Алгоритми за класификация в машинното обучение.

Класификация на текст

1. Алгоритми за класификация

Постановка на задачата за класификация.

Нека е дадено крайно множество от класове C={c_1,c_2,….,c_k} и крайно множество от документи D={d_1,d_2,….,d_k}. Целевата функция f: DxC→{0,1} за всяка двойка <документ, клас> е неизвестна. Необходимо е да се намери класификатор f ‘, т.е. функция, максимално близка към функцията f.

Основната задача на класификацията е приобщаването на даден документ към определен клас в съответствие с набор от признаци. Традиционно в качеството на признаци се използват честотата на срещане на думите.

1.1. Наивен Бейсов класификатор

Един от класическите алгоритми в машинното обучение е Наивният Бейсов класификатор, който се базира на теоремата на Бейс за определяне на апостериорната вероятност за настъпване на дадено събитие.

Теорема на Бейс

$$P(y = c|x) = \frac{P(x|y=c)P(y=c)}{P(x)} \quad (1)$$

където:

P(y=c|x) е вероятността обект да принадлежи на клас c (апостериорна вероятност на класа)

P(x|y=c) – вероятността обекта x да се среща в средата на обекта на класа *c*

P(y=c) – безусловна вероятност да се среща обект y в клас *c* (априорна вероятност на класа)

P(x) – безусловна вероятност на обекта x

Целта на класификацията се състои в това, да се определи към какъв клас принадлежи обекта x. Следователно е необходимо да се намери вероятностен клас на обекта x, т.е. от всички класове да се избере този, който дава максимална вероятност P(y=c|x).

$$c_{opt} = \arg\max_{c \in C} P(x|y = c)P(y = c) \quad (2)$$

Наивният класификатор на Bayes предполага, че атрибутите описващи примерите са независими, въпреки контекста на текста. Това предположение изглежда невярно в повечето реални задачи, но на практика класификатора използващ теоремата на Бейс определя с висока точност верния клас.

Известни алгоритми от типа Наивен Бейсов класификатор, са: Бернулиев и Мултиноминален, свързани с различни предположения за разпределението на признаците.

1.1.1. Бернулиев Бейсов класификатор

Бернулиевият Бейсов класификатор представя документа, като бинарен вектор на пространството. Стойност единица, показва дали думата се появява поне веднъж в документа. С такова представяне, правим наивното предположение на Бейс, че вероятността всяка дума да се среща в текста е независима от появата на други думи.

Нека x∈[0,1] е бинарна променлива, която има вероятност за поява θ.

Тогава:

$$P(x|\theta) = \begin{cases} \theta & \text{, ако x} = 1 \\ 1-\theta & \text{, ако x} = 0 \end{cases} \quad (3)$$

1.1.2. Мултиноминален Бейсов класификатор

Мултиноминалният Наивен Бейсов класификатор, прави предположението, че признаците са разпределени мултиноминално.

Нека $x_i \in \{1,\dots K\}$, имат вероятности за поява $\theta_1, \dots \theta_K$

Тогава, вероятността за поява на събитието x при даден признак θ е:

$$P(\mathrm{x}|\theta) = \frac{n!}{\mathrm{x}_1!\dots \mathrm{x}_K!} \prod_{i=1}^{K} \theta_i^{\mathrm{x}i} \quad (4)$$

където: $n = \sum_{i=1}^{K} \mathrm{x}_i$ (5)

Многономиналният Наивен Бейсов класификатор изчислява честотата на поява на всяка дума в документите. Отново се прави наивното предположение, че вероятността дадена дума да се среща в текста е независима от контекста и от позицията на думата в документа.

1.2. Метод на опорните вектори - SVM (Support Vector Mashine).

Методът на опорните вектори (SVM, Support Vector Machines) представя обучаващите примери като точки в n-мерно пространство. Примерите са проектират в пространството по такъв начин, че да бъдат линейно разделими. При работа с два класа се търси начин да се начертае линия, която да разделя данните от двата класа. Линията, която разделя данните, се нарича разделителна хипер равнина. Тази хипер равнина трябва да се избере по такъв начин, че да се намира възможно най-далеч от примерите и на двата класа.

