



Patrně řekne, že body v rovině jsou uspořádány do dvou kosočtverců. Body  $o_1$  a  $o_2$ , jakkoliv jsou k sobě navzájem blíže, než ke kterémukoliv jinému bodu, umístí tedy do různých shluků. To znamená, že člověk nevytváří tyto shluky na základě vzdálenosti dvojic bodů, nýbrž na základě pojmové sounáležitosti. Body vytvářejí shluk, jestliže společně reprezentují týž pojem – v našem příkladu je tímto pojmem „kosočtverec“. Uvedený příklad ilustruje základní ideu, vedoucí k pojmovému shlukování. Z tohoto pohledu závisí míra podobnosti objektů  $o_1$  a  $o_2$  nikoli pouze na těchto objektech a na objektech z jejich „okolí“ ale též na množině pojmů, pomocí nichž můžeme charakterizovat (popsat)  $o_1$  i  $o_2$  společně. Pojmové shlukování tedy neidentifikuje shluky pouze jako množiny objektů, ale odhaluje také jejich implicitní pojmovou strukturu. Shluky jsou definovány jak výčtem svých prvků (extenzionálně) tak formulí, určující vztah náležení objektu do shluku (intenzionálně). Kvalita vytvořené množiny shluků závisí jak na kvalitě objektů a jejich reprezentace tak na kvalitě pojmů, které mohou být použity při charakterizování shluků (a která se dá měřit např. „jednoduchostí“ pojmů, jejich „vhodností“ apod.).

## 2. Pojmové shlukování jako rozšíření postupů numerické taxonomie..

K metodám pojmového shlukování můžeme přistupovat ze dvou hledisek. S jedním přístupem jsme se již seznámili – pojmové shlukování se chápe jako rozšíření (či analogie) technik numerické taxonomie. Budeme se mu nyní věnovat poněkud podrobněji. Druhým pohledem, který se dívá na pojmové shlukování jako na metodu tzv. učení pozorováním – chápaného v jistém kontrastu k „pouhému“ učení z příkladů – se budeme zabývat později.

Literatura o numerické taxonomii ([ 1 ]) rozlišuje tři skupiny metod. Optimalizační metody vytváří (nijak nestrukturovanou) množinu po dvou disjunktích shluků, jinými slovy rozklad množiny  $O$ . Při své činnosti hledají globálně optimální rozklad na  $k$  shluků ( $k$  je přirozené číslo předem dané uživatelem); to ovšem má za následek jejich značnou výpočetní náročnost a omezuje jejich použití na malé množiny objektů a/nebo malé hodnoty čísla  $k$ .

Hierarchické metody vytváří (kořenové) klasifikační stromy na množině objektů  $O$ , kde listy jsou jednotlivé objekty a „vnitřní“ uzly stromu reprezentují shluky objektů. Každý „řez“ takového stromu na nějaké hladině (tj. množina všech shluků reprezentovaných uzly ležícími na zvolené hladině) je rozkladem  $O$ . Hierarchické metody můžeme dále dělit na rozdělovací (divizibilní), které generují klasifikační strom shora-dolů a slučovací (aglomerativní), které ho generují způsobem zdola-nahoru. Kvalita rozkladu vytvořeného nějakým hierarchickým postupem závisí obvykle na řadě „lokálních“ rozhodnutí. Při použití rozdělovací techniky je uzel částečně vytvořeného klasifikačního stromu dále dělen nezávisle na jiných (nerodičovských) uzlech stromu. Tím, že se v hierarchických postupech užívá „lokálních“ rozhodování, jsou tyto metody v porovnání s optimalizačními méně výpočetně náročné (což ovšem může být znehodnoceno pravděpodobně menší kvalitou výsledného rozkladu).

Třetím typem jsou tzv. skupinové (clumping) metody, vytvářející množinu shluků, jejíž prvky (tedy jednotlivé shluky) nejsou nutně po dvou disjunktí. možnost „překrývání“ některých shluků plyne z toho, že metoda tohoto typu nezávisle zpracovává určitý počet shluků, které mohou být „hostiteli“ (nového) objektu, který potřebujeme klasifikovat (zařadit).

