Evaluation

Caroline Sporleder

Computational Linguistics
Universität des Saarlandes

Sommersemester 2011

26.05.2011

Datensets

Warum evaluieren?

Ziel

- Wie gut funktioniert die Methode (Methode=Lernverfahren plus Attribute etc.)?
- Ist die Methode besser als andere Methoden?
- Ist die Methode unter bestimmten Umständen besser? Wenn 'ja' unter welchen? (wenig annotierte Daten, keine Vorverarbeitungswerkzeuge zur Verfügung, Laufzeit, Speicherplatz etc.)

$\Rightarrow$ verläßliche Aussagen über **zukünftige Performanz**

Naive Möglichkeit

Messen der Performanz auf dem Trainingsset

$\Rightarrow$ **Nicht gut, weil:**

Warum evaluieren?

Ziel

- Wie gut funktioniert die Methode (Methode=Lernverfahren plus Attribute etc.)?
- Ist die Methode besser als andere Methoden?
- Ist die Methode unter bestimmten Umständen besser? Wenn 'ja' unter welchen? (wenig annotierte Daten, keine Vorverarbeitungswerkzeuge zur Verfügung, Laufzeit, Speicherplatz etc.)

$\Rightarrow$ verläßliche Aussagen über **zukünftige Performanz**

Naive Möglichkeit

Messen der Performanz auf dem Trainingsset

$\Rightarrow$ **Nicht gut, weil:**

- Methode ist für Trainingsset optimiert
- Performanz ist daher artifiziell erhöht (höher als auf ungesehenen Daten)

Evaluation auf ungesehenen Daten (1)

Evaluation auf ungesehenem Test-Set ist verläßlicher, sofern:

- Test- und Trainingsset aus derselben statistischen Population stammen (d.h. vergleichbar sind)
  Achtung: was zu einer Population gehört ist manchmal schwierig zu bestimmen (alle Texte? nur Zeitungstexte? nur Wall Street Journal (WSJ)?, nur bestimmte WSJ Kategorien? etc.)
- Test- und Trainingsset statistisch unabhängig sind (ggf. vor dem Splitting Randomisieren)

Evaluation auf ungesehenen Daten (2)

Parameteroptimierung

- manchmal braucht man ein separates Datenset zur Parameteroptimierung (z.B. Pruning von Decision Trees)
- dies sollte nicht auf dem Test-Set geschehen (warum?)
  $\Rightarrow$ es ist ein drittes Datenset notwendig: **Development-Set (auch Validation-Set)**

Datengröße

Wie teilt man die annotierten Daten?

- betreute Lernverfahren brauchen genügend große Trainingsdatenmengen (abhängig von der Lernaufgabe und der gewählten Methode)
- aber die Testdaten müssen auch noch groß genug sein, um verläßliche Ergebnisse zu bekommen
- genaue Größenverhältnisse hängen von individuellen Umständen ab
  $\Rightarrow$ typisch ist **80% Trainings- und 20% Testdaten**, bei zusätzlichem Development-Set: 80%-10%-10% (oder 70%-15%-15%)

Cross-Validation (1)

Was, wenn die Datenmenge relativ klein ist?

- je kleiner das Testset, desto unwahrscheinlicher ist es, dass Testset-Performanz ein guter Indikator für allgemeine Performanz ist (Testdaten können durch Zufall unrepräsentativ sein)
- Abhilfe:

Cross-Validation (1)

Was, wenn die Datenmenge relativ klein ist?

- je kleiner das Testset, desto unwahrscheinlicher ist es, dass Testset-Performanz ein guter Indikator für allgemeine Performanz ist (Testdaten können durch Zufall unrepräsentativ sein)
- Abhilfe:
  - mehr Daten annotieren ($\rightarrow$ teuer)
  - den Trainins-Test-Split zu Gunsten des Testsets verändern ($\rightarrow$ ggf. schlechtere Ergebnisse)
  - Training und Testing mehrfach mit verschiedenen Splits wiederholen und die Durchschnittsperformanz berechnen

Cross-Validation (2)

Komplettes Datenset wird in n Folds gesplittet. Es werden n Training-Test-Durchläufe gemacht, jeweils mit einem Fold zum Testen und $n-1$ Folds zum Trainieren. Standard ist $n = 10$.

