Лекция 2. Пространството на версии и алгоритъм за елиминиране на кандидати

В тази лекция се разглежда друг подход към проблема за научаване на понятия, който премахва някои от ограничения, свързани с използване на алгоритъма FIND-S. Искам да ви напомня, че макар този алгоритъм намира в *H* такава хипотеза, която е съвместима с обучаващите примери, тя е само една от многото хипотези в *H*, които могат еднакво добре да покриват (”обясняват”) обучаващите данни. *Основната идея* на новия алгоритъм, наричан “Алгоритъм за елиминиране на кандидати” (CANDIDATE-ELIMINATION) е намирането на описанието на *множеството от всички хипотези, съвместими с обучаващи примери*. Най-интересно е, че този алгоритъм успява да определи това множество без явното указване (преброяване) на всички членове на това множество! Отново в основата на успеха е частичната наредба *по-обща-от*, която този път се ползва за поддръжка на едно компактно представяне на множеството от непротиворечиви хипотези и инкременталното уточняване на това описание при постъпване на всеки нов обучаващ пример.

2.1. Терминология

Ще въведем няколко термина. Първо, ще казваме, че една хипотеза е *съвместима с* (не противоречи на) *обучаващите* примери, ако тя класифицира правилно всички тези примери. По-формално:

Определение 1: Хипотезата *h* е *съвместима* с множество обучаващи примери *D* тогава и само тогава, когато *h(x) = c(x)* за всеки обучаващ пример <*x, c(x)*> от *D*:

$$Consistent(h, D) \equiv (\forall < x, c(x) > \in D) \quad h(x) = c(x)$$

Обърнете внимание на разликата между понятието “*пример удовлетворява хипотезата*” и “*хипотезата съвместима с примера*”. Ние казваме, че примерът *x удовлетворява* хипотезата *h*, ако *h(x) = 1*, без разликата, дали той е положителен или отрицателен пример на научаваното понятие. Обаче, дали един пример е съвместим с хипотезата, зависи от целевото понятие, в частност, дали *h(x) = c(x)*.

Множеството на всички хипотези от *H,* съвместими с множеството от обучаващите примери *D,* ще наричаме *пространството на версии* относителното *H* и *D*, тъй като то съдържа всички правдоподобни (приемливи) версии на целевото понятие.

Определение 2: Пространството на версии $VS_{H,D}$ относително пространството на хипотезите *H* и обучаващите примери *D* е подмножеството от хипотезите от *H* съвместими с обучаващите примери от *D*.

$$VS_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

2.2. Алгоритъм "Изброи и след това елиминирай" (LIST-THEN-ELIMINATE)

Един очевиден начин за представяне на пространството на версии е да изброим всички неговите членове. Това води до един лесен алгоритъм за научаване на понятия, наречен "Изброи и след това елиминирай", който може да се опише по следния начин:

Алгоритъм LIST-THEN-ELIMINATE

1. $VS_{H,D} \leftarrow$ списък, съдържащ всички хипотези в H
2. За всеки обучаващ пример $<x, c(x)>$ направи:
   Изтрий от $VS_{H,D}$ всяка хипотеза h, за която $h(x) \neq c(x)$
3. Изведи като изход списък на всички хипотези от $VS_{H,D}$

Този алгоритъм инициализира пространството на версиите със списъка на всички възможни хипотези от H, а след това премахва от тях тези, които са несъвместими с всеки нов постъпващ обучаващ пример. По този начин със всеки постъпващ обучаващ пример пространството на версиите се намалява докато, в идеалния случай, в него не се остане само една единствена хипотеза, съвместима с всички обучаващи примери. Предполага се, че това е описанието на желаното целево понятие. Ако наличните обучаващи примери не са достатъчни, за да сведат пространството на версии само до една хипотеза, алгоритмът извежда като решение цялото множество от хипотези, съвместими с наличните данни.

Очевидно е, че този алгоритъм е приложим само, ако пространството на хипотези е крайно. Той гарантира намирането на всички хипотези, съвместими с обучаващите примери, обаче изисква пълното изброяване на всички хипотези в H – едно много не реалистично изискване практически за всички нетривиални реални задачи.

2.3. По-компактно представяне на пространството на версии

Пространството на версии може да бъде представено *неявно*, обаче значително по-компактно, чрез свои най-общи и най-специфични членове. Тези членове формират два гранични множества (най-общото и най-специфичното), които ограничават пространството на версии в частично наредено пространство от хипотези.