Stejným způsobem můžeme třídít i metody a techniky pojmového shlukování. Tak například tzv. shlukovací modul (Clustering Module nebo také Partitioning Module) programu CLUSTER/2 (viz [ 12 ] ; o tomto systému budeme podrobněji mluvit později) realizuje optimalizační metodu; pro danou množinu objektů  $O$  a parametr  $k$  vytváří optimální  $k$ -rozklad (rozklad na  $k$  disjunktích shluků). Tento postup je výpočetně náročný a není realizovatelný pro velké hodnoty čísla  $k$ . Hierarchický modul (Hierarchy-building Module) téhož programu je zase příkladem hierarchické metody. Vytváří klasifikační strom způsobem shora-dolů (je to tedy rozdělovací technika); pro daný uzel klasifikačního stromu volá shlukovací modul s malou hodnotou  $k$  a vybírá jeho optimální rozklad.

Jinými příklady rozdělovacích hierarchických postupů jsou systémy DISCON ([ 7 ]) nebo RUMMAGE ([ 2 ]). Oba vytvářejí monotetický klasifikační strom (hrany vycházející z jednoho uzlu jsou ohodnoceny různými hodnotami téhož (jednoho) atributu. Naproti tomu hierarchický modul z CLUSTER/2 ohodnocuje hrany konjunkcemi hodnot několika atributů (vytváří tedy tzv. polytetickou klasifikaci). DISCON se od obou dalších zde zmiňovaných systémů (a vlastně od většiny hierarchických technik) odlišuje v tom, že vytváří optimální klasifikační strom (měřeno počtem uzlů úplného stromu), zatímco RUMMAGE a CLUSTER/2 dělají pouze „lokálně optimální“ klasifikaci (tj. pro daný uzel vytvářejí jeho optimální dělení nezávisle na dělení jiných uzlů) a „věří“, že výsledný strom bude mít „vysokou kvalitu“. Příkladem systému užívajícího slučovací hierarchickou techniku vytváření klasifikačního stromu je MK 10 ([ 13 ]).

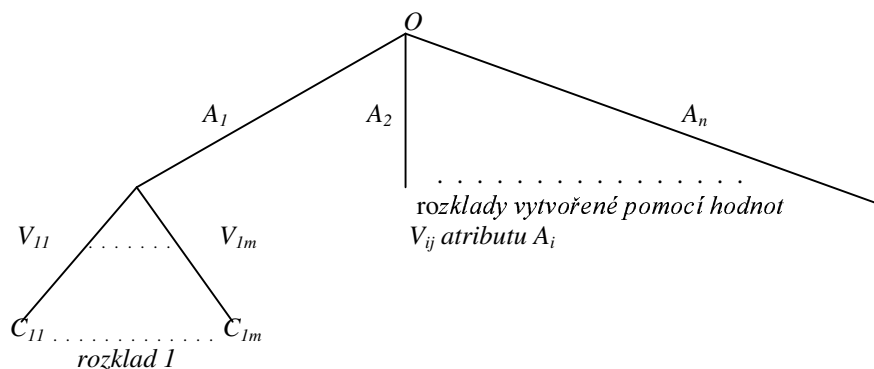
Mezi systémy pojmového shlukování pomocí skupinových (clumping) metod jsou UNIMEM (viz [ 10 ] , [ 5 ] ; k tomuto systému se ještě podrobněji vrátíme), z dalších například GLAUBER ([ 8 ]).

### 3. Pojmové shlukování jako učení pozorováním.

Pohled na pojmové shlukování jako na rozšíření metod numerické taxonomie přibližuje vstupně – výstupní chování řady shlukovacích algoritmů. Pro bližší porozumění jejich činnosti je užitečné podívat se na pojmové shlukování ještě z jiného pohledu – totiž jako na tzv. učení pozorováním (v kontrastu s „pouhým“ učením z příkladů). Při učení z příkladů učitel klasifikuje objekty co do příslušnosti k jednotlivým třídám, zatímco pojmové shlukování je neřízené učení (tedy učení bez učitele), které užívá ohodnocovací funkci, pomocí níž konstituuje třídy s „dobrymi“ pojmovými popisy; metody pojmového shlukování tedy ohodnocují kvalitu shluku na základě jeho pojmového popisu.

