WEKA Projekt

Fabian Langguth, Sebastian Koch

Wintersemester 10/11

Aufgabe 1 Regellernen

Für diese Aufgabe benutzen wir die Datensätze *glass, iris* und *splice. glass* wurde für den Prism-Learner mit dem Discretize-Filter verwendet (*java weka.filters.supervised.attribute.Discretize -i glass.arff -o glass_nom.arff -R 1,2,3,4,5,6,7,8,9 -c last*).

Anzahl der Regeln

	glass	iris	splice
Conjunctive Rule	1	1	1
JRip	8	4	14
Prism	63	16	3176

Gesamtanzahl der Bedingungen

	glass	iris	splice
Conjunctive Rule	2	1	1
JRip	18	3	55
Prism	385	51	3176

Anzahl der vorhergesagten Klassen

	glass	iris	splice
Conjunctive Rule	1	1	1
JRip	6	3	3
Prism	6	3	3

Prism erzeugt bei allen Datensätzen die meisten Regeln und Bedingungen. *Conjunctive Rule* erzeugt fast immer nur eine Regel mit einer Bedingung.

Eine Default Rule existiert nur bei JRip. Dort wird als Defaultklasse üblicherweise die Klasse gewählt, die am häufigsten im Datensatz vorkommt. Um zukÃijnftige Daten zu klassifizieren ist das im Hinblick auf relative HÃďufigkeiten die sinnvollste Entscheidung.

Der Datensatz *iris* lässt sich am einfachsten lernen, da man hier alle Algorithmen besonders wenig Regeln und besonders wenig Bedingungen benötigen.

Auf dem Datensatz *Contact Lenses* erzeugt *JRip* 3 Regeln, wo bei einer davon der Default-Rule entspricht. *Prism* erzeugt hingehen 9 Regeln. Die Anzahl der Bedingungen ist ebenfalls höher für die von *Prism* gefundenen Regeln. Daraus lässt sich folgern, dass *JRip* vermutlich besser veralgemeinert. Dieser Unterschied entsteht hier vorallem durch dir verschiedenen Performance Maße. Da *Prism* Precision verwendet, wird lediglich auf eine hohe Anzahl von korrekt klassifizierten Beispielen optimiert. Deshalb tendiert der Algorithmus eher zum Overfitting und erzeugt viele Regeln mit vielen Bedingungen. *JRip* hingegen zieht durch das Gain Maß auch die Anzahl der Regeln in Betracht und versucht dadurch nur wichtige Regeln zu lernen um Overfitting zu vermeiden.

Aufgabe 2 Evaluation von Regellernern

a

Accuracy

Datensatz	1x5	1x10	1x20	LOO	Trainingsmenge
glass	67.3	61.8	60.7	61.7	85.98
iris	92.0	88.0	96.0	93.3	96.0
audiology	67.3	66.4	69.9	69.9	76.1
ionosphere	89.2	92.0	90.0	89.2	100
yeast	56.5	57.7	57.7	59.4	67.8

Die geschätzte Genauigkeit wird auf der gesamten Trainingsmenge immer höher sein als auf Echtdaten, da die Regeln speziell für diese Daten trainiert wurden. In der Praxis sollte man daher Cross-Validation verwenden um sinnvolle Genauigkeiten zu erhalten. Die Qualität der Abschätzung wird besser, je mehr Folds man für die Cross-Validation verwendet. Am genauesten sollte die Leave-One-Out Methode sein, sie benötigt jedoch auch die meiste Zeit. In der Praxis sollte man darauf Rücksicht nehmen.

b

Datensatz	10x10
glass	67.3
iris	92.0
audiology	67.3
ionosphere	89.2
yeast	56.5

Die Änderung des seeds hat keine signifikante Veränderung der Abschätzung bewirkt. Im Allgemeinen sollte die Auswahl der Random-Seeds auch keinen Einfluss auf die Abschätzung haben.

c

Datensatz	Validierungsmenge
glass	73.1
iris	93.3
audiology	67.3
ionosphere	84.6
yeast	56.1

Die Genauigkeit der Evaluierungsmethoden hängt stark vom entsprechenden Datensatz ab. Im Allgemeinen konnte keine Methode immer gute Abschätzungen liefern.