Beispiel: 3-fold Cross-Validation

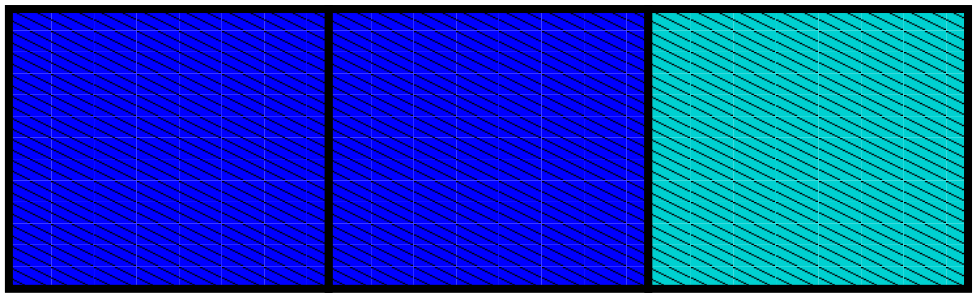
Gesamtdatenset mit 3 Folds

Cross-Validation (2)

Komplettes Datenset wird in n Folds gesplittet. Es werden n Training-Test-Durchläufe gemacht, jeweils mit einem Fold zum Testen und $n-1$ Folds zum Trainieren. Standard ist $n = 10$.

Beispiel: 3-fold Cross-Validation

Durchlauf 1



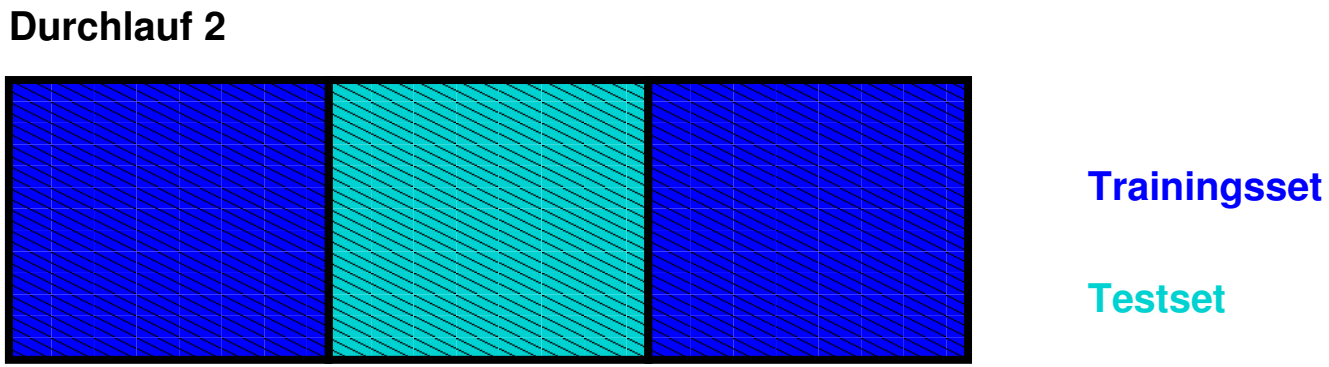
Trainingsset

Testset

Cross-Validation (2)

Komplettes Datenset wird in n Folds gesplittet. Es werden n Training-Test-Durchläufe gemacht, jeweils mit einem Fold zum Testen und $n-1$ Folds zum Trainieren. Standard ist $n = 10$.

