UNIVERSITÄT
LEIPZIG

Evaluierung

INTERACTIVE VISUAL DATA MINING

EVALUIERUNG

- Warum?
  - Systematischer Weg, um zu testen, wie verschiedene Methode funktionieren
  - Vergleich zweier Methoden
- Wie?
  - Führe Experimente durch
- Probleme:
  - Limitierte Datenqualität
  - Methoden haben unterschiedliche Ergebnisse
    - Klassen
    - Wahrscheinlichkeiten
    - Numerische Werte

EVALUIERUNG

- Training und Tests
  - Die Performanz für Klassifizierungsprobleme wird mit der Fehlerrate gemessen
  - Fehlerrate: Anzahl der falschen Vorhersagen über alle Vorhersagen
  - Ziel: Erreiche gute Vorhersagen auf **neuen** Daten
- Die Fehlerrate auf dem Trainingsdatensatz eignet sich nicht als guter Indikator
  - Wird auch *Resubstitution Fehler* genannt
  - Ist nur ein erster Indikator für die Qualität des Schätzers (Klassifizierer)
- Teile den Datensatz in
  - Training Datensatz
  - Test Datensatz

EVALUIERUNG

- Training und Tests
  - Die Testdaten dürfen nicht zur Erstellung des Klassifizierers genutzt werden
  - Nutzung von verschiedenen, unabhängigen Datensätzen ist oft sinnvoll:
    - Trainingsdatensatz: Erstelle die Regeln
    - Validierungsdatensatz:
      - Pruning
      - Vergleich von verschiedenen Methoden
  - Testdatensatz: Test der Performanz des finalen und optimierten Klassifizierers
  - Große Datensätze sind hier hilfreich
  - Bei kleinen Datensätzen müssen reale Daten oft händisch klassifiziert werden

EVALUIERUNG

- Vorhersage Performanz
  - Hängt von der Größe des Testdatensatzes ab
  - Qualität des Klassifizierers hängt vom Trainingsdatensatz ab
  - Nutze Statistik: **Bernoulli-Kette**
    - Abfolge von unabhängigen Versuchen
    - Beispiel: Münzwurf

- Angenommen, die Münze hat einen Bias
  - Führe eine Reihe von Experimenten durch
  - Sage immer Kopf vorher
  - Habe eine 75% Erfolgsrate (Vorhersage Kopf, Ergebnis Kopf)
  - Was ist dann die richtige Erfolgsrate p?

EVALUIERUNG

- Vorhersage Performanz
  - Einzelner Bernoulli Versuch
    - Mittelwert: p
    - Varianz: $p \cdot (1 - p)$
  - N Bernoulli Versuche
    - Mittelwert: p
    - Varianz: $\frac{p \cdot (1-p)}{N}$
- Wie schaut die Verteilung für große N aus?

EVALUIERUNG

- Vorhersage Performanz
  - Einzelner Bernoulli Versuch
    - Mittelwert: p
    - Varianz: $p \cdot (1 - p)$
  - N Bernoulli Versuche
    - Mittelwert: p
    - Varianz: $\frac{p \cdot (1-p)}{N}$
- Wie schaut die Verteilung für große N aus?
  - Es nähert sich der Normalverteilung an

EVALUIERUNG

- Vorhersage Performanz
  - Nimm eine zufällige Variable X an
    - Mittelwert: 0
    - Varianz: 1
  - $\Pr[-z \leq X \leq z] = \mathrm{c}$
- Es gibt Tabellen für die Werte z und c
- Angenommen, die Daten sind normalverteilt:
  - Für einseitige Tests sehen die Tabellen so aus : $\Pr[X \geq z]$
  - Für Normalverteilungen gilt: $\Pr[X \geq z] = \Pr[X \leq z]$
  - $\Pr[X \geq z] = 5\%$: 5% Chance, dass $X > 1.65 \cdot \sigma + \mu$
  - Äquivalent zu: $\Pr[-1.65 \leq X \leq 1.65] = 90\%$
  - Finale Gleichung: $\Pr\left[-z < \frac{f-p}{\sqrt{\frac{p\cdot(1-p)}{N}}} < z\right] = \mathrm{c}$

