Strojové učení

Program se **učí** ze zkušenosti **DATA** vzhledem k nějaké **třídě úkolů T** a **míře úspěšnosti (chyby) U (resp. Err)**, pokud se jeho výkon na úkolech třídy T zlepšuje s přibývající zkušeností `DATA`.

Zkušenosti (data)

- Většinou mám
- nebo postupně získávám – **inkrementální učení**
- někdy dokonce mohu ovlivnit sběr (např. ve zpětnovazebném učení)

Třída úkolů

- klasifikace (např. poskytnout úvěr, rozpoznání psaných číslic)
- regrese (předpovídáme spojitou veličinu)
- hledání optimální strategie
- popis dat: klastrování, asociace (market basket analysis), asociační pravidla, BN

Míra úspěchu (chyby)

- pro každý příklad zvlášť (kvadratická, chyba predikce)
- mám jen kumulovanou (zpětnovazebné učení)
- nemám, musím nějak definovat (např. součet vzdáleností bodů od středů klastru, support asociace atd.

Volba jazyka reprezentace

- logika: konjunkce atributů, rozhodovací stromy, množiny pravidel, logické programy
- funkce: přímka (lineární regrese), transformace prostoru (SVM), neuronová síť
- pravděpodobnostní modely: pravděpodobnosti tříd, naive Bayes, obecná bayesovská síť
- prostě uchovat data
- dvojice stav–zisk, nebo stav–akce–zisk
- středy klastrů

Volba učícího mechanizmu

- prostě spočítat: lineární regrese, pravděpodobnosti, naive Bayes
- gradientně najít minimum chyby: neuronové sítě, GA, klastrování
- prohledávat prostor hypotéz: úplně většinou nereálné; gradientě, paprskovitě (beam search), myopicky

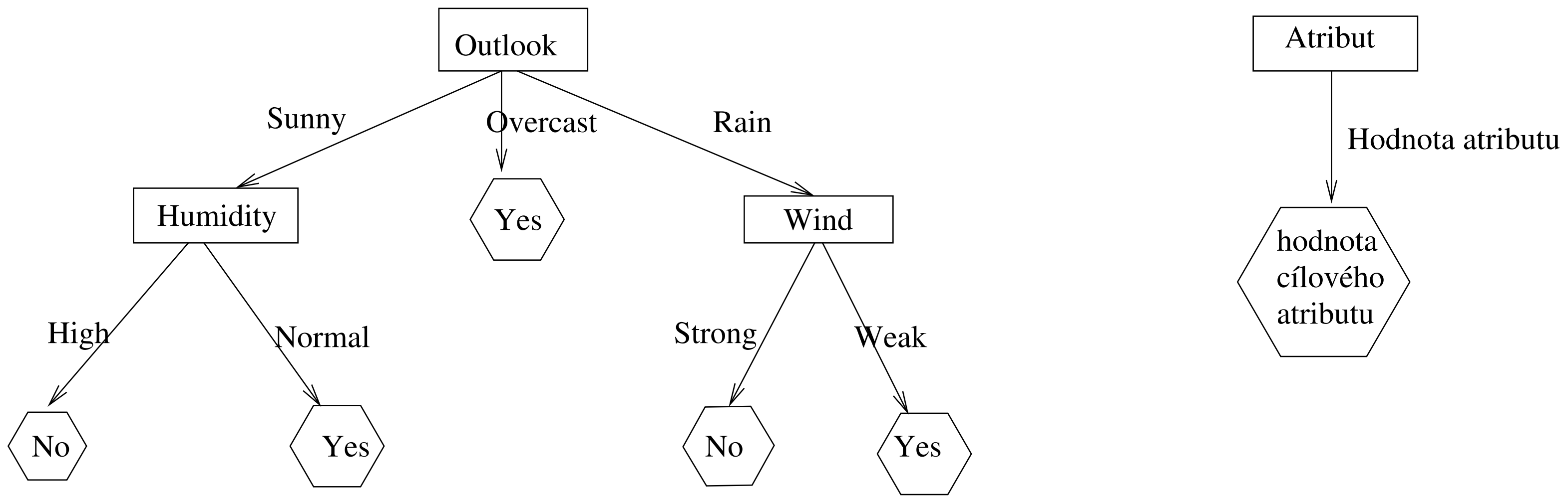
Jak se vyhnout přeučení

Přeučení je přílišná závislost na trénovacích datech; tj. pokud existuje jiná než naučená hypotéza, která sice má chybu na trénovacích datech horší, ale chybu na nově generovaných datech menší.