Beispiel: 3-fold Cross-Validation

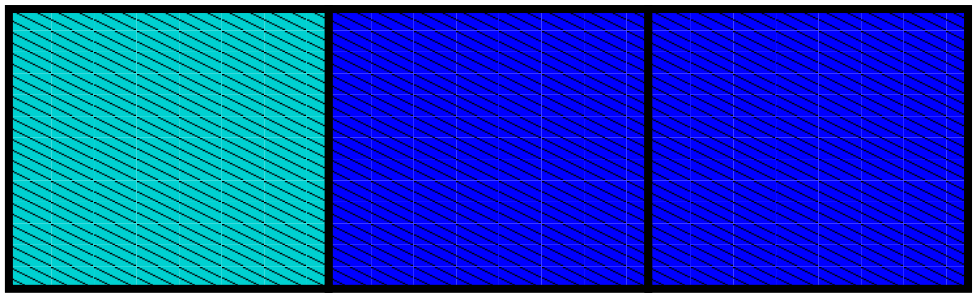


Cross-Validation (2)

Komplettes Datenset wird in n Folds gesplittet. Es werden n Training-Test-Durchläufe gemacht, jeweils mit einem Fold zum Testen und $n-1$ Folds zum Trainieren. Standard ist $n = 10$.

Beispiel: 3-fold Cross-Validation

Durchlauf 3



Trainingsset

Testset

Cross-Validation (3)

Eigenschaften

- auf allen Instanzen wird einmal getestet
- die Ergebnisse können dadurch zuverlässiger sein als bei einem zufällig augewählten, kleinen Testset (voraussgesetzt das Trainingsset ist repräsentativ für die Population)
- aber es muss sichergestellt sein, dass das Testset in jedem Durchlauf tatsächlich "ungesehen" ist (z.B. müssen Parameter für jeden Durchlauf neu auf einem Teil der jeweiligen Trainingsdaten optimiert werden).

Cross-Validation (4)

Sonderfälle

- ein Extremfall der Cross-Validation ist **Leave-One-Out Cross-Validation**, dabei n = *Anzahl der Instanzen*
  Vorteil: Ergebnisse nicht abhängig von zufälligen Datensplits
- **Stratified Cross-Validation:** Instanzen werden nicht zufällig auf Folds aufgeteilt, sondern es wird sichergestellt, dass die anteilige Verteilung der Ausgabeklasse in jedem Fold gleich ist (generell eine gute Idee; warum?)

Cross-Validation (5)

Wie wählt man ein geeignetes n?

Cross-Validation (5)

Wie wählt man ein geeignetes n?

Je größer n, desto grösser das Trainingset, aber desto länger dauern auch die Experimente

Evaluationsmaße

Performanz und Klassifikationsfehler (1)

Performanz eines Klassifikators

- (proportional) wieviele Fehler (bzw. wieviele korrekte Klassifikationen)
- was für Fehler (verschiedene Fehler können unterschiedlich teuer sein, vgl. Spam Detection)

Performanz und Klassifikationsfehler (2)

Man unterscheidet verschiedene Fehlerarten

z.B. bei einer binären Ausgabeklasse (Spam vs. kein Spam):

		Vorhergesagte Ausgabeklasse	
		positiv	negativ
tatsächliche Klasse	positiv	true positive (TP)	false negative (FN)
	negativ	false positive (FP)	true negative (TN)

Evaluationsmaße (2)

Accuracy

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Precision (P)

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

Recall (R)

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

F-Score

$$F = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}$$

wenn $\alpha = 0.5$ dann:

$$F = \frac{2\times P\times R}{P+R}$$

Evaluationsmaße (3)

Welches Maß wann?