EVALUIERUNG

- Vorhersage Performanz
  - Konfidenzgrenzen:
  - $p = \dfrac{f + \frac{z^2}{2N} \pm z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$
  - Erhalte von c (Konfidenz) z aus der Tabelle
  - Nutze die Formel mit:
    - Erfolgsrate $f = \frac{S}{N}$
    - S: Anzahl der Erfolge
    - N: Anzahl der Versuche
    - z: siehe Tabelle

EVALUIERUNG

- Vorhersage Performanz
  - $f = 75\%, N = 1000, c = 80\%$ ergibt $[0.732, 0.767]$
  - $f = 75\%, N = 100, c = 80\%$ ergibt $[0.691, 0.801]$
- Erinnerung:
  - Die Annahme einer Normalverteilung ist nur bei großen N schlüssig!

EVALUIERUNG

- *Holdout* Methode
  - Datenmengen für das Lernen und Testen sind oft limitiert verfügbar
  - Reserviere vorab einen Teil der Daten für das Testen
  - Nutzung der restlichen Daten für das Training
- Manchmal müssen auch Daten für die Validierung reserviert werden
- Oft:
  - Ein Drittel für das Testen
  - Zwei Drittel für das Training

EVALUIERUNG

- *Holdout* Methode
  - Problem: Die Datensätze können nicht repräsentativ sein
    - Im Trainingsdatensatz fehlt eine Klasse
    - Eine Klasse ist im Trainingsdatensatz unterrepräsentiert
- Lösung: *Stratifikation*
  - Random sampling
  - Garantiere, dass jede Klasse in Trainingsdatensatz und Testdatensatz gleich repräsentiert ist
  - Methode: *Stratified Holdout*
  - Ist nur eine simple Absicherung gegen ungleiche Verteilungen

EVALUIERUNG

- Repeated Holdout
  - Wiederhole die Analyse (mit Stratifikation) mehrere Male
  - Wähle jedes Mal eine Teilmenge der Daten zufällig für das Testen
  - Nutze die restlichen Daten als Trainingsdatensatz
- Zusammengefasste Fehlerrate:
  - Bilde den Durchschnitt über alle einzelnen Fehlerraten

EVALUIERUNG

- Kreuzvalidierung
  - Erweiterung der holdout Methode
  - Wähle eine feste Anzahl an *folds* (Partitionen) für die Daten
  - Normalerweise sind alle folds gleich groß
  - Jedes fold wird für das Testen genutzt
  - Die übrigen werden für das Training genutzt
  - Bei drei folds:
    - Dreifache Kreuzvalidierung
    - Stratifizierte dreifache Kreuzvalidierung
  - Bilde den Durchschnitt der Fehlerraten

EVALUIERUNG

- Kreuzvalidierung
  - Standard: 10 folds
  - Warum 10?
    - Praktische Erfahrungen
    - Theoretische Nachweise
    - Muss dennoch nicht als “Standardwert” angenommen werden!

EVALUIERUNG

- Leave-one-out Kreuzvalidierung
  - Datensatz: n Instanzen
  - n-fold Kreuzvalidierungen
    - Lasse jede Instanz nacheinander aus
    - Nutze n-1 Instanzen für das Training
    - Nutze eine Instanz für das Testen
    - Bilde den Durchschnitt über alle n Bewertungen

EVALUIERUNG

- Leave-one-out Kreuzvalidierung
  - Vorteile:
    - Nutzung der maximal möglichen Datenmenge für das Training
    - Dadurch kann es zur Steigerung der Klassifikatorperformanz kommen
    - Deterministisch (kein random sampling notwendig)
    - Optimal für kleinere Datensätze
  - Nachteile
    - Rechenzeit!
    - Unmöglich für große Datensätze
    - Kann nicht stratifiziert werden
      - Garantiert nicht stratifiziertes Ergebnis
    - Annahme: Datensatz mit gleicher Anzahl von Instanzen für 2 Klassen
      - Berechne die Mehrheitsklasse:
        - Echter Fehlerrate: 50%
        - Berechnete Fehlerrate: 100%

