ROZPOZNÁVANIE OBRAZU

Rozpoznávanie objektov, rozpoznávanie obrazcov

- Rozpoznávanie obrazcov sa používa na **klasifikáciu** oblastí a objektov a predstavuje dôležitú súčasť zložitejších systémov počítačového videnia.

- Príklad o uvádzačovi na oslave basketbalistov a džokejov.

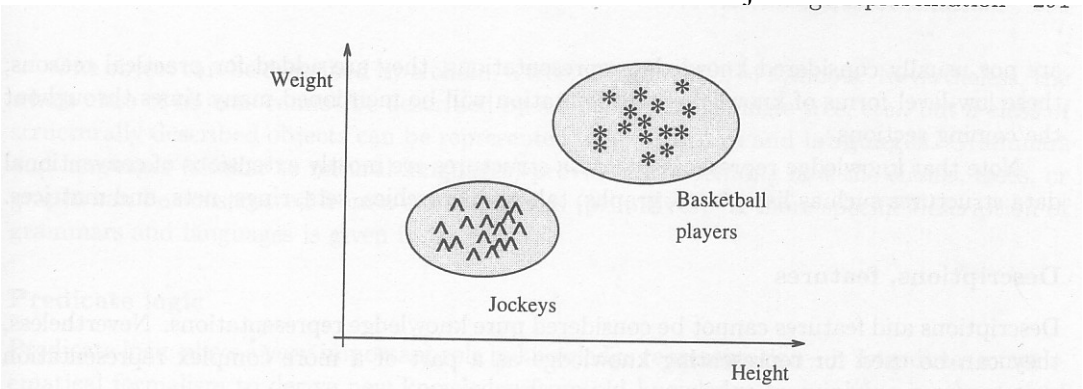

Figure 7.1: *Recognition of basketball players and jockeys; features are weight and height.*

- Žiadne rozpoznávanie nie je možné bez znalostí. Vyžadujú sa špecifické znalosti tak o objektoch, ktoré sa rozpoznávajú, ako aj všeobecnejšie znalosti o triedach objektov.

Reprezentácia znalostí

- Reprezentácia je súhrn syntaktických a sémantických pravidiel, ktorú umožňujú popisovať veci.

- Využívajú sa techniky umelej inteligencie:
  - Popisy a príznaky
  - Gramatiky a jazyky
  - Predikátová logika
  - Produkčné pravidlá
  - Fuzzy logika
  - Sémantické siete
  - Rámce, skripty

- ***Popisy*** reprezentujú nejaké skalárne vlastnosti objektov a nazývajú sa ***príznaky***. Jeden popis na objekt nestačí, preto sa príznaky organizujú do tzv. príznakových vektorov.

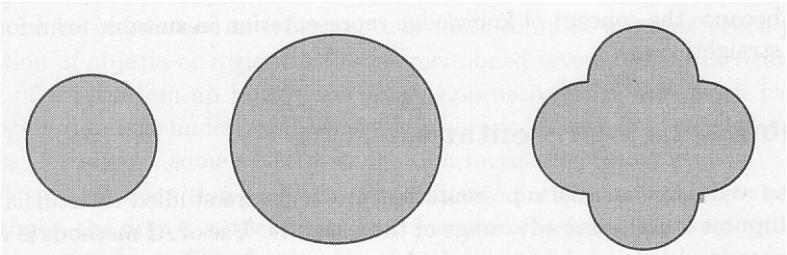

Figure 7.2: *Feature description of simple objects.*

- Na popis objektov na obrázku stačí dvojrozmerný vektor príznakov *x = (veľkosť, kompaktnosť),* čo umožní ich zaradenie do tried objektov: veľký, malý, okrúhly, neokrúhly apod.

- ***Gramatiky a jazyky.*** Ak sa má popísať štruktúra objektu, príznakový popis nestačí. Primitívum je najmenšia, ďalej nedeliteľná časť obrazu a štrukturálne popisy sa vytvárajú pomocou relácií medzi nimi. Môžu sa využiť reťazcové kódy, stromy, vo všeobecnosti grafy symbolov. Gramatiky predstavujú pravidlá ako sa z abecedy vytvárajú slová jazyka.

1. Štatistické rozpoznávanie obrazcov

- Objekt je fyzická jednotka, obvykle reprezentovaná oblasťou v segmentovanom obraze. Množinu všetkých objektov možno rozdeliť na podmnožiny so spoločnými príznakmi, ktoré sa nazývajú triedy.

- **Rozpoznanie objektu** je založené na priradení triedy k neznámemu objektu a zariadenie, ktoré vykonáva toto priradenie sa nazýva klasifikátor. Počet tried sa obyčajne vie dopredu a obvykle sa dá odvodiť zo špecifikácie problému.

- Klasifikátor nerozhoduje o zaradení do triedy na základe objektu priamo, ale na základe vlastností objektu, ktoré sa nazývajú obrazec.

