Mathematische Grundlagen III

Evaluation

Garance PARIS

16. Juli 2011

Training Set und Test Set

- Ein fairer Test gibt an,
  wie gut das Modell im Einsatz ist
- Resubstitution:
  Evaluation auf den Trainingsdaten
- Resubstitution sagt uns nichts darüber, wie gut das Modell verallgemeinern kann
- Da statistische Modelle von Daten lernen,
  neigen sie dazu, neue Ereignisse zu sehr wie bereits gesehene Ereignisse zu modellieren

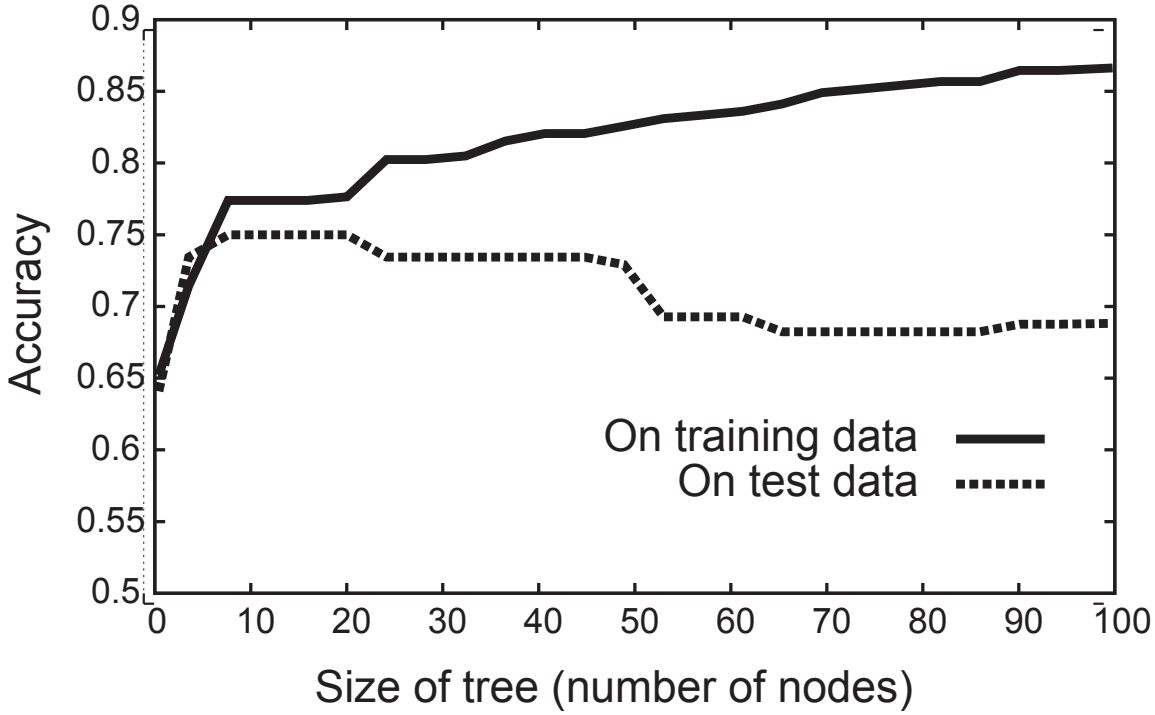
Training Set und Test Set

- Überanpassung (Engl. “overfitting”):
  Die genaue Eigentschaften des Training Sets
  zu genau widerspiegeln, anstatt Verallgemeinerung
- Im schlimmsten Fall könnte ein Modell einfach
  die Beispiele speichern und später nachschlagen
- Zum Testen braucht man also einen unabhängigen
  Datensatz, das Test Set, das aus ungesehenem
  Material besteht
- Es sollte groß genug sein, um eine zuverlässige
  Einschätzung des Modells zu ermöglichen

Evaluierung auf dem Test Set

- Die Daten werden anfangs in ein *"Training Set"* und ein *"Test Set"* geteilt, z. B. mit dem Verhältnis 9:1
- Die Parameter werden auf dem Training Set geschätzt (Lernphase)
- Dann wird jeweils auf dem Training Set und auf dem Test Set getestet
- Die Differenz gibt an, wie gut das Modell verallgemeinern kann:
  Je kleiner der Verlust, umso besser

Overfitting: Beispiel bei Entscheidungsbäumen



(Mitchell, 1997, S. 67)

Occam's Razor: Das Sparsamkeitsprinzip

Von mehreren Theorien, die den gleichen Sachverhalt erklären, ist die einfachste zu bevorzugen

- Wir wollen die allgemeine Tendenz mit unserem Modell fassen, nicht die zufälligen Umstände der Trainingsdaten
- Weniger Parameter führen weniger zu Overfitting und generalisieren besser

