INF FAKULTÄT FÜR INFORMATIK

Intelligente Systeme

Fallbasiertes Schließen

Prof. Dr. R. Kruse **C. Braune**

{kruse,cbraune}@iws.cs.uni-magdeburg.de

Institut für Wissens- und Sprachverarbeitung

Fakultät für Informatik

Otto-von-Guericke Universität Magdeburg

Motivation

In regelbasierten Systemen:

Regeln als Repräsentanten von Wissen

Regeln abstrahieren vom speziellen Kontext, drücken generisches Wissen aus

Regelbasis wird durch Wissensakquisition aufgebaut

- schwer automatisierbarer Prozess
- zeitaufwendig, teuer (Experten)
- Problembereich muss formalisierbar sein, das ist jedoch nicht immer gegeben

Die Regelbasis wird mit Hilfe des *Schließens* nutzbar gemacht

Motivation

Fallbasiertes Schließen (Cased-based reasoning, CBR)

Erfahrungswissen aus konkreten Beispielen und Situationen nutzen

Motivation

Wissensbasis besteht nicht aus generischen Regeln, sondern aus einer Sammlung von Fällen (cases)

Fälle stellen spezifische frühere Erfahrungen dar

Problemlösung:

- suche relevantesten Fall aus der Wissensbasis
- übertrage dessen Lösung geeignet auf das neue Problem

Kein regelbasierter, sondern *erinnerungsbasierter* Prozess

Grundzüge des CBR

CBR wird benutzt, um:

- eine neue Aufgabe zu bewältigen,
- eine Lösung an eine neue Situation anzupassen,
- vor falschen Entscheidungen zu warnen,
- eine Situation zu analysieren.

Es beruht auf zwei grundsätzlichen Annahmen:

- Ähnliche Probleme haben ähnliche Lösungen.
- Jedes Problem ist anders, aber der Typ der Aufgabenstellung wiederholt sich.

Erinnern ist nützlich, und Erfahrungen dürfen genutzt werden.

Vorteile einer fallbasierten Vorgehensweise

Schnelle Präsentation von Problemlösungen

Übertragung von Problemlösungen auf andere Bereiche

Evaluation von Problemlösungen, auch wenn eine algorithmische Methode fehlt

Schwach strukturierte und vage Konzepte lassen sich interpretieren

Wesentliche Charakteristika einer Aufgabe werden deutlich

Warnung vor Fehlschlägen

Behandlung komplexer Aufgabenstellungen

Qualität eines CBR-Systems

Die Qualität eines CBR-Systems ist hauptsächlich abhängig von:

- seiner Erfahrung, also dem Umfang der Fallbasis;
- seiner Fähigkeit, neue Situationen mit gespeicherten Fällen geschickt in Verbindung zu bringen;
- seiner Fähigkeit, alte Lösungen an neue Probleme anzupassen;
- der Art und Weise, wie neue Lösungen evaluiert und bei Bedarf korrigiert werden;
- der geschickten und passenden Integration neuer Erfahrungen in die Fallbasis.

Anwendungsgebiete

Rechtsprechung:
Die Argumentation erfolgt häufig und explizit nach der Sachlage früherer, vergleichbarer Fälle.

Medizin:
Jeder Patient ist ein *Fall*, auf den später als Erfahrungswissen zurückgegriffen werden kann.

e-commerce:
Virtuelle Verkaufsagenten, die den Kundenwünschen entsprechende Produkte aus dem Angebot heraussuchen, indem sie beispielsweise ähnliche Kunden vergleichen ("Kunden, die Produkt A kauften, kauften häufig auch Produkt B").

Wichtig: die Erklärungskomponente ist beim CBR sehr ausgeprägt, d.h. die Problemlösungen sind meist sehr gut verständlich.

