Vorlesungsplan

- 17.10. Einleitung
- 24.10. Ein- und Ausgabe
- 31.10. Reformationstag, Einfache Regeln
- 7.11. Naïve Bayes, Entscheidungsbäume
- 14.11. Entscheidungsregeln, Assoziationsregeln
- 21.11. Lineare Modelle, Instanzbasiertes Lernen
- 28.11. Clustering
- 5.12. **Evaluation**
- 12.12. Lineare Algebra für Data Mining
- 19.12. Statistik für Data Mining
- 9.1. Entscheidungsbäume, Klassifikationsregeln
- 16.1. Lineare Modelle, Numerische Vorhersage
- 23.1. Clustering
- 30.1. Attribut-Selektion, Diskretisierung, Transformationen
- 6.2. Kombination von Modellen, Lernen von nicht-klassifizierten Beispielen

Glaubwürdigkeit:
Was wurde gelernt?

- Trainieren, Testen, Verbessern
- Vorhersage der Performanz: Konfidenzintervalle
- Zurückbehalten von Beispielen, Kreuz-Validierung, Bootstrap
- Vergleich von Schemata: t-Test
- Vorhersage von Wahrscheinlichkeiten: Verlust-Funktionen
- Kostenabhängige Maße
- Evaluation nummerischer Vorhersagen
- Prinzip der Minimalen Beschreibungslänge (MDL)

Evaluation: der Schlüssel zum Erfolg

- Wie gut eignet sich das gelernte Modell zur Vorhersage?
- Fehlerrate auf Trainingsdaten ist kein guter Indikator für Performanz auf zukünftigen Daten
  - Sonst würde 1-NN der optimale Klassifikator sein!
- Einfache Lösung, falls viele Beispiel-Instanzen vorhanden sind:
  - Teile Instanzenmenge in Trainings- und Testdaten
- Aber: (vorklassifizierte) Daten sind oft sehr begrenzt
  - Komplizierte Techniken werden gebraucht

Fragen zur Evaluation

- Statistische Zuverlässigkeit von geschätzten Unterschieden ($\rightarrow$ significance tests)
- Wahl des Performanz-Maßes:
  - Anzahl der korrekt klassifizierten Instanzen
  - Genauigkeit der Schätzungen für die Klassenwahrscheinlichkeiten
  - Fehler bei nummerischer Vorhersage
- Kosten für verschiedene Fehlertypen
  - Viele Anwendungen beinhalten Kosten

Training und Testen I

- Natürliches Performanz-Maß für Klassifikations-Probleme: *Fehlerrate*
  - *Erfolg*: Klasse der Instanz ist korrekt vorhergesagt
  - *Fehler*: Klasse der Instanz ist falsch vorhergesagt
  - Fehlerrate: Anteil der Fehler an der Gesamtmenge der Instanzen
- *Resubstitutionsfehler:* Fehlerrate auf den Trainingsdaten
- Resubstitutionsfehler ist (hoffungslos) optimistisch!

Training und Testen II

- *Test-Menge*: unabhängige Instanzen, die nicht zum Aufbau der Klassifikators genutzt wurden
  - Annahme: Trainings- und Testdaten sind repräsentative Stichproben des DM - Problems

- Test- und Trainingsdaten können sich in ihrer Herkunft unterscheiden
  - Beispiel: Klassifikatoren für Kundendaten aus zwei verschiedenen Städten *A* und *B*
    - Zum Schätzen der Performanz des Klassifikators für *A*, teste ihn auf Daten aus *B*

Bemerkung über Einstellen von Parametern

- Es ist wichtig, daß die Test-Daten in keiner Weise benutzt werden, um Klassifikator zu erstellen
- Einige Lernschemata arbeiten in zwei Schritten :
  - Schritt 1: baue Basis-Struktur
  - Schritt 2: optimiere Parametereinstellungen
- Die Testdaten dürfen nicht zur Parameter-Einstellung genutzt werden!
- Richtig ist, drei Mengen zu nutzen: *Trainings*-, *Validierungs*- und *Testdaten*
  - Validierungsdaten sind für die Optimierung der Parameter

Alles aus den Daten herausholen

- Sobald die Evaluierung beendet ist, können *alle Daten* für den endgültigen Klassifikator genutzt werden
- Im allg., je mehr Trainingsdaten desto besser der Klassifikator (aber die Verbesserung steigt langsamer an)
- Je mehr Testdaten desto genauer die Fehlerschätzung
- *Zurückbehalten*: Teile Originaldaten in Trainings- und Testdaten
  - Dilemma: idealerweise sollten beide Mengen groß sein!