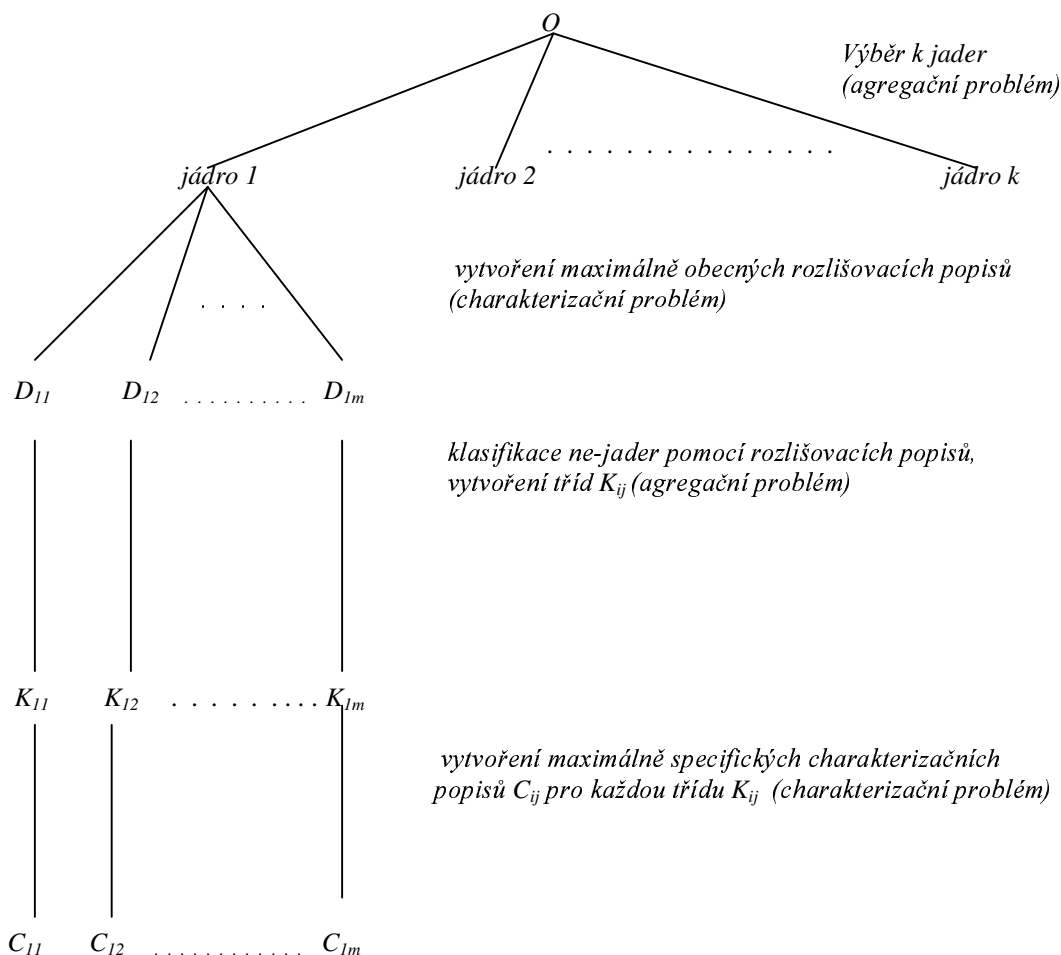
Každý algoritmus pojmového shlukování se musí vypořádat se dvěma úkoly. Prvním je řešení shlukovacího (agregačního) problému, spočívající ve vytvoření množiny shluků na dané množině objektů; každý shluk je definován extenzionálně, tedy výčtem svých prvků. Tento úkol namůže být řešen učením z příkladů, při kterém předpokládáme, že množina shluků je poskytnuta externím zdrojem – učitelem. Druhým úkolem je řešení charakterizačního problému, tedy určení pojmů, charakterizujících jednotlivé (extenzionálně definované) shluky a vytvářejících jejich pojmové popisy. Tento úkol je vlastně problém učení z příkladů. Oba problémy nejsou ovšem při pojmovém shlukování nezávislé; výsledek charakterizace (tedy množina pojmů) musí být použit pro určení kvality shluků objektů (tedy výsledku agregace).

