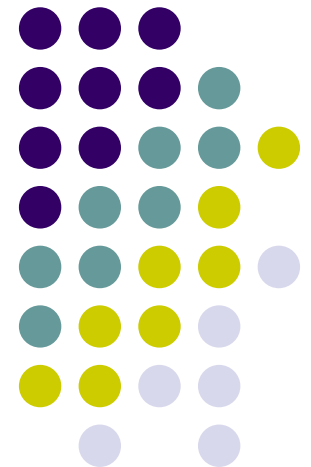


Incremental Learning from Noisy Data

Im Rahmen des Seminars:

Seminar aus maschinellem Lernen –
Learning from Data Streams

WS 2006/2007



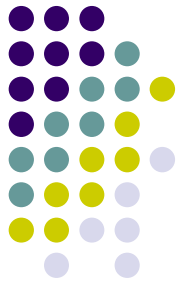
Peter Jaremenko 10.01.2007

Thema & Herausforderung



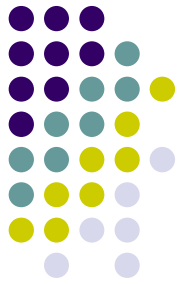
- Einflussfaktoren auf reale Systeme, Beschreibungskonzepte bei rauschenden Daten schwierig zu verwirklichen
- Problematik verschärft durch Änderung im Konzept
- Herausforderung an intelligente Mechanismen sich der Umwelt anzupassen

Inhalte der Präsentation



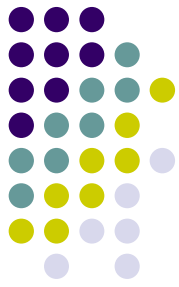
- Grundlagen
- Realisierung durch STAGGER
- Anwendungsbeispiel Wettervorhersage
- Noisy data & concept drift
- Verwandte Arbeiten
- Zusammenfassung

Paper zur Thematik



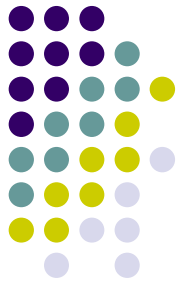
- Vorgehensweise zur Erlangung von Lerntechniken fällt unter den Begriff „concept attainment“ (Konzepterlangung):
 - Im voraus Aufteilung des Untersuchungsgegenstandes in mind. zwei Kategorien
 - Ziel des Lernens ist nützliche Beschreibung, um die Kategorie einer zuvor nicht gesehenen Instanz vorausszusagen – aus Beispielen lernen
 - Lernende muss mit Toleranz umgehen können, jedoch auch Beschreibung zu Weilen revidieren
 - Umsetzung erfolgt durch das Programm STAGGER

Komponenten zum Lernen aus Erfahrung



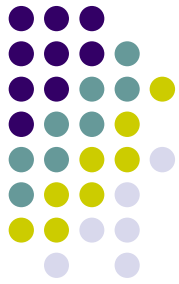
- 1. Clustering (gruppieren von Objekten in Klassen)
- 2. Initialization (Erstellen von initialen Beschreibungen der gruppierten Objekte)
- 3. Projection (Übereinstimmung von Beschreibung und nachfolgender Erfahrung)
- 4. Evaluation (Bestimmung der Effektivität der Beschreibung)
- 5. Refinement (Verbesserung durch Modifikation)
- 6. Aggregation und 7. Storage

Realisierung in STAGGER



- Clustering Lösung: Einteilung in positive und negative Vertreter (üblich bei Konzepterlangungssystemen)
- Eingabe besteht aus Beschreibungen von Instanzen anhand von Merkmalen mit Kennzeichnung der Klasse

STAGGER - Initialization

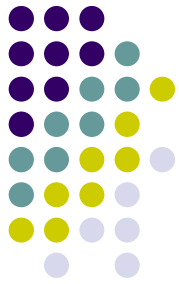


- Initiale Konzeptbeschreibung ist Sammlung einfachster Eigenschaften – diese besitzen Paare von Gewichten
- Elemente werden anhand boolischer Funktionen mit gewichteten Attributwerten beschrieben
- Bsp.: (size = small and color = red) or shape = (circle or triangle)

Characterization	Weights	
size = small	1	1
size = medium	1	1
size = large	1	1
color = red	1	1
.	.	.
.	.	.
.	.	.
shape = triangle	1	1