Weitere Aufteilungen: Das Validation Set

- Bei manchen Lernmethoden werden die Parameter in zwei Schritten festgelegt
- Beispiel bei N-Gramm-Modellen: Auszählung von Häufigkeiten gefolgt von Smoothing
- Hier braucht man einen unbekannten Datensatz für den 2. Schritt, der üblicherweise als “Validation Set” bezeichnet wird
- Es ist unwahrscheinlich, dass das Validation Set die selben zufälligen Schwankungen enthält als das Training Set

Weitere Aufteilungen: Das Development Test Set

- Bei der Entwicklung statistischer Modelle arbeitet man häufig iterativ:
  Algorithmus implementieren, trainieren, testen,
  Fehler anschauen (Fehleranalyse),
  Algorithmus verbessern, und das ganze wiederholen
- Wenn dieser Zyklus zu oft wiederholt wird,
  steigt die Gefahr einer Überanpassung des Modells
  an das Test Set
- Lösung: Das Test Set in zwei teilen, ein Teil für die Entwicklung und ein Teil für die allerletzte Testrunde
- Das Ergebnis bei der letzten Testrunde ist
  meist niedriger

Stratifizierung

- Alle Datensätze müssen repräsentative Stichproben der Daten sein
- *Stratifizierung* (Schichtung): Instanzen aus allen Kategorien werden in Untermengen gleich verteilt
- Beispiel: Ein Datensatz mit 600 Instanzen, 250 in Klasse A und 350 in Klasse B

  Wenn 10 % der Daten fürs Testen vorenthalten werden, würde ein stratifiziertes Test Set aus 25 Instanzen aus Klasse A und 35 Instanzen aus Klasse B bestehen

Problem: Kleine Datenmengen

- Allgemein gilt: Je größer das Training Set, umso zuverlässiger die Parameter, die geschätzt werden
- Ähnlich dazu gilt für das Test Set:
  Je größer, desto genauer die Fehlerabschätzung
- Aber:
  Häufig müssen Daten im Training und Test Set im Vorhinein von Hand klassifiziert werden, was sehr aufwendig sein kann (s. Annotation)

Mehr über Evaluation

Kreuzvalidierung ("k-fold crossvalidation")

Lösung, um die Verwendung der Daten zu optimieren:

- Die Daten werden in k Untermengen gleicher Größe unterteilt (typischerweise 10)
- Es wird jeweils eine Untermenge zum Testen bei Seite gelassen, und die restlichen Daten zum Trainieren verwendet
- Die Performanz wird gemittelt
- Wenn $k = N$ ist, spricht man von "Leaving-One-Out"
- Bsp.: 20 Instanzen unterteilt in 4 *"Folds"* (grau = Test)

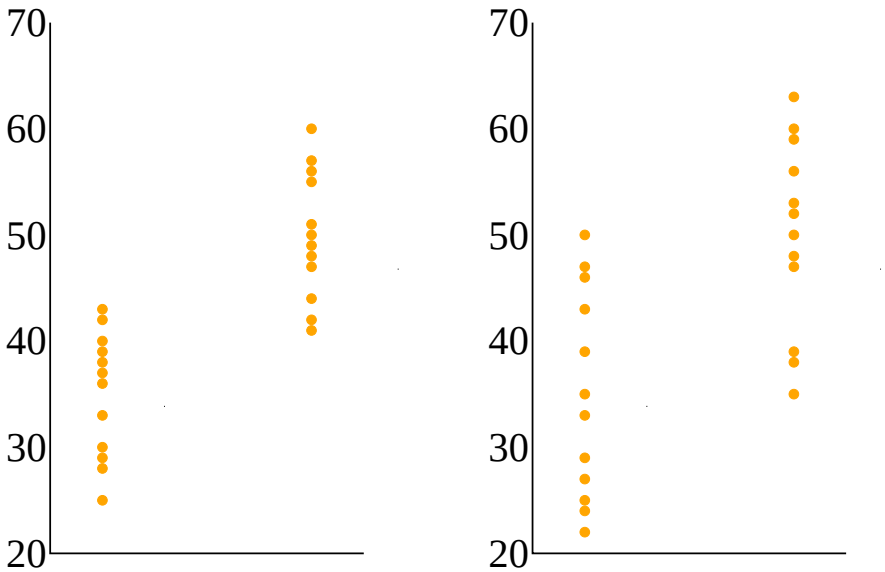
$i_1 \ldots i_5$	$i_1 \ldots i_5$	$i_1 \ldots i_5$	$i_1 \ldots i_5$
$i_6 \ldots i_{10}$	$i_6 \ldots i_{10}$	$i_6 \ldots i_{10}$	$i_6 \ldots i_{10}$
$i_{11} \ldots i_{15}$	$i_{11} \ldots i_{15}$	$i_{11} \ldots i_{15}$	$i_{11} \ldots i_{15}$
$i_{16} \ldots i_{20}$	$i_{16} \ldots i_{20}$	$i_{16} \ldots i_{20}$	$i_{16} \ldots i_{20}$

