

Naive Bayes

Michael Staniek

University of Heidelberg

November 16, 2017

Inhalt

- 1 Klassifikation und Naive Bayes
 - Recap
- 2 Multiclass Klassifikation
- 3 Naive Bayes

Recap vom Ende letzter Woche

- Perzeptron
- trainiertes Gewicht $w=(0.1, 1)$
- $x_1=(1, 1)$
- $x_2=(1, -1)$
- $w*x_1=1.1$
- $w*x_2=-0.9$
- $w*x_1$ hat ein positives Vorzeichen
- $w*x_2$ hat ein negatives Vorzeichen

Lossfunktion fürs Klassifizieren

$$\max(0, -y*w*x)$$

Lossfunktion fürs Klassifizieren

$$\max(0, -y * w * x)$$

$$\begin{cases} -y * w * x & \text{if } -y * w * x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Lossfunktion fürs Klassifizieren

$$\max(0, -y * w * x)$$

$$\begin{cases} -y * w * x & \text{if } -y * w * x > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{cases} -y * w * x & \text{if } y * w * x \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

Lossfunktion fürs Klassifizieren

$$\begin{cases} -y * w * x & \text{if } y * w * x \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

Lossfunktion fürs Klassifizieren

$$\begin{cases} -y * w * x & \text{if } y * w * x \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{cases} -y * x & \text{if } y * w * x \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

Demotime

Gradient Descent

Learningrate 0.1

x	y	w	$\max(0, -y * w * x)$	$-y * x$ if $y * w * x < 0$
(0.9,0.8)	1	(1,0.1)	0	(0, 0)
(0.9,-0.8)	-1	(1,0.1)	0.82	(0.9, -0.8)
(0.9,0.8)	1	(0.91,0.18)	0	(0, 0)
(0.9,-0.8)	-1	(0.91,0.18)	0.675	(0.9, -0.8)
(0.9,0.8)	1	(0.82,0.26)	0	(0, 0)
(0.9,-0.8)	-1	(0.82,0.26)	0.52	(0.9, -0.8)
(0.9,0.8)	1	(0.73,0.34)	0	(0, 0)
(0.9,-0.8)	-1	(0.73,0.34)	0.385	(0.9, -0.8)

- Wir haben bisher 1 Gewichtsvektor, der + und - unterscheiden kann
- Wie kriegen wir es hin, mehrere Klassen zu unterscheiden?

Multiclass Klassifikation

- Wir haben bisher 1 Gewichtsvektor, der + und - unterscheiden kann
- Wie kriegen wir es hin, mehrere Klassen zu unterscheiden?
- Mehrere Gewichtsvektoren!
- $w_1=(0.1, 1)$, $w_2=(10, -2)$
- Welcher Gewichtsvektor für ein Trainingsbeispiel den höchsten Score liefert, gewinnt.
- $x=(1, 1)$
- $w_1*x=1.1$
- $w_2*x=8$
- Klasse von w_2 wird genommen.
- Sollte aber w_1 sein, was tun?

- Wir haben bisher 1 Gewichtsvektor, der + und - unterscheiden kann
- Wie kriegen wir es hin, mehrere Klassen zu unterscheiden?
- Mehrere Gewichtsvektoren!
- $w_1=(0.1, 1)$, $w_2=(10, -2)$
- Welcher Gewichtsvektor für ein Trainingsbeispiel den höchsten Score liefert, gewinnt.
- $x=(1, 1)$
- $w_1*x=1.1$
- $w_2*x=8$
- Klasse von w_2 wird genommen.
- Sollte aber w_1 sein, was tun?
- $w_1=w_1+x$
- $w_2=w_2-x$

Ausblick

- Und das kann man in einem einzigen Gewichtsvektor ausdrücken!

Multiclass Classification

Videotime!

Naive Bayes

$$p(c|d) = \frac{p(d|c)p(c)}{p(d)}$$

Ein einfaches Textklassifikationsbeispiel

- Pizza ist toll +
- Lasagne ist toll +
- Pizza ist bäh -
- ist der Satz 'Lasagne ist bäh' eher Positiv oder Negativ?