Přeučení se vyhýbám tím, že

- mám omezený prostor hypotéz
- preferuji jednodušší hypotézy (Occamova břitva, minimální délka popisu, ...)

Rozhodovací stromy



Rozhodovací strom pro daný cílový atribut G je kořenový strom tvořený z

- kořene a vnitřních uzlů označených atributem; ze kterého vede jedna hrana pro každou možnou hodnotu tohoto atributu;
- listy jsou označeny předpokládanou hodnotou cílového atributu G za předpokladu, že ostatní atributy nabývají hodnot na cestě

od kořene do listu. Pokud se některé atributy na cestě nevyskytují, na jejich hodnotě nezáleží.

Základ algoritmu tvorby rozhodovacího stromu z dat je následující:

1. vyber atribut; vytvoř z něj uzel a rozděl data podle hodnoty tohoto atributu
2. pro každou hodnotu atributu vytvoř podstrom z dat s odpovídající hodnotou
3. pokud data obsahují jen jednu hodnotu cílové třídy či pokud došly atributy k dělení, vytvoř list s hodnotou nejčetnější cílové třídy.

Otázkou je, jak vybírat atribut k dělení.

Entropie

Po míře entropie rozdělení hodnot daného atributu A (tj. míře nejistoty, negativní míře informace) chceme, aby:

- byla nula, pokud jsou všechny hodnoty cílové třídy stejné
- byla největší, pokud je stejně hodnot všech cílových tříd (tj. nevíme nic)
- aby rozhodnutí ve dvou krocích vedlo ke stejnému výsledku jako rozhodnutí naráz, tj.

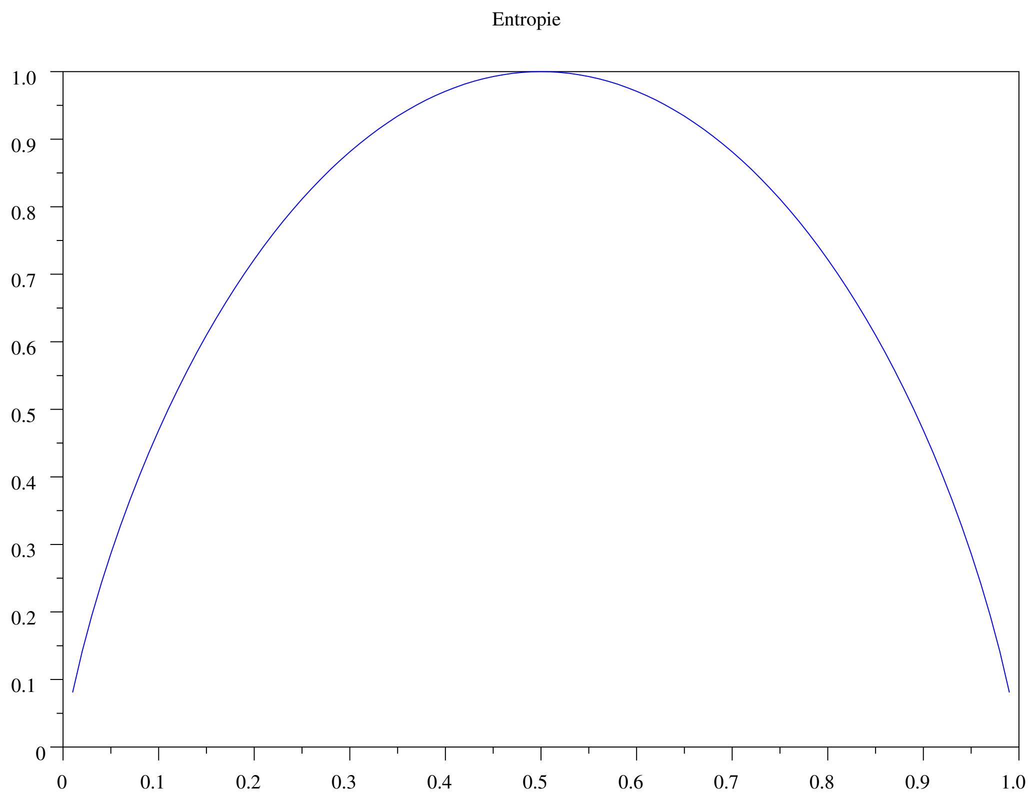
$$E([2,3,4]) = E([2,7]) + \frac{7}{9} \cdot E([3,4])$$

Toto splňuje pouze **entropie** $E([p_1, \ldots, p_n]) = -\sum_{i=1}^{n} p_i \log p_i$, logaritmus se bere většinou dvojkový.