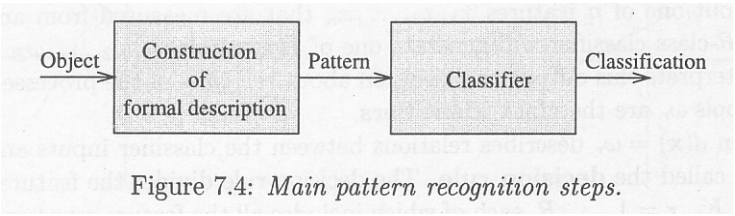

Figure 7.4: *Main pattern recognition steps.*

- Pre štatistické rozpoznávanie obrazcov je charakteristický **kvantitatívny** popis objektov, obyčajne sa používajú numerické príznaky. **Obrazec** je potom príznakový vektor **x** = $(x_1, x_2, ..., x_n)$. Množina všetkých možných obrazcov sa nazýva priestor obrazcov alebo **príznakový priestor.** Triedy obrazcov vytvárajú v príznakovom priestore zhluky, ktoré je možné rozdeliť diskriminačnými (oddeľujúcimi) hyperkrivkami.

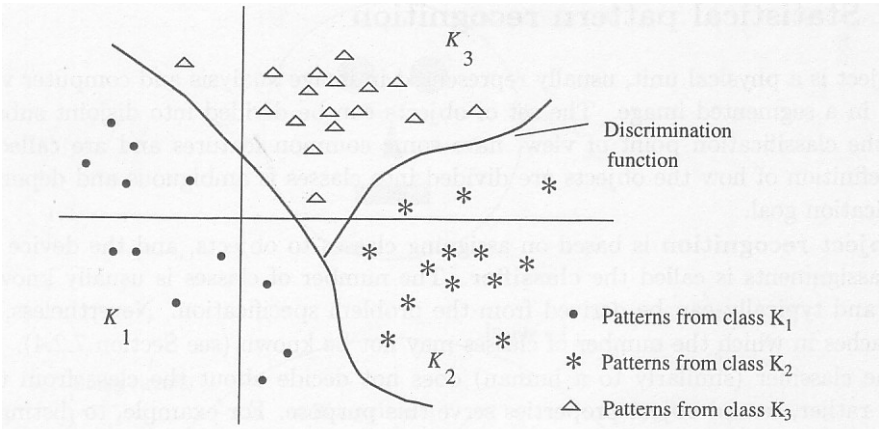

Figure 7.5: *General discrimination functions.*

- **Separabilné triedy** – také, pre ktoré existujú oddeľujúce nadplochy. Lineárne separabilné – také, pre ktoré existujú oddeľujúce nadroviny. Väčšina reálnych problémov je reprezentovaná ako neseparabilné triedy, teda klasifikátor sa pomýli.

- **Štatistický klasifikátor** je zaradenie s *n* vstupmi a 1 výstupom. Každý vstup sa používa na získanie informácie o jednom príznaku z *n* príznakov meraných na objekte, ktorý sa má klasifikovať. *R-triedny* klasifikátor generuje jeden z *R* symbolov ω_r, predstavujúcich identifikátory triedy.

- Oddeľujúce nadplochy sa definujú pomocou *R* skalárnych funkcií $g_1(\mathbf{x}), g_2(\mathbf{x}),..., g_R(\mathbf{x})$, ktoré sa nazývajú diskriminačné funkcie. Musia spĺňať takú podmienku, že ak $\mathbf{x} \in \omega_r$, tak pre každé $s = \{1, ..., R\}$, $s \neq r$: $g_r(\mathbf{x}) \geq g_s(\mathbf{x})$. Potom je nadplocha medzi triedami ω_r a ω_s určená rovnicou $g_r(\mathbf{x}) - g_s(\mathbf{x}) = 0$. Z toho vyplýva rozhodovacie pravidlo

$$d(\mathbf{x}) = \omega_r \Leftrightarrow g_r(\mathbf{x}) = \max_{s=1,2,...,R} g_s(\mathbf{x})$$

- Lineárne diskriminačné funkcie sú najjednoduchšie a sú široko používané

$$g_r(\mathbf{x}) = q_{r0} + q_{r1}x_1 + q_{r2}x_2 + ... + q_{rn}x_n$$

- Druhá možnosť je založená princípe **minimálnej vzdialenosti.** Predpokladajme, že je definovaných *R* etalónov $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_R$, ktoré reprezentujú ideálnych reprezentantov *R* tried. Rozhodovacie pravidlo je potom

$$d(\mathbf{x}) = \omega_r \Leftrightarrow |\mathbf{v}_r - \mathbf{x}| = \min_{s=1,2,...,R} |\mathbf{v}_s - \mathbf{x}|$$