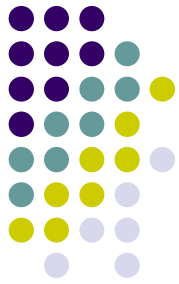
STAGGER - Projection



- Übereinstimmung beschriebener Konzepte gegenüber neuer Instanzen
- Bei Übereinstimmung wird die Erwartung einer positiven Instanz erhöht
- STAGGER benutzt Bayessche Statistiken (Schlussfolgerungen bei unsicherem Wissen)
- Zwei Formeln elementar:
 - Logical Sufficiency (LS) misst den Grad, inwiefern die Gegenwart einer Eigenschaft die Erwartung für ein Ergebnis erhöht
 - Logical Necessity (LN) misst den Grad, inwiefern das Fehlen einer Eigenschaft die Erwartung für ein Ergebnis verringert

$$LS = \frac{p(F|O)}{p(F|\neg O)}$$

$$LN = \frac{p(\neg F|O)}{p(\neg F|\neg O)}$$



STAGGER – Projection(2)

- Bei Werten von 1 ist die Eigenschaft irrelevant für das Ergebnis (Unabhängigkeit)
- Bei der Projection werden Erwartungswerte für eine Klassenzugehörigkeit durch Multiplikation der bisher bekannten Wahrscheinlichkeit für eine positive Instanz mit den LS Gewichtungen aller „gematchten“ Beschreibungen sowie den LN Gewichtungen aller „ungematchten“ Beschreibungen multipliziert
- Das ermittelte Ergebnis entspricht der Wahrscheinlichkeit, die ein positive Instanz begünstigt (Maßzahl eines Prototypen)

$$Odds(positive|instance) = Odds(positive) \times \prod_{\forall matched} LS \times \prod_{\forall unmatched} LN$$

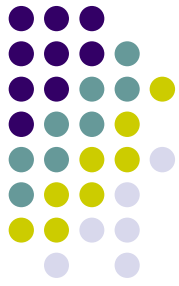
STAGGER – Projection (3)



- Besonderheiten:
 - STAGGER versucht möglichst viele Konzeptbeschreibungen mit einer Instanz abzugleichen
 - Entsprechend der Berechnungen werden sowohl Gewichtungen einzelner Merkmale als auch deren Strukturen modifiziert (ausgereiftes Konzept)

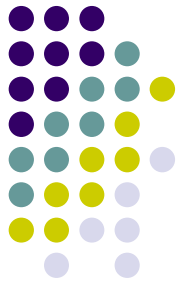


STAGGER – Evaluation

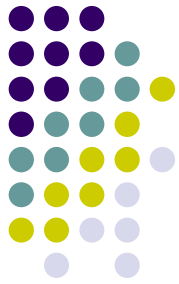


- Beschreibungen werden kontinuierlich durch Anpassungen der Gewichte bewertet
- Bewertung basiert auf psychologischen Erkenntnissen sowie dem Erfolg/Misserfolg der Vorhersage
- Tierversuche sowie Menschenversuche über Zusammenhang zwischen unbekanntem Reiz und unangenehmem Reiz
- Ergebnis der Untersuchung : Wahrscheinlichkeit bzw. Erwartungswert des unangenehmen Reizes muss in Gegenwart eines neuen Reizes größer sein als beim Abwesendheit eines neuen Reizes

STAGGER – Evaluation (2)



- Vier unterschiedliche Fälle des Zufallsexperimentes unterscheidbar
 - Perfekter Zusammenhang zwischen neuem und unangenehmem Reiz (kein rauschen)
 - Partial Reinforcement: unangenehmer Reiz folgt immer dem neuen Reiz, jedoch tritt der neue Reiz auch alleine auf – dennoch Lernen möglich
 - Partial Warning: Der neue Reiz führt immer zu einem unangenehmen Reiz, jedoch tritt der unangenehme Reiz auch alleine auf – Lernen auch möglich
 - Neuer und unangenehmer Reiz treten sowohl gemeinsam als auch isoliert voneinander auf – Lernversuche scheitern bereits bei wenigen ungepaarten Beobachtungen

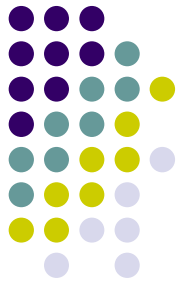


STAGGER – Evaluation (3)