Přirozeným postupem (při pojmovém shlukování) se tedy jeví nejprve vyřešit problém agregace a poté použít nějakou metodu učení z příkladů pro řešení charakterizačního problému. Takto se skutečně dá strukturovat činnost řady stávajících systémů (MK 10, GLAUBER, UNIMEM). Všimněme si ještě alespoň stručnou poznámkou systémů RUMMAGE a DISCON. Oba používají seznamu atributů (specifikovaných uživatelem) k vytváření možných rozkladů množiny objektů. RUMMAGE uvažuje řadu rozkladů (každý z nich je určen různými hodnotami jednoho zvoleného atributu) a vybírá z nich takový, který má „nejlepší“ pojmový popis vytvořený pomocí zbývajících atributů. Jinými slovy, nejprve řeší agregační problém užitím hodnot jednoho atributu a pak volá (jako subrutinu) učení z příkladů, aby charakterizoval shluky v termínech ostatních atributů. RUMMAGE aplikuje tento postup rekurzivně na každý ze shluků vybraného rozkladu a tím vytváří jednu hierarchickou klasifikaci. DISCON také používá hodnot jednotlivých atributů k vytvoření možných rozkladů. Nevytváří ale explicitní popis vzniklých shluků pomocí zbývajících atributů, nýbrž pouze se rekurzivně volá na každý z nich a tím vytváří pomocí zbývajících atributů klasifikační strom na množině objektů sdružených v tomto shluku. Oba tyto systémy jsou založeny na známém algoritmu ID3 učení z příkladů. Následující obrázek schematicky znázorňuje jejich agregační proces.



Důmyslněji kombinuje agregační a charakterizační proces systém CLUSTER/2 (přesněji jeho shlukovací modul). Má-li rozložit vstupní množinu pozorovaných objektů (treningovou množinu) do  $k$  disjunktních tříd (shluků), zvolí nejprve  $k$  jader (zárodečných objektů). Systém chápe každé jádro  $i$  jako pozitivní příklad nějaké třídy (nějakého pojmu) a ostatní jádra jako negativní příklady této třídy. Pro každou takto indukovanou (určenou) třídu program generuje tzv. maximálně obecné rozlišovací (diskriminační) popisy  $D_{i1}, \dots, D_{im}$ ; jinak řečeno pro každé jádro  $i$  dostaneme jistý počet  $im$  popisů (tj. pojmů), z nichž každý „pokrývá“ toto jádro (tj. jádro  $i$  splňuje popis) a „nepokrývá“ žádné jiné jádro (žádné jiné jádro  $l$  různé od  $i$  popis nesplňuje). Každý popis  $D_{ij}$  daného jádra  $i$  pokrývá také další objekty (které nejsou jádra) a tyto objekty přiřadíme spolu s ním do  $j$ -té třídy  $K_{ij}$ . Jsou-li všechny objekty (jádra i ne-jádra) klasifikovány pomocí těchto rozlišovacích popisů, pak z toho, že zmíněné popisy jsou maximálně obecné vyplývá, že pro každý rozklad určený jejich libovolnou kombinací (pro každé jádro jeden popis) platí, že pro libovolný objekt obsahuje alespoň jeden shluk, do kterého tento objekt náleží. Nyní systém „zapomene“ maximálně obecné rozlišovací popisy a pro každou třídu  $K_{ij}$

vytvoří tzv. maximálně specifický charakterizační popis  $C_{ij}$ . Pro každé jádro vybere jeden z těchto popisů, čímž dostane soubor (ne nutně po dvou disjunktních) shluků, které dávají klasifikaci všech prvků vstupní množiny objektů. Maximálně specifické popisy jsou vytvářeny proto, aby se zmenšila možnost „překrývání“ shluků pro objekty, které nebyly obsaženy v trénigové množině. Závěrem je aplikována procedura, vytvářející z tohoto souboru shluků disjunkttní rozklad vstupní množiny objektů. Schematicky je popsán postup agregace a charakterizace v CLUSTER/2 znázorněn na následujícím obrázku.