Pozn. nemusíme normalizovat, pak dostaneme entropii násobenou součtem všech p_i.

Pokud chceme uvést, přes který atribut entropii počítáme, používáme dolní index, např. E_A, resp. E_G pro cílový atribut.

Entropie pro dvouhodnotový atribut



vodorovná osa: p_i, svislá: entropie.

ID3 algorithmus

Uzel, který dáme do kořene (pod)stromu, vybíráme podle maximálního **informačního zisku** (information gain), definovaného pro množinu dat *data* a atribut X_j jako:

$$Gain(data, X_j) = E_G(data) - \sum_{x_j \in X_j} \frac{|data_{X_j=x_j}|}{|data|} E_G(data_{X_j=x_j})$$

kde $data_{X_j=x_j}$ je podmnožina *data*, kde atribut X_j má hodnotu x_j, entropie je definovaná

$$E_G(data) = \sum_{g \in G} -\frac{|data_{G=g}|}{|data|} \cdot log_2 \frac{|data_{G=g}|}{|data|} = \sum_{i=1}^{|G|} -p_i \cdot log_2 p_i$$

kde p_i je počet dat v *data* patřící do třídy g_i dělený celkovým počtem dat v *data*.

Algoritmus **ID3 algorithm(**$data$**,** G **cíl,** $Attributes$ **vstup. atributy)**

Vytvoř kořen $root$

Pokud mají všechna data stejné g, označ kořen g a konec,

Pokud došly $Attributes$, označ $root$

    nejčastější hodnotou g v $data$ a konec

jinak

  X_j= atribut z $Attributes$ s maximálním $Gain(data, X_j)$

  označ $root$ atributem X_j

  pro každou hodnotu x_j atributu X_j,

    přidej větev pod $root$, odpovídající testu $X_j = x_j$

    $data_{X_j=x_j}$= podmnožina $data$, kde $X_j = x_j$

    Je–li $data_{X_j=x_j}$ prázdné, přidej list označený

      nejčastější hodnotou g v $data$ a konec

    jinak přidej podstrom $ID3(data_{X_j=x_j}, G, Attributes \setminus \{X_j\})$

vrať $root$

Klastrování

k–means (průměry)

- Učení bez učitele
- minimalizujeme ”chybovou funkci”, součet vzdáleností bodů od středů klastrů
- iterativně zlepšujeme, než dojdeme do lokálního minima

1. zadej počet klastrů k
2. náhodně zvol k bodů, resp. k instancí, které zvolíme za počáteční středy klastrů
3. inkrementálně pročítej data, dokud se mění přiřazení bodů ke klastrům
   (a) nový bod zařaď k nembližšímu klastru
   (b) spočti nový průměr či centroid

Prohledávání prostoru hypotéz

- **DATA**: příklad na tabuli,
- jeden atribut je **cílový**, u nás `EnjoySport`, ostatní jsou **vstupní**
- cílem učení je najít hypotézu – funkci, která na základě vstupních parametrů správně určí cílový atribut
- **positivní příklady** jsou data s hodnotou cílového atributu `Yes`, **negativní příklady** jsou data s hodnotou cílového atributu `No`.

Prostor hypotéz

- **Hypotézy** formulujeme v určitém vyjadřovacím jazyce v našem případě konjunkce testů vstupních atritutů, které charakterizují hodnotu cílového atributu `Yes`
  - hypotézy jsou formátu $\langle ?, Cold, High, ?, ?, ? \rangle$, kde
  - znak na pozici odpovídá podmínce na odpovídající vstupní (ne–cílový) atribut
  - znakem je buď konkrétní hodnota atributu, znak ? nekladoucí žádnou podmínku na daný atribut, znak $\emptyset$ odpovídající nesplňitelné podmínce
- Pro binární atributy máme $4^{|\text{počet atributů}|}$ hypotéz, hypotézy obsahující $\emptyset$ jsou ekvivalentní, tj. máme $3^{|\text{počet atributů}|} + 1$.
- Budeme prohledávat systematicky.