- Abgleich zwischen Beschreibungen und Instanzen
- Positive Instanzen haben positive Beweiskraft und können die Vorhersage entweder bestätigen (Übereinstimmung) oder schwächen, entsprechend negative Instanzen
- „Positiv infirming instance“ bezeichnet als „error of omission“
- „negativ infirming instance“ bezeichnet als „error of comission“

Instance	Characterization	
	Matched	Unmatched
Positive	Confirming (C_p)	Infirming (I_p)
Negative	Infirming (I_N)	Confirming (C_N)

STAGGER – Evaluation (4)



- Verbindung der vorgestellten Situationen zu den Gewichtungen LS und LN

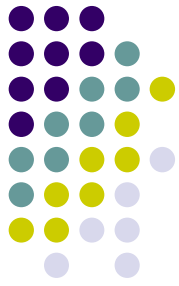
$$LS = \frac{C_P(I_N + C_N)}{I_N(C_P + I_P)} \quad LN = \frac{I_P(I_N + C_N)}{C_N(C_P + I_P)}$$

- Ebenso kann die bisherige Wahrscheinlichkeit mit Hilfe des Quotienten aus der Summe der positiven Beweise durch die Summe der negativen Beweise ermittelt werden:

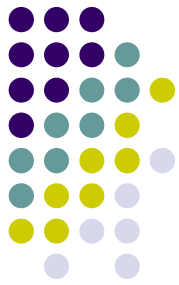
$$(C_P + I_P)/(I_N + C_N)$$



STAGGER – Refinement

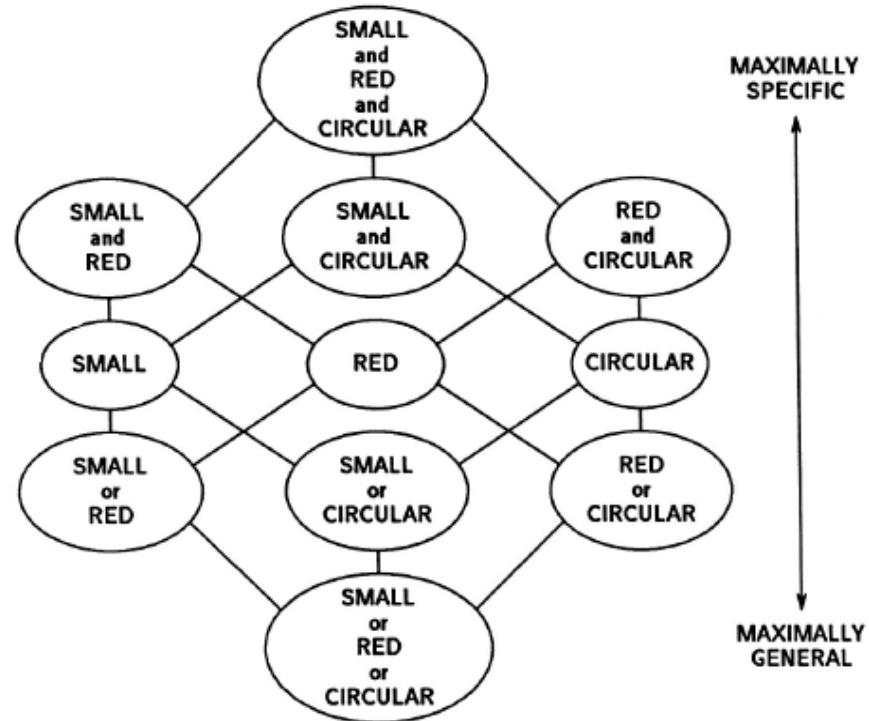


- Beschränkung des Lernens auf Anpassung der Gewichte von Beschreibungen lediglich ausreichend bei „linear teilbaren“ Konzepten
- Einzelne Beschreibungen werden zu komplexen boolischen Funktionen kombiniert, Ziel ist die Effektivität der Größen bei der Evaluation zu steigern
- Änderungen in der Struktur werden durch Fehler bei der Projektion ausgelöst und durch Heuristiken der Bayesschen Statistik unterstützt
- Zur Verbesserung der bestehenden Beschreibung wird der Suchraum angepasst und Beschreibungselemente modifiziert bzw. neue hinzugefügt



STAGGER – Refinement (2)

- Jeder Attributwert der Beschreibung kann als Knoten einer boolischen Funktion dargestellt werden – spezifische Beschreibungen durch „und“, allgemeine durch „oder“ verknüpft
- Initiale Konzeptbeschreibung besteht aus einfachen Elementen
- Drei Suchoperatoren entsprechen Spezialisierung, Generalisierung sowie Invertierung der Beschreibungen



