, je-li už dána značka, předpokládáme, že pravděpodobnosti jednotlivých rysů jsou vzájemně nezávislé. Podle definice statistické nezávislosti (rovnice 1.1), rozepíšeme podmíněnou pravděpodobnost takto:

$$\mathsf{P}(x|y) = \mathsf{P}(x_1, \ldots, x_n|y) \approx \mathsf{P}(x_1|y) \cdot \mathsf{P}(x_n|y).$$

Předpoklad podmíněné pravděpodobnosti je pochopitelně značně zjednodušující. Rysy, které spolu systematicky souvisí, přináší modelu částečně tu samou informaci. Protože ale předpokládáme vzájemnou nezávislost rysů, je tato informace ve skutečnosti započítána víckrát a zkresluje rozhodnutí modelu. V tomto případě mohou být takovými rysy věk pacientky a informace o tom, zda prošla menopazou, které spolu souvisí.

Podmíněnou pravděpodobnost pro i-tý $\mathsf{P}(x_i|y)$ odhadneme už jednoduše pomocí četnosti výskytu rysu v trénovacích datech. Je-li tímto rysem například věk ženy, vezmeme zvlášť ty, u kterých došlo k recidivě nádoru ($y = +$) a ty, kde nikoli ($y = 0$) a v rámci těchto kategorií (za těchto podmínek) spočítáme pravděpodobnost pro jednotlivé věkové kategorie. Pro každou věkovou kategorii a odhadneme

$$\mathsf{P}(x_0 = a|y = 0) = \frac{c(x_0 = a \wedge y = 0)}{c(y = 0)}$$

a stejně pro $y = 1$, kde funkcí c se myslí počet trénovacích příkladů, splňující podmínku uvedenou jako argument funkce.

Učení modelu spočívá ve spočítání četností, které potřebujeme pro výpočet pravděpodobností. Kód může vypadat například takto.

```
class NaiveBayes(object):
    def __init__(self, training_data):
        self.feature_count = len(training_data[0][0])
        self.data_size = len(training_data)
        # Tabulka s pocty instanci v jednotlivych tridach
        self.labels_counts = {}
        # Tabulka s hodnotami rysu v jednotlivych tridach
        self.cond_feature_counts = {}

        for feature_vector, label in training_data:
            # Pokud jsme label, jeste nevideli, zalozime ji polozku v obou
            # dvou tabulkach
            if label not in labels_counts:
                # Znacku jsme jeste nevideli ani jednou
                self.labels_counts[label] = 0
                # Pripravime zvlastni tabulku pro kazdy rys
                self.cond_feature_counts[label] = [{}] * feature_count

            self.labels_counts[label] += 1
```

```
            for i in range(self.feature_count):
                feature_value = feature_vector[i]
                if feature_value not in self.cond_feature_counts[label][i]:
                    self.cond_feature_counts[label][i][feature_value] = 1
                else
                    self.cond_feature_counts[label][i][feature_value] += 1
```

Model implementujeme v Pythonu jako třídu. Při inicializaci objektu se spočítají parametry modelu, které později využívají při klasifikaci. Metoda, která provádí samotnou klasifikaci se dá naprogramovat takto.

```
    def classify(self, instance):
        # Pro zajisteni numericke stability si rozepiseme nasobeni
        # pravdepodobnosti jako nejprve nasobeni vsech citatelu a potom deleni vsemi
        # jmenovateli
        best_score = 0
        best_label = None

        # Projdu vsechny mozne labely a pro kazdy spocitam skore
        for label in self.labels_counts:
            label_count = self.labels_counts[label]
            count_product = np.prod(
                [ self.cond_feature_counts[label][i].get(feature_value, default=0.0)
                        for i, feature_value in enumerate(instance) ])
            score = count_product /
                labels_counts ** (self.feature_count - 1.0) / self.data_size
            if score > best_score:
                best_score = score
                best_label = label

        return best_label
```

JL: Výsledky klasifikátoru a porovnat s jinými: decision tree, knn Hamming

1.3 Cvičení

JL: Tabulka: heatmapa četnosti jednotlivých rysů

V trénovacích datech jsou kombinace rysy, které nejsou zastoupené ...vedou k odhadu nulové pravděpodobnosti v obou případech ... zkuste transformovat množinu rysů tak, aby tyto případy zmizely. Dokážete takto zvýšit správnost klasifikace?

V tabulce **??** uvádíme výsledek učení perceptronovým algoritmem. Jakým způsobem je potřeba transformovat rysy, aby bylo možné perceptron použít?

Shrnutí

- *Pravděpodobnostní model* –
- *Bayesova věta* –