FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KIV/PRO

Algoritmus strojového učení na základě analýzy konvexního obalu

28. 10. 2021

Lukáš Varga

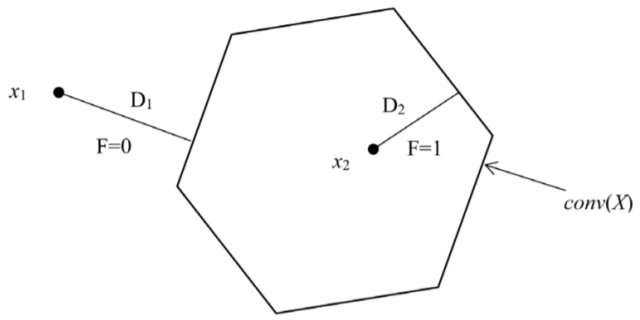
Obsah

1 Úvod

Semestrální práce je založena na článku *Machine learning algorithm based on convex hull analysis* [1] publikovaného ve vědeckém časopis *Procedia Computer Science*. Snaha je kladena na srozumitelnost, aby hlavní myšlenku článku pochopil běžný student 2. ročníku informatiky.

U strojového učení se často setkáváme s následujícím postupem. Analyzované objekty jsou popsány pomocí množin vektorů ve vícedimenzionálních příznakových prostorech. Jinými slovy jsou odchyceny určité rysy chování modelu a ty jsou poté zaznamenány na různých úrovních podle toho, jak spolu souvisejí. Tento postup využívá k řešení problému početní matematiku a statistiku.

Ačkoliv tyto modely mají mnoho výhod, někdy je jejich výsledkem pouze hrubý odhad ve srovnání s reálnými daty. Autoři článku [1] se proto zabývají ne tak častými metodami strojového učení, které využívají početní geometrie. Přesněji se zaměřují na metody vytvoření konvexního obalu. Tento postup byl použit pro redukci prostoru, zkoumání průniků tříd, klasifikaci a shlukování. V příznakových prostorech pak nejsou analyzovány jednotlivé body, ale místo toho se pracuje přímo s konvexními obaly, jejich průniky a také jejich vzdáleností od testovaných bodů jako vidíme na Obr. 1 (převzato z [1]).



Obr. 1: Vzdálenost bodů x1 a x2 od konvexního obalu conv(X)

2 Definice

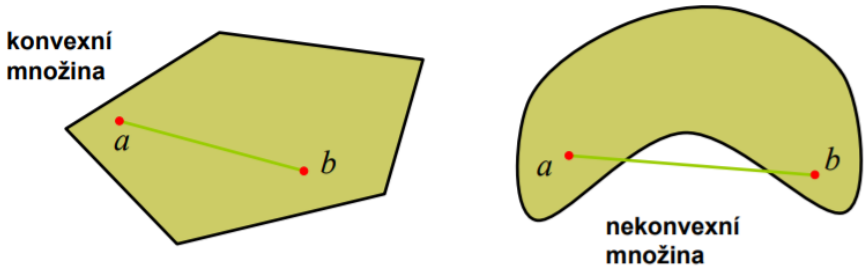
2.1 Klasifikace na základě nejbližšího konvexního obalu

Jak jsem již zmínil v úvodu, mnoho početních metod v geometrii je realizováno pomocí konvexního obalu, který úzce souvisí s pojmem konvexní množina. Nejprve si však musíme onen koncept popsat, abychom lépe pochopili algoritmy s ním spjaté. *„Množina se nazývá konvexní, jestliže úsečka spojující libovolné dva její body je částí této množiny“* [2], tj.

$$\forall a, b \in M, \forall t \in \langle 0,1 \rangle : ta + (1 - t)b \in M$$

Rovnice 1: Definice konvexní množiny

To se dá lépe uchopit vizuálně, jak se lze přesvědčit na Obr. 2 (převzato z [2]).