Der CBR-Zyklus

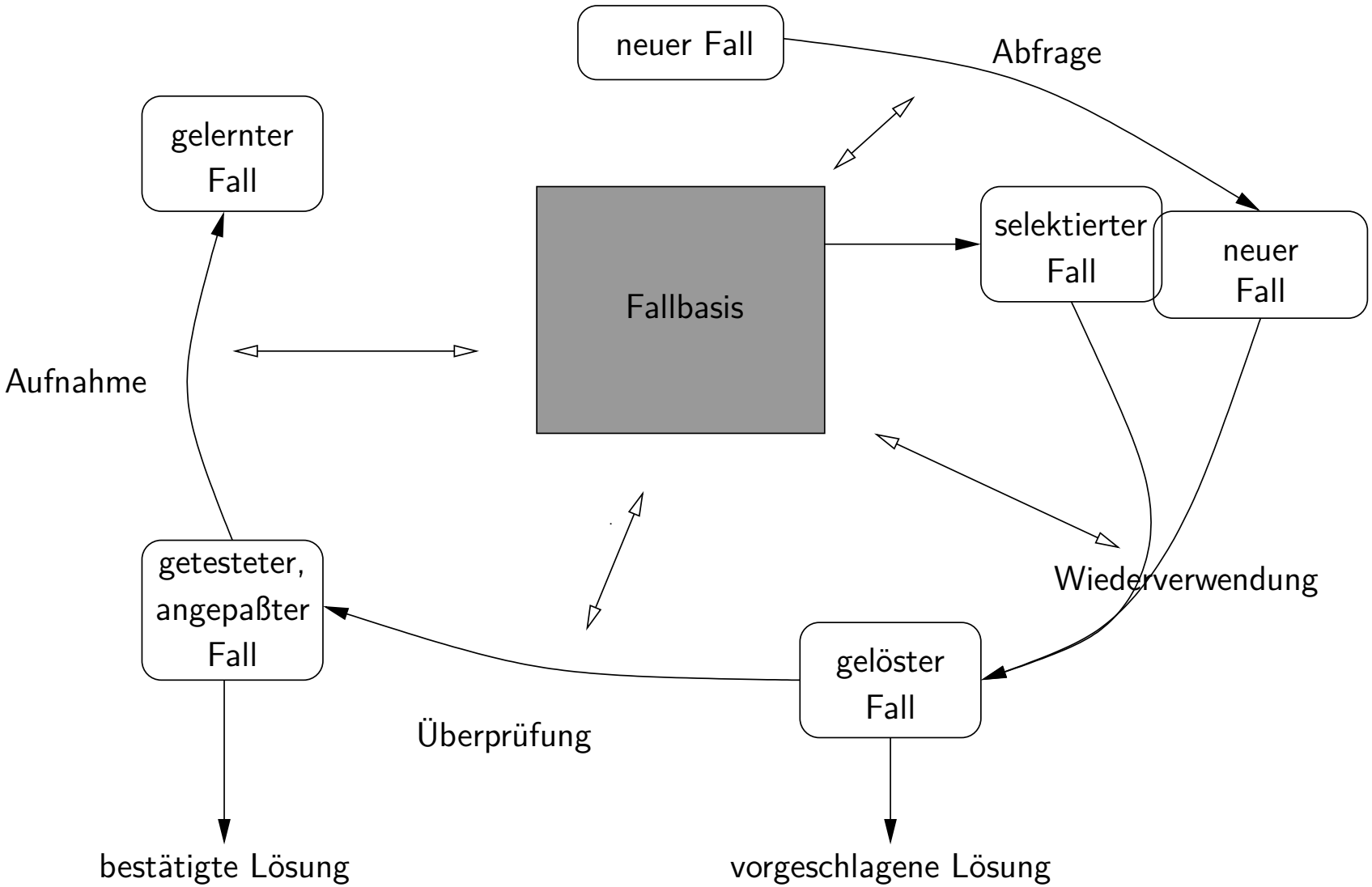
Selektierung (retrieve) des ähnlichsten Falls bzw. der ähnlichsten Fälle

Wiederverwendung (reuse) des in den gefundenen Fällen gespeicherten Wissens, um die Aufgabenstellung zu lösen