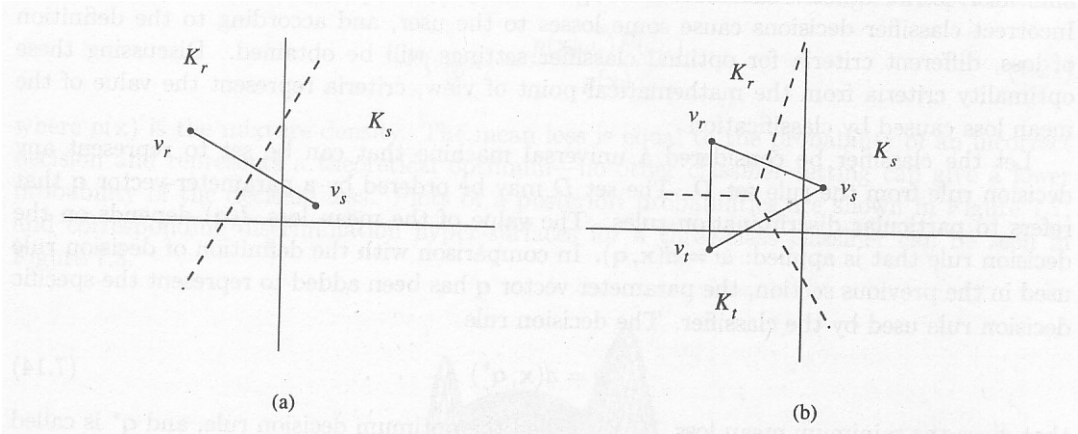

Figure 7.6: *Minimum distance discrimination functions: (a) two-class problem; (b) three-class problem.*

- Ak neexistujú lineárne diskriminačné funkcie, tak sa zobrazí príznakový priestor Φ-prevodníkom na iný priestor, v ktorom je možné použiť lineárne funkcie.

- Klasifikátor je deterministické zariadenie, to znamená, že ten istý vektor **x** zaradí vždy do tej istej triedy, hoci vzhľadom na neseparabilitu môže tento vektor reprezentovať objekt z inej triedy. Preto musí byť nastavenie optimálneho klasifikátora pravdepodobnostné.

- Ak definujeme stratu *J(***q***)* ako stratu spojenú s použitím konkrétneho rozhodovacieho pravidla ω = *d(***x**,**q***)* v prípade nesprávnej klasifikácie, kde parameter **q** zvýrazňuje použitie konkrétneho pravidla. Potom pravidlo ω = *d(***x**, **q****)* sa nazýva optimálne rozhodovacie pravidlo, ak dáva minimálnu strednú stratu, teda platí, že $J(\mathbf{q}^*) = \min_q J(\mathrm{q})\,, \forall\, \mathrm{d}(\mathbf{x},\mathbf{q})$.

- Klasický príklad štatistického rozpoznávania je Bayesovo pravidlo, ktoré pracuje s dvoma apriórnymi a jednou aposteriórnou pravdepodobnosťou. Jedna apriórna pravdepodobnosť je $P(\omega_s)$, ktorá vyjadruje pravdepodobnosť objavenia sa triedy. Druhá je pravdepodobnosť $p(\mathbf{x}|\omega_r)$, ktorá vyjadruje podmienenú pravdepodobnosť príznakového vektora **x** v triede ω_r. Ak zvolíme veľkosť straty pri nesprávnej klasifikácii ako jednotku a pri správnej ako nulu, potom optimálne rozhodovacie pravidlo je založené na diskriminačných funkciách $g_r(\mathbf{x}) = p(\mathbf{x}|\omega_r)\,P(\omega_r)$, pričom $g_r(\mathbf{x})$ zodpovedá aposteriórnej pravdepodobnosti $P(\omega_r|\mathbf{x})$, ktorá je pravdepodobnosťou toho, že vektor **x** patrí do triedy ω_r. Takže rozhodovacie pravidlo zaradí objekt popísaný vektorom **x** do tej triedy, ktorá má najvyššiu aposteriórnu pravdepodobnosť $P(\omega_r|\mathbf{x})$.

- Aposteriórne pravdepodobnosti sa počítajú z apriórnych pravdepodobností podľa Bayesovej formuly:

$$P(\omega_r|\mathbf{x}) = \frac{\mathrm{p}(\mathbf{x}|\omega_s)P(\omega_s)}{p(\mathbf{x})}$$, kde *p(***x***)* je zmiešaná hustota, vypočítaná cez všetky *s*.