Функцията f(x) на линейната класификация е във вида:

$$f(x) = w^T x + b \quad (3)$$

където: w^T *е* тегловен вектор, а b е отклонението

След като се определи, къде да бъде поставена разделителната линия, се калкулира допустимата границата:

$$label * (w^T x + b) \quad (4)$$

Целта е да се намерят стойностите на w^T и b, които ще определят класификатора. За да се направи това, е необходимо да се намерят точките с най-малко отклонение, който трябва да се максимизира. Това може да се запише по този начин:

$$arg \max_{w,b} \left\{ \min_n label * (w^T + b). \frac{1}{\|w\|} \right\} \quad (5)$$

1.3. Метод на К най-близкия съсед (KNN - k nearest neighbours).

Методът на К най-близкия съсед е метрически алгоритъм за класификация на обекти, основан на оценка на сходство на обекти. Класифицираният обект се отнася към този клас, към който принадлежат най-близките до него обекти на обучаващата извадка (KNN - k nearest neighbours).

Алгоритъм

- Определяне на К броя обекти, които ще участват в класификацията

- Изчисляване на дистанцията между всеки два обекта от обучаващата извадка, чрез използване на подходяща функция за измерване на разстояние между две точки
- Избор на K обекти от обучаващата извадка, разстоянието до което е минимално
- Класа на класифицирания обект – това е класа на новия обект, намерен въз основа на заключенията, направени по отношение на K най-близки съседи

Видове функции, изчисляващи разстояние:

- Евклидово

$$\rho(x_i, x_j) = \sqrt{\sum_{k=1}^{m} w_k \, (x_i^{(k)} - x_j^{(k)})^2} \quad (6)$$

- L_p – метрика

$$\rho(x_i, x_j) = (\sum_{k=1}^{m} w_k \, |x_i^{(k)} - x_j^{(k)}|^p)^{1/p} \quad (7)$$

- L_∞ – метрика

$$\rho(x_i, x_j) = \max_{k=1,2,\dots,m} |x_i^{(k)} - x_j^{(k)}| \quad (8)$$

- L_1 – метрика

$$\rho(x_i, x_j) = \sum_{k=1}^{m} |x_i^{(k)} - x_j^{(k)}| \quad (9)$$

- Ланс – Уилямс

$$\rho(x_i, x_j) = \frac{\sum_{k=1}^{m} |x_i^{(k)} - x_j^{(k)}|}{\sum_{k=1}^{m} (x_i^{(k)} - x_j^{(k)})} \quad (10)$$

- Косинусова мяра

$$\rho(x_i, x_j) = \frac{\sum_{k=1}^{m} x_i^{(k)} x_j^{(k)}}{\sqrt{\sum_{k=1}^{m} (x_i^{(k)})^2} \sqrt{\sum_{k=1}^{m} (x_j^{(k)})^2}} \quad (11)$$

$x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(m)})$ – вектор от m-признаци на i- я обект
$x_j = (x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(m)})$ – вектор от m-признаци на j- я обект

Важен въпрос при работа с метода на K най-близкия съсед е изборът на числото K – това е броя на най-близките съседи. Евристични техники, като кръстосано валидиране могат да помогнат за получаването на подходящи стойности на K.

2. Класификация на текст

2.1. Библиотека Natural Language Toolkit

Практическата работа при обработка на естествен език, обикновено използва големи обеми от лингвистични данни или корпуси. **NLTK (Natural Language Toolkit)** е open source библиотека на Python, предоставящ безплатно онлайн книги, статии и данни. [1]

2.1.1. Текстови корпуси на NLTK

- **Корпус Гутенберг**

NLTK включва малка колекция от текстове от електронния текстов архив на Project Gutenberg, който съдържа около 25 000 безплатни електронни книги, които се намират на http://www.gutenberg.org/

[1] http://www.nltk.org/book/ch02.html

```
>>> import nltk
>>> from nltk.corpus import gutenberg
```

- **Уеб и чат текстове**

Малка колекция от уеб текстове на NLTK включва съдържание от дискусионен форум на Firefox, разговори, които са чути в Ню Йорк, филмовия скрипт на Carribean Pirates , лични реклами и рецензии за вино.

```
>>> from nltk.corpus import webtext
```