Überprüfung (revise) der vorgeschlagenen Lösung

Aufnahme (retain) des neuen Falls in die Fallbasis durch Integration

CBR-Zyklus schematisch



Übersicht

1. **Fallrepräsentation**
   - Indizierung von Fällen
   - Suche nach geeigneten Fällen
   - Organisation der Fallbasis
2. **Bestimmung der Ähnlichkeit**
3. **Adaption**

Repräsentation von Fällen

Ein Fall soll Wissen in einem bestimmten Kontext repräsentieren;

Ein Fall kann Erfahrungen dokumentieren, die von der Norm abweichen oder die einen anderen Verlauf nahmen als erwartet;

Ein Fall soll dabei helfen, ein gewisses Ziel zu erreichen;

Ein Fall kann ein warnendes Beispiel liefern.

Ein *Fall* repräsentiert *Wissen im Kontext* und dokumentiert eine Erfahrung, die ein wichtige *Aussage* bzw. *Lehre* im Hinblick auf gewisse Ziele beinhaltet.

Komponenten eines Falles

Problem- bzw. Situationsbeschreibung

Lösung: Begründungen, Lösungsweg

Resultat: Feedback aus der realen Welt

Problem- und Situationsbeschreibung

Beispiel: Es soll ein Kochrezept ausgearbeitet werden, das Rindfleisch und Broccoli verwendet. Das Gericht soll würzig schmecken und in der Pfanne zubereitet werden.

Formale Problemstellung:

Ziel:

kreiere Kochrezept

Bedingungen:

Zutat: Broccoli

Zutat: Rindfleisch

Geschmack: würzig

Zubereitungsart: Pfannengericht

Problem- und Situationsbeschreibung

Ausgangssituation:

- vorrätig: Broccoli
- vorrätig: Rindfleisch
- eingefroren: Rindfleisch
- nicht vorrätig: rote Paprika
- vorrätig: Ingwer
- vorrätig: Schalotten
- verfügbare Zeit: 2h
- verfügbares Geld: 20 EUR

Recht einfache Problemlösungsaufgabe mit dem Schwerpunkt der Formulierung der Ziele und der Bedingungen an die Ziele.

Repräsentation von Lösungen

Meist wird das Aussehen der Lösung schon durch die Formulierung des Ziels festgelegt.

Wichtige Bestandteile der Lösung:

- Lösungsweg:

  Wie? Argumente? Inferenzschritte?

- Begründungen:

  Welche Entscheidungen liegen der Lösung zugrunde?
  Warum wurde nicht anders entschieden?
  Welche anderen Lösungen gibt es?

Resultat eines Falles

Feedback/Ausgang: Was geschah, als die Lösung angewandt wurde?

Erfüllung von Zielen: Wie wurden gesteckte Ziele erreicht? Wurden die Erwartungen erfüllt?

Erfolg oder Fehlschlag: Wie erfolgreich war die Lösung?

Erklärung: Wie konnte es zum Fehlschlag kommen?

Korrektur: Was kann man tun, um einen Irrtum zu korrigieren?

Vermeidung: Was hätte man tun können, um den Fehlschlag zu verhindern?

Indizierung von Fällen

Ähnliche Aufgabe: Indizierung der Bücher einer Bibliothek

Indizes enthalten alle wichtigen Angaben, die Fälle voneinander unterscheiden

Fallsuchalgorithmen benutzen die Indizes, um Fälle auszuwählen

- **Indexvokabular:**
  fester Rahmen zur Beurteilung und Klassifizierung von Fällen
- **Fallkennzeichnung:**
  jeder Fall wird durch eine geeignete Indexkombination gekennzeichnet

Indexvokabular

Das Indexvokabular sollte:

Konzepte verwenden, die der Terminologie des zu behandelnden Gebiets folgen;

Begriffe vorwegnehmen, die bei einer Fallselektion benutzt werden;