#### 4. Pojmové shlukování jako prohledávání.

Dosud jsme posuzovali a rozlišovali metody pojmového shlukování jednak z hlediska struktury rozkladů, které vytvářejí jednak z hlediska jejich činnosti tj. způsobů, kterými se vyrovnávají s řešením problémů agregace a charakterizace. Nyní zvolíme jiný (možno říci sjednocující a zobecňující) pohled, vycházející z jednoho ze základních paradigmat umělé inteligence a chápající učení jako prohledávání; z tohoto zorného úhlu nyní popíšeme jednotlivé podprocesy, obsažené v úloze pojmového shlukování. V zásadě se jedná o prohledávání prostoru možných rozkladů (agregací) a jemu podřízené prohledávání prostoru pojmů (možných charakterizací); v případě hierarchických metod jde navíc o (oběma) nadřazené prohledávání prostoru hierarchií. Je pravda, že řada existujících systémů pojmového shlukování nebyla původně koncipována jako systémy prohledávání, nicméně je možné se na ně zpětně takto dívat (lze v nich nalézt charakteristické rysy, sledované a hodnocené při posuzování prohledávání). Výhodou takového pohledu je to, že vyžaduje explicitní vyjádření cíle učení a schopností systému potřebných k dosažení (nebo alespoň aproximaci) tohoto cíle; to dále umožňuje odpovídat na otázky po kvalitě získaného rozkladu.

##### 4.1. Prohledávání prostoru pojmů (charakterizací).

Jde vlastně o učení z příkladů, což je oblast, v níž byl pohled „učení jakožto prohledávání“ prvně uplatněn. Mezi základní kriteria, podle nichž můžeme algoritmy třídit, patří jistě směr prohledávání. Toto kriterium je založeno na faktu, že popisy pojmů jsou uspořádány (ne nutně lineárně) co do obecnosti. Prostorem pojmů tedy můžeme postupovat od velmi speciální hypotézy (charakterizace) směrem k obecnějším popisům, pokrývajícím

příklady z trénigové množiny; v takovém případě mluvíme o učení zobecňováním. Opačný postup začíná s velmi obecnou hypotézou a postupuje ke specifitějším popisům pokrývajícím data – tzv. učení rozlišováním (by discrimination). Konečně je možné postupovat v obou směrech a konvergovat ke konzistentní hypotéze – to je strategie spojená s Mitchellovým prostorem verzí. Ze systémů, o nichž jsme se již zmínili, UNIMEM a GLAUBER užívají při vytváření charakterizací (popisů svých množin shluků) zobecňování. CLUSTER/2 vytváří charakterizaci během své činnosti dvakrát. Jednak vytváří maximálně obecné rozlišovací popisy diskriminačním postupem jednak odvozuje maximálně specifické charakterizační popisy zobecňováním. RUMMAGE a DISCON vytváří pomocí hodnot jednotlivých atributů řadu možných rozkladů; každou hodnotu můžeme chápat jako maximálně obecný rozlišovací pojem pro skupinu objektů, kterou určuje. V tomto procesu se nevyskytuje žádné učení zobecňováním ani rozlišováním. RUMMAGE ale užívá zobecňování v jiné souvislosti a to při odvozování charakterizace shluků – prvků rozkladu – pomocí těch atributů, které se neužívaly při vytváření tohoto rozkladu.

Druhým kritériem je řídicí strategie prohledávání. Jejich spektrum zahrnuje jak metody slepého prohledávání (do hloubky i do šířky) tak i heuristické prohledávání, závislé na nějaké funkci ohodnocující hypotézy; užívá se horolezecká metoda (UNIMEM, COWEB – viz [ 4 ] ; třetí a poslední systém, kterým se budeme zabývat podrobněji), tzv. beam search, neboli vějířovité / paprskovité prohledávání (CLUSTER/2) nebo best-first prohledávání.