Aufgabe 3 ROC-Kurven

Datensatz: glass

Fläche unter ROC Kurve

Regellerner	*build wind float*	*containers*	*tableware*
J48	0.81	0.87	0.93
Naive Bayes	0.71	0.84	0.98

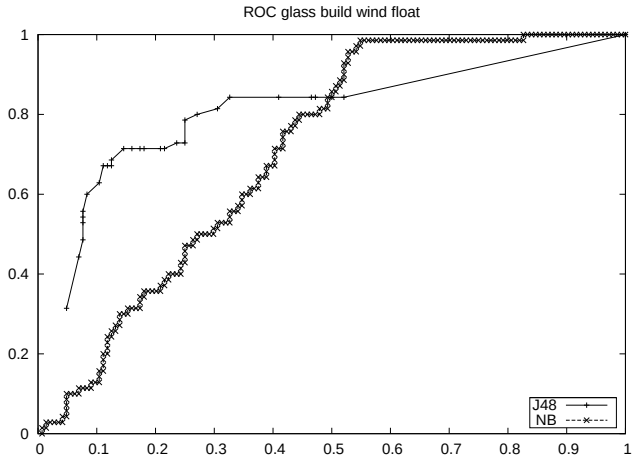

Figure 1: ROC-Kurve für *Naive Bayes* und *J48* über das Attribut *build_wind_float*

Bis auf wenige Klassen hat *J48* eine höhere Fläche unter der ROC Kurve. Anhand der ROC Kurven kann man also sagen, dass für eine uniforme Klassenverteilung und für eine Verteilung mit mehr negativen als positiven Beispielen *J48* bessere Ergebnisse liefert. Bei Besonders vielen positiven Beispielen könnte der Naive Bayes Klassifizierer bessere Ergebnisse erzielen.

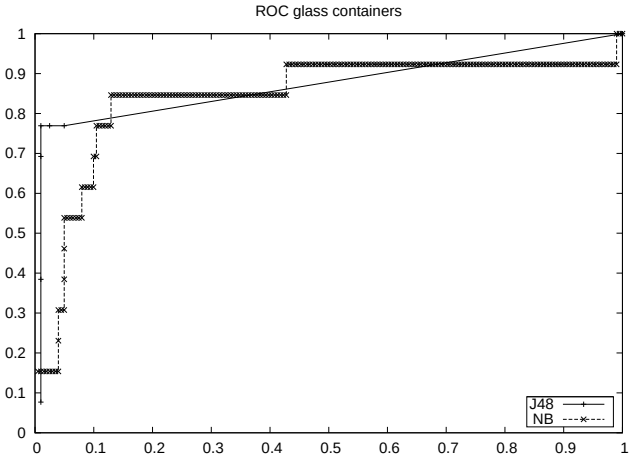

Figure 2: ROC-Kurve für *Naive Bayes* und *J48* über das Attribut *containers*

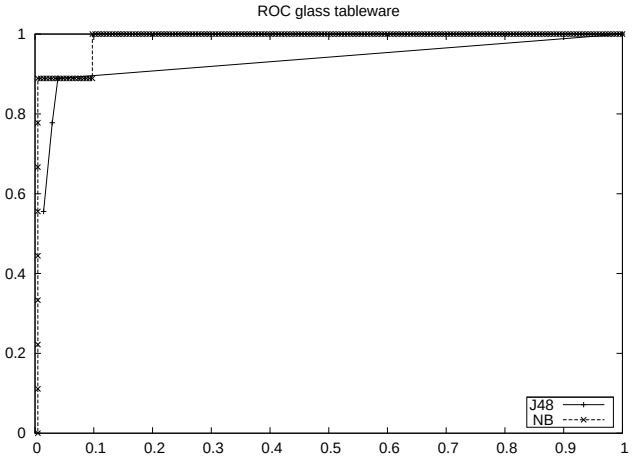

Figure 3: ROC-Kurve für *Naive Bayes* und *J48* über das Attribut *containers*

Aufgabe 4 Entscheidungsbäume

Datensätze: glass (discretize filter), kr-vs-kp

Fläche unter ROC Kurve

Regellerner	*bwf*	*bwn*	*vwf*	*cont*	*table*	*head*
J48 - unpruned	0.73	0.70	0.71	0.87	0.92	0.92
J48 - pruned	0.77	0.73	0.73	0.86	0.87	0.84
ID3	0.62	0.70	0.59	0.81	0.77	0.88

