Analyse und Visualisierung Zusammenfassung

Maximilian Ortwein

5. Februar 2012

Inhaltsverzeichnis

1 Data Types

Nominal (z.B. Haarfarbe): Symbolisch, nicht numerisch wörter und so; Operationen sind nur überprüfung auf Gleich und ungleich

Ordinal (z.B. Schulnoten): Werte die Auf oder absteigend sortiert werden können bzw in einer reihenfolge. Operationen: Gleichheit, Größer Kleiner

Numeric: Zahlenwerte, Abstände sind von Bedeutung, Mathematische operationen sind möglich
Operationen:
Gleichheit, Größer kleiner
Interval Scale: +,- (Z.B. Datum)
Ratio scale: +,-,/,* (z.B. Größe)

2 Preprocessing

2.1 Fehlerarten

Random Error: wird als noise oder Rauschen bezeichnet, es beinflusst nicht den mittelwert aber die varianz um den durchschnitt
Systematical Error: WIrd als Bias bezeichnet, oder Verzerung bezeichnet, es Beeinflusst den Mittelwert

Missing Value: Wenn ein Wert exitistiert aber dieser in den Daten noicht vorhanden ist

Empty Value: Ein Wert der nicht Existiert

2.1.1 behandeln fehlender Werte

- Tupel igonrieren
- Globale Konstante einfügen
- Fehlenden Werte von Hand einfügen
- Mittelwert einfügen
- den Warscheinlichsten Wert einfügen

Eingefügte Werte sollten die Informationen nach Möglichkeit nicht beeinflussen

2.2 Data Cleaning

Rauschen entsteht z.b. durch messfehler, ungenaue Hardware und so weiter

2.2.1 Binning

Binning glättet die Daten oder stuft sie ab

equal-width Binning: N intervalle mit gleicher größe $weite = \frac{\{max-min\}}{N}$ ist einfach aber Ausreißer können das ergebnis stark beeinflussen.

equal-depth: N intervalle mit nahezu der gleichen Anzahl an Einträgen

2.2.2 Interpolation / Approximation

Interpolation:
- Polynomial Funktionen
- Teilweise Polynomial
- Orthogonal Polynomial
- Trigonometrische funktionen

Approximation:
- Regressionen

2.2.3 Regression

Lineare Regression versucht eine Funktion zu finden die möglichst nah an allen Werten ist. Lineare Funktion: $y = a + bx$

b Bestimmen:

$$b = \frac{\sum_{i=1}^{n}(x_i - x_r)(y_i - y_r)}{\sum_{i=1}^{n}(x_i - x_r)^2}$$

x_r ist der Mittelwert aller x und y_r ist der Mittelwert aller y

$a = y_r - bx_r$

2.3 Normalisierung

Der versuch alle Daten zwischen 0 und 1 mappen, Kann durch folgende 33 funktionen erfolgen:

- Lineares Mapping: $f(v) = \frac{v-min}{max-min}$
- Wurzel-Mapping: $f(v) = \frac{\sqrt{v}-\sqrt{min}}{\sqrt{max}-\sqrt{min}}$

- Logarithmische Mapping: $f(v) = \frac{\ln(v)-\ln(min)}{\ln(max)-\ln(min)}$

2.4 Segmentation

Ziel: finde eine natürliche Aufteilung in k Cluster und Rauschen. Das kann Manuell oder Automatisch erfolgen.

- Automatisch:
-k -means
- Linked-Based

2.4.1 k-means

1. partitionieren der daten - means als zentren wählen
2. jedes objekt dem naheliegensten zentrum zuweisen
3. das zentrum an den neuen mean der zugehörenden objekte verschieben
4. bei 2 weiter machen, bis sich nciht mehr ändert

2.5 Data Reduction

Methoden:
- Teilmengen bestimmen z.B. durch SQL
- Anzahl der Dimensionen verringern
Entweder durch entfernen von unwichtigen oder redundanten attributen und dimensionen oder durch reduktion von dimensionen durch zum beispiel PCA oder MDS u.s.w.

2.5.1 Sampling

- Nicht Warscheinlichkeitsbasiertes sampling: Eine nicht zufällig gewählte Baisis wird ausgesucht.