Algorithmen vergleichen

- Abhängig vom Test Set kann die Performanz eines Systems schwanken
- Daher erlaubt uns ein einziges Performanzergebnis nicht zwei Systeme zu vergleichen
- Besser ist es, wenn man das Test Set unterteilt, für jede Unterteilung die Performanz beider Systeme ausrechnet und die Ergebnisse mit einem T-Test vergleicht

Der T-Test

- Der Durchschnitt allein erlaubt uns nicht zu sagen, welches System besser ist
- Die Varianz innerhalb der Performanzergebnisse beider Systeme muss berücksichtigt werden
- Der T-Test vergleicht zwei Mittelwerte mit Rücksicht auf die Varianz in den Daten
- Berechnung: s. 1. Teil des Kurses



Schätztheorie (statistische Tests)

- Mit statistischen Tests entscheidet man, ob eine Hypothese gültig oder ungültig ist
- Da Daten empirisch erhoben werden und nur eine Stichprobe der Bevölkerung darstellen, lassen sich nie sicheren Schlüsse ziehen (Bevölkerung = Wörter, Zeichen, Sätze, ...)
- Stattdessen gibt man an, wie wahrscheinlich es ist, dass man keine Fehlentscheidung trifft
- Diese Wahrscheinlichkeit entspricht das sogenannten *Signifikanzniveau*
- Daher spricht man von einem Hypothesentest oder Signifikanztest

Beweisbarkeit

- Wenn A zu beweisen ist, geht man immer anfangs davon aus, dass $\sim A$ gilt (vgl. Schuld des Angeklagten bei einem Gerichtsverfahren)
- Nullhypothese: $H_0 = \sim A$
  Alternativhypothese: $H_1 = A$
- Nur bei überwiegenden Belegen nimmt man H_1 an, ansonsten bleibt es bei H_0
- D. h., *„False Positives"* sollen unbedingt vermieden werden (auch Fehler erster Art oder α-Fehler genannt)
- Dadurch wächst die Wahrscheinlichkeit eines *„False Negatives"* (Fehler zweiter Art oder β-Fehler)
- Negative Tatsachen oder *Nulleffekte* kann man nicht beweisen

Schätztheorie (statistische Tests)

Signifikanz: Einführendes Beispiel

- Frage: Ist eine Person hellsehfähig?
- Experiment: 25 Karten werden ihr verdeckt gezeigt; sie soll raten welche Farbe jede Karte hat
- Wir gehen davon aus, die Testperson sei nicht hellsehend:
  - Chance Baseline oder Nullhypothese: $H_0 : p = \frac{1}{4}$
  - $H_1 : p > \frac{1}{4}$
- Wenn die Farbe bei allen 25 Karten richtig benannt wird, betrachten wir die Person als Hellseher
- Bei 24 oder 23 Treffern auch
- Bei 5 oder 6 Treffern nicht
- Aber wie ist es bei 12 Treffern? Bei 17 Treffern?

Schätztheorie (statistische Tests)

Signifikanz: Einführendes Beispiel

- Wo liegt die *kritische* Anzahl an Treffern c, von der an wir nicht mehr glauben, es seien reine Zufallstreffer?
- Wie kritisch in Bezug auf α-Fehler wollen wir sein?
- Die Wahrscheinlichkeit einer Fehlentscheidung ist...

Bei $c = 25$:

$$\begin{aligned} P(H_0 \text{ ablehnen} | H_0 \text{ ist richtig}) &= P(X \geq 25 | p = \tfrac{1}{4}) \\ &= \left(\tfrac{1}{4}\right)^{25} \approx 10^{-15} \end{aligned}$$

Bei $c = 10$:

$$\begin{aligned} P(H_0 \text{ ablehnen} | H_0 \text{ ist richtig}) &= P(X \geq 10 | p = \tfrac{1}{4}) \\ &= \left(\tfrac{1}{4}\right)^{10} \approx 0,07 \end{aligned}$$

Schätztheorie (statistische Tests)

Signifikanz: Einführendes Beispiel

- Typischerweise wird eine Wahrscheinlichkeit von $0{,}01$ oder $0{,}05$ festgelegt
- Wenn wir das Signifikanzniveau festgelegt haben, können wir die Gleichung lösen und c ermitteln:

$$P(X \geq c | p = \tfrac{1}{4}) \leq 0,01 \Leftrightarrow c = 12$$

- 12 ist in diesem Beispiel der *„kritische Wert"*

Baselines

- Eine Zahl allein ist nicht sehr aussagekräftig
- Um ein Modell zu bewerten muss man auch wissen, wie schwierig das zu lösende Problem an sich ist
- Beispiel 1: Heutige Taggers erreichen 96 %–97 % richtig