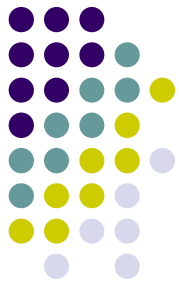


STAGGER – Refinement (3)



- Anhand des Fehlers in der Projektion kann eine Aussage über notwendige Änderungen vorgeschlagen werden
- Neben dem Fehler haben auch die Operatoren der bestehenden Konzeptbeschreibung Einfluss auf die Vorgehensweise
- Zwei Heuristiken werden für die Auswahl guter Komponenten angewendet:
 - Nominierung von Kandidaten
 - Auswahl entsprechend der Situation (nach LN und LS Richtwerten)

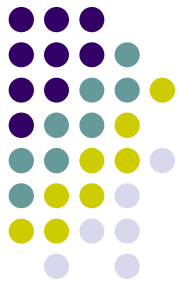
Projection	Actually	Error type	Search direction	Boolean function
Positive	Negative	Commission	Specialize	AND[c1, c2]
Negative	Positive	Omission	Generalize	OR[c1, c2]
—	—	Either	Invert	NOT[c]



STAGGER – Refinement (4)

- Neue Beschreibungen werden durch generieren und testen in den Suchbereich aufgenommen
- „Pruning“ erfolgt, falls sich eine beteiligte Beschreibungs-komponente nicht im Suchbereich befindet und sofern nicht mindestens eine neue Komponente in die Beschreibung aufgenommen wird
- Grenzwerte im Zusammenhang mit Näherungsverfahren unterstützen die Suche, Tests überprüfen die Wirksamkeit und leiten ggf. ein „Backtracking“ ein.

Boolean function	Threshold	Prune	Establish	Impeach
AND [c1, c2]	$T = \min\{LN(c1), LN(c2)\}$	$\leftarrow T$	$< T$	$> T$
OR [c1, c2]	$T = \max\{LS(c1), LS(c2)\}$	$\rightarrow T$	$> T$	$< T$
NOT [c]	$T = 1/LN(c)$ $T = 1/LS(c)$	$\leftarrow T$ $\rightarrow T$	$< T$ $> T$	$> T$ $< T$



Beispiel Wettervorhersage

- Wettervorhersage untersteht ständigen Schwankungen und umfasst auch Konzeptänderungen
- Mögliche Beschreibung von STAGGER für die 14 Instanzen: (humidity=normal and windy=false) or outlook=overcast

Outlook	Attributes			Instance
	Temperature	Humidity	Windy	
Sunny	Hot	High	True	Negative
Sunny	Hot	High	False	Negative
Sunny	Mild	High	False	Negative
Sunny	Mild	Normal	True	Positive
Sunny	Cool	Normal	False	Positive
Overcast	Hot	High	False	Positive
Overcast	Hot	Normal	False	Positive
Overcast	Mild	High	True	Positive
Overcast	Cool	Normal	True	Positive
Rain	Mild	High	True	Negative
Rain	Mild	High	False	Positive
Rain	Mild	Normal	False	Positive
Rain	Cool	Normal	True	Negative
Rain	Cool	Normal	False	Positive



Wettervorhersage (2)

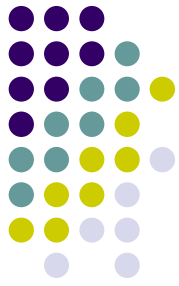


- Beschreibung eines initialen Wetterkonzeptes durch Messgrößen
- Für eine neue Instanz sind LS-Gewichte der „gematchten“ mit den LN-Gewichten der „ungematchten“ Beschreibungen zu multiplizieren

C_p	Characterization			LS	LN
	C_N	I_p	I_N		
outlook = sunny 10.48	10.78	15.35	12.99	0.74	1.31
outlook = overcast 12.56	14.49	13.28	10.00	1.19	0.87
outlook = rain 11.75	12.99	14.17	10.78	1.00	1.00
temperature = cool 11.36	13.51	14.48	10.78	0.99	1.01
temperature = mild 12.58	13.43	13.26	10.57	1.11	0.92
temperature = hot 11.32	11.55	15.07	12.45	0.83	1.19
humidity = normal 14.19	13.51	11.64	10.78	1.24	0.81
humidity = high 11.64	10.78	14.19	13.51	0.81	1.24
windy = true 10.65	12.00	15.18	11.77	0.83	1.16
windy = false 15.18	11.77	10.65	12.00	1.16	0.83



