Maschinelles Lernen und Data Mining

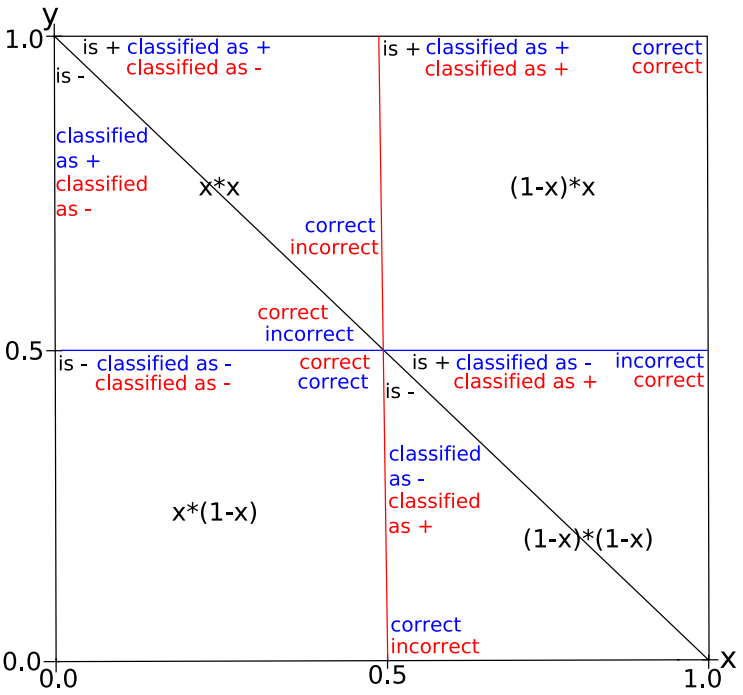
Übungsblatt für den 30.01.2007

Aufgabe 1

Gegeben sei ein Datenset mit 2 numerischen Attributen x, y mit jeweils Wertebereich [0,1]. Das gesuchte Konzept sei $y > 1 - x$ (wie auf Folie 16 dargestellt). Nehmen Sie weiters an, die Trainingsbeispiele sind gleichmäßig über den Bereich verteilt (d.h. alle x und y-Koordinaten sind gleich wahrscheinlich). Ihr Lernalgorithmus findet Decision Stumps mit nur einem binären Knoten, wobei er versucht, die Anzahl der misklassifizierten Beispiele minimieren.

- Bestimmen Sie den Fehler eines Splits, abhängig von seiner Position auf der x-Achse (bzw. y-Achse). Den Fehler können Sie durch die Größe der Fläche der Region von falsch klassifizierten Beispielen bestimmen. Wo werden die Splits in etwa gesetzt werden? Was ist der erwartete Fehler dieses Algorithmus?

  Lösung: Decision Stumps sind Entscheidungsbäume mit nur einem Knoten. Daher kann ein Decision Stump in diesem Szenario entweder einen Wert von x oder von y prüfen. Mögliche Decision Stumps wären also beispielsweise $x > 0.3 \rightarrow + \; else \; -$, $x > 0.7 \rightarrow + \; else \; -$ oder auch $y > 0.2 \rightarrow + \; else \; -$. Zu beachten ist, dass die Anzahl der misklassifizierten Beispiele minimiert werden soll, woraus folgt, dass im Decision Stump immer beim Operator $>$ der Wert $+$ vorhergesagt werden muss (da die Fläche der positiven Beispiele dort immer größer ist). Die Trennung ist nun nicht genau spezifiziert. Es muss beispielsweise nicht der Abstand zu den positiven und negativen Beispielen maximiert werden, sondern es sollen nur möglichst misklassifizierte Beispiele vermieden werden. Die folgende Grafik veranschaulicht die Situation:



Der Decision Stump der nach y klassifiziert ist blau gezeichnet und der der nach x klassifiziert ist rot markiert. Die schwarzen Werte entsprechen den jeweiligen Flächeninhalten der Quadrate. Beide Klassifizierungsmethoden benutzen in der Grafik als Trennlinie den Wert 0.5. An diesem Wert wird für beide Klassifizierer der kleinste Fehler erreicht (bei x und y ist jeweils die Hälfte des *linken oberen* und *rechten unteren* Quadrats falsch klassifizert, wobei bei x *oben links* die rechte Hälfte und bei y die linke Hälfte, sowie bei x *unten rechts* die linke Hälfte und bei y die rechte Hälfte fehlerhaft vorhergesagt werden). Daher ist der Fehler unabhängig davon welchen Decision Stump man verwendet. Die Summe der beiden Hälften ist genau der Fehler:

Fehler $= 1/4$

Nimmt man den Punkt $(0, 1)$, so werden bei der Decision Stump für x alle Beispiel positiv klassifiziert und bei der für y alle negativ. Das entspricht dem schlechtesten Fall, da die Hälfte falsch klassifiziert wird, der Fehler also $1/2$ ist.

Aus diesem Grund gilt: $1/4 \leq Fehler \leq 1/2$.

Um den Fehler nun in Abhängigkeit des Punktes $(x, 1 - x)$ anzugeben, addiert man einfach beide Flächeninhalte (die Hälfte der *linken oberen* und die Hälfte der *rechten unteren* Fläche) auf:

Fehler $= x^2/2 + (1-x)^2/2$

- Nehmen Sie an, sie verwenden Bagging, um die Performanz zu verbessern. Überlegen Sie sich, wie die Classifier der einzelnen Iterationen aussehen werden.

  Lösung: Die Idee bei Bagging ist, immer ein Sample (einen Teil) der Trainingsdaten zu verwenden, auf diesen einen Klassifizierer zu lernen, dann die Daten wieder zu den anderen zurückzulegen und wieder ein zu Sample ziehen, usw. Am Ende bekommt man m Klassifizierer heraus (wobei man darauf achten sollte, dass m ungerade ist) und führt ein simples Voting durch.

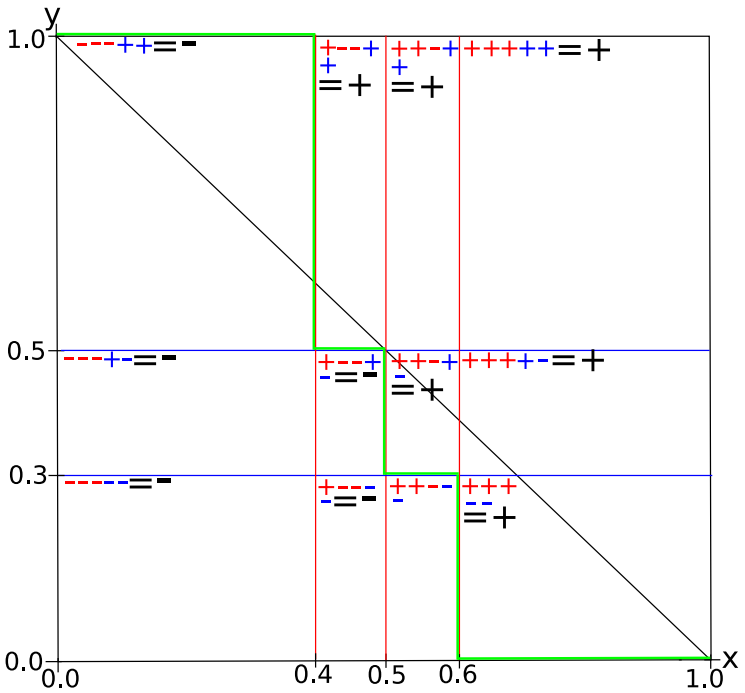
  Die Klassifizierer der einzelnen Iterationen sind wie bereits oben erwähnt Linien, die entweder parallel zur y-Achse sind (wenn x abgefragt wird) oder parallel zur x-Achse wenn y abgefragt wird. Da die Daten gleichverteilt sind, wird üblicherweise die Linie im Bereich von 0.5 liegen (da der Lerner versucht den Fehler zu minimieren). Wenn das Sample aber ungünstig gewählt ist, können natürlich auch andere Linien entstehen.

- Überlegen Sie sich grafisch, wie der Gesamt-Klassifizierer nach einigen wenigen Iterationen aussehen muß.