- Accuracy ist wenig informativ, wenn eine Ausgabeklasse überproportional häufig vorkommt
  $\longrightarrow$ Accuracy ist dann auch bei einem naiven Klassifikator, der immer die Mehrheitsklasse zuweist, hoch
- Precision, Recall, F-Score lassen sich direkt auf binäre Klassifikationsprobleme anwenden, bei mehreren Ausgabeklassen muß man die Definition erweitern (mehr später)

Evaluationsmaße (4)

Beispiel

Eine Wahrheitsmatrix (engl. confusion matrix) listet auf, wie oft welche Ausgabewerte vom Klassifikator für welche tatsächlichen Ausgabewerte vorhergesagt werden.
Ein Klassifikator hat die folgende Confusion Matrix:

		Vorhergesagte Ausgabeklasse	
		positiv	negativ
tatsächliche Klasse	positiv	500	100
	negativ	500	10,000

Ist der Klassifikator gut oder schlecht?

- precision=50%
- recall=83%
- f-score=62%
- accuracy=95%

F-Score für multiple Ausgabeklasse

Der F-Score wird für jede Ausgabeklasse, $c_i \in C$, separat berechnet. Dabei ist:
$c_i = positive$ und $\{c_j \in C | c_j \neq c_i\} = negativ$
Der F-Score über alle Klassen ist dann der Durchschnitt, berechnet entweder als macro averaged F-score oder micro averaged F-score.

Macro Averaged F-Score

$\frac{\Sigma_{c_i \in C} F(c_i)}{|C|}$ wobei $|C|$ die Anzahl der Ausgabeklassen ist und $F(c_i)$ der F-Score für Klasse c_i.

Micro Averaged F-Score

$\frac{\Sigma_{c_i \in C} F(c_i) \times |c_i|}{|I|}$ wobei $|c_i|$ die Anzahl der Instanzen mit Ausgabeklasse c_i ist und $|I|$ die Anzahl der Instanzen insgesamt im Testset.

Weitere Maße

Neben accuracy, precision, recall und f-score gibt es noch eine Anzahl weiterer Maße, die in NLP allerdings weniger häufig verwendet werden.

Zum Beispiel kann man sogenannte ROC curves (Receiver Operating Charecteristic) berechnen, die für alle Parameterwerte den Recall gegen die Falsch-Positiv-Rate ($\frac{FP}{TN+FP}$) darstellen. Die Fläche unter der Kurve (Area under Curve, AUC) kann dann als parameterunabhängiges Performanzmaß aufgefaßt werden (je größer desto besser).

Signifikanz

Zuverlässigkeit von Ergebnissen (1)

Ziel der Evaluation ist es die vermutliche Erfolgsrate (engl. success rate) eines Klassifikators auf der Gesamtpopulation herauszufinden.

Beispiele

Zuverlässigkeit von Ergebnissen (1)

Ziel der Evaluation ist es die vermutliche Erfolgsrate (engl. success rate) eines Klassifikators auf der Gesamtpopulation herauszufinden.

Beispiele

- Wie hoch ist der Anteil der von einem Spamfilter korrekt als Spam-vs.-Nicht-Spam klassifizierten Emails in der Population aller Emails der Welt (oder eines bestimmten Nutzers)?
- Wie hoch ist der Anteil der korrekten Part-of-Speech Tags eines bestimmten Taggers in der Gesamtpopulation aller Nachrichtentexte?
- Wie hoch ist der Anteil der von einem Named-Entity-Tagger korrekt erkannten NEs in der Gesamtpopulation der biomedizinischen Texte zum Thema 'Tuberkulose'?

Zuverlässigkeit von Ergebnissen (2)

Evaluiert wird jedoch auf einem Sample der Gesamtpopulation, dem Testset. Wie wahrscheinlich ist es, dass die Erfolgsrate auf dem Testset der Erfolgsrate auf der Gesamtpopulation entspricht?

Beispiel

Ein Klassifikator erreicht auf dem Testset 75% Accuracy. Wie wahrscheinlich ist es, dass dies der tatsächlichen Accuracy entspricht?

Faktoren, die die Verlässlichkeit der Ergebnisse beeinflussen

Zuverlässigkeit von Ergebnissen (2)

Evaluiert wird jedoch auf einem Sample der Gesamtpopulation, dem Testset. Wie wahrscheinlich ist es, dass die Erfolgsrate auf dem Testset der Erfolgsrate auf der Gesamtpopulation entspricht?