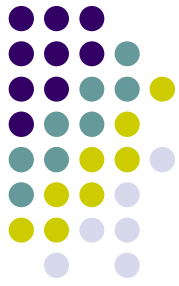
Wettervorhersage (3)



- Bsp.: outlook = rain, temperature = cool, humidity = normal and wind = true – Instanz ist negativ
- Für jedes Merkmal wird die negative Beweiskraft entweder bestätigt oder abgeschwächt – bei Merkmal „gematcht“ (outlook = rain) wird I_N erhöht, bei Merkmal „ungematcht“ (outlook = overcast) wird C_N erhöht
- Einleiten des Refinement für ein spezifischere Beschreibung als Folge des „error of commission“ – das niedrigste „gematchte“ LN-Merkmal wird verknüpft mit dem niedrigsten „ungematchten“ LS-Merkmal (humidity = normal and windy = false wird ergänzt)
- Grenzwert, um im Suchbereich zu verbleiben ist $\min(LN)$ der zwei unterstützenden Merkmale ($\min\{0,81;0,83\} = 0,81$) – folgend entsprechendes „Pruning“



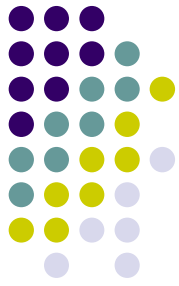
Toleranz von noisy data



- In Realwelt wie bei Wetter Ungenauigkeiten kaum vermeidbar
- Unterteilung in systematische und zufällige Schwankungen
- Bsp. einer zufälligen Schwankung Thermometer, welches beeinflusst durch Umgebung manchmal zu hohe und manchmal zu niedrige Temperaturen anzeigt
- Systematische Schwankungen entstehen bspw. bei einem Leck in der Regentonne, zeigt nie mehr Regen an – entweder nur positive oder nur negative Abweichungen
- Wenn alle Instanzen nur noch zufälligen Schwankungen unterworfen sind, besteht kein Unterschied mehr zwischen Konzept und „Rauschen“



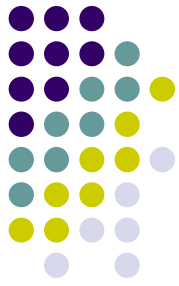
Bewertung von noisy data



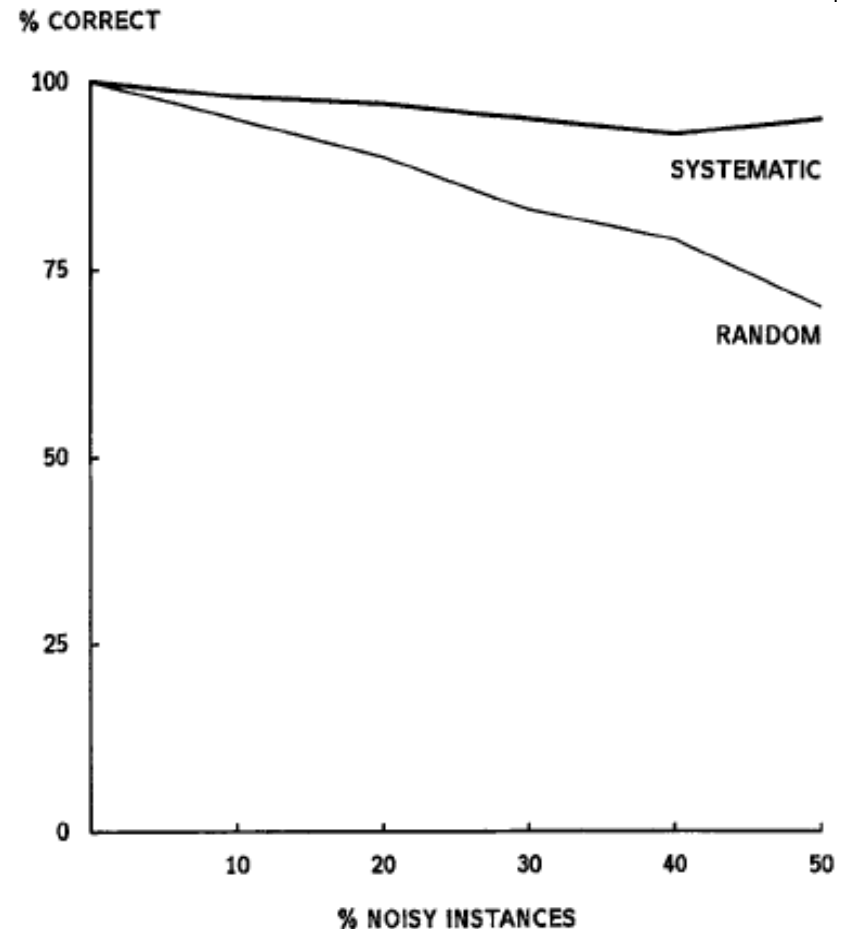
- Die Maßgrößen LS und LN bewerten systematische Schwankungen mit sehr großen Werten, zufällige Schwankungen hingegen mit kleinen Werten
- Unterschiede bestehen bei der Toleranz gegenüber positiven und negativen „infirmität“ – LS toleriert positives „infirmität“, LN toleriert hingegen negatives „infirmität“
- Auf diesen Analysen basiert die Heuristikenwahl von LN bei „und-Verknüpfungen“ und von LS bei „oder-Verknüpfungen“
- Die Toleranz gegenüber jeweils einer Art „infirmität“ deckt sich mit den Zufallsexperimenten
 - in drei Fällen war ein Lernen möglich -dabei trat maximal auch nur eine Art „infirmität“ auf
 - der vierte Fall entspricht der Intoleranz von LS und LN bei Vermischung