- Prostor hypotéz je **částečně uspořádaný inkluzí** $h_1 >_g h_2$ hypotéza h_1 je **obecnější** než h_2 (píšeme $h_1 >_g h_2$), pokud každý příklad splňující h_1 splňuje i h_2. V tom případě se h_2 nazývá **specifičtější** než h_1.
- Např. $\langle ?, ?, \ldots, ? \rangle$ je obecnější než $\langle Sunny, ?, \ldots, Same \rangle$.
- **Nejobecnější hypotéza** je $\langle ?, ?, \ldots, ? \rangle$, tu splňují všechna data
- **maximálně specifická hypotéza** je $\langle \emptyset, \emptyset, \ldots, \emptyset \rangle$, kterou nesplňuje žádný záznam.
- Prostor všech hypotéz tvoří svaz, viz. obrázek na tabuli.

Ohodnocovací funkce

- **ohodnocovací funkce** určuje, nakolik hypotéza odpovídá datům.
- Hledáme takovou hypotézu, kterou by splňovaly všechny pozitivní příklady a nesplňoval žádný negativní příklad.
- tj. aby byla implikace $hypoteza \Rightarrow (EnjoySport = Yes)$ pro všechna data pravdivá
- (to lze, pokud máme data bez náhody a šumu).

Nalezení maximálně specifické hypotézy odpovídající datům

Algoritmus FIND–S

1. $h \leftarrow \langle \emptyset, \ldots, \emptyset \rangle$ max. specifická hypotéza
2. pro každý positivní příklad x v datech

   pro každou podmínku na atribut $A_i = a_i$ v h

   Pokud příklad x nesplňuje $A_i = a_i$

   nahraď podmínku nejbližší obecnější podmínkou,

   kterou x splňuje

   jinak nech h beze změny
3. vydej hypotézu h

Ale:

- Je hypotéza nalezená FIND–S jediná konzistentní s daty?
- Proč tedy volit ji, ne nějakou maximálně obecnou či něco mezi?
- V jiném prostoru hypotéz nemusí být ani maximálně specifická hypotéza jednoznačná.
- **Budeme hledat všechny hypotézy konzistentní s daty.**
- Pokud nejsou trénovací data konzistentní, máme problém. Řešení je jiný typ hypotéz a jiná ohodnocovací funkce.

Prostor verzí

Prostor verzí vzhledem k prostoru hypotéz H a trénovacích dat D je podmnožina hypotéz z H konzistentní s trénovacími daty D,

$$VS_{H,D} = \{h \in H | Consistent(h, D)\}$$

- Tento prostor může být charakterizován obecnou a specifickou hranicí; každá hypotéza mezi těmito hranicemi spadá do prostoru verzí.
- **Obecná hranice G** vzhledem k H a D je množina maximálně obecných hypotéz z H konzistentních s daty, tj.

$$\begin{aligned} G = \quad \{ \quad & g \in H | Consistent(g, D) \& \\ & (\neg \exists g^{|} \in H)[(g^{|} >_g g) \& Consistent(g^{|}, G)]\} \end{aligned}$$

- **Specifická hranice S** vzhledem k H a D je množina maximálně specifických hypotéz z H konzistentních s daty, tj.

$$
\begin{aligned}
S = \quad \{ \quad & s \in H | Consistent(g, D) \& \\
& (\neg \exists s^{|} \in H)[(s >_{g} s^{|}) \& Consistent(s^{|}, G)]\}
\end{aligned}
$$