Beispiel

Ein Klassifikator erreicht auf dem Testset 75% Accuracy. Wie wahrscheinlich ist es, dass dies der tatsächlichen Accuracy entspricht?

Faktoren, die die Verlässlichkeit der Ergebnisse beeinflussen

- die Auswahl des Testsets (wie repräsentativ ist das Testset für die Gesamtpopulation?)
  - zufällig?
  - Verteilung der Ausgabeklassen im Testset verglichen mit Gesamtpopulation?
- die Größe des Testsets

Zuverlässigkeit von Ergebnissen (3)

Beispiel

Euer Spamfilter erreicht auf dem Testset 75% Accuracy. Wie sicher könnt ihr sein, dass der Spamfilter auch auf der Gesamtpopulation aller Emails eine Erfolgsrate von 75% hat?

Intuitiv

- wenn dass Testset 100 Instanzen enthält, könnt ihr euch nicht sehr sicher sein
- wenn dass Testset 100.000 Instanzen enthält, könnt ihr euch relativ sicher sein

$\Rightarrow$ **Wie kann man die Zuverlässigkeit der** auf dem Testset erreichten **Erfolgsrate mathematisch berechnen?**

Zuverlässigkeit von Ergebnissen (4)

Erfolgsrate als Zufallsvariable

- jeder **Klassifikationsversuch** kann als **ein Bernoulli-Experiment** angesehen werden
- die Gesamtheit, N, der Klassifikationen der Testinstanzen ist damit eine Sequenz von Bernoulli-Experimenten
- d.h. wir können uns vorstellen, dass es eine **Zufallsvariable**, X, gibt, **die die Anzahl der Klassifikationserfolge in N Versuchen modelliert**.
- X hat dabei eine verteilung
- **für große N** (=großes Testset) **approximiert X die Normalverteilung**
- unter der Voraussetzung, dass X normal verteilt ist, gibt es eine mathematische Formel um **Konfidenzintervalle** für ein bestimmtes Ergebnis zu berechnen
- zudem gibt es statistische Tests, um die Wahrscheinlichkeit zu berechnen, dass die ermittelte Erfolgsrate einer bestimmten tatsächlichen Erfolgsrate entspricht (Einstichprobentest)

Zuverlässigkeit von Ergebnissen (4)

Erfolgsrate als Zufallsvariable

- jeder **Klassifikationsversuch** kann als **ein Bernoulli-Experiment** angesehen werden
- die Gesamtheit, N, der Klassifikationen der Testinstanzen ist damit eine Sequenz von Bernoulli-Experimenten
- d.h. wir können uns vorstellen, dass es eine **Zufallsvariable**, X, gibt, **die die Anzahl der Klassifikationserfolge in N Versuchen modelliert**.
- X hat dabei eine Binomialverteilung
- **für große N** (=großes Testset) **approximiert X die Normalverteilung**
- unter der Voraussetzung, dass X normal verteilt ist, gibt es eine mathematische Formel um **Konfidenzintervalle** für ein bestimmtes Ergebnis zu berechnen
- zudem gibt es statistische Tests, um die Wahrscheinlichkeit zu berechnen, dass die ermittelte Erfolgsrate einer bestimmten tatsächlichen Erfolgsrate entspricht (Einstichprobentest)

Vergleichen von Ergebnissen

Konfidenzintervalle für ein Ergebnis sind nützlich, oft ist es aber interessanter zwei Ergebnisse miteinander zu vergleichen (Zweistichprobentests).

Beispiel

Auf einem gegebenen Testset hat Klassifikator A hat eine Erfolgsrate/Accuracy von 67%, während Klassifikator B eine Erfolgsrate von 69% hat. Ist B wirklich besser als A?