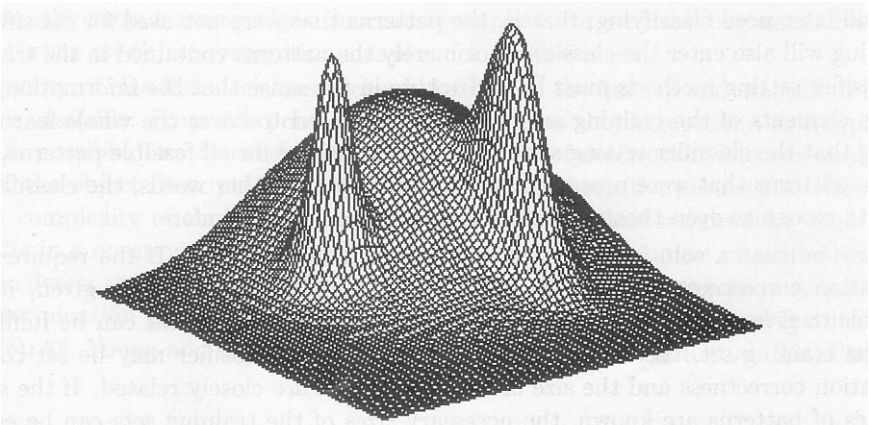

Figure 7.7: *Minimum error classifier: a posteriori probabilities.*

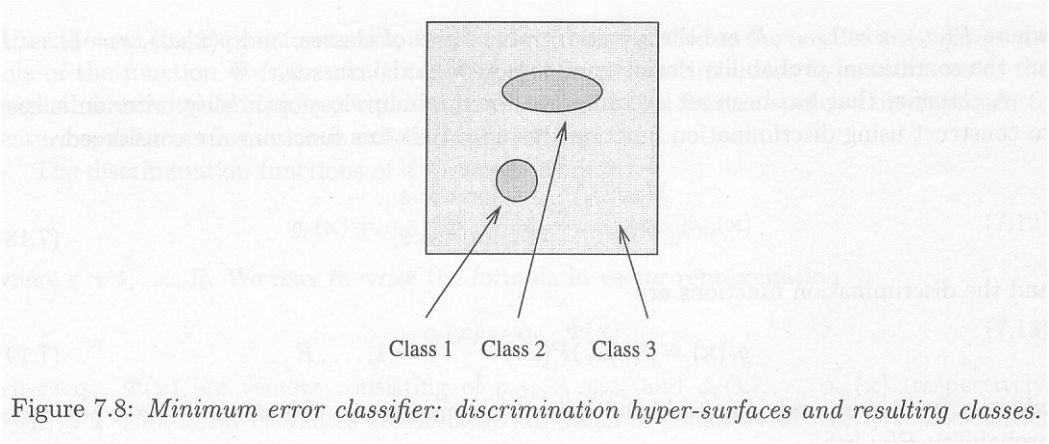

Figure 7.8: *Minimum error classifier: discrimination hyper-surfaces and resulting classes.*

- Klasifikačné parametre sa určujú z **trénovacej množiny** príkladov počas **učenia** klasifikátora. Učenie je založené na sekvenčnom predkladaní príkladov. Cieľom učenia je minimalizovať optimalizačné kritérium. Charakter učenia je induktívny, pretože musí naučiť z príkladov aj na situácie, ktoré neboli v trénovacej množine. Ide o učenie s učiteľom, ktorý každému príkladu priradí informáciu o jeho zaradení do triedy.

- Kvalita učenia závisí od veľkosti trénovacej množiny a od kvality konštrukcie príznakového vektora. Jednotlivé príznaky musia byť informatívne a diskrimina-tívne.

- Jedna obvyklá stratégia učenia je odhad hustoty pravdepodobnosti, kde sa využívajú znalosti o rozdelení pravdepodobnosti (obvykle sa používa normálne rozdelenie).

- Niektoré klasifikačné metódy nevyžadujú trénovacie množiny na učenie, konkrétne nepotrebujú informáciu o zaradení príznakového vektora do triedy (vtedy hovoríme

o učení bez učiteľa). Metódy **zhlukovej analýzy** rozdeľujú množiny obrazcov do podmnožín (zhlukov) založených na vzájomnej podobnosti prvkov takejto podmnožiny.

- Sú dve skupiny metód zhlukovej analýzy: hierarchická a nehierarchická. Hierarchické metódy vytvárajú zhlukový strom. Množina všetkých obrazcov sa rozdelí na dve rozdielne podmnožiny a každá z nich sa rozdelí na ďalšie rozdielne podmnožiny, atď.

- Ako príklad nehierarchickej klastrovej analýzy je **MacQeenova metóda *k* prostriedkov.** Pracuje v dvoch krokoch. Na začiatku sa vyberie *k* počiatočných etalónov z množiny všetkých obrazcov. V prvom kroku sa zaradia všetky obrazce do nejakého zhluku podľa princípu najmenšej vzdialenosti a prepočítajú sa etalóny ako ťažiská zhlukov. Tie sa použijú v druhom kroku na klasifikáciu všetkých obrazcov (niekedy sa druhý krok opakuje až do konvergencie). Existujú aj iné vylepšenia.

2. Neurónové siete

- Viaceré **neurónové prístupy** sú založené na kombinácii elementárnych procesorov (neurónov), z ktorých každý má viacero vstupov a jeden výstup. Každému vstupu je priradená váha a výstup je sumou váhovaných výstupov. Rozpoznávanie obrazcov je jednou z mnohých oblastí použitia neurónových sietí.