Table 1: glass

Regellerner	*won*	*nowin*
J48 - unpruned	1.0	1.0
J48 - pruned	1.0	1.0
ID3	1.0	1.0

Table 2: kr-vs-kp

Accuracy

Regellerner	*glass*	*kr-vs-kp*
J48 - unpruned	57.94	99.41
J48 - pruned	57.94	99.44
ID3	50.47	99.69

Grösse der entstandenen Bäume

Regellerner	*size of the tree*	*number of leaves*
J48 - unpruned	221	199
J48 - pruned	81	73
ID3	550	496

Table 3: glass

Regellerner	*size of the tree*	*number of leaves*
J48 - unpruned	82	43
J48 - pruned	59	31
ID3	95	49

Table 4: kr-vs-kp

Betrachtet man die Fläche unter der ROC Kurve so hat *ID3* im Allgemeinen schlechtere Werte. Dies spiegelt sich auch in der Accuracy wieder. Außerdem erzeugt *ID3* immer einen größeren Baum. Man kann also deutlich erkennen, das ein großer Baum schlecht veralgemeinert und somit auch schlechtere Performancewerte erzielt. Das Pruning von *J48* erzeugt einen kleineren Baum, der allerdings nicht deutlich bessere Performance

liefert. Dieser Effekt ist dabei wahrscheinlich stark abhängig von den von uns gewählten Datensätze.

Aufgabe 5 Nearest Neighbour

Accuracy

k-NN	*glass*	*kr-vs-kp*
k = 1	59.35	96.28
k = 3	57.01	96.50
k = 5	58.88	96.03
k = 7	57.01	95.40
k = 9	56.07	95.24
k = 11	56.07	95.06

Höchste cross validation performance liegt bei $k = 1$ bzw $k = 3$. Damit ist der *k-NN* Klassifizierer für den *glass* Datensatz besser und für den *kr-vs-kp* Datensatz schlechter als die Entscheidungsbaumlerner.

Aufgabe 6 Regressionsbäume

Mean Absolute Error

Datensatz	*R P MAE*	*R U MAE*	*M P MAE*	*M U MAE*
auto-price	2096.37	2075.07	1403.20	1466.56
concrete	6.77	6.48	4.27	4.74
housing	3.29	3.20	2.39	2.50
stock	1.19	1.17	0.67	0.67
winequality	0.55	0.53	0.51	0.55

Table 5: R: *Regression-Trees* or M: *Model-Trees*
U: *unpruned* or P: *pruned*

Root Mean Squared Error:

Datensatz	*R P RMSE*	*R U RMSE*	*M P RMSE*	*M U RMSE*
auto-price	3336.37	3287.12	2094.59	2171.16
concrete	8.68	8.33	5.89	6.36
housing	4.82	4.72	3.71	3.75
stock	1.60	1.59	0.93	0.94
winequality	0.72	0.70	0.68	0.71

Table 6: R: *Regression-Trees* or M: *Model-Trees*
U: *unpruned* or P: *pruned*

Aus den Tabellen können wir ablesen, dass bei Regression-Tasks keine Verbesserung bringt. *Model-Trees* haben auf allen Datensätze einen kleineren Fehler als *Regression-Trees*. Verwendet man Pruning bei *Model-Trees* liefert es eine leichte Verbesserung.

M5P liefert einen dem aus der Übung ähnlichen Baum: Auf den ersten beiden Ebenen wird an denselben Attributen gesplittet. Danach erzeugt *M5P* direkt Blätter und löst den Baum nicht feiner auf. Der *M5P*-Baum ist kleiner und wahrscheinlich allgemeiner.

Der *Root Mean Squared Error* des Baumes aus der Übung betrugt 0.75, der jetzt gelernte Baum hat einen *RMSE* von 0.64 auf den Testdaten. Dies bestätigt unsere Vermutung, dass der *M5P*-Baum allgemeiner ist.