Определение 3: **Общата граница G** по отношение на пространството на хипотези *H* и обучаващите данни *D* е *множеството от най-общи членове на H, съвместими с D.*

$$G \equiv \{g \in H \mid Consistent(g, D) \wedge (\neg \exists g' \in H)[(g' >_g g) \wedge Consistent(g', D)]\}$$

Определение 4: **Специфичната граница S** по отношение на пространството на хипотези *H* и обучаващите данни *D* е *множеството от най-малко общи (т.е. най-специфични) членове на H, съвместими с D.*

$$S \equiv \{s \in H \mid Consistent(s, D) \wedge (\neg \exists s' \in H)[(s >_g s') \wedge Consistent(s', D)]\}$$

Тези два множества *G* и *S напълно определят пространството на версии*. Може да се докаже, че пространството на версии е равно на множеството от хипотези, съдържащи се в *G*, плюс хипотезите, съдържащи се в *S*, плюс хипотезите, лежащи между тези два множества в частично наредено пространство от хипотези.

Теорема за представяне на пространството на версии. Нека *X* е едно произволно множество от примери, а *H* – множеството от двоични хипотези, определени на *X*. Нека $c: X \rightarrow [0, 1]$ е произволно целево понятие, определено на *X*, и нека *D* е множество от обучаващи примери *{<x, c(x)>}*. За всички *X, H, c* и *D*, такива, че *S* и *G* са добре определени, следва, че :

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}$$

2.4. Алгоритъм за елиминиране на кандидати

Алгоритъм CANDIDATE-ELIMINATION

Инициализирай *G* с множеството от най-общи хипотези в *H*
Инициализирай *S* с множеството от най-специфични хипотези в *H*
За всеки обучаващ пример *d* направи:

- Ако *d* е *положителен* пример, то
  - Изтрий от *G* всички хипотези, несъвместими с *d*
  - За всяка хипотеза *s* от *S*, която несъвместима с *d*, направи:
    - Изтрий *s* от *S*
    - Добави към *S* всички *най-малки обобщения h* на *s*, такива, че:
      - *h* е съвместима с *d* и някои членове на *G* са по-общи от *h*
    - Изтрий от *S* всяка хипотеза, която е *по-обща* от някоя друга хипотеза в *S*
- Ако *d* е *отрицателен* пример, то
  - Изтрий от *S* всички хипотези, несъвместими с *d*
  - За всяка хипотеза *g* от *G*, която несъвместима с *d*, направи:
    - Изтрий *g* от *G*

- Добави към G всички *най-малки специализации* h на g, такива, че:
  - h е съвместима с d и има някой член на S, който е по-специфичен от h
- Изтрий от G всяка хипотези, която *по-малко обща* от някоя друга хипотеза в G

Алгоритмът изчислява пространството на версии, съдържащо всички хипотези от H, съвместими с подавана последователност от обучаващи примери. Той започва работата си с инициализация на пространството на версии с цялото пространство от хипотези. След постъпване на всеки нов обучаващ пример той обобщава и специализира съответните множества S и G, представящи в явен вид границите на пространството на версии, като целта е новото пространство на версии *да не съдържа* хипотези, несъвместими с наблюдавани до момента обучаващите данни. След като бъдат обработени всички обучаващите примери, останалото пространство на версии ще съдържа всички (и само такива) хипотези, които са съвместими със всички подадени данни.

Обърнете внимание, че алгоритмът е описан в термините на операции, изчисляващи най-малки обобщения и специализации на хипотези, както и на операции, идентифициращи неминимални и немаксимални хипотези. Реалната имплементация на тези операции зависи от избраното представяне на примери и хипотези. Обаче, самият алгоритъм може да се прилага към произволна задача за научаване на понятия, за която пространството на хипотези е добре дефинирано.

