Învățare Automată - Tema 3
Varianta C:
Clasificatorii Naïve Bayes și K-Nearest Neighbors

Tudor Berariu
Laboratorul AIMAS
Facultatea de Automatică și Calculatoare

17 aprilie 2014

1 Pe scurt, ...

... se cere să se implementeze doi clasificatori, Naïve Bayes și K-Nearest Neighbors și să se folosească pentru clasificarea mailurilor (*spam* / *ham*) și a recenziilor de film (*negative* / ... / *positive*).

Argumente și contraargumente pentru alegerea temei:

+ învățarea a doi algoritmi de învățare automată simpli, dar puternici;
+ clasificarea a două seturi de date reale;
- seturile de date necesită prelucrare;
- este necesar un timp de rulare mare.

2 Motivația temei

Scopul acestei teme îl reprezintă înțelegerea ipotezelor și a funcționării algoritmilor Naïve Bayes și K-Nearest Neighbors, precum și aplicarea acestora pentru **clasificare de texte** pe două seturi de date reale. Rezultatele

aplicării celor două metode de clasificare nu vor fi cele mai bune cu putință, dar cei doi algoritmi reprezintă un punct de plecare bun pentru abordarea unor astfel de seturi de date.

Clasificarea textelor are aplicații importante în: detectarea mesajelor de tip spam, identificarea limbii în care a fost scris un text sau detectarea autorului unui text (sau a vârstei sau a sexului acestuia). Multe dintre aceste probleme au fost rezolvate satisfăcător cu ajutorul Naïve Bayes sau K-Nearest Neighbors.

3 Descrierea celor două metode

Pornind de la un set de date $\mathbf{X}$ în care fiecare exemplu $x^{(i)}$ format dintr-o serie de atribute $x_1^{(i)}, x_2^{(i)}, \ldots, x_k^{(i)}$ este etichetat cu o clasă $c^{(i)} \in \mathbf{C}$, se dorește construirea unui clasificator care să eticheteze date noi.

3.1 Naïve Bayes

Naïve Bayes reprezintă o metodă statistică de clasificare, bazată pe Teorema lui Bayes (Formula 1) pentru exprimarea relației dintre probabilitatea *a priori* și probabilitatea *posterioară* ale unei ipoteze.

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)} \tag{1}$$

În Teorema lui Bayes, (Formula 1):

- $P(c)$ reprezintă probabilitatea *a priori* a clasei c,
- $P(c|x)$ reprezintă probabilitatea *a posteriori* a ipotezei "x aparține clasei c",
- $P(x|c)$ reprezintă probabilitatea ca exemplul x să aparțină clasei c (*verosimilitatea*).

Un clasificator Naïve Bayes funcționează pe principiul *verosimilității maxime* (eng. *maximum likelihood*), deci alege clasa c pentru care probabilitatea $P(c|x)$ este maximă (MAP = *maximum a posteriori*):

$$c_{MAP} = \underset{c \in \mathbf{C}}{\operatorname{argmax}}\, P(c|x) = \underset{c \in \mathbf{C}}{\operatorname{argmax}}\, \frac{P(x|c) \cdot P(c)}{P(x)} = \underset{c \in \mathbf{C}}{\operatorname{argmax}}\, P(x|c) \cdot P(c) \tag{2}$$

Fiecare exemplu x este descris printr-un număr de atribute $x_1, \ldots x_k$, de aceea Formula 2 se poate rescrie astfel:

$$c_{MAP} = \underset{c \in \mathbf{C}}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_k | c) \cdot P(c) \tag{3}$$

Cuvântul *naiv* provine de la faptul că Naïve Bayes nu ține cont de posibilele relații de dependență între atribute. Practic, presupunerea făcută este aceea că atributele sunt condițional independente odată ce clasa este cunoscută.

$$P(x_1, x_2, \ldots, x_k | c) = P(x_1|c) \cdot P(x_2|c) \cdot \ldots \cdot P(x_k|c) \tag{4}$$

Aplicând presupunerea exprimată de Formula 4, clasa prezisă de un clasificator Naïve Bayes devine:

$$c_{NB} = \underset{c \in \mathbf{C}}{\operatorname{argmax}} P(c) \cdot \prod_{x_i} P(x_i|c) \tag{5}$$

O prezentare mai detaliată a algoritmului puteți vizualiza aici: `https://www.youtube.com/watch?v=8yvBqhm92xA`.