EVALUIERUNG

- Bootstrapping
  - Sampling mit Ersatz
  - Vorherige Verfahren wählten Datenpunkte ohne Ersatz
  - Variante: *0.632 bootstrap*

- Ein Datensatz mit n Instanzen wird n mal mit Ersatz gesampelt
- Liefert einen Trainingsdatensatz, welcher n Instanzen hat
- Nutzt einige Instanzen im neuen Datensatz mehrfach
- Ungenutzte Instanzen werden dann im Testdatensatz genutzt

EVALUIERUNG

- Bootstrapping
  - Jede Instanz hat eine Wahrscheinlichkeit von:
    - $\frac{1}{n}$, um gezogen zu werden
    - $1 - \frac{1}{n}$, um nicht gezogen zu werden
  - Jedes Item kann n mal gezogen werden
  - Wahrscheinlichkeit, dass eine Instanz nicht gezogen wird:

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

- Trainingsdatensatz
  - 63.2% Instanzen
- Testdatensatz
  - 36.8% Instanzen
- Einige Instanzen sind mehrfach im Trainingsdatensatz vertreten und ergeben so n Instanzen

EVALUIERUNG

- Bootstrapping
  - Schätzung der Fehlerrate ist pessimistisch
    - 63% Instanzen gegenüber der 90% Instanzen der 10-fold Kreuzvalidierung
  - Idee: Kombiniere des Testdatensatzfehler mit dem Fehler des Models
    - Normalerweise zu optimistisch
    - Sollte nicht alleine genutzt werden
- Bootstrap Fehlerrate:
- $e = 0{,}632\ e_{test\ instances} + 0.368\ \cdot e_{training\ instances}$
- Wiederhole das bootstrapping mehrmals
- Berechne den Durchschnitt über alle Wiederholungen

EVALUIERUNG

- Bootstrapping
  - Vorteil
    - Eine der beste Lösungen für sehr kleine Datensätze
  - Nachteile
    - Liefert schlechte Ergebnisse für spezielle Fälle
    - Annahme:
      - Zwei Klassen
      - Gleiche Anzahl von Instanzen für jede Klasse
- $e_{training\ instances} = 0$
- Schätzung der Fehlerrate:
  $0.632 \cdot 50\% + 0.368 \cdot 0\% = 31.6\%$
- Echte Fehlerrate: 50%

EVALUIERUNG

- Vergleich verschiedener Data Mining Verfahren
  - Simple Methode:
    - Nutze zum Beispiel Kreuzvalidierung
    - Nutze dann das Verfahren mit der geringsten Fehlerrate
  - Problem:
    - Geschätzte Fehlerrate kann aber nicht korrekt sein
    - Wie kann man beurteilen, dass ein Verfahren besser als das andere ist?
- Insbesondere für neue Verfahren muss man dies evaluieren
- Lösung:
  - Statistische Tests, die auf den Konfidenzintervallen basieren

EVALUIERUNG

- Vergleich verschiedener Data Mining Verfahren
  - Nutzung des t-Tests (auch bekannt als Student’s t-test)
  - Wenn der Trainingsdatensatz und der Testdatensatz in beiden Verfahren gleich sind, wird der gepaarte t-Test verwendet
  - Evaluierung wird später erklärt
    - Wurde für die Visualisierung auch angepasst, ist aber generell anwendbar