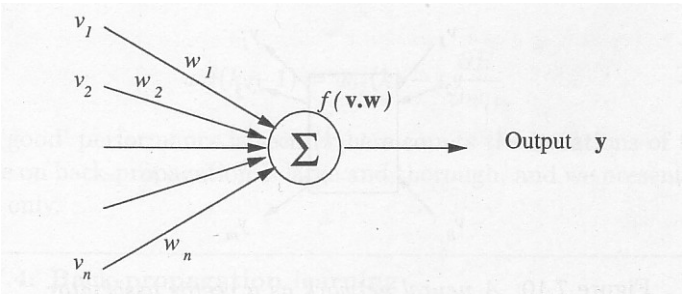

Figure 7.9: *A simple (McCulloch-Pitts) neuron.*

- Vstup do neurónu je definovaný ako $x = \sum_{i=1}^{n} v_i w_i - \theta$, kde θ je nejaký prah. Výstup neurónu je spojený s transfer funkciou *f(x),* napríklad

$f(x)=0$ ak $x \leq 0$ a $f(x)=1$ inak.

$$f(x) = \frac{1}{1+e^{-x}}.$$

- Príkladov využitia neurónovej siete môže byť viacero:
  klasifikácia: ak výstupný *m*-rozmerný vektor je binárny a obsahuje iba jednu jednotku, tak tá reprezentuje klasifikáciu vstupu do jednej z *m* kategórií.

všeobecná asociácia: vstupný vektor **v** a výstupný vektor **y** reprezentujú obrazce v rôznych oblastiach a neurónová sieť vytvára vzťah medzi nimi. Napr. čítanie textu.

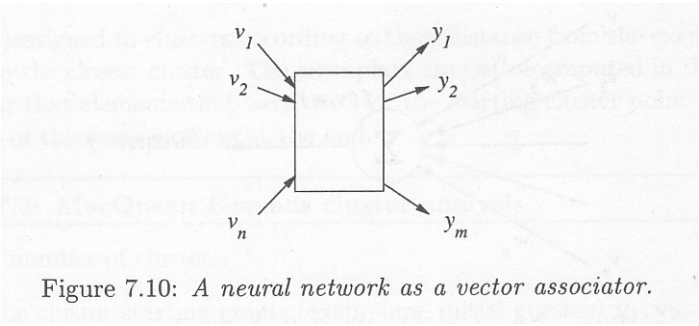

Figure 7.10: *A neural network as a vector associator.*

- Najprv sa používali siete bez skrytej vrstvy, ale ukázali sa ako veľmi obmedzujúce. Dá sa ukázať, že reálne stačia dve skryté vrstvy.

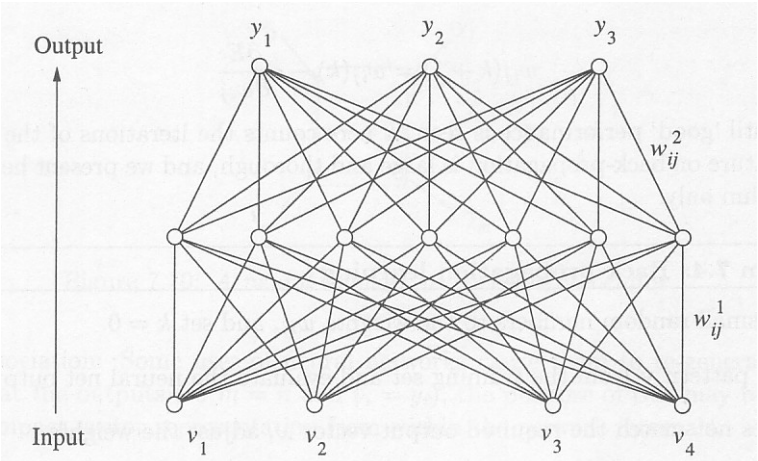

Figure 7.11: *A three-layered neural net structure.*

- V problémoch rozpoznávania obrazcov sa bežne používajú ***dopredné siete***. Ich učenie požíva trénovaciu množinu príkladov a je často založené na algoritme spätného posunu. Vyhodnocuje sa výstup siete a očakávaný výstup a počíta sa chyba založená na štvorci rozdielov

$$E = \sum_{\mathrm{i}} \sum_{j} (y_j^i - \omega_j^i)^2 \ ,$$

kde $\mathbf{y}^i$ je aktuálny výstup a ω^i je želaný výstup. Potom algoritmus urobí spätné korekcie

$$w_{ij}(k+1) = \mathrm{w}_{\mathrm{ij}}(k) - \varepsilon \frac{\partial E}{\partial w_{ij}}$$

kde *k* zodpovedá počtu iterácií opráv koeficientov.