Съществува и корпус от чат-сесии от незабавни съобщения, първоначално събрани от Военноморското училище за следдипломна квалификация за изследване на автоматичното откриване на интернет хищници. Корпусът съдържа над 10 000 публикации, като са заменени потребителските имена с общи имена "UserNNN" и ръчно редактирани, за да премахната идентифицираща информация. Корпусът е организиран в 15 файла, където всеки файл съдържа няколкостотин публикации, събрани за дадена дата, за възрастова група (тийнейджъри, 20, 30 и 40, както и общ чат за възрастни). Името на файла съдържа датата, чата и броя публикации; напр. 10-19-20s_706posts.xml съдържа 706 публикации, събрани от 20-те години на чат на 10/19/2006.

```
>>> from nltk.corpus import nps_chat
```

- **Корпус Браун**

Браун Corpus е първият милион електронен корпус на английски, създаден през 1961 г. в университета Браун. Този корпус съдържа текст от 500 източника и източниците са категоризирани по жанр, като: *новини, мнения, хобита, мистерии, научна фантастика, хумор и други*. Пълен списък на всички раздели, може да видите на следния адрес: http://icame.uib.no/brown/bcm-los.html

```
>>> from nltk.corpus import brown
```

- **Корпус Ройтерс**

Reuters Corpus съдържа 10,788 новинарски документа, възлизащи на 1,3 милиона думи. Документите са класифицирани в 90 теми и групирани в два набора, наречени "обучение" и "тест".

```
>>> from nltk.corpus import reuters
```

- **Корпус: Inaugural Address Corpus**

Колекция от 55 текста, известни като президентските уводни речи на САЩ от 1789 година до 2009 година, събрани и предоставени от Kathleen Ahrens.

```
>>> from nltk.corpus import inaugural
```

- **Корпус: mouvie_review**

Колекция от 1000 положителни и 1000 отрицателни отзиви на филми, използвани за анализ на полярността на настроенията, предоставени през 2004 година от Bo Pang и Lillian Lee .

```
>>> from nltk.corpus import movie_reviews
```

- **Корпус: names**

Колекция от 5001 английски женски имена и 2943 мъжки имена, сортирани по азбучен ред, предоставени през 1991 година от Mark Kantrowitz.

- **Зареждане на собствен корпус**

Ако имате своя собствена колекция на текстови файлове, можете да ги заредите с помощта на NLTK на PlaintextCorpusReader.

```
>>> from nltk.corpus import PlaintextCorpusReader
```

2.1.2. Структура на текстовия корпус

Най-простият вид корпус е колекция от текстове без конкретна организация. Често текстовете се групират в категории, които биха могли да съответстват на жанра, източника, автора, езика и т.н. Понякога тези категории се припокриват, особено в случаите на актуални категории, тъй като текстът може да е релевантен за повече от една тема. Повече документация може да се намери с помощта на nltk.corpus.reader или http://nltk.org/howto.

3. Задачи.

При т.нар. класификация с учител (Supervised Classification) се използват две множества от данни – обучаващи данни (training set) и тестови данни (testing set). Обучаващите данни служат като „учител“ по отношение на тестовите данни. Изгражда се модел за предсказване на нови данни въз основа на алгоритми на машинното обучение, използващи обучаващите данни и тестващи модела върху тестовите данни.

В сайта http://www.nltk.org/book/ch06.html е представена схема описваща т.нар. класификация с учител - „Надзорна класификация“

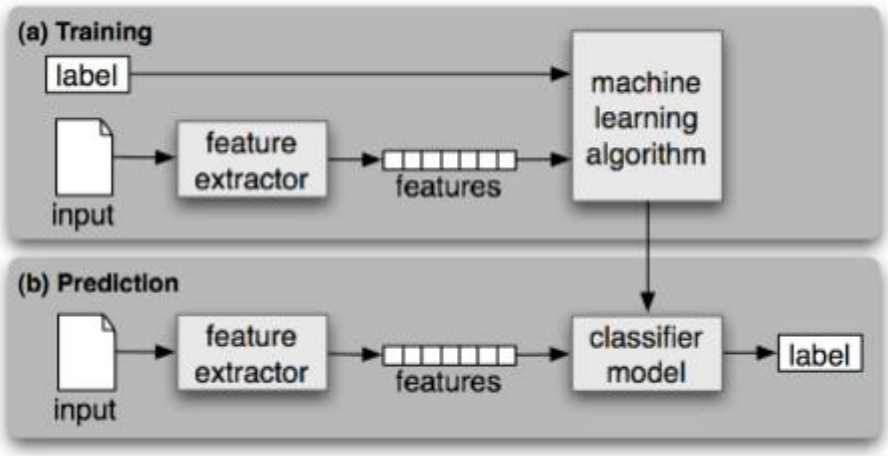

***Figure 1.1**: Supervised Classification. (a) During training, a feature extractor is used to convert each input value to a feature set. These feature sets, which capture the basic information about each input that should be used to classify it, are discussed in the next section. Pairs of feature sets and labels are fed into the machine learning algorithm to generate a model. (b) During prediction, the same feature extractor is used to convert unseen inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels.*[2]