  Lösung: Angenommen es gilt $m = 5$ und die 5 Klassifizierer sind wie folgt:

  1. $x > 0.5 \rightarrow + \; else \; -$
  2. $x > 0.6 \rightarrow + \; else \; -$
  3. $x > 0.4 \rightarrow + \; else \; -$
  4. $y > 0.5 \rightarrow + \; else \; -$
  5. $y > 0.3 \rightarrow + \; else \; -$

  wobei die Klassifizierer, die nach x entscheiden wieder rot und die die nach y entscheiden wieder blau markiert sind.



In der Grafik ist diese Situation (nach 5 Iterationen von Bagging) veranschaulicht. Jeder Klassifizierer gibt eine Stimme ab (die drei die nach x entscheiden also 3 "rote" Stimmen und die 2 die nach y entscheiden 2 "blaue" Stimmen). Dann wird die Klasse (schwarz) vorhergesagt, die die meisten Stimmen erhält. Wie man sehen kann, ergibt sich eine treppenförmige Annäherung an das zu lernende Konzept wobei im Bereich um 0.5 die Granularität (die Länge einer "Treppenstufe") geringer ist als am Rand, da bei gleichverteilten Daten häufiger Klassifizierer um diesen Wert gelernt werden. Bei einer ausreichenden Iterationsanzahl sollte das Konzept hinreichend mit Bagging gelernt werden können.

- Wenn Sie statt der Decision Stumps einen Lernalgorithmus haben, der völlig zufällig einen Split auswählt, und dann die beiden Blätter so markiert, daß der Fehler minimiert wird. Was ist der erwartete Fehler dieses Algorithmus? Wenn Sie mit diesem Algorithmus mehrere Theorien lernen und diese kombinieren, erwarten Sie bessere oder schlechtere Ergebnisse als mit Bagging?

  Lösung: Zur Berechnung des erwarteten Fehlers ist die durchschnittliche Fläche unter der Kurve gesucht. Die Funktion des Fehlers für den Punkt $(x, 1-x)$ wurde bereits im Aufgabenteil 1 berechnet:

  $f(x) = {x^2}/{2} + {(1-x)^2}/{2}$

  Gesucht ist also nun $\frac{\int_0^1 \frac{x^2}{2} + \frac{(1-x)^2}{2}\, dx}{1}$.

$\frac{\int_0^1 \frac{x^2}{2} + \frac{(1-x)^2}{2}\ dx}{1} = \frac{x^3}{3} - \frac{x^2}{2} + \frac{x}{2}|_0^1 = \frac{1}{3} - \frac{1}{2} + \frac{1}{2} = 1/3$

Der erwartete Fehler beträgt folglich $1/3$.

Da man keine Aussage darüber treffen kann, welche Splits von dem zufälligen Algorithmus ausgewählt werden, kann man nur eine Abschätzung machen: Mit ausreichend vielen Iterationen werden beide Algorithmen in etwa gleich gut abschneiden, wobei Bagging in den Randbereichen jeweils schlechter abschneidet (wie im vorherigen Aufgabenteil erwähnt) und der zufällige Algorithmus diese Ungenauigkeit auf die gesamte Diagonale $(x, 1-x)$ ausdehnt, also einen konstanten Fehler aufweist. Da der Bereich am Rand, der fehlerhaft klassifiziert ist, bei Bagging sehr groß ist, also einen hohen Einfluß auf den gesamten Fehler hat und der Fehler des zufälligen Algorithmus konstant gering ist, könnte man abschätzen, dass der Gesamtfehler des zufälligen Algorithmus kleiner als der von Bagging ist. Grundsätzlich gilt aber, dass je nachdem welche Gewichtung auf die verschiedenen Bereiche der Diagonalen vergeben wird, der jeweilige Algorithmus zu bevorzugen wäre.

Aufgabe 2

Rechnen Sie das AdaBoost-Beispiel aus der Vorlesung nach. Verwenden Sie für die einzelnen Datenpunkte die folgenden Koordinaten (x, y, Klasse):

```
1,  5,  +          3,  1,  -
2,  2,  +          4,  6,  -
5,  8,  +          7,  4,  -
6, 10,  +          9,  3,  -
8,  7,  +         10,  9,  -
```

Der Basis-Lerner wählt unter allen möglichen Splits jenen aus, bei dem die Gesamtsumme der Gewichte der falsch klassifizierten Beispiele minimiert wird.