Performanz vorhersagen

- Angenommen die geschätzte Fehlerrate ist 25%. Wie nah ist dies an der wahren Fehlerrate?
  - Hängt von der Größe der Testdaten ab
- Vorhersage ist wie würfeln mit einer beeinflußten Münze
  - “Kopf” ist “Richtig”, “Zahl” ist “Fehler”
- In Statistik, mehre unabhängige Ereignisse wie diese werden *Bernoulli Prozess* genannt
  - Statistische Theorie liefert Konfidenzintervalle für die wahre Verteilung

Vertrauensintervalle

- p liegt in einem bestimmen Intervall mit einer bestimmten Konfidenz
- Beispiel: S=750 Erfolge bei N=1000 Versuchen
  - Geschätzte Erfrolgsrate: 75%
  - Wie nah ist das an der wahren Erfolgsrate p?
    - Antwort: mit 80% Konfidenz $p \in [73.2,76.7]$
- Anderes Beispiel: S=75 und N=100
  - Geschätzte Erfrolgsrate : 75%
  - mit 80% Konfidenz $p \in [69.1,80.1]$

Durchschnitt und Varianz

- Durchschnitt und Varianz für Bernoulli-Versuch: p, $p\,(1-p)$
- Erwartete Erfolgsrate $f=S/N$
- Durchschnitt und Varianz für f : p, $p\,(1-p)/N$
- Für große N: f folgt einer Normalverteilung
- c% Konfidenz-Intervall $[-z \le X \le z]$ für Zufallsvariable mit Erwartungswert 0 ist gegeben durch:

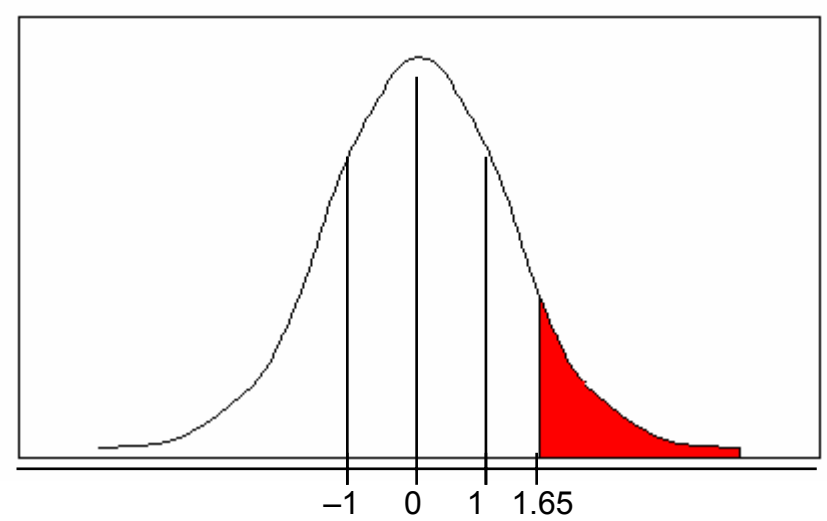
$$\Pr[-z \le X \le z] = c$$

- Bei symmetrischer Verteilung:

$$\Pr[-z \le X \le z] = 1 - 2 \times \Pr[X \ge z]$$

Konfidenz-Grenzen

- Konfidenz-Grenzen für Normalverteilung mit Erwartungswert 0 und Varianz 1:



Pr[$X \geq z$]	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

- Deshalb:

$$\Pr[-1.65 \leq X \leq 1.65] = 90\%$$

- Um die Tabellen zu nutzen, muß die Zufallsvariable *f* umskaliert werden, um Erwartungswert 0 und Einheitsvarianz zu haben

Transformation von *f*

- Transformierter Wert von *f* : $\frac{f-p}{\sqrt{p(1-p)/N}}$

  (i.e. subtrahiere Durchschnitt und dividiere durch Std. Abweichung)

- Erhaltene Ungleichung: $\Pr\left[-z \le \frac{f-p}{\sqrt{p(1-p)/N}} \le z\right] = c$