[2] http://www.nltk.org/book/ch06.html

Задача 1. Класификация на текст

С помощта на класификаторите: Бернулиев Бейсов класификатор, Мултиноминален Бейсов класификатор, Метод на опорните вектори и Метод на К най-близкия съсед да се изчисли и сравни точността на разпознаване на мненията на потребителите въз основа на филмови отзиви, предоставени в корпуса: movie_reviews.

```
import nltk
from nltk.classify.scikitlearn import SklearnClassifier
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB,BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from nltk.corpus import movie_reviews
import random

#създаване на списък от всички документи и тяхната категория:
#neg - негативно мнение и pos - позитивно мнение
docs = [(list(movie_reviews.words(fileid)), category)
        for category in movie_reviews.categories()
        for fileid in movie_reviews.fileids(category)]

#Разместване на случаен принцип всички документи
random.shuffle(docs)

#създаване на списък от всички думи и превръщането им в малки букви
all_words = []
for w in movie_reviews.words():
   all_words.append(w.lower())

#Изчисляване на честотата на срещане на всяка дума
all_words=nltk.FreqDist(all_words)

#списък с първите 3000 най-често използвани думи в корпуса
word_features = list(all_words.keys())[:3000]
```

```
#създаване на екстрактор, който проверява дали всяка от думите присъства
#в даден документ
def find_features(document):
   words = set(document)
   features={}
   for w in word_features:
      features[w] = (w in words)
   return features

featuresets = [(find_features(rev),category) for (rev,category) in docs]

#обучаващо множество
training_set = featuresets[:1900]
#тестващо множество
testing_set = featuresets[1900:]

BNB_classifier = SklearnClassifier(BernoulliNB())
BNB_classifier.train(training_set)
print("BernoulliNB accuracy percent:",(nltk.classify.accuracy(BNB_classifier,
testing_set))*100)

MNB_classifier = SklearnClassifier(MultinomialNB())
MNB_classifier.train(training_set)
print("MultinomialNB accuracy
percent:",(nltk.classify.accuracy(MNB_classifier, testing_set))*100)

LinearSVC_classifier = SklearnClassifier(LinearSVC())
LinearSVC_classifier.train(training_set)
print("LinearSVC accuracy
percent:",(nltk.classify.accuracy(LinearSVC_classifier, testing_set))*100)
```

```
KNN_classifier = SklearnClassifier(KNeighborsClassifier())

KNN_classifier.train(training_set)

print("KNN accuracy percent:",(nltk.classify.accuracy(KNN_classifier,
testing_set))*100)
```

Задача 2.1. Идентификация на пол.

В българския език имената на жените обикновено завършват на двете гласни: а и я. Например: Яна, Ива, Мария, Иванова, Ковачева и т.н. Имената на мъжете завършват обикновено на съгласна: Иван, Драган, Петкан, Драгомир, Иванов, Коларов и т.н.

Напишете функция, която според името на човек с българско име, извежда информация, от какъв пол е.

Задача 2.2. Идентификация на пол.

В английския език имената на мъжете и жените имат отличителни черти. Имената на жените завършват на гласните: *a, e i*, докато имената завършващи на *k* , *o* , *r* , *s* и *t* , вероятно са мъжки.

Да се създаде скрип, който според името на човек с английско име, извежда информация, от какъв пол е. Да се използва корпуса names и вградения класификатор nltk.NaiveBayesClassifier и да се изчисли колко процента е точността на върнатата информация.

Задача за домашна работа (по желание).

Да се имплементира един от следните класификатори: Бернулиев Бейсов класификатор, Мултиноминален Бейсов класификатор, Метод на опорните вектори или Метод на К най-близкия съсед за класифициране на текст на тема по избор, без да се използват вградените библиотеки. Да се създаде собствен корпус от текстове.