2.5. Илюстративен пример

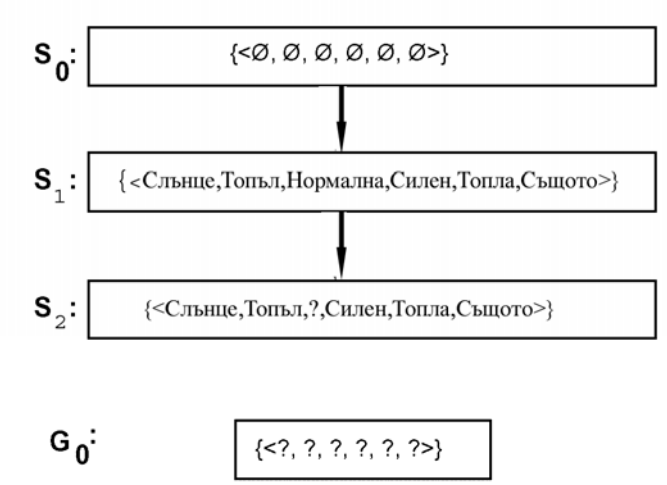
Да проследим работата на алгоритъма на примера на задачата за научаване на понятие *Харесва*, описано чрез Табл. 1-2, при условие, че обучаващите примери се подават в съответствие с Табл. 1-1.

Първата стъпка е инициализирането на граничните множества, съдържащи най-общите и най-специфичните хипотези от H:

$$G_0 \leftarrow \{<?,?,?,?,?,?>\} \qquad S_0 \leftarrow \{<\varnothing,\varnothing,\varnothing,\varnothing,\varnothing,\varnothing>\}$$

При постъпване на първия пример (който в случая е положителен), алгоритмът проверява границата S и намира, че тя е прекалено специфична – тя не може да покрие конкретния положителен пример. По тази причина границата се ревизира чрез преместване към най-малко по-обща хипотеза, която покрива примера. Най-общата граница G_0 не се ревизира, тъй като тя коректно покрива този пример.

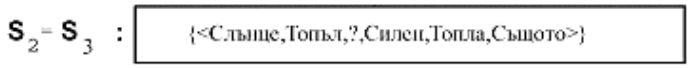
При постъпване на втория (също положителен) пример, граница S_1 се обобщава до S_2, а G отново остава непроменена. Обърнете внимание, че на този етап обработка на примери работа на алгоритъма много напомня работата на FIND-S.

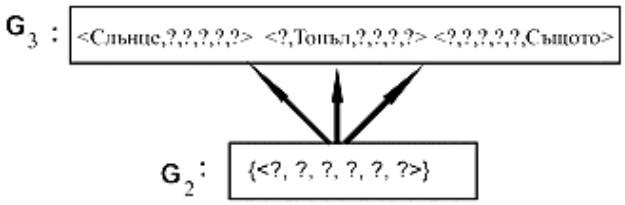


Обучаващи примери:

1. <Слънце,Топъл,Нормална,Силен,Топла,Същото> Харесва = да
2. <Слънце,Топъл,Висока,Силен,Топла,Същото> Харесва = да

Третия пример е отрицателен. Той показва, че текущата най-обща граница *G* е прекалено обща – тя грешно покрива този отрицателен пример. Следователно, границата трябва да бъде специализирана, за да класифицира правилно този пример като отрицателен. Тъй като има няколко възможни минимално по-специфични хипотези, отговарящи на този критерий, всички те трябва да станат членове на тази нова, по-специфична най-общата граница G_3:



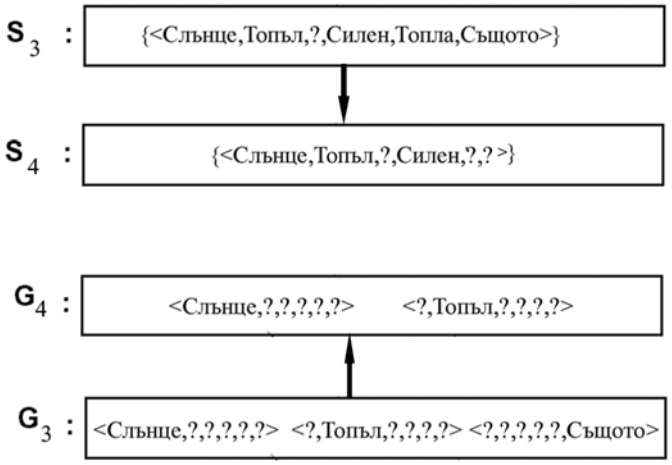


Обучаващи примери:

3. <Дъжд,Студено,**Нормална**,Силен,Топла,Промяна> Харесва = не

При условие, че разполагаме с шест различни атрибута, които могат да бъдат фиксирани, за да специализираме G_2, защо само три хипотези са включени в G_3? Например, хипотезата $h \leftarrow$ *<?,?,Висока,?,?,?>* е също минималната специализация

на G_2, която правилно класифицира новия пример като отрицателен. Обаче, тя не е включена в G_3 – защо? Причината е, че тази хипотеза е несъвместима с предишните положителни примери. Алгоритмът разпознава това чрез определяне, че тази хипотеза не е по-обща от текущата специфична граница S_2. Текущата граница S на пространството на версии *формира "резюме" на всички срещнати по-рани положителни примери*, които могат да се ползват за определяне на това, дали някоя конкретна хипотеза е съвместима с всички тези положителни примери. Всяка хипотеза, по-обща от S, по определение ще покрива всеки пример, който е покрит от S, т.е. ще покрива всеки положителен пример, срещнат по-рано. Аналогично, границата G *резюмира информацията за всички срещнати по-рано отрицателни примери*. Всяка хипотеза, по-специфична от G, е гарантирано съвместима с всички известни отрицателни примери. Това е така, тъй като всяка такава хипотеза, по определение, не може да покрива примери, които не се покриват от G.



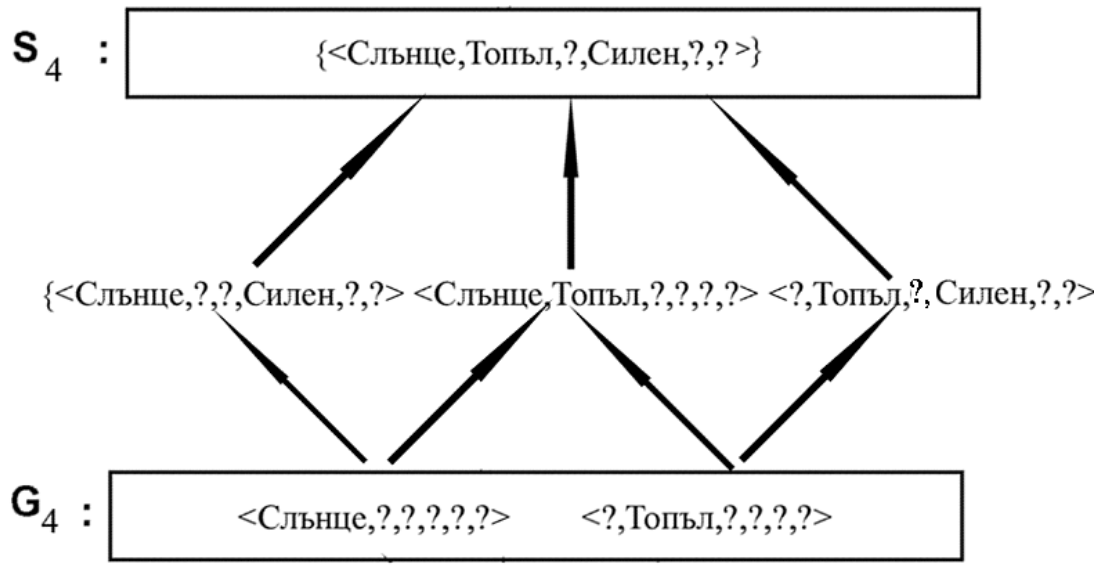
Обучаващи примери:

4. <Слънце,Топъл,Висока,Силен,Студена,Промяна> Харесва = да

Четвъртия обучаващ пример (положителен) води до по-нататъшното обобщаване на границата S. Обаче, той също така води и до премахване на един член (хипотеза) от границата G, тъй като тя не покрива този нов пример (виж първата стъпка под условието "ако d е положителен пример" в описанието на алгоритъма за елиминиране на кандидати). Защо тази хипотеза трябва да бъде премахната? Обърнете внимание, че тя не може да бъде специализирана, тъй като след нейното специализиране тя повече няма да може да покрива новия пример. Тя също така не може и да бъде обобщена, тъй като, по определение на G, всяка по-обща хипотеза ще покрива най-малко един отрицателен пример. Следователно, тази хипотеза трябва да бъде премахната от G, водейки по този начин до премахване на целия клон от частичната наредба на хипотези в разглежданото пространство на версии.