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten
  - Bisher wurde nur Korrektheit geprüft
    - Korrekt im Sinne, dass die Vorhersage mit dem gesetzten Wert übereinstimmt
    - Ansonsten ist es ein Fehler
  - Das ist in vielen Fällen passend
  - 0-1 Verlustfunktion
    - Korrekte Vorhersage: loss = 0
    - Fehlerhafte Vorhersage: loss = 1
- Viele Lernverfahren geben aber zudem eine Wahrscheinlichkeit für die Vorhersage an
  - Zum Beispiel Naïve Bayes
- Nützlich, wenn die Ergebnisse in weiteren Schritten genutzt werden sollen

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten: Quadratische Verlustfunktion
  - Für eine einzelne Instanz
  - Annahme:
    - *k* mögliche Ergebnisse
    - Wahrscheinlichkeitsvektor $(p_1, \dots, p_k)$, $\sum_{i=1}^{k} p_i = 1$
  - Tatsächliches Ergebnis: $(a_1, \dots, a_k)$
    - $a_i = \begin{cases} 1 & i \text{ is the correct class} \\ 0 & \text{else} \end{cases}$
- Quadratische Verlustfunktion:
  $\sum_j (p_j - a_j)^2 = 1 - 2p_i + \sum_j {p_j}^2$
  - Fehlerhafte Vorhersagen:
    ${p_j}^2$
  - Korrekte Vorhersagen:
    $(p_i - 1)^2 = 1 - 2p_i + {p_i}^2$
- Für mehrere Instanzen wird die Verlustfunktion summiert

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten: Quadratische Verlustfunktion
  - Minimierung des quadratischen Fehlers ist gut erforscht
  - In diesem Falle soll der Klassifizierer die beste Schätzung der wahren Wahrscheinlichkeiten vornehmen
  - Häufig verwendet zur Vorhersage von Wahrscheinlichkeiten

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten: Informational Loss Function
  - $-\log_2 p_i$
  - *i.* Vorhersage ist korrekt
  - Negative log likelihood
    - Modulo eines konstanten Faktors, welcher durch die Basis des Logarithmus bestimmt wird
  - Repräsentiert die Informationen, welche notwendig sind, um die tatsächliche Klasse *I* auszudrücken
  - Einheit: bit
  - Wahrscheinlichkeitsverteilung: $p_1, \ldots, p_k$
  - Minimale Anzahl von bits, die benötigt werden, um die tatsächlich aufgetretene Klasse auszudrücken.
    - Unter Beachtung der gegebenen Wahrscheinlichkeitsverteilung
  - Minuszeichen: Wahrscheinlichkeiten sind kleiner als 1

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten: Informational Loss Function
  - $-\log_2 p_i$
  - *i.* Vorhersage ist korrekt
  - Negative log likelihood
    - Modulo eines konstanten Faktors, welcher durch die Basis des Logarithmus bestimmt wird
  - Repräsentiert die Informationen, welche notwendig sind, um die tatsächliche Klasse *I* auszudrücken
  - Einheit: bit
  - Wahrscheinlichkeitsverteilung: $p_1, \ldots, p_k$
  - Minimale Anzahl von bits, die benötigt werden, um die tatsächlich aufgetretene Klasse auszudrücken.
    - Unter Beachtung der gegebenen Wahrscheinlichkeitsverteilung
  - Minuszeichen: Wahrscheinlichkeiten sind kleiner als 1
    - Negative Logarithmen!

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten: Informational Loss Function
  - Beispiel:
    - Kopf oder Zahl braucht 1 bit
      - $-\log_2 \frac{1}{2} = 1$
  - Erwarteter Wert
    - $\sum_{j=1}^{k} -p_j' \cdot \log_2 p_j$
    - p_j' ist die wahre Wahrscheinlichkeit für die Klasse *j*
    - Ausprägung ist minimal, wenn $p_j = p_j'$
- Entropie: Durchschnittliche Information
- Probleme:
  - Wenn die Wahrscheinlichkeit 0 ist, wird der Information loss ∞
  - Auch bekannt als *zero-frequency problem*
  - Mögliche Lösung:
    - Laplace estimator