Aufgabe 7 Ensemble-Lernen

Performance Normal

	yeast	vowel	vehicle	sick	abalone
J48	56.0	81.5	72.5	98.9	21.2

Performance Bagging

Iterations	yeast	vowel	vehicle	sick	abalone
10	60.8	90.4	76.6	98.7	23.1
20	61.0	91.3	75.9	98.9	23.2
50	62.0	91.9	76.2	98.9	23.5
100	61.3	92.5	76.0	98.8	23.5

Performance AdaBoost

Iterations	yeast	vowel	vehicle	sick	abalone
10	56.4	93.3	76.2	99.2	21.7
20	58.1	95.9	77.0	99.2	21.9
50	58.9	96.0	78.4	99.2	22.6
100	58.6	96.5	78.8	99.0	22.7

Performance Random Forests

Number of Trees	yeast	vowel	vehicle	sick	abalone
10	57.9	96.0	77.0	98.4	22.4
20	61.2	98.0	76.5	98.4	23.4
50	61.3	98.2	76.7	98.5	23.4
100	61.4	98.5	76.5	98.4	23.8

Generell verbessern alle 3 Ensemble Lerner die Performance gegenüber dem regulären *J48*. Bei *Bagging* bringt eine erhöhte Anzahl der Iteration nur sehr geringe Verbesserungen auf unseren Datensätze. Durch *Boosting* kann die Performance um mehrere Prozent steigen, wenn man die Zahl der Iterationen erhöht. Bei *Random Forests* hängt eine Steigerung der Performance durch eine erhöhte Anzahl von Trees stark vom Datensatz ab.

Aufgabe 8 Entdecken von Assoziationsregeln

Der Datensatz *adult* enthält sehr viele Attribute, die stark miteinander korrelieren, wie zum Beispiel *sex* und *marital-status*. Daher haben wir, um spannende Regeln zu finden sehr viele Attribute entfernt, und listen nur die Attribute auf, die wir zum Finden von Regeln verwendet haben.

Attribute: age, workclass, education, occupation, relationship, race, hoursperweek, native-country, class
Sortierung: lift

relationship = **Not-in-family** $\Longrightarrow$ **class** = **<=50K** (confidence: 0.9 lift: 1.18)

Attribute: workclass, education, occupation, race, hoursperweek, class
Sortierung: confidence

class = **>50K** $\Longrightarrow$ **race** = **White** (confidence: 0.91)

Da etwa 85 % der Teilnehmer der Studie **race** = **white** hatten, wurden sehr viele Regeln der Form $\Longrightarrow$ **race** = **white** gefunden. Entfernt man dieses Attribut, findet man noch weitere interessante Regeln.

Attribute: age, education, marital-status, occupation
Sortierung: confidence

age = **0** $\Longrightarrow$ **marital-status** = **Never-married** (confidence: 0.85)
age = **3** $\Longrightarrow$ **marital-status** = **Married-civ-spouse** (confidence: 0.62)

Wobei hier **age** wahrscheinlich nicht für das tasächliche alter steht, sondern für bestimmte Altersgrenzen. Junge Menschen sind nicht oft verheiratet. Alte Männer sind oft verheiratet.

Attribute: marital-status, relationship, class
Sortierung: confidence

relationship = **Own-child** $\Longrightarrow$ **class** = **<=50K** (confidence: 0.99)
class = **>50K** $\Longrightarrow$ **marital-status** = **Married-civ-spouse** (confidence: 0.85)

Erwachsene die ein eigenes Kind haben, verdienen eher wenig. Erwachsene die viel verdienen sind oft verheiratete Männer.

Aufgabe 9 Pre-Processing

Datensätze: ionosphere, iris, yeast, letter.

Accuracy

	ionosphere	iris	yeast	letter
J48 unfiltered	91.5	96.0	56.0	88.0
J48 filtered	89.2	94.0	59.1	78.6
FilteredClassifier	91.2	93.3	57.0	78.7

Number of Leaves

	ionosphere	iris	yeast	letter
J48 unfiltered	18	5	185	1226
J48 filtered	21	3	64	9624
FilteredClassifier	21	3	64	9624

Der *FilteredClassifier* liefert im Durchschnitt eine geringe Accuracy, die aber Vermutlich realistischer ist, da die Informationen für das Filtering ausschließlich aus dem Testteil der Cross-Validation gewonnen werden, und nicht aus dem Trainingsteil. Filtert man die Daten vor dem Lernen benutzt man Informationen aus dem gesamten Datensatz.

Die Grösse der Bäume hängt nicht von der Art des Filterings ab.