Obr. 2: Konvexní a nekonvexní množina

Pokud tedy máme množinu *X*, naším cílem je vytvořit nejmenší možný konvexní obal (množinu), který ale zároveň obsahuje všechny prvky množiny *X*. Pro tento účel máme hned několik algoritmů, které toto dovedou ve dvou i tří dimenzionálním prostorech. Mezi tyto algoritmy patří například Graham [3], Jarvis [4], Chan [5] a další. Existují také algoritmy pro vyšší dimenze jako například Quickhull [6], ale ty jsou zase limitované dimenzí příznakového prostoru a velikostí tréninkové množiny.

Nyní se dostanu ke konkrétnější aplikaci ve strojovém učení. Klasifikátor nejbližšího konvexního obalu nejprve zkoumá, jak daleko je testovaný bod od konvexních obalů jednotlivých tříd. Hledáme pak tedy nejbližší konvexní obal tomuto bodu. Vzdálenost testovaných bodů od obalu můžeme vidět na Obr. 1. Jak jistě víme, euklidovská vzdálenost se obecně udává v absolutní hodnotě a proto i při tomto výpočtu jsou vzdálenosti bodů x_1 a x_2 od obalu obě kladné. Pro nás je ovšem také užitečné vědět, zda-li je bod uvnitř nebo vně konvexního obalu. Proto zde použijeme popisek *F*, kdy *F=1* pro body uvnitř obalu a *F=0* pro body mimo obal.

Hlavním problémem všech algoritmů založených na tomto principu je samotná metoda zjišťování vzdálenosti testovaného bodu od konvexního obalu.

2.2 Metody měření vzdálenosti bodu od konvexního obalu

Pro lineárně separovatelné třídy je problém nejkratší vzdálenosti mezi konvexními obaly tříd řešen pomocí algoritmů SVM [7, 8], SK-algoritmus [9], MDM-algoritmus [10]. Tyto algoritmy lze také použít, pokud zjišťujeme vzdálenost bodů od konvexního obalu, pokud je bod vně obalu.

Bavíme-li se o průniku konvexních obalů a také pokud je testovaný bod uvnitř obalu, musíme použít koncept hloubky průniku [11,12]. Toto měření určuje stupeň, ve kterém se objekty protínají. Pro lepší uchopení myšlenky si představme následující modelovou situaci. Pokud budeme mít dva protínající se objekty - A (pohyblivý) a B (statický) - můžeme popsat hloubku průniku následovně, hloubka průniku se rovná nejmenší vzdálenosti vektoru takového, že objekt A posuneme tak, aby s objektem B měly prázdný průnik. Tato technika je opět použitelná i pro lineárně separovatelné třídy. Hledáme minimální vzdálenost, o kterou se A musí posunout, aby měla prázdný průnik s B a aby se zároveň dotýkala B.

Opět existují algoritmy, které hloubku průniku dokáží řešit ve dvou a tří dimenzionálních prostorech, ale pro více rozměrné prostory je výpočet obtížný a časově náročný. Proto lze použít lehce upravenou verzi hledání hloubky průniku, abychom posléze vypočetli onu kýženou minimální vzdálenost bodu od

konvexního obalu. V tomto případě bereme v potaz projekci konvexního obalu ve vícedimenzionálním prostoru místo toho, abychom ve všech dimenzích přenášeli veškeré body množiny *X*. Tímto způsobem lze i ve vícedimenzionálních prostorech použít zmíněný algoritmus. Místo zpracování všech bodů množiny v daném prostoru zpracujeme pouze vrcholy konvexního obalu. Tyto vrcholy ideálně spočteme z tréninkové množiny ještě předtím, než budeme body používat jak pro fázi tréninku modelu, tak posléze pro klasifikaci testovaných bodů. Tento přístup nám nakonec ušetří nemalé množství výpočtů.