Performance bei noisy data



- Betrachtung einer simplen Verknüpfung
- Performance bei einer Art „infirming“ besser, fällt mit steigendem „Rauschen“
- Auch bei zufälligen Abweichungen zumutbar





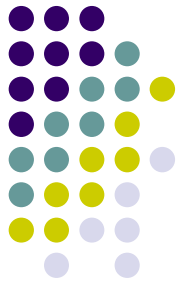
Concept drift



- Fähigkeit auf Veränderungen in der Umwelt zu reagieren ist elementares Merkmal beim Erlernen von Mechanismen (Bsp. Fuchs und seine Beute)
- Unterscheidung zwischen Zufall und wahren Wechsel notwendig
- Kenntnisse des alten Konzeptes zum Erlernen des neuen Konzeptes ausnutzen
- STAGGER realisiert Konzeptwechsel bei der Evaluation durch den „impeach“ Wert

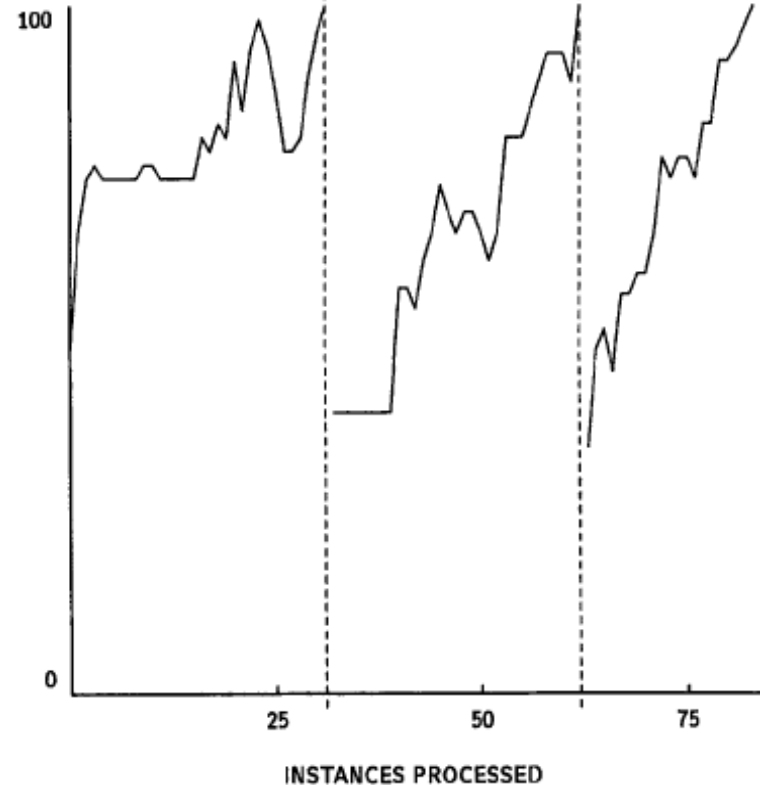


Concept drift (2)