- Auflösen nach *p* :

$$p = \left(f + \frac{z^2}{2N} \pm z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Big/ \left(1 + \frac{z^2}{N} \right)$$

Beispiele

- f = 75%, N = 1000, c = 80% (so daß z = 1.28):

$$p \in [0.732, 0.767]$$

- f = 75%, N = 100, c = 80% (so daß z = 1.28):

$$p \in [0.691, 0.801]$$

- Beachte das Normalverteilungsannahme nur für große N gilt (i.allg. $N > 100$)
- f = 75%, N = 10, c = 80% (so daß z = 1.28):

$$p \in [0.549, 0.881]$$

(sollte mit etwas Vorsicht genossen werden)

Holdout, Schätzen der Fehlerrate

- Was tun, wenn Beispieldaten begrenzt sind?
- Reserviere eine bestimmte Menge zum Testen, der Rest kann zum Training genutzt werden
  - Daumenregel: ein Drittel Testdaten, Rest für Training
- Problem: die Teilmengen können nicht repräsentativ sein
  - Beispiel: Klasse kann in Testdaten fehlen
- Erweiterte Version nutzt *Stratifizierung*
  - Sichert daß jede Klasse mit etwa gleichen Anteilen in beiden Teilmengen vertreten ist

Wiederholtes Zurückbehalten

- Schätzung der Fehlerrate kann verbessert werden, wenn verschiedene Aufteilungen genutzt werden
  - In jeder Iteration, wird ein bestimmter Anteil zufällig zum Training gewählt (mit/ohne Stratifizierung)
  - Die Fehlerraten aus den verschiedenen Iterationen werden gemittelt, um den Gesamtfehler zu bestimmen
- Noch nicht optimal: die verschiedenen Testmengen können überlappen

Kreuz-Validierung

- Kreuz-Validierung vermeidet überlappende Testmengen
  - a) teile Daten in *k* Teilmengen gleicher Größe
  - b) nutze jede Teilmenge einmal zum Testen und den Rest zum Training
- *k-fache* Kreuz-Validierung
- Teilmengen werden vor Kreuz-Validierung stratifiziert
- Einzelnen Fehlerraten werden gemittelt

Mehr über Kreuz-Validierung

- Standard Methode für Evaluierung: stratifizierte 10-fache Kreuz-Validierung
- Warum 10?
  - durch viele Experimente als gute Wahl belegt
- Stratifizierung reduziert die Varianz der Shätzung der Fehlerrate
- Noch besser: wiederholte stratifizierte Kreuz-Validierung
  - z.B. 10-fache Kreuz-Validierung wird 10 mal wiederholt und Ergebnisse werden gemittelt (reduziert die Varianz noch mehr)
  - hoher Aufwand: 100 mal den Lernalg. auf 9/10 der Daten laufen lassen

Leave-One-Out Kreuz-Validierung

- Leave-One-Out:
  Spezialfall der Kreuz-Validierung :
  - Setze Anzahl der Teilmengen auf Anzahl der Instanzen
  - für *n* Instanzen, konstruiere den Klassifikator *n* mal
- Macht starken Gebrauch von den Daten
- Keine zufälligen Partitionierungen
- Sehr berechnungsaufwändig

Leave-One-Out-KV und Stratifizierung

- Nachteil der Leave-One-Out-KV: Stratifizierung ist nicht möglich
  - Die Test-Menge ist immer nicht-stratifiziert, da sie nur aus einer Instanz besteht!
- Extremes Beispiel: zufällige Daten werden in genau zwei Klassen geteilt
  - Bester Klassifikator sagt immer die Mehrheitsklasse vorher
  - 50% Genauigkeit auf frischen Daten
  - Leave-One-Out-KV Schätzung ist immer 100% Fehler!