EVALUIERUNG

- Vorhersage von Wahrscheinlichkeiten
  - Welche Verlustfunktion sollte man nutzen?
  - Unterschiede:

Quadratische Verlustfunktion	Information loss function
Beachtet alle Wahrscheinlichkeiten	Basiert nur auf den Wahrscheinlichkeiten der aktuell auftretenden Klassen
Obere Schranke: 2	Keine obere Schranke

EVALUIERUNG

- Berechnung der Kosten
  - Vorgestellte Evaluierungen beachten nicht falsch klassifizierte Daten
  - Das kann zu fragwürdigen Ergebnissen führen!
- Annahme: Ein Ergebnis tritt in 97% der Fälle auf
- Das Modell sagt dieses Ergebnis immer voraus
  - Ergebnis ist zu 97% korrekt
- Sind die Fälle, in denen das Ergebnis nicht auftritt vielleicht interessanter?

EVALUIERUNG

- Berechnung der Kosten
  - True positive rate: $\frac{TP}{TP+FN}$
  - False positive rate: $\frac{FP}{FP+TN}$
  - Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

		Vorhergesagte Klasse	
		ja	nein
Tatsächliche Klasse	ja	true positive	false negative
	nein	false positive	true negative

EVALUIERUNG

- Berechnung der Kosten
  - Multiklassen Vorhersage: Nutzung einer 2 dimensionalen *confusion* Matrix
  - Jede Zelle repräsentiert die Anzahl der Instanzen mit:
    - Zeile: tatsächliche Klasse
    - Spalte: vorhergesagte Klasse
- Gute Ergebnisse:
  - Große Zahlen auf der Diagonalen
  - Kleine Zahlen außerhalb
- Kann gut gegen Zufallsvorhersagen verglichen werden

EVALUIERUNG

- Berechnung der Kosten
  - Kappa Statistik: $\frac{p-r}{\mathrm{m}-r}$
    - p: Vorhersagen
    - m: Maximal erfolgreiche Vorhersagen
    - r: Zufällige erfolgreiche Vorhersagen
  - Maximaler Wert: 100%
  - Zufallsvorhersage ergibt 0%
- Beispiel vom Anfang:
  - Kappa Wert würde bei 0% liegen
- Kosten sind aber immer noch nicht berücksichtigt!

EVALUIERUNG

- Berechnung der Kosten: Kostensensitive Klassifikation
  - Nutzung einer Kostenmatrix
    - Jede Zelle repräsentiert die Kosten, welche diese Entscheidung des Klassfikators verursacht
    - Unterschiedliche Zellen können unterschiedliche Kosten haben
- Berechnung der Kosten
  - Summiere alle Zellen der Kostenmatrix für eine Testinstanz auf
  - Kosten werden bei der Vorhersagen ignoriert
  - Kosten werden aber bei der Evaluierung in Betracht gezogen

EVALUIERUNG

- Berechnung der Kosten: Kostensensitive Klassifikation
  - Ergebnisse mit Wahrscheinlichkeiten:
    - Vorhersage der Klasse mit den geringsten Fehlerkosten
    - Gegeben:
      - Kosten Matrix $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$
      - Wahrscheinlichkeiten für die Klassen a, b, c: p_a, p_b, p_c
  - Vorhersage a: $1 - p_a$
    - Multipliziere $[p_a \quad p_b \quad p_c]$ mit $[0 \quad 1 \quad 1]$: $p_b + p_c = 1 - p_a$
  - Vorhersage b: $1 - p_b$
  - Vorhersage c: $1 - p_c$
  - Auswahl der Vorhersage mit der geringsten Fehlerkosten:
    - Geringste Wahrscheinlichkeit
  - Die meisten Klassifikatoren können angepasst werden, dass sie Wahrscheinlichkeiten mit ausgeben