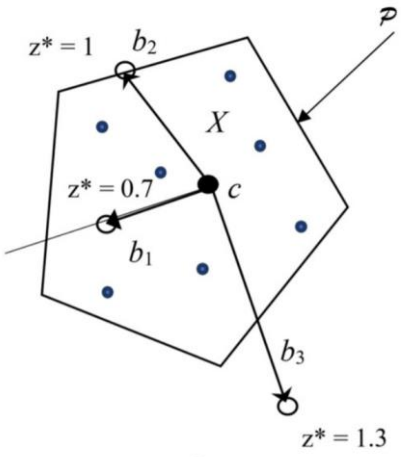
3 Nově navržená metoda

3.1 Postup

Nyní se podíváme na to, jak autoři článku navrhnuli novou metodu pro odhad vzdálenosti testovaného bodu od konvexního obalu. Vycházeli z lineárního programu (LP1), jehož cílem je efektivně nalézt body na hranici konvexního obalu [13]. Bodům se snažíme optimálně nalézt informaci o jejich umístěné vůči hranici konvexního obalu. Máme-li testovaný bod *b*, množinu *X* a těžiště konvexního obalu *c*, poté pro popis situace užijeme vektor *b*, který je v tomto případě spojnicí bodu *b* a těžiště *c*. Následně můžeme v závislosti na vektoru *b* odhadnout řešení lineárního programu pomocí hodnoty z^*, která určí pozici bodu vůči konvexnímu obalu následovně [1]:

- $z^* < 1$: $Bod\ b\ leží\ uvnitř\ conv(X)$
- $z^* = 1$: $Bod\ b\ leží\ na\ hranici\ conv(X)$
- $z^* > 1$: $Bod\ b\ leží\ vně\ conv(X)$

Tento vztah vektoru *b* a odhadu řešení z^* si můžeme lépe představit díky Obr. 3. Na něm vidíme tři body (b_1, b_2, b_3), které se vyskytují ve všech třech odlišných pozicích vůči konvexnímu obalu *P*. Tento obal je vytvořen nad množinou bodů *X*. Optimální hodnotu z^*, dostaneme tak, že budeme náš vektor *b* zvětšovat či zmenšovat, dokud se nedostaneme na hranici konvexního obalu, kde vytvoříme nově bod s pozicí $(1/z^*)b$.



*Obr. 3: Množina X (2D); konvexní obal P; těžiště c; testované body/vektory b_1, b_2, b_3; optimální hodnoty LP pro tyto body označené z**

Pokud budeme chtít zjistit euklidovskou vzdálenost nově vytvořeného bodu od původního bodu *b*, využijeme poměr hodnoty $|z^* - 1|$ a hodnoty z^* a dostaneme tak po menších úpravách následující vztah [1]:

$$D = \|b\| \, |z^* - 1| / z^*$$

Rovnice 2: Určení vzdálenosti bodu b od hranice konvexního obalu

Tady vidíme, že vektor nezáleží na tom, kde přesně se bod nachází, tedy zda je $F = 0$ nebo $F = 1$. Hodnota *D* tedy bude vzdálenost bodu *b* od místa, který je průnikem polopřímky definované těžištěm *c* a bodem *b*. Díky použití těžiště dostaneme relativně vhodné odhady bodu na konvexním obalu blízkého k našemu bodu *b*. Toto zjednodušení výrazně sníží výpočetní složitost celého algoritmu.

3.2 Pseudokód

Následující pseudokód popisuje algoritmus určující přibližnou vzdálenost testovaného bodu od konvexního obalu. Po jeho běhu se výstup použije pro klasifikaci bodu v rámci principu strojového učení.