3.2 K-Nearest Neighbors

K-Nearest Neighbors[CH67] este un algoritm *leneș* de învățare automată care urmărește un principiu simplu: fiecare exemplu nou z este etichetat cu acea clasă c care este cea mai frecventă între cele mai apropiate k exemple din setul de învățare $\mathbf{X}$ de z.

Observați că K-Nearest Neighbors nu are o etapă de antrenare separată în care să fie construit un model, ci întreg procesul are loc atunci când un punct nou trebuie clasificat.

K-Nearest Neighbors este un algoritm simplu, însă necesită alegerea *a priori* a lui k, precum și definirea unei distanțe între două puncte din spațiul de intrare (ceea ce nu este întotdeauna ușor).

Pentru un exemplu nou z și un set de antrenare $\mathbf{X}$:

1. $neighbors \longleftarrow nearest(z, k, \mathbf{X})$
2. $c \longleftarrow most\_frequent\_class(neighbours)$

Pentru o explicație mai detaliată a algoritmului urmăriți această prezentare: `https://www.youtube.com/watch?v=4ObVzTuFivY`.

4 Clasificarea Textelor

Clasificarea textelor presupune gruparea documentelor în categorii prestabilite pe baza conținutului lor. Setul de date de învățare va conține documente cu etichete atașate.

În cadrul acestei teme se face o presupunere simplificatoare, aceea că pozițiile cuvintelor în cadrul textului nu contează. Documentele sunt reprezentate ca multimulțimi de cuvinte (eng. *bag of words*) [1] sau ca vectori ponderați `tf-idf` [2] [3].

5 Cerințe

3 puncte: Să se implementeze algoritmul de clasificare Naïve Bayes într-un limbaj de programare la alegere.

3 puncte: Să se implementeze algoritmul K-Nearest Neighbors într-un limbaj de programare la alegere.

2 puncte: Să se aplice cei doi algoritmi deja implementați asupra setului de date SpamAssasin (descris în Anexa A.1). Să se descrie în fișierul `README` cum anume a fost împărțit setul de date (pentru testare și antrenare) și care a fost acuratețea clasificatorului (o matrice de confuzie, eng. *confusion matrix*). Pentru algoritmul K-Nearest Neighbors trebuie precizate rezultatele pentru fiecare valoare a parametrului k alesă, precum și modul în care a fost calculată distanța între două exemple.

2 puncte: Să se aplice cei doi algoritmi deja implementați asupra setului de date Rotten Tomatoes (descris în Anexa A.2). Să se scrie toate detaliile despre antrenare, acuratețe, etc. în fișierul `README`.

Bonus, 2 puncte: Să se aplice un alt algoritm de clasificare pentru a obține un scor mai bun pe unul dintre cele două seturi de date. Se poate alege orice metodă studiată sau implementată la curs sau laborator sau un algoritm nou, dar atunci acesta trebuie descris în fișierul `README`

[1] `http://www.stanford.edu/class/cs124/lec/naivebayes.pdf`

[2] `https://www.usenix.org/legacy/event/sec02/full_papers/liao/liao_html/node4.html`

[3] `http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html`

ce însoțește tema. Algoritmul poate fi și o versiune mai sofisticată a celor două metode implementate deja (de exemplu Weighted K-Nearest Neighbors). Tot în `README` se va explica și alegerea făcută (preferința pentru acel algoritm anume).

6 Trimiterea temei

În arhiva temei includeți:

- toate fișierele sursă (eventual cu `Makefile` / script de compilare și rulare),
- fișier `README` (în format pdf) în care să includeți toate detaliile cerute în cerințele temei.

A Seturile de date

A.1 SpamAssasin

Un set de date ce cuprinde mailuri `spam` și `ham` (non-spam) se găsește aici: `https://spamassassin.apache.org/publiccorpus/`. Detalii despre conținutul setului de date se găsesc în fișierul `readme.html`. Este un set de date folosit pentru testare de majoritatea celor ce implementează filtre anti-spam.

A.2 Sentiment Analysis on Movie Reviews

Un set de date ce conține poziționări afective față de filme este cel implicat într-un concurs pe `kaggle.com`: *Sentiment Analysis on Movie Reviews*. Acesta conține fraze din recenzii de film de pe Rotten Tomatoes etichetate cu valori între 0 (comentariu negativ) și 4 (comentariu pozitiv).

Setul de date și detaliile despre conținutul acestuia se pot obține de aici: `http://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data`.

Bibliografie

[CH67] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.