Bootstrap

- KV nutzt *auswählen ohne Zurücklegen*
  - dieselbe Instanz, einmal gewählt kann nicht nochmal für eine andere Test/Traingsmenge gewählt werden
- *Bootstrap* nutzt *auswählen mit Zurücklegen,* um die Trainingsmenge zu bestimmen
  - Wähle eine Datenmenge mit *n* Instanzen *n* mal mit Zurücklegen um eine neue Menge mit *n* Instanzen zu erzeugen
  - Nutze diese Daten zum Training
  - Nutze die Instanzen aus den Originaldaten, die nicht in der Trainingsmenge auftauchen zum Testen

0.632 Bootstrap

- *0.632 Bootstrap*
  - Jede Instanz hat eine Whr. von 1–1/*n* nicht gewählt zu werden
  - Deshalb ist die Whr. in der Testmenge zu landen:

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

  - Das heißt, die Trainingsdaten enthalten etwa 63.2% der Instanzen

Fehlerrate schätzen mit Bootstrap

- Fehlerrate auf Test-Daten ist sehr pessimistisch
  - Es wurde nur auf ~63% der Instanzen trainiert
- Kombiniere diese FR mit Resubstitutionsfehler:

$$err = 0.632 \cdot e_{\text{test instances}} + 0.368 \cdot e_{\text{training instances}}$$

- Der Resubstitutionsfehler hat ein kleineres Gewicht als der Fehler auf den Testdaten
- Wiederhole den Prozeß mehrmals mit verschiedenen Trainingsdaten, mittele die Ergebnisse

Mehr über Bootstrap

- Gute Methode die Performanz für sehr kleine Datenmengen zu schätzen
- Es gibt auch Probleme
  - Gegeben seien zufällige Daten wie vorhin
  - Durch Auswendiglernen erreicht man
    0% Resubstitutionsfehler und
    ~50% Fehler auf Testdaten
  - Bootstrap Schätzung für diesen Klassifikator:

$$err = 0.632 \cdot 50\% + 0.368 \cdot 0\% = 31.6\%$$

  - Wahrer erwarteter Fehler: 50%

Vergleich von Data Mining Schemata

- Häufige Frage: welches von zwei Lernschemata ist besser?
- Bemerkung: dies ist Anwendungsabhängig!
- Einfacher Weg: vergleiche 10-fach KV Schätzungen
- Problem: Varianz der Schätzung
- Varianz kann durch wiederholte KV reduziert werden
- Aber, wir wissen nicht, ob der Vergleich zuverlässig ist

Signifikanz-Test

- Signifikanz-Tests sagen uns wie sicher ein Unterschied ist
- *Null Hypothese*: es gibt keinen “realen” Unterschied
- *Alternativ Hypothese*: es gibt einen Unterschied
- Signifikanz Test mißt wie stark die Belege sind, um die Null Hypothese abzulehnen
- Zum Beispiel, wir nutzen 10-fache KV
- Frage: unterscheiden sich die Durchschnitte der 10 fachen KV Schätzungen signifikant?

Gepaarter t-Test

- *Student's t-Test* zeigt ob die Erwartungswerte zweier Stichproben sich signifikant unterscheiden
- Benutze individuelle Stichproben der KV
- Nutze gepaarten t-Test, da individuelle Stichproben gepaart sind
  - Dieselbe KV wird auf die zwei DM–Methoden angewendet

William Gosset

Born: 1876 in Canterbury; Died: 1937 in Beaconsfield, England

Obtained a post as a chemist in the Guinness brewery in Dublin in 1899. Invented the t-test to handle small samples for quality control in brewing. Wrote under the name "Student".

Verteilung der Durchschnitte

- $x_1\ x_2\ \ldots\ x_k$ und $y_1\ y_2\ \ldots\ y_k$ *sind die Fehlerraten der* $2k$ Stichproben für eine k-fache KV
- m_x und m_y sind die Durchschnitte
- Bei hinreichend vielen Stichproben ist der Durchschnitt einer Menge von unabhängigen Stichproben normal verteilt
- Geschätzte Varianzen der Durchschnitte sind σ_x^2/k und σ_y^2/k
- Falls μ_x und μ_y die wahren Erwartungswerte sind, dann sind $\frac{m_x - \mu_x}{\sqrt{\sigma_x^2 / k}}$ und $\frac{m_y - \mu_y}{\sqrt{\sigma_y^2 / k}}$ *ungefähr* normal verteilt mit Erwartungswert 0 und Varianz 1

Student's Verteilung

- Für kleine Stichproben ($k < 100$) folgt der Erwartungswert *Student's* Verteilung mit $k-1$ *Freiheitsgraden*
- Konfidenzgrenzen:

9 Freiheitsgrade

$\Pr[X \geq z]$	z
0.1%	4.30
0.5%	3.25
1%	2.82
5%	1.83
10%	1.38
20%	0.88

Normalverteilung

$\Pr[X \geq z]$	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84

Verteilung der Differenzen

- Sei $d_i = x_i - y_i$ und m_d der Erwartungswert der Differenzen
- m_d ist auch Student verteilt mit k–1Freiheitsgraden
- Sei σ_d^2 die Varianz der Differenz
- Die standardisierte Version von m_d ist die t-Statistik:

$$t = \frac{m_d}{\sqrt{\sigma_d^2 / k}}$$

- Wir nutzen t für den t-Test

Durchführung des t-Test

- Lege Signifikanz Nivau α fest
  - Falls Unterschied signifikant auf Nivau α% ist, dann ist der Unterschied mit Whr. (100-α)% real
- Halbiere Signifikanznivau, weil der Test zweiseitig ist
  - die wahre Differenz kann +ve oder – ve sein
- Schlage Wert z nach, der zu $\alpha/2$ gehört
- Falls $t \leq -z$ oder $t \geq z$ dann ist der Unterschied signifikant
  - Null Hypothese wird abgelehnt

Ungepaarte Beobachtungen

- Fall KV Schätzungen von verschiedenen Zufallspartitionen kommen, sind sie nicht mehr gepaart
- (oder wir nutzen *k* -fache KV für Schema a, und *j* -fache KV für Schema b)
- Dann müssen wir den ungepaarten t-Test nutzen mit min(*k* , *j*) – 1 Freiheitsgraden
- Die *t*-Statistik wird dann:

$$t = \frac{m_d}{\sqrt{\sigma_d^2 / k}} \Rightarrow t = \frac{m_x - m_y}{\sqrt{\frac{\sigma_x^2}{k} + \frac{\sigma_y^2}{j}}}$$

Interpretation des Ergebnis

- Alle KV Schätzungen basieren auf derselben Datenmenge
- Stichproben sind nicht unabhängig
- Eigentlich sollten k verschiedene Stichproben genutzt werden
- Alternative: nutze heuristischen Test, z.B. *corrected resampled t-test*

Vorhersage von Wahrscheinlichkeiten

- Performanz Maß bisher: Erfolgsrate
- Auch genannt *0-1 Verlustfunktion*:

$$\sum_i \begin{cases} 0 \text{ falls Vorhersage korrekt} \\ 1 \text{ falls Vorhersage falsch} \end{cases}$$

- Viele Klassifikatoren geben Klassen-Whr. zurück
- In Abhängigkeit von der Anwendung können wir auch die Genauigkeit der Klassen-Whr. testen
- 0-1 Verlust ist nicht das richtige Performanzmaß in diesem Fall

Quadratische Verlustfunktion

- $p_1 \ldots p_k$ sind Whr.schätzungen für eine Instanz
- c ist der Index der Klasse der Instanz
- $a_1 \ldots a_k = 0$, außer für a_c der 1 ist
- *Quadratischer Verlust* ist: $\sum_j (p_j - a_j)^2 = \sum_{j \neq c} p_j^2 + (1 - p_c)^2$
- Minimiere $E\left[\sum_j (p_j - a_j)^2\right]$
- Es kann gezeigt werden, daß der quadratische Verlust minimiert wird, wenn $p_j = p_j^*$, die wahren Whr. sind

Informations-Verlustfunktion

- Informationsverlust ist: $-\log(p_c)$
  wobei *c* der Index der Klasse der akt. Instanz ist
- Anzahl der Bits um die akt. Klasse zu kommunizieren
- Seien $p_1^* \ldots p_k^*$ die wahren Klassen-Whr. mit $\sum_{i=1}^{k} p_i^* = 1$
- Erwartungswert für Informationsverlust ist:

$$-p_1^* \log_2 p_1 - \ldots - p_k^* \log_2 p_k$$

- Minimal wenn $p_j = p_j^*$
- Problem: *Whr. einer Klasse ist Null*

Diskussion

- Welche Verlustfunktion?
  - Beide sind sinnvoll
  - Quadratischer Verlust beachtet alle Klassenwahrscheinlichkeiten einer Instanz
  - Informationsverlust focusiert nur auf die Whr. der richtigen Klasse einer Instanz
  - Quadratischer Verlust ist beschränkt: $1+\sum_j p_j^2$ *nie größer als 2*
  - Informationverlust kann unendlich sein
- Informationsverlust ist in Beziehung zum *MDL Prinzip* [später]