Algoritmus DISTANCE_LP

[VSTUP:] X_i, b // Matice množiny bodů X_i třídy i, testovaný bod b

[VÝSTUP:] F_i, D_i // Popisek pozice F_i, vzdálenost D_i (bodu b od konvexního obalu) třídy i

```
1.0 Inicializace:
    1.1 Převod množiny bodů do matice dat Xi.
    1.2 Výpočet těžiště c množiny Xi.
    1.3 Nastavení popisku Fi = 0.
    1.4 Přenesení bodů pomocí středové souměrnosti: X'i = Xi − c ; b' = b − c
2.0 Určení odhadu LP u X'i, b' pro získání z*
3.0 If (z* < 1) then Fi = 1.
4.0 Výpočet Di pomocí (Rovnice 2).
5.0 Konec.
```

Výše zmíněné proměnné v programu a jejich smysl jsou popsané v přechozí *Kapitole 3.1*. Co však stojí za zmínku, je bod programu **1.4**. Autor článku zde přenáší body množiny na jejich středový obraz, ačkoliv se to tom blíže nezmiňuje. V doložené literatuře [13] se naopak autor původního algoritmu LP1, ze kterého se vychází, zmiňuje, že toto přenesení bodů pomocí středové souměrnosti je potřeba, jen pokud se dané těžiště *c* nachází mimo konvexní obal.

Dále je dobré zmínit, že při odhadu vzdálenosti bodu *b* od konvexního obalu množiny **X** se jako konvexní obal považují pouze body v množině **X**, které nejsou uvnitř konvexního obalu a tvoří tak hranici obalu. Zrychlí to výpočet a zároveň předejdeme nepředvídatelným efektům testovaných bodů v blízkosti těžiště *c*.

Po dokončení běhu programu musíme řešit problém *„mnoha tříd“*. V jednotlivých třídách *i* ke každé množině X_i odpovídá spojení (F_i,D_i). Finální klasifikaci bodu provedeme následovně:

1. Jestliže žádné spojení neobsahuje $F = 1$, poté popisek odpovídá třídě s nejmenší vzdáleností od testovaného bodu.
2. Jestliže právě jedno spojení obsahuje $F = 1$, potom popisek odpovídá právě tomuto jedinému spojení.
3. Jestliže několik spojení (nebo dokonce všechny) obsahují $F = 1$ a zároveň indexy *i* těchto spojení vytváří množinu *G*, pak popisek odpovídá třídě v *G* s nejmenší vzdáleností k testovanému bodu

3.3 Výsledky experimentu

Navrhovaný klasifikační algoritmus *NCH* byl testován na známém problému diagnózy rakoviny prsu [14]. Výsledky jsou ukázány v Tabulce 1. Sloupce p_1 a p_2 odpovídají hodnotám rozhodovací chyby pro třídy B (nezhoubný) a třídy M (zhoubný). Aritmetický průměr těchto dvou chyb je obsažen ve sloupci *p*. V tabulce jsou rovněž pro porovnání uvedeny ostatní algoritmy, které byly popsány v *Kapitole 2.2*. Nově navržený algoritmus byl použit jednak s křížovou validací *NCH-1* tak bez ní *NCH-2*. V obou případech bylo v datech 444 případů nezhoubného nádoru a 239 případů zhoubného nádoru. Stejně tak se v obou případech data rozdělila na jednu polovinu pro tréninkovou sadu a na druhou polovinu pro testovací sadu.

Data	p_1	p_2	p
SVM (linear)	4.10	2.00	3.05
SNCH [3]	–	–	2.70
LNCH [6]	2.93	4.18	3.56
kNN ($k = 5$)	2.25	3.15	2.70
NCH-1	0.45	0	0.23
NCH-2	1.8	2.5	2.15

Tabulka 1: Rozpoznání chyb pro jednotlivé algoritmy.
NCH-1 – bez křížové validace, NCH-2 – s křížovou validací

4 Závěr

V článku byla navrhnuta nová metoda pro určení vzdálenosti bodu od konvexního obalu. Metoda se zakládá na odhadnutí hodnoty v rámci lineárního programování, kde algoritmus určuje vzdálenost pomocí těžiště konvexního obalu. Zrychlení ve výpočtech je zároveň dosaženo pomocí toho, že algoritmus pracuje s konvexním obalem a přesněji s body, které tvoří jeho ohraničení, místo práce se všemi body. Výsledky experimentu ukázaly, že navrhovaná metoda vykazuje zlepšení týkající se kvality rozpoznání oproti běžně užívané metodě *LNCH*. Nová metoda *NCH* je relativně jednoduchá na implementaci a nevyžaduje definici

vstupních parametrů uživatelem. V budoucnu se zkoumání může zaměřit především na efektivní algoritmy získání hranice konvexního obalu.