EVALUIERUNG

- Berechnung der Kosten: Kostensensitive Klassifikation
  - Nutzung der Kostenmatrix während der Trainingsphase
  - Ignoriere Kosten bei der Vorhersage
  - Simple Methode:
    - Variiere die Größen der Instanzen im Trainingsdatensatz
    - Z.B. Duplikate der Instanzen
- Viele Lernmethoden erlauben gewichtete Instanzen
  - Instanzen werden normalerweise auf 1 initialisiert
  - Instanzen können aber auch die relative Kosten der false positives und false negatives initialisiert werden

EVALUIERUNG

- Evaluierung numerischer Vorhersagen
  - Grundideen funktionieren auch hier
  - Messung der Fehlerrate muss angepasst werden:
    - Vorhergesagte Werte in den Testinstanzen: $p_1, \ldots, p_n$
    - Tatsächliche Werte: $a_1, \ldots, a_n$

EVALUIERUNG

- Evaluierung numerischer Vorhersagen
  - Mittlerer quadratischer Fehler:

$$\frac{\sum_{j=1}^{n}(p_j - a_j)^2}{n}$$

    - Wird sehr häufig genutzt
    - Reagiert empfindlich auf Ausreißer

  - Wurzel der mittleren Fehlerquadratsumme:

$$\sqrt{\frac{\sum_{j=1}^{n}(p_j - a_j)^2}{n}}$$

    - Hat die gleiche Dimension wie der vorgesagte Wert
    - Mathematisch unproblematischer

EVALUIERUNG

- Evaluierung numerischer Vorhersagen
  - Mittlerer absoluter Fehler:

$$\frac{\sum_{j=1}^{n}|p_j - a_j|}{n}$$

  - Unempfindlich gegen Ausreißer
- Relative Fehler
  - Unabhängig von der Größe der Werte
  - Nutzung von relativen Fehlern statt der absoluten Fehler in den Berechnungen

EVALUIERUNG

- Evaluierung numerischer Vorhersagen
  - Relativer-quadratischer Fehler:

$$\frac{\sum_{j=1}^{n}(p_j - a_j)^2}{\sum_{j=1}^{n}(a_j - \bar{a})^2}$$

    - Relativ zu den Ergebnissen eines simplen Klassifikators: $\bar{a}$
    - $\bar{a}$ ist der mittlere Wert über dem Trainingsdatensatz
  - Wurzel des relativen quadratischen Fehler:

$$\sqrt{\frac{\sum_{j=1}^{n}(p_j - a_j)^2}{\sum_{j=1}^{n}(a_j - \bar{a})^2}}$$

    - Analog zum relativen quadratischen Fehler
  - Relativer absoluter Fehler:

$$\frac{\sum_{j=1}^{n}|p_j - a_j|}{\sum_{j=1}^{n}(a_j - \bar{a})^2}$$

EVALUIERUNG

- Evaluierung numerischer Vorhersagen: Korrelation Koeffizienten
  - Berechnung der statistischen Korrelation zwischen vorhergesagten und tatsächlichen Wert
  - 1: Perfekte Korrelation
  - 0: Keine Korrelation
  - -1: Perfekte negative Korrelation
  - Unabhängig von der Skalierung, wenn man die vorhergesagten Werten mit einer Konstante multipliziert

EVALUIERUNG

- Evaluierung numerischer Vorhersagen: Korrelation Koeffizienten

  - $\frac{S_{PA}}{\sqrt{S_P \cdot S_A}}$

  - $S_{PA} = \frac{\sum_{i=1}^{n}(p_i - \bar{p})(a_i - \bar{a})}{n-1}$

  - $S_P = \frac{\sum_{i=1}^{n}(p_i - \bar{p})^2}{n-1}$

  - $S_A = \frac{\sum_{i=1}^{n}(a_i - \bar{a})^2}{n-1}$

EVALUIERUNG

- Evaluierung numerischer Vorhersagen
  - Sinn des berechnete Maßes hängt von der Anwendung ab:
    - Was soll minimiert werden?
    - Was sind die Kosten der verschiedenen Fehlerarten?
  - Meistens liefern alle Fehlermessarten das gleiche Ergebnis