Umstände einer Fallselektion vorwegnehmen, d.h. den Aufgabenkontext

sowohl *hinreichend abstrakt* als auch *hinreichend konkret* sein

Erstellung des Indexvokabulars

1. Sammlung einer repräsentativen Menge von Fällen (Probleme, Kontexte, Lösungen, Resultate)
2. Identifizierung der Besonderheiten/Lehren jedes Falls
3. Charakterisierung besonderer Situationen aus 2.
4. Formulierung von Indizes, die 3. vollständig abdecken und hinreichend konkret/abstrakt sind
5. Bildung des Indexvokabulars aus den verwendeten Begriffen

Indexvokabular als Checkliste zur Indizierung von Fällen

Kennzeichnung eines Falles durch Indizes

Bestimmung der relevanten Aspekte eines Falles im Hinblick auf die vom System zu behandelnden Aufgaben

Bestimmung der näheren Umstände, unter denen der Fall für die einzelnen Aufgaben von Interesse ist

Übertragung dieser Umstände in das Vokabular des Systems, um sie identifizierbar zu machen

Bearbeitung der Beschreibungen, um sie so breit anwendbar wie möglich zu machen

Beispiel: Indizierung

Problem: Zwanzig Leute kommen zum Abendessen; es ist Sommer und Tomatenzeit; geplant ist ein vegetarisches Mahl; eine Person reagiert allergisch auf Milchprodukte.

Lösung: Es wurden Tomaten-Pies aus Tomaten und Käse serviert; dabei wurde in einem der Pies Tofu-Käseersatz statt Käse verwendet, um den Milchallergiker in der Runde zu berücksichtigen.

Beispiel: Indizierung

Schritt 1: Verwendung des Falls in zweierlei Hinsicht

Erfolgsbedingung, falls die Aufgabe darin besteht, ein vegetarisches Hauptgericht mit Tomaten auszusuchen: *Wähle einen Tomaten-Pie für ein Essen mit Vegetariern im Sommer.*

Erfolgsbedingung, falls ein HG mit Käse an einen Milchallergiker angepasst werden muss: *Wähle Tofu statt Käse, wenn das Gericht für einen Milchallergiker zu modifizieren ist.*

Beispiel: Indizierung

Schritt 2: Konstruktion/Adaption einer Lösung

Nutzung des Falls zur Konstruktion einer Lösung:

Ziel: ein Hauptgericht bzw. ein vegetarisches Gericht bzw. ein Gericht mit Tomaten auszusuchen

Ziel: ein Hauptgericht bzw. ein vegetarisches Gericht bzw. ein Gericht im Sommer auszusuchen

Nutzung des Falls für die Adaption einer Lösung:

Das Hauptgericht enthält Käse als Zutat. Ein Gast oder mehrere Gäste sind Milchallergiker und das Ziel besteht darin, diese Gäste zu berücksichtigen

Beispiel: Indizierung

Schritt 3: Formalisierung, Erstellung des Indexvokabulars

aktives Ziel

Gäste

Gastgeber

Kochkultur

Zutaten

Zubereitungsart

Jahreszeit

Gerichte

- Salat
- Hauptgericht
- Beilagen
- Getränke
- Dessert

Einschränkungen

Ergebnisse

(eventuelle Unterstrukturen sind nicht dargestellt)

Beispiel: Indizierung

Index1 Aktives Ziel: Auswahl eines Hauptgerichts

Gerichte:

- Hauptgericht:

  Einschränkungen: vegetarisch

  Zutaten: Tomaten

Index2 Aktives Ziel: Auswahl eines Hauptgerichts

Gerichte:

- Hauptgericht:

  Einschränkungen: vegetarisch

  Jahreszeit: Sommer

Beispiel: Indizierung

Index3 Aktives Ziel: Adaption eines Hauptgerichts

Gerichte:

- Hauptgericht:

  Einschränkungen: keine Milch

  Zutaten: Käse

Beispiel: Indizierung

Schritt 4: Anwendung von Vereinfachungen und Generalisierungen:

Index1, Index2:

- ein *Hauptgericht* ist ein *Gericht* und erscheint als Teil des Menüs
- *Tomate* gehört zum *Gemüse*
- *Auswahl eines Hauptgerichts* allgemeiner als "Zusammenstellung eines Menüs"
- neu: Index4, Index5

Index3:

- Adaption eines beliebigen Gerichts mit Käse, nicht nur eines Hauptgerichts
- neu: Index6

Beispiel: Indizierung

Die so generalisierten Indizes haben nun die folgende Form:

Index4 Aktives Ziel: Zusammenstellung eines Menüs

Gerichte:

Einschränkungen: vegetarisch
Zutaten: Gemüse

Index5 Aktives Ziel: Zusammenstellung eines Menüs

Gerichte:

Einschränkungen: vegetarisch
Jahreszeit: Sommer

Beispiel: Indizierung

Index6 Aktives Ziel: Adaption eines Gerichts

Gerichte:

Einschränkungen: keine Milch

Zutaten: Käse

Suche nach geeigneten Fällen

Auftretende Probleme:

Suchproblem:
Die Fallbasis muss effektiv durchsucht werden.

Ähnlichkeitsproblem:
Zur Beurteilung der Ähnlichkeit von Fällen muss es eine Art *Ähnlichkeitsmaß* geben.

Beides leistet ein *Selektierer*.

Fallselektions-Prozess

Der ideale Fallselektionsprozess sieht wie folgt aus:

Weiterleitung der neuen Situation und des aktuellen Ziels an den Selektierer

Analyse der neuen Situation; Entwicklung der formalen Situationsbeschreibung in Form einer zur Aufgabe passenden Indizierung

...

Suche nach geeigneten Fällen

...

Ausgangspunkt der Suche: (neuer) Fall und errechnete Indizierung

Nutzung von Matching-Algorithmen, die den Grad der Übereinstimmung berechnen

Ausgabe der Fälle mit hinreichend großer Übereinstimmung

Festlegung der Rangordnung der Fälle (bspw. Sortierung nach Nutzungspotential)

Rückgabe der besten Fälle an das System

Ähnlicher Prozessablauf bei der Aufnahme neuer Fälle in die Datenbank.

Organisationsformen der Fallbasis

Serieller Suchalgorithmus bei flacher Fallbasis:

Eingabe:	Eine flache Fallbasis; ein neuer Fall.
Ausgabe:	Diejenigen Fälle der Fallbasis, die am besten zu dem neuen Fall passen.

Vergleiche jeden Fall in der Fallbasis mit dem neuen Fall;

gib diejenigen Fälle der Fallbasis aus, die dem neuen Fall am ähnlichsten sind.

Organisationsformen der Fallbasis

Vorteile der flachen Fallbasis:

Die besten Fälle werden auf jeden Fall gefunden, da die gesamte Fallbasis durchsucht wird;

Die Aufnahme neuer Fälle ist unproblematisch; sie müssen nicht eingegliedert werden, sondern werden einfach angehängt.

Nachteil der flachen Fallbasis:

Die Suche ist ineffektiv und aufwendig.

Organisationsformen der Fallbasis

Suchalgorithmus bei hierarchischer Fallbasis:

Eingabe: Eine hierarchische Fallbasis; ein neuer Fall.
Ausgabe: Diejenigen Fälle der Fallbasis, die am besten zu dem neuen Fall passen.

Setze $N :=$ Wurzel der Hierarchie;

repeat until N ist ein Blattknoten

Finde den Konten unter N, der am besten zum neuen Fall passt;

return N.

Organisationsformen der Fallbasis

Nachteile der hierarchischen Fallbasis:

Größerer Speicherplatzbedarf als bei flachen Strukturen

Mehr Sorgfalt bei der Integration neuer Fälle notwendig

(großer) Vorteil der hierarchischen Fallbasis:

Die Suche kann sehr effektiv ablaufen, was die Nachteile meist mehr als aufwiegt.