3. Syntaktické rozpoznávanie obrazcov

- Pre syntaktické rozpoznávanie obrazcov je charakteristický kvalitatívny popis objektov. Syntaktický popis treba použiť vtedy, ak príznakový popis nie je schopný popísať zložitosť objektu alebo keď sa dá objekt zapísať ako hierarchická štruktúra pozostávajúca z jednoduchších častí, napríklad slovom alebo grafom. Elementár-nymi vlastnosťami syntakticky popísaných objektov sú primitíva. Na popis vzťahov medzi primitívami objektu sa používajú relačné štruktúry. Neexistuje postup na vytvorenie primitív a ich vzťahov. Vo všeobecnosti:
  - počet primitív má byť relatívne malý,
  - musia byť schopné vytvoriť vhodnú reprezentáciu objektu,
  - musia sa dať ľahko segmentovať z obrazu,
  - musia byť ľahko rozpoznateľné nejakou štatistickou metódou
  - musia zodpovedať prirodzeným dôležitým súčastiam objektu, ktorý sa opisuje.

- Množina všetkých primitív sa nazýva abeceda. Množina všetkých slov vytvorených z abecedy, ktoré opisujú objekt z jednej triedy, sa nazýva popisný jazyk. Gramatika predstavuje množinu pravidiel, podľa ktorých sa vytvárajú slová nejakého jazyka z prvkov abecedy.

- Gramatika je štvorica (V_t, V_n, P, s), kde V_t je množina terminálnych symbolov, V_n je množina neterminálnych symbolov, $s \in V_n$ je počiatočný symbol a P je konečná množina pravidiel odvodenie typu

  (r) $\alpha \to \beta$

kde tvar ľavej strany a pravej strany pravidiel určuje typ gramatiky: regulárne, bezkontextové, kontextové a frázové (tzv. Chomského hierarchia).

- Syntaktické rozpoznávanie pozostáva z nasledovných krokov:
  1. Definuj primitíva a vzťahy medzi nimi
  2. Zostroj gramatiku pre každú triedu objektov.
  3. Pre každý objekt vytiahni primitíva rozpoznaj ich a vzťahy medzi nimi a zostroj slovo reprezentujúce objekt.
  4. Na základe syntaktickej analýzy zaraď objekt do tej triedy, ktorej gramatika ho generuje.

- Vytvorenie gramatiky si obyčajne vyžaduje výraznú ľudskú interakciu. Niekedy je možné použiť automatický proces vytvárania gramatiky na základe príkladov, ktorý sa nazýva inferencia gramatiky. V zmysle teórie učenia je potrebné predkladať pozitívne príklady aj negatívne príklady. Problémom býva prílišná generatívna sila gramatiky, aj keď sa požívajú regulárne a bezkontextové gramatiky.

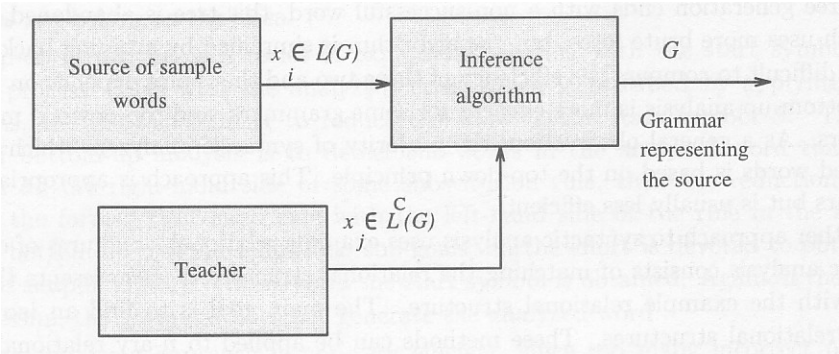

Figure 7.17: *Grammar inference.*

- Vytvárajú sa ďalšie typy gramatík – plošné gramatiky a programové gramatiky. Prvé preto, aby zachytili plošnú štruktúru vytváraného slova, druhé preto, aby obmedzili generatívnu silu známych typov gramatík.

- Programová gramatika je sedmica $(V_t, V_n, J, f_1, f_2, P, s)$, kde V_t je množina terminálnych symbolov, V_n je množina neterminálnych symbolov, $s \in V_n$ je počiatočný symbol, J je konečná množina čísiel pravidiel odvodenia, P je konečná množina pravidiel odvodenie typu

  $(r) \qquad \alpha \to \beta \qquad f_1(r) \qquad f_2(r)$,

  kde r je číslo odvodenia, $\alpha \to \beta$ je jadro pravidla a $f_1(r)$, $f_2(r) : J \to 2^J$.

- Rozhodnutie, či neznáme slovo môže alebo nemôže byť generované danou gramatikou sa robí syntaktickou analýzou, buď zdola nahor alebo zhora nadol. Čistý prístup zhora nadol nie je veľmi efektívny, pretože sa generuje príliš veľa nekorektných ciest. Často sa využíva apriórna informácia na odbúranie časti analýzy s rizikom, že nenájdeme riešenie vôbec.