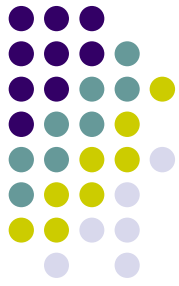
- Verfolgung eines concept drift
 - (1) size = small and color = red
 - (2) color = green or shape = circular
 - (3) size = (medium or large)
- Gestrichelte Linie markiert den Konzeptwechsel
- Frage nach der Empfindlichkeit der Lernmethode von der Reihenfolge der Instanzen
- Lernmethoden durch disjunkte Beschreibungen von vermischten Instanzen zum Teil besser anwendbar

% CORRECTLY CLASSIFIED





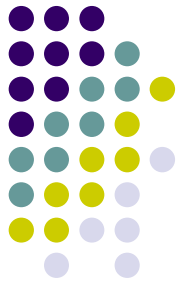
Verwandte Arbeiten



- Eingesetzte Lernmethoden von STAGGER ähnlich dem ACT* Modell von Anderson
- Beide benutzen eine gewichtete Konzeptbeschreibung und unterstützen eine bidirektionale Suche von einfachen Konzepten hin zu allgemeineren sowie spezifischeren Konzepten
- Das Konzept von ACT* besteht aus Erzeugnisregeln, welche eine Handlung spezifizieren sowie einer Situation, unter welchen Umständen die Handlung geeignet ist
- Zwei verschiedene Arten von Modifikationen der Lernkomponente: „compilation“ und „tuning processes“
- Compilation processes führt mehrmalige Erzeugnisse zu längeren Handlungssequenzen zusammen und ersetzt des Weiteren Variablen durch spezifische Werte
- Die tuning processes generalisieren, spezialisieren und verstärken bestehende Erzeugnisse, welche dann gegen andere konkurrieren



Verwandte Arbeiten (2)



- Güte der Erzeugnisse wird bei Anwendung und entsprechend des Erfolges angepasst
- Erweiterung von Langley durch Verbesserung der Toleranz gegenüber „noise“ in einem Lernsystem, verfolgt ebenso einfache Konzeptveränderungen
- ID3 Beschreibung wird als Entscheidungsbaum dargestellt
- Aufgrund Baumstruktur Probleme mit refinement und concept drift
- Der genetische Algorithmus von Holland dupliziert und kombiniert „Bit-Muster“ miteinander, Einsatzhäufigkeit wird durch Erfolge bei Projektion und Evaluation bestimmt
- Stark verbreitete Muster erlangen unter Umständen zu viel Dominanz (Suboptimale Ergebnisse)

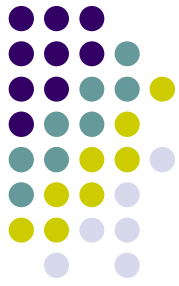
Zusammenfassung



- Programm STAGGER wurde im Hinblick auf Komponenten zum Lernen aus Erfahrungen untersucht
- Projektion gelernter Konzepte auf neue Instanzen
- Evaluierung der Beschreibungen und suche nach geeigneteren Beschreibungen (Unterscheidung nach zufälligen und systematischen Abweichungen)
- Refinement durch Generalisierung, Spezialisierung und Invertierung sowie entsprechende Heuristiken
- Backtracking erlaubt Reaktion sowohl auf zufällige als auch tatsächliche Konzeptveränderungen
- STAGGER unterstützt fundamentale, robuste Mechanismen, welche beim Lernen sowohl Rauschen als auch Konzeptveränderungen berücksichtigen



Quellen

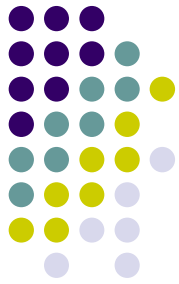


- J. C. Schlimmer, R. H. Granger (jr): „**Incremental Learning from Noisy Data**“. Departement of Information and Computer Science, University of California, Irvine, 1986.
- J. R. Anderson: „**The architecture of cognition**“. Harvard University Press, Cambridge, 1983.
- J. D. Easterlin: „**A survey of concept formation in machine learning**“. Departement of Information and Computer Science, University of California, Irvine, 1985.
- J. H. Holland.: „**Adaption in natural and artifical systems**“. University of Michigan Press, Ann Arbor, 1975.



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vortragsende



- Vielen Dank für die Aufmerksamkeit!
- Fragen ???