Übersicht

1. Fallrepräsentation

2. **Bestimmung der Ähnlichkeit**

3. Adaption

Bestimmung der Ähnlichkeit von Fällen

Repräsentation eines Falls $\boldsymbol{x}$ durch ein Tupel

$$\boldsymbol{x} = (x_1, \ldots, x_n)$$

wobei jedes x_i aus dem Wertebereich des i-ten Merkmals stammt

Grundidee: die Gesamtähnlichkeit zweier Fälle lässt sich durch einen Abgleich der einzelnen Merkmale bestimmen:

$$\mathbf{sim}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{function}(\mathrm{sim}_1(x_1, y_1), \ldots, \mathrm{sim}_n(x_n, y_n))$$

wobei $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{y} = (y_1, \ldots, y_n)$ zwei Fälle repräsentieren

und die sim_i reelle Funktionen sind, die jeweils die Ähnlichkeit zwischen verschiedenen Werten eines Merkmals bestimmen

Hamming-Ähnlichkeit

Annahme zweiwertiger Attribute (wahr/falsch, ja/nein, Frau/Mann, 0/1)

$x_i, y_i \in \{0, 1\}$ mit entsprechender Interpretation der Werte

Hamming-Distanz zweier Fälle $\boldsymbol{x}, \boldsymbol{y}$:

$$d_H(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

Stimmen $\boldsymbol{x}$ und $\boldsymbol{y}$ in allen Komponenten überein, so ist der Hamming-Abstand 0, bei vollkommener Verschiedenheit (d.h. $x_i \neq y_i$ für alle i), so ist der Abstand n, also maximal

Hamming-Ähnlichkeit

Umrechnung in ein Ähnlichkeitsmaß zwischen 0 und 1:

$$\begin{aligned} \mathrm{sim}_H(\boldsymbol{x}, \boldsymbol{y}) &= 1 - \frac{d_H(\boldsymbol{x}, \boldsymbol{y})}{n} \\ &= 1 - \frac{\sum_{i=1}^{n} |x_i - y_i|}{n} \end{aligned}$$

Allerdings können einige Attribute wichtiger sein und andere weniger wichtig $\rightarrow$ gewichtete Hamming-Ähnlichkeit

Gewichtete Hamming-Ähnlichkeit

Einige Attribute können wichtiger sein als andere

Priorisierung der Ähnlichkeitsbemessung durch *Gewichte*

Änderung der Schreibweise für die Hamming-Ähnlichkeit in:

$$\begin{aligned} \mathrm{sim}_h(\boldsymbol{x}, \boldsymbol{y}) &= \frac{n - \sum_{i=1}^{n} |x_i - y_i|}{n} \\ &= \frac{\sum_{i=1}^{n} (1 - |x_i - y_i|)}{n} \end{aligned}$$

Einführung nichtnegativer Gewichtsfaktoren w_i führt zu gewichteter Hamming-Ähnlichkeit für binäre Attribute: $(w_i \geq 0)$

$$\mathrm{sim}_H^w(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{n} w_i (1 - |x_i - y_i|)}{\sum_{i=1}^{n} w_i}$$

Gewichtete Hamming-Ähnlichkeit

$1/\sum_{i=1}^{n} w_i$ als Normierungsfaktor; so gilt $0 \leq \mathrm{sim}_H^w(\boldsymbol{x}, \boldsymbol{y}) \leq 1$

Gewichte w_i drücken aus, wie stark der Ähnlichkeitswert des jeweiligen Merkmals die Gesamtähnlichkeit beeinflussen soll und werden oft (nach Angaben) von Experten bestimmt

Umformung in eine günstigere Form für Berechnungen:

$$\begin{aligned}
\mathrm{sim}_H^w(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\sum_{i=1}^{n} w_i(1 - |x_i - y_i|)}{\sum_{i=1}^{n} w_i} \\
&= \frac{sum_{i=1}^{n} w_i - \sum_{i=1}^{n} w_i|x_i - y_i|}{\sum_{i=1}^{n} w_i} \\
&= 1 - \frac{\sum_{i=1}^{n} w_i|x_i - y_i|}{\sum_{i=1}^{n} w_i} \\
&= 1 - \frac{\sum_{i:x_i \neq y_i} w_i}{\sum_{i=1}^{n} w_i}
\end{aligned}$$