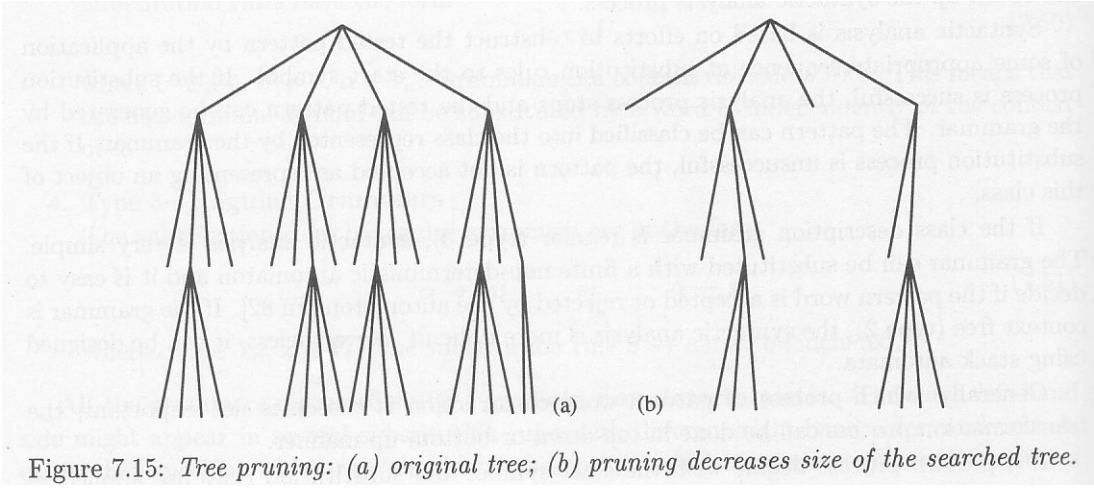

Figure 7.15: *Tree pruning: (a) original tree; (b) pruning decreases size of the searched tree.*

- Na opravu sledovania zlej cesty je možné použiť návrat (back-tracking) alebo sa mu vyhnúť, rozvíjať viacero ciest naraz a ak niektorá neuspeje, tak sa zahodí. Nie je možné

jednoznačne povedať, ktorý prístup je lepší, či zhora nadol alebo zdola nahor, väčšinou sa používa zhora nadol.

- Alternatívou syntaktickej analýzy je porovnávanie so vzorom, príkladom triedy.

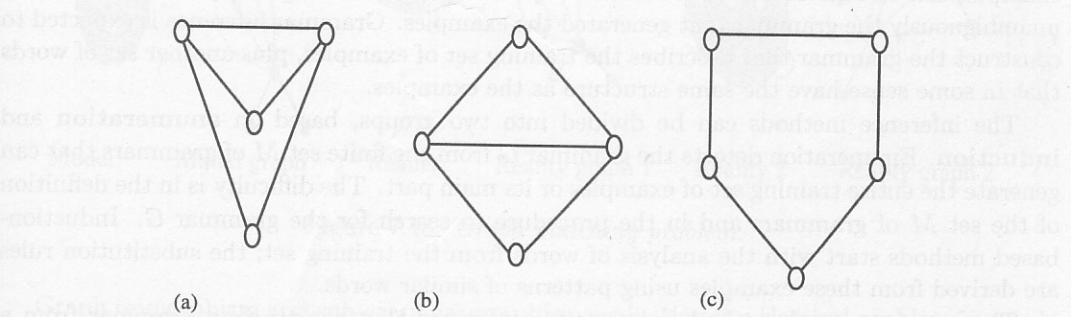

Figure 7.16: *Matching relational structures: (a) and (b) match assuming nodes and relations of the same type; (c) does not match either (a) or (b).*

4. Rozpoznávanie ako porovnávanie zhody dvoch grafov

- Porovnávanie zhody (matching) modelu a objektu popísaných grafom sa dá využiť na rozpoznávanie.