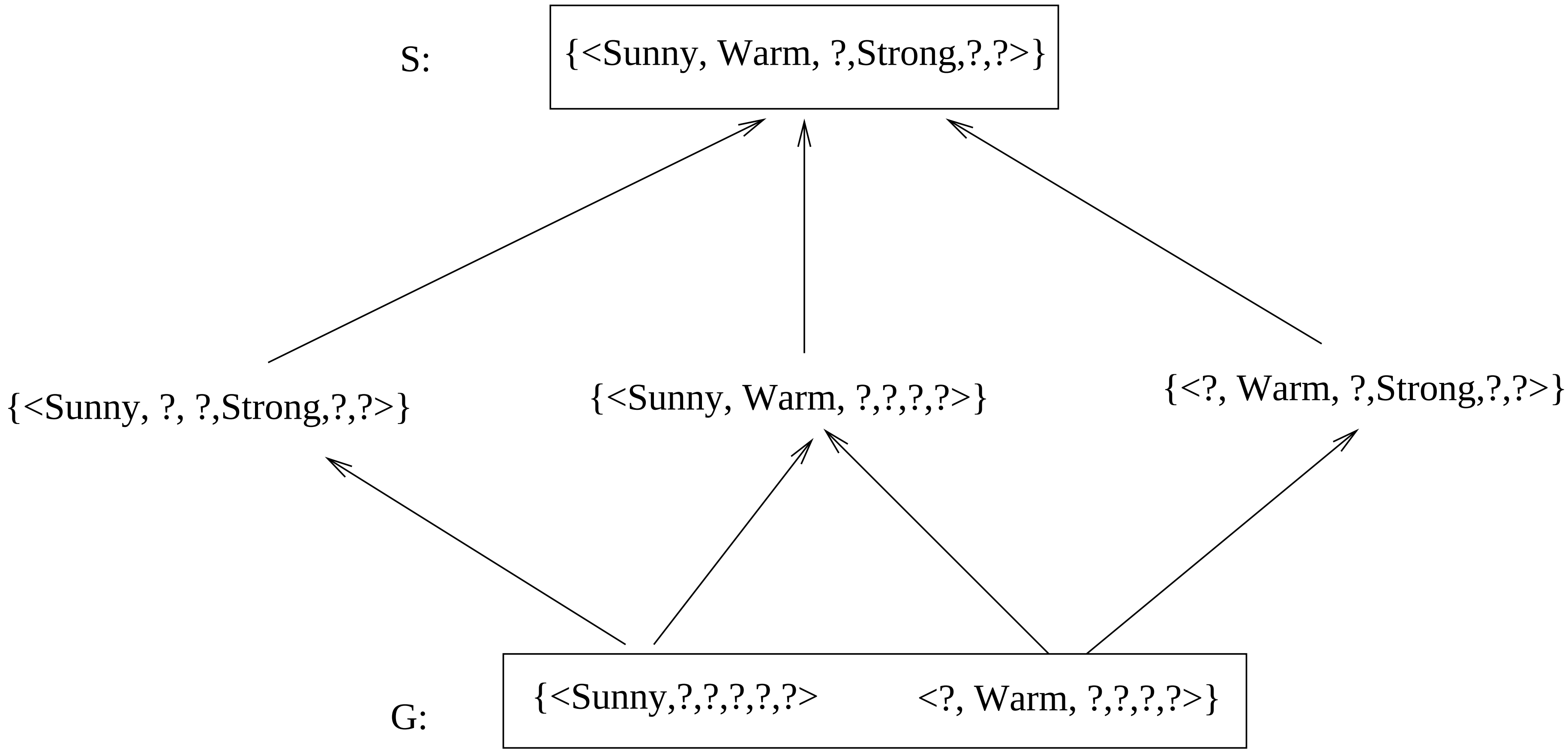


Figure 1: Prostor verzí s částečným uspořádáním inkluzí.

Algoritmus **Candidate–Elimination**

$G \leftarrow$ maximálně obecné hypotézy v H
$S \leftarrow$ maximálně specifické hypotézy v H

pokračuje

Pro každý trénovací příklad d, **do**

If d je positivní příklad

Odstaň z G všechny hypotézy nekonzistentní s d

For each $s \in S$, s nekonzistentní s d

Odstraň s z S

Přidej do S všechna h; minimální zobecnění s taková, že

h je konzistentní s d a zároveň $\exists g \in G; g >_g h$

Odstraň z S hypotézy, které nejsou maximálně specifické v S

If d je negativní příklad

Odstaň z **S** všechny hypotézy nekonzistentní s d

For each $g \in G$, g nekonzistentní s d

Odstraň g z G

Přidej do G všechna h; minimálně specifičtější než g taková, že

h je konzistentní s d a zároveň $\exists s \in S; h >_g s$

Odstraň z G hypotézy, které nejsou maximálně obecné v G

Zkoušky:

- středa nebo čtvrtek?
- 9 nebo 10?