Lösung: In jeder Iteration sind jeweils 5 horizontale (y $\leq$ Wert) und 5 vertikale Splits möglich. Für jeden dieser Splits addieren wir die Gewichte der Beispiele auf, die durch den Split falsch klassifiziert werden. Anschließend wählen wir denjenigen Split aus, der die geringste Summe aufweist. Wir berechnen danach die Gewichtung α_m des aus dem Split resultierenden Klassifizierer (Decision Stump). Die Gewichte der Beispiele werden unter Verwendung der Gewichtung α_m erhöht bzw. gesenkt, falls sie falsch bzw. richtig klassifiziert werden. Am Ende jeder Iteration werden die Gewichte der Beispiel so normiert, daß ihre Summe eins ergibt. Die Folien 11-15 illustrieren diese Vorgehensweise.

Hinweis: *Bei den berechneten Fehlern können abhängig von der verwendeten Methode (Berechnung einer oder beider Tabellenspalte(n)) Abweichungen auftreten, da die Gesamtsumme der Beispielsgewichte bedingt durch die Rundung der Werte selten 1 beträgt.*

Beginnen wir nun mit den Berechnungen. Am Anfang haben alle Beispiele das Gewicht $1/10$ (10 Beispiele). Betrachten wir nun zuerst alle vertikalen Splits.

	Fehler	
Wert	x $\leq$ Wert $\Rightarrow$ +	x > Wert $\Rightarrow$ +
1	$4/10$	$6/10$
2	$3/10$	$7/10$
3	$4/10$	$6/10$
4	$5/10$	$5/10$
5	$4/10$	$6/10$
6	$3/10$	$7/10$
7	$4/10$	$6/10$
8	$3/10$	$7/10$
9	$4/10$	$6/10$
10	$5/10$	$5/10$

Entsprechend erhalten für die horizontalen Splits die folgenden Fehler.

	Fehler	
Wert	y $\leq$ Wert $\Rightarrow +$	y $>$ Wert $\Rightarrow +$
1	$^6/_{10}$	$^4/_{10}$
2	$^5/_{10}$	$^5/_{10}$
3	$^6/_{10}$	$^4/_{10}$
4	$^7/_{10}$	$^3/_{10}$
5	$^6/_{10}$	$^4/_{10}$
6	$^7/_{10}$	$^3/_{10}$
7	$^6/_{10}$	$^4/_{10}$
8	$^5/_{10}$	$^5/_{10}$
9	$^6/_{10}$	$^4/_{10}$
10	$^5/_{10}$	$^5/_{10}$

Betrachten wir beide Tabellen, sehen wir, daß es 4 Splits mit minimalen Fehler gibt (rot markiert). Wir entscheiden uns für den zuerst gefundenen Split ($x \leq 2 \Rightarrow +$). Berechnen wir nun das Gewicht des resultierenden Klassifizierers. Hierfür benötigen wir zunächst den Fehler err_1:

$$err_1 = \frac{3}{10}$$

Hiermit können wir nun das Gewicht α_1 des Klassifizierers berechnen:

$$\alpha_1 = \frac{1}{2}\log\left(\frac{1 - err_m}{err_m}\right) = \frac{1}{2}\log\left(\frac{7}{3}\right) \approx 0,424$$

Damit ergeben sich die folgenden Faktoren, mit denen die einzelnen Gewichte multipliziert werden:

$$w_i \leftarrow \begin{cases} w_i \cdot e^{-\alpha_1} \approx 0,654, & \text{falls } w_i \text{ korrekt klassifiziert wird} \\ w_i \cdot e^{\alpha_1} \approx 1,528, & \text{falls } w_i \text{ falsch klassifiziert wird} \end{cases}$$

Da sieben Beispiele korrekt und drei falsch klassifizert wurden, erhalten wir:

$$3 \cdot 1,528 + 7 \cdot 0,654 = 9,162$$

als Gesamtsumme der Gewichte. Diese wollen wir auf eins normieren, aus diesem Grund teilen wir alle Gewichte durch 9,162. Damit erhalten wir folgende Gewichte:

$$w_i = \begin{cases} 0,071, & \text{falls } w_i \text{ korrekt klassifiziert wird} \\ 0,167, & \text{falls } w_i \text{ falsch klassifiziert wird} \end{cases}$$

und folgende Tabelle:

x	y	Gewicht
1	5	0,071
2	2	0,071
3	1	0,071
4	6	0,071
5	8	0,167
6	10	0,167
7	4	0,071
8	7	0,167
9	3	0,071
10	9	0,071