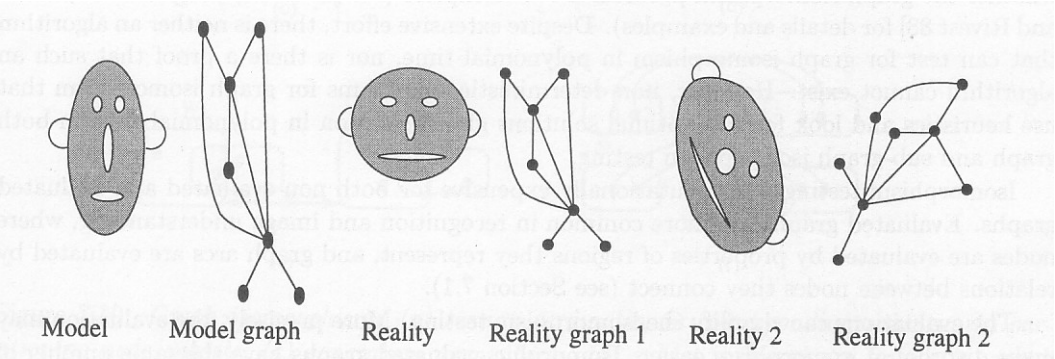

Figure 7.18: *Graph matching problem.*

- Presná zhoda grafov sa nazýva grafový izomorfizmus. Určenie grafového izomorfizmu je výpočtovo náročné. Ak máme dva grafy $G_1 = (V_1, E_1)$ a $G_2 = (V_2, E_2)$, potom izomorfizmus f je bijektívne zobrazenie medzi V_1 a V_2 také, že pre každú hranu z E_1 spájajúcu vrcholy $v, v' \in V_1$ existuje hrana z E_2 spájajúca $f(v)$ a $f(v')$. Podgrafový izomorfizmus sa definuje ako izomorfizmus medzi grafom G_1 a podgrafom druhého grafu G_2. Problém podgrafového izomorfizmu je NP úplný. Na zníženie výpočtovej náročnosti sa robia rôzne testy, ktorých neúspech umožní nepokračovať v procese.

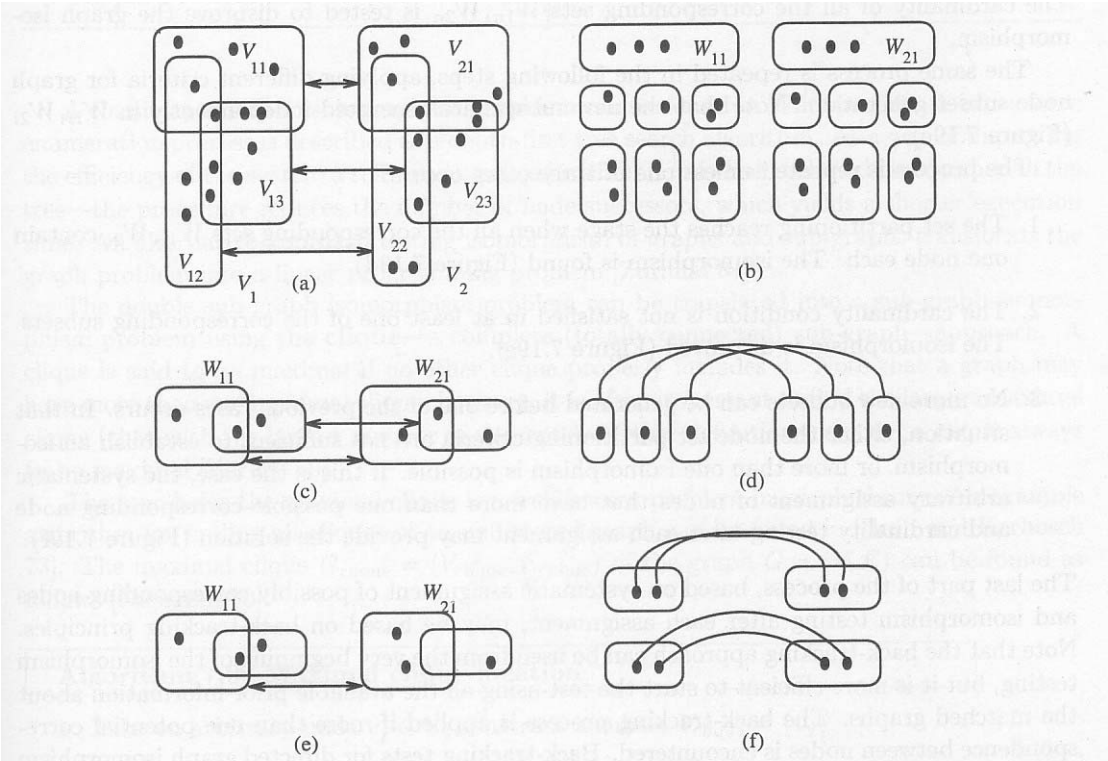

Figure 7.19: *Graph isomorphism: (a) testing cardinality in corresponding subsets; (b) partitioning node subsets; (c) generating new subsets; (d) subset isomorphism found; (e) graph isomorphism disproof; (f) situation when arbitrary search is necessary.*

- V reálnom svete sa nezhoduje graf objektu s grafom modelu úplne dokonale. Grafový izomorfizmus nevie určiť mieru nezhody. Na rozpoznanie objektov reprezentovaných podobnými grafmi sa dá využiť pojem podobnosti grafov, ktorý využíva pojem ***Levenshteinovej vzdialenosti,*** ktorá pre dva reťazce je definovaná ako najmenší počet vsunutí, vypustení a substitúcií potrebných na to, aby sa jeden reťazec zmenil na druhý.