Třetí kritérium spočívá v rozlišení systémů řízených daty a řízených modelem. V prvním případě (zahrnujícím například systémy UNIMEM, CLUSTER/2, COWEB) operátory definované a pracující na prostoru hypotéz vyžadují jako vstup data (která řídí prohledávání). Ve druhém se pro generování nových hypotéz užívá nějaká jiná znalost a data se používají pouze ve fázi ohodnocování; sem spadají např. DISCON nebo RUMMAGE, i když v jejich případě se „model“ neboli „jiná znalost“ redukuje pouze na seznam uživatelem předem daných atributů, kterých systém může použít při vytváření klasifikačního schématu.

#### **4.2. Prohledávání prostoru možných rozkladů (agregací).**

Co se řídicí strategie týká, žádný z uváděných systémů nepoužívá exhaustivního (slepého) prohledávání, při kterém by systematicky generoval možné rozklady, ohodnocoval je a vybíral nejlepší z nich. CLUSTER/2 užívá horolezeckou metodu k nalezení akceptovatelného rozkladu, přičemž používá charakterizačních technik k ohodnocování svého výběru. Ostatní ze zmiňovaných systémů provádějí pouze „degenerované prohledávání“ (hloubky jedna) prostoru agregací, protože vybírají svůj rozklad v jednokrokovém procesu. Například v systémech RUMMAGE a DISCON se operátor generující kandidáta rozkladu omezuje pouze na výběr jednoho atributu; systém pak automaticky generuje odpovídající rozklad ( pro každou hodnotu vybraného atributu jeden shluk), který je následně ohodnocen. GLAUBER, UNIMEM a další docílují stejného výsledku (pouze s tím rozdílem, že jejich postup je řízen daty). Pouze CLUSTER/2 má k dispozici operátor, který není takto striktně omezen; místo atributu vybírá jádra (zárodečné objekty), která mohou ale také nemusí vést k „dobrému“ rozkladu.

#### **4.3. Prohledávání prostoru hierarchií.**

Na rozdíl od učení z příkladů, při kterém se vytváří pojmy na jedné úrovni, generuje se při pojmovém shlukování obvykle hierarchie pojmů (strom klasifikací). Dříve diskutovaná prohledávání jsou tedy vnořena v tomto nadřazeném (nejvýše postaveném) prohledávání prostoru klasifikačních stromů. Při něm opět sledujeme (a používáme při rozlišování systémů pojmového shlukování) směr prohledávání a jeho řídicí strategii.

Pokud jde o směr prohledávání, většina systémů (například CLUSTER/2, DISCON, RUMMAGE) používají rozdělovacích (shora-dolů) metod. Začínají s jednou množinou objektů a pokračují jejich rozkladem do podmnožin, rozkladem těchto podmnožin, atd. Je možno si ovšem představit i obrácený postup začínající jednoprvkovými podmnožinami množiny všech objektů, které se postupně „spojují“ do větších množin. Takovouto slučovací (zdola-nahoru) metodu používá např. GLAUBER a MK 10. V psychologické literatuře ([ 11 ] ) se objevuje i návrh jakéhosi „smíšeného“ postupu, při kterém se nejprve vytváří množiny „střední obecnosti“ a teprve poté obecnější i speciálnější shluky. Zhruba tímto způsobem se chová UNIMEM; v každém okamžiku může ke stávající klasifikaci přidat obecnější i speciálnější podmnožiny.

Při řízení prohledávání prostoru hierarchií je v některých případech (CLUSTER/2, RUMMAGE, GLAUBER a MK 10) toto řízení „degenerované“. Jmenované systémy totiž používají operátorů, nalézajících optimální rozklad (agregací) a charakterizaci. Prohledávání se tedy děje na těchto nižších úrovních a jejich výsledkem je optimální klasifikační strom. Naproti tomu systém DISCON má degenerované prohledávací schéma na nižších úrovních avšak provádí best-first prohledávání prostoru hierarchií (realizuje exhaustivní „look-ahead“ proces – ohodnotí celé podstromy a preferuje ty, které mají nejmenší počet uzlů). Podobně i UNIMEM provádí prohledávání na této úrovni.