Autoři vypracovali poměrně zajímavou metodu výpočtu, kde dokázali oproti běžně používaným algoritmům ušetřit výpočetní výkon a zároveň zmenšit chybu při výsledné klasifikaci. Dle mého názoru algoritmus určitě najde uplatnění nejen ve zdravotnictví, ale také v celé řadě úkonů, které se nutně nemusí zabývat jen strojovým učením, ačkoliv tam lze pravděpodobně očekávat největší přínos článku. Jelikož jsou obecně disciplíny spojené s umělou inteligencí a strojovým učením poměrně mladým oborem, můžeme v příštích dekádách předvídat desítky nových zajímavých nápadů, které posunou hranice vědění zase o krok dál.

Reference

[1] **A. P. Nemirko, J. H. Dulá** (2021). Machine learning algorithm based on convex hull analysis. *Procedia Computer Science*. [Online] [Citace: 23. 10. 2021.] https://www.sciencedirect.com/science/article/pii/S1877050921009911

[2] **RNDr. Petra Surynková, Ph.D.**. Konvexní obal a množina. *Počítačová geometrie – Přednáška 8*. [Online] [Citace: 26. 10. 2021.] http://surynkova.info/dokumenty/mff/PG/Prednasky/prednaska_8.pdf

[3] **R.L. Graham** (1972). An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*. 1(4), pp.132–133. [Citace: 26. 10. 2021.]

[4] **R.A. Jarvis** (1973). On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*. 2(1), pp.18–21. [Citace: 26. 10. 2021.]

[5] **T.M. Chan** (1995). Output-sensitive construction of convex hulls. University of British Columbia. [Citace: 26. 10. 2021.]

[6] **C. Barber, D.P. Dobkin and H. Huhdanpaa** (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4), pp.469–483. [Citace: 26. 10. 2021.]

[7] **V. Vapnik** (1998). Statistical learning theory. *Wiley*. [Citace: 26. 10. 2021.]

[8] **K.P. Bennett and E.J. Bredensteiner** (2000). Duality and geometry in SVM classifiers. *ICML*, pp.57–64. [Citace: 26. 10. 2021.]

[9] **V. Franc and V. Hlaváč** (2003). An iterative algorithm learning the maximal margin classifier. *Pattern Recognition*, 36(9), pp.1985–1996. [Citace: 26. 10. 2021.]

[10] **B.F. Mitchell, V.F. Demyanov and V.N. Malozemov** (1974). Finding the point of a polyhedron closest to the origin. *SIAM Journal on Control*, 12(1), pp.19–26. [Citace: 26. 10. 2021.]

[11] **R. Weller** (2013). New geometric data structures for collision detection and haptics. *Springer Science & Business Media*. [Citace: 26. 10. 2021.]

[12] **M.C. Lin, D. Manocha, and Y.J. Kim** (2018). Collision and proximity queries. Handbook of Discrete and Computational Geometry. *CRC Press*, pp.1029–1056. [Citace: 26. 10. 2021.]

[13] **J.H. Dula and R.V. Helgason** (1996). A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space. *European Journal of Operational Research*, 92, pp.352–367. [Citace: 28. 10. 2021.]

[14] **UCI Machine Learning Repository** (2020). Breast Cancer Wisconsin (Original) Data Set. [Online] [Citace: 28. 10. 2021.] https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)