  Das scheint sehr gut, aber:
  - Ein “dummer” Tagger, der immer jedem Wort seine häufigste Kategorie zuweist, erreicht 90 %
  - Ein Tagger mit 97 % Korrektheit hat 63 % Chancen alle Wörter eines 15-Wort-Satzes richtig zu annotieren
  - Ein Tagger mit 98 % Korrekt. dagegen 74 % Chancen
  - Jedes halbe Prozent zählt!
- Beispiel 2:
  - Bei mehrdeutigen Wörtern sind sich menschlichen Annotatoren nur in 65 %-70 % der Fällen einig
  - Ein automatisches System wird diese obere Schranke also sicher nicht übersteigen

Baselines

Chance Baseline

Die Performanz eines Modells, der jeder Instanz zufällig eine Kategorie zuweisen würde, z. B. bei zwei Kategorien 50 %

Frequency Baseline

Performanz, wenn allen Instanzen die häufigste Kategorie zugewiesen würde

Beispiel:
Datensatz mit 600 Instanzen,
250 in Kategorie A,
350 in Kategorie B

Erwartete Korrektheit: $\frac{350}{600} = 58\,\%$

Konfusionsmatrix

- Instanzen mit dem Modell klassifizieren
- Überprüfen, wie oft das Ergebnis mit der Annotation übereinstimmt
- Das Ergebnis kann dann als Tabelle angezeigt werden:

		Ermittelte Kategorie	
		A	$\neg A$
Tatsächliche Kategorie	A	True Positive	False Negative
	$\neg A$	False Positive	True Negative

- Die Korrektheit bzw. Fehlerrate (Anteil der korrekt bzw. falsch klassifizierten Instanzen) lassen sich auf der Diagonalen ablesen

  $\text{accuracy} = \frac{TP+TN}{N}$

Präzision und Recall

- Korrektheit und Fehlerrate allein sagen uns nicht, wie kostspielig Fehler sind
- Wir müssen bei den Fehlern zwischen *"False Positives"* und *"False Negatives"* unterscheiden
- Beispiel 1:
  - Aus einer chemischen Analyse des Meereswasser soll ermittelt werden, ob irgendwo ein Ã–lteppich liegt oder nicht
  - *"False Positive":* Die Möglichkeit eines Ã–lteppichs vorhersagen, der bei weiterer Untersuchung nicht bestätigt wird
  - *"False Negative":* Einen existierenden Teppich übersehen und keine Maßnahmen einleiten
  - False Negatives sind in diesem Fall deutlich schlimmer als False Positives

Präzision und Recall

- Beispiel 2: Spam filtern
  - Unerwünschte Werbenachrichten werden von Spam-Filter in einem sogenannten Spam- oder Junk-Ordner verschoben bzw. manchmal sogar sofort gelöscht
  - Alle verschobenen bzw. gelöschten Nachrichten sollten tatsächlich Spam sein, damit der Empfänger keine wichtige Nachricht verpasst
  - False Positive:
    Eine echte Nachricht, die als Spam klassifiziert wird
  - False Positives sind hier ein größeres Problem als False Negatives

Präzision und Recall

Präzision

- Anteil der als positiv klassifizierten Datenpunkte, die richtig sind
- Wächst bei geringer Anzahl von False Positives

$$Precision = \frac{TP}{TP + FP}$$

Recall

- Anteil der positiven Datenpunkte in den Daten, die das Modell als solche erkannt hat
- Wächst bei geringer Anzahl von False Negatives

$$Recall = \frac{TP}{TP + FN}$$

Präzision und Recall

- Beide Größen sind zwischen 0 und 1 beschränkt
- Abhängig von der Aufgabe kann Präzision oder Recall wichtiger sein:
  - Beispiel 1: Recall wichtiger als Präzision
  - Beispiel 2: Präzision wichtiger als Recall

Präzision und Recall

- Beide Größen sind zwischen 0 und 1 beschränkt
- Abhängig von der Aufgabe kann Präzision oder Recall wichtiger sein:
  - Beispiel 1: Recall wichtiger als Präzision
  - Beispiel 2: Präzision wichtiger als Recall

Der F-Measure

Wenn Precision und Recall für eine Aufgabe gleich wichtig sind, können sie im *F-Measure* kombiniert werden

$$F = \frac{2 \cdot recall \cdot precision}{recall + precision} = \frac{2TP}{2TP + FP + FN}$$