#### 4.4. Inkrementální a neinkrementální učení.

Závěrem se zmíníme ještě o jednom rozlišovacím kritériu, vztahujícímu se k systému pojmového shlukování jako celku. Jde o to, zda učení je či není inkrementální. Jsou-li všechny instance (objekty) z trénigové množiny k dispozici současně (na začátku učení), mluvíme o neinkrementálním přístupu. Jeho příkladem je valná část (především starších) systémů; z těch, o nichž jsme se zmiňovali, jsou to CLUSTER/2, DISCON, RUMMAGE, GLAUBER a MK 10. Systém UNIMEM (a některé novější, např. COWEB a GCC ([9])) je naopak příkladem inkrementálního učení. Při něm systém přijímá a zpracovává instance (objekty) postupně – v každém okamžiku (kroku, taktu) jeden. Důležité přitom je, že při zpracování jednoho objektu se nemění zásadním způsobem dosud vytvořená struktura. Bez této podmínky bychom každou – i neinkrementální – metodu mohli považovat za „inkrementální“; stačilo by po přidání nové instance znovu aplikovat neinkrementální metodu na rozšířenou množinu instancí. Zaměření na inkrementální učení (které patrně spíše odpovídá učení v reálném světě) vede přirozeně k užšímu propojení prováděcího a učícího modulu. V inkrementálním systému akce prováděcího modulu (například klasifikace dané instance) řídí učící modul (například při modifikování hierarchie pojmů). Na druhé straně v neinkrementálních systémech jsou tyto moduly od sebe odděleny.

#### Literatura.

- [1] Everitt, B.: Cluster Analysis; Heinemann Educational Books, Ltd., 1980
- [2] Fisher, D.H.: A Hierarchical Conceptual Clustering Algorithm; Technical Report, Dept. of Information and Computer Science, University of California, Irvine, 1984
- [3] Fisher, D.H., Langley, P.: Approaches to Conceptual Clustering; In Proc. of the 9<sup>th</sup> International Joint Conf. on AI, Los Angeles, CA, 1985
- [4] Fisher, D.H.: Knowledge Acquisition via Incremental Conceptual Clustering; Machine Learning 2, 1987
- [5] Gennari, J.H., Langley, P., Fisher, D.H.: Models of Incremental Concept Formation; Artificial Intelligence 40, 1989
- [6] Gowda, K.C., Krishna, G.: Disagregative Clustering Using the Concept of Mutual Nearest Neighborhood; Man and Cybernetics, IEEE Transactions on Systems, Vol. SMC-8, No.12, 1978
- [7] Langley, P., Sage, S.: Conceptual Clustering as Discrimination Learning; In Proc. of the 5<sup>th</sup> Biental Conference of the Canadian Society for Computational Studies of Intelligence, 1984
- [8] Langley, P., Zytkow, J., Simon, H., Bradshaw, G.: The Search for Regularity : Four Aspect of Scientific Discovery ; In Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) : Machine Learning : An artificial intelligence approach , Vol. 2, Tioga Publishing Co., Palo Alto, CA, 1985
- [9] Maarek, Y.S.: An Incremental Conceptual Clustering Algorithm that Reduces Input-Ordering Bias;
- [10] Lebowitz, M.: Concept Learning in a Rich Input Domain; In Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) : Machine Learning: An artificial intelligence approach, Vol. 2, Tioga Publishing Co., Palo Alto, CA, 1985
- [11] Mervis, C., Rosch, E.: Categorization of Natural Object; In Annual Review of Psychology, 32, 1981
- [12] Michalski, R.S., Stepp, R.E.: Learning from Observation: Conceptual Clustering ; In Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) : Machine Learning : An artificial intelligence approach, Vol. 1, Tioga Publishing Co., Palo Alto, CA, 1983
- [13] Wolff, J.: Data Compression, Generalization and Overgeneralization in an Evolving Theory of Language Development; In Proc. of the AISB-80 Conference on Artificial Intelligence, 1980