- Warscheinlichkeitsbasiertes Sampling: Die basis wird zufällig ausgewählt, so das jedes element die gleiche warscheinlichkeit hat gewählt zu werden.
Arten:
- Einfaches zufallsbasiertes sampling = EInfache Zufallsauswahl aus den Daten
- Systematisches zufallsbasiertes Sampling: Es wird darauf geachtet, das vorbedingungen erfüllt sind
- Schichtweises zufallsbasiertes sampling: Es werden Gruppen gebildet in denen zufällig gesampelt wird
- Cluster zufallsbasiertes sampling: In erzeugten Clustern wird zufällig gesampled - Verzerttes Sapmpling

2.6 Dimensionen Reduktion

2.6.1 Hauptprobleme

- Objekte werden durch eine Große Anzahl von Attributen dargestellt
- Die Daten sind schwer zu Visualisieren
- Unwichtige attribute können die Genauigkeit des Algorithmus verringern

2.6.2 Wie werden Dimensionen Reduziert?

Durch Projektion, dabei werden die wichtigsten attribute identifiziert um den Prozess zu vereinfachen ohne Qalitätsverlust und die wichtigsten zwei bis drei attribute direkt zu visualisieren.

2.6.3 PCA- Hauptkomponenten Analyse

Ziel ist es Dimensionen zu veringern und versteckte Faktoren in den Daten zu finden.
Es werden die unwichtigsten Eigenvektoren weggelassen und dadurch die Dimensionen reduziert
PCA kann nur auf Numerische Daten angewendet werden
Wird benutz wenn Daten mit möglichst geringer Korrelation Dargestellt werden sollen.
Varianz wird maximiert.
PCA ist nicht gut für Daten mit gleichmäßiger abweicheichung
Nicht Robust gegen Ausreißer
Ist Robust gegen Rauschen
Ist Robust gegen Rotationenn im Datenraum

- Eigenwerte:

$$\begin{pmatrix} 4 & 2 \\ 3 & 3 \end{pmatrix}$$

$$Determinante(A - \lambda I) = \begin{pmatrix} 4-\lambda & 2 \\ 3 & 3-\lambda \end{pmatrix}$$

$$= (4-\lambda)(3-\lambda) - 6$$
$$= \lambda^2 - 7\lambda + 6$$
$$= (\lambda - 1)(\lambda - 6) = 0$$

Eigenwerte sind die 0-Stellen der Funktion. In diesem Fall $\lambda_1 = 1$ und $\lambda_2 = 6$

- Eigenvectoren:

Exemplarisch für $\lambda = 1$:

$$\begin{pmatrix} 3 & 2 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Durch Multiplikation ergibt sich: $3x_1 + 2_x 2 = 0$ und Damit als einen Eigenvektor $\begin{pmatrix} 2 \\ -3 \end{pmatrix}$

3 Klassifikation

3.1 Train and Test

- Es werden Trainingsdaten verwendet um Klassifikatoren aufzubauen
- Durch Testdaten werden die Klassifikatoren Validiert

3.2 m-fold cross validation

- Daten werden in m Teilmengen mit gleicher Größe aufgeteilt
- m Klassifikatoren werden trainiert in dem jeder Klassifikator mit m-1 der Teilmengen als Trainingsdaten trainiert wird. Als Testdaten verwendet man die übrig gebliebene Teilmenge
- Die Auswertung der m Klassifikatoren wird kombiniert

Kriterien für die Auswertung

- Genauigkeit der Klassifikation
- Nachvollziehbarkeit
- Effizienzder Model-Konstruktion und Anwendung
- Eigung für große Datenmengen
- Robustheit

3.2.1 Klassifikationsgenauigkeit

$$G_{TE}(K) = \frac{\text{Anzahl Korrekt klassifizierter Objekte}}{\text{gesammtzahl aller Objekte}}$$

3.2.2 Klassifikations-Fehler

$$F_{TE}(K) = \frac{\text{Anzahl Falsch klassifizierter Objekte}}{\text{gesammtzahl aller Objekte}}$$

3.2.3 Confusion Matrix

	Predicted as positiv	Predicted as Negatie
Actually Positive	True Positive (TP)	False Negative (FN)
Actually negative	False Positive (FP)	True Negative (TN)

$\text{Precision}(K) = \frac{|TP|}{|TP|+|FP|}$
$\text{Recall}(K) = \frac{|TP|}{|TP|+|NP|}$

Beide Maße Stehen im Umgekehrten Verhältnis zueinander

$\text{F-Measure}(K) = \frac{2 \cdot \text{Precision(K)} \cdot \text{Recall(K)}}{\text{Precision(K)} + \text{Recall(K)}}$
F-Measure sollte möglichst hoch sein

3.2.4 True Classification Error

Es werden zufällig n Elemente aus einer Menge gewählt und Klassifiziert, Der Anteil an Falscher Klassifikationen ist der True Classification Error

3.3 Entscheidungsbäume

Werden durch den vergleich von Attributen aufgebaut
Die Blätter entsprechen den Gegebenen Klassen.