Verallgemeinerte Ähnlichkeiten

Allgemeines Ähnlichkeitsmaß:

$$\mathrm{sim}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{n} w_i \, \mathrm{sim}_i(x_i, y_i)}{\sum_{i=1} n w_i}$$

sim_i: bei reellen Attributwerten bspw. normierte Differenzen, sonst auch Klasseneinteilungen oder subjektives Urteil

Übersicht

1. Fallrepräsentation
2. Bestimmung der Ähnlichkeit
3. **Adaption**

Adaption

Lösung des (soeben gefundenen) alten Falls ist eine mögliche Lösung des aktuellen Problems

Bei Klassifikation oder Diagnose bspw. ist damit schon die Lösung gefunden

Häufig allerdings liegen neue Aspekte vor, vollständige Gleichheit ist nicht gegeben

Adaption als Vorgang, eine (vorgeschlagene) Lösung, die sich als nicht ganz richtig für eine (gegebene) Problembeschreibung erweist, so zu manipulieren, dass sie besser zu dem Problem passt.

Adaption, Einteilung

Etwas Neues wird in die alte Lösung eingefügt.

Etwas wird aus der alten Lösung entfernt.

Eine Komponente wird gegen eine andere ausgetauscht.

Ein Teil der alten Lösung wird transformiert.

Adaption, Methoden

Substitutionsmethoden:

- Reinstantiierung
- Lokale Suche
- Parameteranpassung
- Fallbasierte Substitution

Transformationsmethoden:

- Common-Sense-Transformation
- Modellbasierte Transformation

Spezielle Adaption und Verbesserung

Derivationswiederholung

Adaption, Substitutionsmethoden

Reinstantiierung wird angewendet, wenn offensichtlich die Rahmenbedingungen übereinstimmen, jedoch Komponenten oder Rollen mit anderen Objekten besetzt sind. Die alte Lösung wird mit den neuen Objekten instantiiert.

Bei der **Lokalen Suche** werden komplette Rollen nicht neu besetzt, sondern nur kleinere Teile einer Lösung geändert; Hilfsstrukturen wie semantische Netze oder Abstraktionshierarchien werden genutzt.

Adaption, Substitutionsmethoden

Parameteranpassung:
numerische Parameter von Lösungen werden verändert, indem Unterschiede in den Eingaben zu Unterschieden in der Ausgabe in Beziehung gesetzt werden.

Fallbasierte Substitution
andere Fälle werden genutzt, um Substitutionen vorzuschlagen; unpassende Teile werden gegen geeignete ausgetauscht.

Adaption, andere Methoden

Common-Sense-Transformation
Heuristiken, die auf Allgemeinwissen basieren, werden der Adaption zugrunde gelegt; häufig werden sekundäre, d.h. als nicht-essentiell eingestufte Komponenten der Lösung entfernt.

Spezielle Adaptions- und Verbesserungsmethoden
bereichsspezifische und strukturmodifizierende Anpassungen durch Heuristiken

Derivationswiederholung
Ableitung einer alten Lösung wird genutzt, um eine neue Lösung zu finden; vgl. Aufgabenlösung im Mathematik-Unterricht

Wie lernt ein fallbasiertes System?

Ein fallbasiertes System lernt:

Hauptsächlich durch Aufnahme und Integration neuer Fälle in seine Fallbasis

Besondere Situationen:

- ein ungewöhnlicher Fall tritt auf
- eine vorgeschlagene Lösung schlägt fehl

Fehlschläge liefern wertvolle Informationen

Erweiterung des Erwartungshorizonts

Weiterführende Literatur

Beierle, C. and Kern-Isberner, G. (2008).
Methoden wissensbasierter Systeme : Grundlagen, Algorithmen, Anwendungen.
Vieweg+Teubner, Wiesbaden, 4., überarbeitete und erweiterte Auflage edition.