И така, след обработката на четири примера граничните множества S_4 и G_4 ограничават пространството на версии на *всички* хипотези, съвместими с тези

инкрементално постъпили примери. Пълното пространство на версии, заключени между тези граници, *е независимо* от реда на постъпване на обучаващите примери (тъй като на края, то съдържа всички хипотези съвместими с цялото множество от обучаващите примери).



Фиг. 2-1. Окончателното пространство на версии за задачата за научаване на понятие *Харесва*.

2.6. Забележки

2.6.1. Ще има ли алгоритмът за елиминиране на кандидати сходимостта към правилната хипотеза?

Пространството на версиите, научавано от алгоритъма за елиминиране на кандидати, ще се приближава към хипотезата, която правилно описва целевото понятие, ако изпълнени две условия: 1) обучаващите примери не съдържат грешки и 2) в H съществува някоя хипотеза, правилно описваща целевото понятие. При постъпване на достатъчен брой обучаващите примери пространството на версии инкрементално се "стяга", намалявайки неопределеността в описание на целевото понятие. То се научава точно, когато граничните множества S и G се "стягат" до една и съща единствена хипотеза.

Какво ще стане, ако обучаващите данни съдържат грешки? Например, нека втория обучаващ пример от разгледаната поредица, е грешно представен като отрицателен, вместо положителен. За съжаление, в този случай алгоритмът със сигурност ще изтрие правилното описание на целевото понятие от пространството версии! Причината е, че той изтрива от пространството на версии всички хипотези, несъвместими с обучаващите данни, т.е. истинското целево понятие ще бъде

изтрито от пространството на версии веднага след постъпване на сгрешен пример. Естествено, ако разполагаме с достатъчен брой обучаващите примери, ние ще сме в състояние да разберем за съществуващата противоречивост в данни – това ще стане, когато границите S и G ще се стегнат до празното множество, което означава, че няма нито една хипотеза, съвместима с всички данни.

Същото ще стане, когато данните са верни, обаче целевото понятие не може да бъде описано с избрания език за представяне на хипотези. Например, ако то се описва чрез дизюнкцията на атрибути, а хипотезите – чрез тяхната конюнкция). Ще разгледаме подобни ситуации по-късно.

2.6.2. Кой следващ обучаващ пример трябва да бъде научен?

До сега ние предполагахме, че обучаващите примери се дават на обучаемия от някой външен учител. Да предположим, че обучаемия може сам да прави експерименти, в които той може да избира следващия пример, а след това да получава неговата правилна класификация (от някой външен източник – оракул, или от учител). За подобни примери се използва термин "заявка".

Да разгледаме пак пространството на версии, научено след четири обучаващи примера на целевото понятие *Харесва*. Каква добра заявка би трябвало да изпрати обучаемият в следващия момент? Каква трябва да бъде добрата стратегия за изпращане на заявки?

Ясно е, че обучаемият трябва да опита да избира сред алтернативни състезаващи се хипотези от пространството на версии, т.е. той трябва да избере пример, който се класифицира като положителен от една част от хипотези, и като отрицателен – от другата. Един от такива примери е

<Слънце,Топъл,Нормална,Слаб,Топла,Същото>

Този пример удовлетворява 3 от 6 хипотези в текущото пространство на версии. Ако учителят класифицира този пример като положителен, границата *S* на *VS* може да бъде обобщена. Ако учителят класифицира примера като отрицателен – ще бъде специализирана границата *G* на пространството на версии. И в двата случая обучаемият ще напредне в научаването на целевото понятие, намалявайки пространството на версии от 6 хипотези до три!

В общия случай, оптималната стратегия за заявки се състои в генериране на примери, които удовлетворяват точно половината от хипотези в текущото пространство на версии. Ако това е възможно, то пространството на версии ще се намалява на половина след всяка нова заявка, и следователно, целевото понятие ще може да бъде научено само за $\lceil \log_2 |VS| \rceil$ брой експерименти.

2.6.3. Как да се използват частично научени понятия?

Да предположим, че не разполагаме с никакви други обучаващи примери, извън тези 4 от разгледания вече случай, обаче трябва да класифицираме нови примери, които досега на нас не са известни. Макар, че наученото от нас пространството на версии съдържа няколко хипотези, показвайки по този начин, че целевото понятие още не е научено абсолютно точно, ние все пак можем да го използваме за класификация на новопостъпилите примери с определена степен на увереност в тяхната класификация. За илюстрация, да разгледаме няколко нови примера за класифициране, събрани в Таблица 2.1.