Bäume werden entweder durch Trainingsdaten oder durch die Top-Down strategier aufgebaut

Indem man von der Wurzel bis zu einer Klasse durchläuft kann man Datensätze klassifizeren

3.3.1 Construction Algorithmus

Am anfang gehören Alle daten Zur wurzel. Man wählt ein Wichtiges Attribut und führt einen sinnvollen Split des Attributs durch. Die Trainingsdaten werden dann nach den Splits aufgeteilt, Das wendet man auf alle Weiteren Attribute der Teilmenge an

Der Algorithmus terminiert, wenn keine Attribute mehr zum spliten vorhanden sind, oder die Meisten Daten eines Knotens zur selben Klasse gehören.

3.3.2 typen von Splits

- Kategorisch(== oder !=) es sind sehr viele Teilmengen möglich
- Numerisch (==, !=, $<$, $>$,$\leq$ oder $\geq$), viele Splits möglich

T ist Datenmenge p_i ist häufigkeit des vorkommens einer klasse c_i in T

3.3.3 Information Gain

Entropie: $\text{entropy(T)} = -\sum_{i=1}^{k} p_i \cdot \log_2 p_i$

Information Gain: $\text{InformationGain(T,A)} = \text{entropy(T)} - \sum_{i=1}^{m} \frac{|T_i|}{|T|} \cdot \text{entropy}(T_i)$

3.3.4 Gini-Index

$\text{gini(T)} = 1 - \sum_{j=1}^{k} p_j^2$

Kleiner Gini-Index = geringe Unreinheit
Großer Gini-Index= hohe Unreinheit

Gini-Index für ein bestimmtes Attribut in Abhängigkeit der Klasse:
$\text{gini}_a(T) = \sum_{i=1}^{m} \text{gini}(T_i)$

3.3.5 Overfitting

Wenn die genauigkeit der Klassifikation zu hoch ist, hat man zwar ein gutes Ergebnis für die trainingsdaten aber das Erbenis wird für die Testdaten schlechter

Wie verhindern?
- Entfernen von fehlerhaften Trainingsdaten
- Trainingsdatengröße geeignet Auswählen
- Minimal Anzahl an Trainingsdaten die zu einem Blatt gehören geeignet auswählen (minimal Support)
- Auswahl geeigneter minimal Confidence (Meist vorkommende Klasse soll geringen anteil an Blättern haben)
- Säubern des Entscheidungsbaums (zweige weglassen)
- Cross-Validation (Daten Aufteilen in Trainings und Testdaten (9 zu 1 z.B.))
- Anzahl der betrachteten Attribute verringern

3.3.6 Säuberung zum verringern von Fehlern

Train-and-Test verfahren:
Zweige abschneiden, Testen, dann wieder zweige entfernen und wieder testen und so weiter

3.3.7 Minimaler Aufwand zum Säubern der Daten

Cross-Validation: Anwendbar wenn nur wenige Daten vorhanden sind.
pruning: Man säubert den Entscheidungsbaum mit den Trainingsdaten und kann den Klassifikationsfehler nicht als Qualitätsmaß nutzen

Kleinere Entscheidungsbäume sind tendetiell besser für noch nicht dagewesene Daten.

3.3.8 C4.5 vorteile gegenüber ID3

- Man kann mit Zahlenwerten Arbeiten
- Gain-Ratio = Modifizierte Split-Kriterien
- Regeln extrahieren - Stopt das verarbeiten von Nodes die keinen Gewinn bringen - Nutzen von post-prunning Methoden - Windowing

3.3.9 Gain-Ratio

Der Informationsgehalt ist durch Tests mit vielen Ergebnissen verzerrt durch die Gain Ratio wird versucht das zu beheben.