Suchen wir nun den nächsten Split. Wir betrachten zuerst vertikale Splits

	Fehler	
Wert	x $\leq$ Wert $\Rightarrow +$	x $>$ Wert $\Rightarrow +$
1	0,572	0,428
2	0,501	0,499
3	0,572	0,428
4	0,643	0,357
5	0,476	0,524
6	0,309	0,691
7	0,38	0,62
8	0,213	0,787
9	0,284	0,716
10	0,355	0,645

und anschließend die horizontalen.

	Fehler	
Wert	y $\leq$ Wert $\Rightarrow +$	y $>$ Wert $\Rightarrow +$
1	0,286	0,714
2	0,643	0,357
3	0,286	0,714
4	0,215	0,785
5	0,286	0,714
6	0,215	0,785
7	0,382	0,618
8	0,549	0,451
9	0,478	0,522
10	0,645	0,355

Der beste Split ist $x \leq 8 \Rightarrow +$. Das heißt die Punkte (3, 1), (4, 6) und (7, 4) werden falsch und alle anderen richtig klassifiziert. Demnach hat err_2 den folgenden Wert:

$$err_2 \approx 0,213$$

Mit err_2 können wir α_2 berechnen:

$$\alpha_2 = \frac{1}{2} \log \left(\frac{0,787}{0,213} \right) \approx 0,652$$

Berechnen wir nun die Faktoren e^{α_2} bzw. $e^{-\alpha_2}$, mit denen wir die Gewichte multiplizieren:

$$e^{-\alpha_2} = 0,521$$
$$e^{\alpha_2} = 1,919$$

Damit ergibt sich die folgende Tabelle:

	Neues Gewicht	
Altes Gewicht	Korrekt klassifiziert	falsch klassifiziert
0,071	0,037	0,136
0,167	0,087	0,32

Multiplizieren wir die Gewichte der korrekt (blau) und falsch (rot) klassifizierten Beispiele mit den entsprechenden Faktoren und normieren diese, erhalten wir folgende Gewichte:

			Neues Gewicht	
x	y	Altes Gewicht	Nicht normiert	Normiert
1	5	0,071	0,037	0,045
2	2	0,071	0,037	0,045
3	1	0,071	0,136	0,166
4	6	0,071	0,136	0,166
5	8	0,167	0,087	0,106
6	10	0,167	0,087	0,106
7	4	0,071	0,136	0,166
8	7	0,167	0,087	0,106
9	3	0,071	0,037	0,045
10	9	0,071	0,037	0,045

Für den letzten Klassifizierer betrachten wir wiederum zuerst die vertikalen Splits

	Fehler	
Wert	$x \leq$ Wert $\Rightarrow +$	$x >$ Wert $\Rightarrow +$
1	0,363	0,637
2	0,318	0,682
3	0,484	0,516
4	0,650	0,350
5	0,544	0,456
6	0,438	0,562
7	0,604	0,396
8	0,498	0,502
9	0,543	0,457
10	0,588	0,412

und anschließend die horizontalen

	Fehler	
Wert	y $\leq$ Wert $\Rightarrow +$	y $>$ Wert $\Rightarrow +$
1	0,574	0,426
2	0,529	0,471
3	0,574	0,426
4	0,740	0,260
5	0,695	0,305
6	0,861	0,139
7	0,755	0,245
8	0,649	0,351
9	0,694	0,306
10	0,588	0,412

Der beste Split ist $y \geq 6 \Rightarrow +$. Berechnen wir nun das Gewicht des Klassifizierers. Es gilt

$$err_3 = 0,139$$

und damit

$$\alpha_3 = 0,912.$$

Damit haben wir die drei Klassifizierer (und deren Gewichte) des Beispiels aus der Vorlesung berechnet. Eine Illustration des resultierenden Klassifizierers befindet sich wie bereits erwähnt auf Folie 15.