$\Rightarrow$ **es gibt verschiedene statistische Tests um Ergebnisse zu vergleichen**

Statistische Tests (1)

Grundprinzip

Um zu zeigen, dass etwas wahr ist, nimmt man das Gegenteil an (Nullhypothese) und berechnet dann, mit welcher Wahrscheinlichkeit die Nullhypothese abgelehnt werden kann.

Beispiel

Nullhypothese: Klassifikator A und B haben eine identische Performanz, d.h. die zugrunde liegenden Zufallsvariablen sind gleich: $X_A = X_B$.

Statistische Tests (2)

Statistische Tests schätzen die Irrtumswahrscheinlichkeit ab, d.h. die Wahrscheinlichkeit, **dass die Nullhypothese fälschlicherweise abgelehnt wird**. Die maximal zulässige Irrtumswahrscheinlichkeit wird als Signifikanzniveau (engl. significance level), α, bezeichnet. Zum Beispiel wird bei $\alpha = 0,05$ die maximale Wahrscheinlichkeit, dass die Nullhypothese fälschlich abgelehnt wird auf 5% gesetzt.

Das Signifikanzniveau kann (theoretisch) frei gewählt werden. In der Praxis wird oft $\alpha = 0,05$ (schwach signifikant) oder $\alpha = 0,01$ (stark signifikant) gesetzt.

Statistische Tests berechnen einen sogenannten p-Wert. Wenn der p-Wert kleiner als α ist, kann die Nullhypothese auf dem festgelegten Signifikanzniveau abgelehnt werden.

Statistische Tests (3)

Achtung: Statistische Tests geben nur Hinweise, keine absoluten Wahrheiten. D.h. statistische Signifikanz muss nicht zwangsläufig bedeuten, dass die zugrunde liegenden Distributionen tatsächlich unterschiedlich sind. Umgekehrt können zwei Ergebnisse, die nicht signifikant unterschiedlich sind, trotzdem aus verschiedenen Distributionen stammen.

Beispiel: t-Test (1)

Wir haben zwei Klassifikatoren A und B. Für beide lassen wir auf einem gegebenen Testset ein Cross-Validation Experiment laufen.

Aus den Ergebnissen können wir die durchschnittliche Erfolgsquote für beide Klassifikatoren berechnen: $\overline{x}_A$ und $\overline{x}_B$. Ebenso können wir die Varianz jedes Klassifikators abschätzen: s_A^2 und s_B^2.

Der sogenannte t-Wert berechnet sich dann, wie folgt:

$$t = \frac{\overline{x}_A - \overline{x}_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}}$$

(wobei n_A und n_B die Anzahl der Folds für A und B sind)

Der p-Wert kann dann anhand des t-Wertes in einer statistischen Tabelle nachgeschaut werden.

Beispiel: t-Test (2)

Wenn die Cross-Validation Folds für beide Klassifikatoren konstant gehalten werden, kann man auch den t-Test für **gepaarte Stichproben** anwenden. Dieser wird etwas anders berechnet und ist empfindlicher als der ungepaarte t-Test.

t-Test vs. χ^2-Test

t-Test

- setzt eine Normalverteilung der Ergebnisse voraus
- ist auch für kleine Samplegrössen zuverlässig

χ^2-Test

- setzt keine Normalverteilung voraus
- nicht gut für kleine Samplegrössen

Neben t-Test und χ^2-Test gibt es noch eine ganze Reihe weiterer statistischer Tests.

Zusammenfassung

Was ihr gelernt haben solltet

Experimentelles Design

- welche Datensets brauche ich unter welchen Bedingungen?
- welchen Einfluß hat die Größe eines Datensets?
- wie wähle ich Test- und Trainingsdaten aus?
- Cross-Validation und Leave-one-out

Evaluation

- welche Evaluationsmaße gibt es?
- wann wird welches Maß angewendet?
- wie kann man die Zuverlässigkeit von Ergebnissen abschätzen?