$\text{SplitInformation(S,A)} = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \cdot \log_2 \frac{|S_i|}{|S|}$

$\text{GainRatio(S,A)} = \frac{\text{InformationGain(S,A)}}{\text{SplitInformation(S,A)}}$

3.4 Bayesian Classification

Für jedes zu klassifizierende Objekt wird die Warscheinlichkeit berechnet.
Der Optimale Bayes Klassifikator ist wie folgt definiert:
$\text{argmax}_{c_j \in C} \sum_{h_i \in H} P(c_j|h_j) \cdot P(h_i|o)$

Ist die zu Wählende Hypothese bekannt, kann man direkt den MAximum-Likelihood-Classifier nutzen, der vom optimalen Bayes-Classifier abgeleitet ist.

3.4.1 Bayesian Networks

Knoten sind Zufallsvariablen, Kanten sind die Abhängigkeiten. Jede Zufallsvariable ist bedingt unabhängig von den anderen Variablen die nicht erfolgreich sind. Für jeden Knoten wird eine Tabelle der bedingten Warscheinlichkeiten erzeugt.

Trainieren solcher Netzwerke geht sowohl mit als auch ohne vorwissen.

3.4.2 Vor- und Nachteile

+ Optimalitätseigenschaft (wird zum Vergleich mit anderen Klassifikatoren verwendet)
+ Hohe Klassifikationsgenauigkeit

+ Inkrementierbar
+ Integration of domain knowledge
- Anwendbarkeit
- Ineffizient

3.5 Neuronale Netzwerke

3.5.1 Vor- und Nachteile

+ hohe Klassifikationsgenauigkeit
+ Robust gegen rauschen
+ Effizientes Anwenden
- Aufbau dauert lange
- Schwer nachzuvollziehen
- kein bekanntes Wissen kann genutzt werden, da nicht nachvollziehbar

3.6 nearest-Neighbor Classification

Für alle Objekte einer Klasse die MEan-Verktoren berechnen, dann das zu Klassifizierende Objekt der Klasse zuweisen, deren Mean-Vektor dem Objektvektor am nächsten ist. Dabei können auch mehr als ein Nachbar berücksichtigt werden und die Klassen können gewichtet werden. Zur berechnung wird eine distanzfunktion benötigt. Und es wird die die Anzahl (k) der berücksichtigten nachbarn benötigt. wenn k zu klein dann sehr anfällig gegen Outlier, wenn zugroß werden zu viele Objekte andere Klassen beachtet, k muss für die besten ergebnisse in der Mitte liegen.

3.6.1 Vor und Nachteile

+ Lokale Methode
+ Hohe Klassifikationsgenauigkeit
+ Inkrementell
+ kann für Vorhersagen benutzt werden
- teure Anwendung (insbesondere das Prüfen der Nachbarn)
- kein genaues Wissen über die Klassen

3.7 Support Vector Machines

Sucht die Hyperebene, die die den Datenraum am Besten zerteil und es wird der Hyperplane mit maximiertem Abstand zu den nächsten Trainingsobjekten ausgewählt.

3.7.1 Vor und Nachteile

+ starke mathematische Grundlage
+ Findet das globale Optiimum
+Skaliert gut bei sehr Hochdimensionalen Daten
+ Hohe Genauigkeit
- Ineffizienter Modelaufbau
- Modell kkann kaum interpretiert werden (lernt ausschließlich Gewichte) gewichte tendieren dazu, gleichmäßig verteilt zu sein

4 Clustering

Clustering identifiziert eine Menge an Kategorien, Klassen oder Gruppen, wobei OBjekte innerhalb eines Clusters ähnlich sein sollen und welche die Außerhalb eines clusters liegen möglichst wenig gemeinsam haben.

4.1 Ziele

- Data Understanding, das finden natürlicher Cluster
- Data Class Identifikation dinden nützlicher und passender Cluster
- Data Reduction Repräsentanten von clustern finden
- Outlier Detection es gibt lokale und globale Ausreißer
- Rauschen erkennen

4.2 Distanzfunktionen

L_p-Metric: $dist(x,y) = \sqrt[p]{\sum_{i=1}^{d} |X_i - y_i|^p}$

Euclidean Distance (p=2): $dist(x,y) = \sqrt{\sum_{i=1}^{d} (X_i - y_i)^2}$

Manhatten-Distance (p=1): $dist(x,y) = \sum_{i=1}^{d} |x_i - y_i|$

Maximum-Distanze($p = \infty$): $dist(x,y) = \max\{|x_i - y_i||1 \leq i \leq d\}$