Ein einfaches Textklassifikationsbeispiel

- Pizza ist toll +
- Lasagne ist toll +
- Pizza ist bäh -
- ist der Satz 'Lasagne ist bäh' eher Positiv oder Negativ?
- Benutzen wir nur Wörter, die nicht Stopwörter sind: [Pizza, Lasagne, toll, bäh]
- mit dieser Erklärung erstellen wir nun Vektoren für die einzelnen Sätze.

Ein einfaches Textklassifikationsbeispiel

- (Pizza, Lasagne, toll, bäh)
- Pizza ist toll +
- (1, 0, 1, 0)
- Lasagne ist toll +
- (0, 1, 1, 0)
- Pizza ist bäh -
- (1, 0, 0, 1)

Jetzt summieren wir für jede Klasse die Vektoren auf.

Ein einfaches Textklassifikationsbeispiel

- (Pizza, Lasagne, toll, bäh)
- Pizza ist toll +
- (1, 0, 1, 0)
- Lasagne ist toll +
- (0, 1, 1, 0)
- Pizza ist bäh -
- (1, 0, 0, 1)

Jetzt summieren wir für jede Klasse die Vektoren auf. $(1, 1, 2, 0)$
 $+ (1, 0, 0, 1) -$

Ein einfaches Textklassifikationsbeispiel

- $(1, 1, 2, 0)$ +
- $(1, 0, 0, 1)$ -
- Der Testsatz 'Lasagne ist bäh' hat den Vektor $(0, 1, 0, 1)$
- + hat 2 Sätze, - hat 1 Satz.
- Wir brauchen auch noch wieviele Sätze wir Total haben. $T=3$
- die Größe unseres Vokabulars ist 4.

Ein einfaches Textklassifikationsbeispiel

- $(1, 1, 2, 0) +$
- $(1, 0, 0, 1) -$
- Der Testsatz 'Lasagne ist bäh' hat den Vektor $(0, 1, 0, 1)$
- $+$ hat 2 Sätze, $-$ hat 1 Satz.
- Wir brauchen auch noch wieviele Sätze wir Total haben. $T=3$
- die Größe unseres Vokabulars ist 4.
- Unsmoothed:
- $p(+|\text{testsatz})=p(+)*p(\text{Lasagne}|+)*p(\text{bäh}|+)$
- $p(+|\text{testsatz})=\frac{2}{3} * \frac{1}{4} * \frac{0}{4} = 0$
- $p(-|\text{testsatz})=p(-)*p(\text{Lasagne}|-)*p(\text{bäh}|-)$
- $p(-|\text{testsatz})=\frac{1}{3} * \frac{0}{2} * \frac{1}{2} = 0$

Ein einfaches Textklassifikationsbeispiel

- (1, 1, 2, 0) +
- (1, 0, 0, 1) -
- Der Testsatz 'Lasagne ist bäh' hat den Vektor (0, 1, 0, 1)
- + hat 2 Sätze, - hat 1 Satz.
- Wir brauchen auch noch wieviele Sätze wir Total haben. $T=3$
- die Größe unseres Vokabulars ist 4.
- $p(+|\text{testsatz})=p(+)*p(\text{Lasagne}|+)*p(\text{bäh}|+)$
- Smoothed:
- $p(+|\text{testsatz})=\frac{2}{3} * \frac{1+1}{4+4} * \frac{0+1}{4+4} = 0.0208$
- $p(-|\text{testsatz})=p(-)*p(\text{Lasagne}|)*p(\text{bäh}|)$
- $p(-|\text{testsatz})=\frac{1}{3} * \frac{0+1}{2+4} * \frac{1+1}{2+4} = 0.0185$

Wie soll euer Datenset aussehen?

Lasagne ist bäh\t-

Pizza ist toll\t+

Lasagne ist toll\t+

Pizza ist bäh\t-

Wie soll euer Output aussehen?

```
python3_Aufgabe3MichaelStaniek.py
```

```
+_0.0
```

```
-_0.0
```

```
python3_Aufgabe3MichaelStaniek.py_unsmoothed
```

```
+_0.0
```

```
-_0.0
```

```
python3_Aufgabe3MichaelStaniek.py_smoothed
```

```
+_0.020833333333333332
```

```
-_0.018518518518518517
```