Пример	*Небе*	*Въздух*	*Влажност*	*Вятър*	*Вода*	*Прогноза*	*Харесва*
A	Слънце	Топъл	Нормална	Силен	Студена	Промяна	?
B	Дъжд	Студен	Нормална	Слаб	Топла	Същото	?
C	Слънце	Топъл	Нормална	Слаб	Топла	Същото	?
D	Слънце	Студен	Нормална	Силен	Топла	Същото	?

Примерът A ще се класифицира като положителен от *всички* хипотези от текущото пространство на версии. Следователно, обучаемият трябва да го класифицира като положителен, при това със същата степен на убеденост, като все едно, че знае абсолютно точно описание на целевото понятие.

Трябва да отбележим, че за да проверим, дали всяка хипотеза от пространството на версии класифицира примера като положителен, изобщо не е необходимо явно да разписваме всяка хипотеза от *VS*. Напълно е достатъчно, да проверим, дали този пример удовлетворява всички хипотези от граничното множество *S*! Причината е, че всяка друга хипотеза от *VS* е по-обща или равна на някой член на *S*. Следователно, съгласно определението на релацията *по-обща-от* получаваме, че ако новият пример удовлетворява всички членове на *S*, той ще удовлетворява и всички по-общи хипотези.

Аналогично, *примерът B* ще се класифицира като отрицателен от всички хипотези в *VS* и, следователно, трябва да бъде класифициран от обучаемия като отрицателен. Ефективният тест за проверка на това условие е, че примерът не трябва да удовлетворява нито един от членове на множеството *G*.

Ситуацията с *примера C* е съвсем друга – едната половина от хипотезите го класифицира като положителен, а другата – като отрицателен. Затова обучаемият не може със сигурност да го класифицира, до като не бъдат достъпни нови обучаващи примери.

Примерът *D* се класифицира като положителен от 2 хипотези от *VS*, и като отрицателен – от останалите 4 хипотези. В този случай имаме по-малко убеденост в нашата класификация от случаите *A* и *B*. Ако предположим, че всички хипотези са априори равновероятни, можем да приложим подход на "*гласуване по мнозинство*", който осигурява най-вероятната класификация на примера. Освен

това, пропорцията на хипотези, гласуващи за положителната класификация, може да се интерпретира като вероятността, че дадения пример е положителен при налични обучаващи данни.

2.7. Индуктивното пристрастие

Както беше отбелязано по-рано, алгоритмът за елиминиране на кандидати гарантира научаване на целевото понятие, ако обучаващите примери са зададени правилно и началното пространство на хипотези съдържа целевото понятие. А ако понятието не се съдържа в пространството на хипотези? Можем ли ние да избегнем този проблем чрез използване на такова пространство от хипотези, което съдържа всички възможни хипотези? Как размерът на подобното пространство от хипотези ще повлияе на възможността на алгоритъма да обобщава нови примери? Как размерът на подобното пространство от хипотези ще повлияе на броя обучаващите примери, необходими за научаване на понятие? Това са въпроси, фундаментални за индуктивния извод като цяло. Ще опитаме да отговорим на тях в контекста на алгоритъма за елиминиране на кандидати, обаче получените изводи са валидни за всяка система за научаване на понятия, която извежда хипотези, съвместими с обучаващи данни.

2.7.1. Пространство на "пристрастни" хипотези

Да предположим, че искаме да бъдем сигурни, че пространството на хипотези съдържа търсеното целево понятие. Едно очевидно решение е да обогатим пространството на хипотези чрез включването в него на *всички възможни* хипотези. За илюстрация да разгледаме примера с понятие *Харесва*, в който ние сме ограничили пространството на хипотези само до тези, които са направени от конюнкция от ограничения върху атрибути. Поради това ограничение пространството на хипотези не е в състояние да представи, например, едно просто дизюнктивно понятие като *"Небе = Слънце" или "Небе = Облаци*". Фактически, имайки три обучаващи примера на това дизюнктивно понятие, нашият алгоритъм ще намери, че пространството от хипотези е празно.

Пример	*Небе*	*Въздух*	*Влажност*	*Вятър*	*Вода*	*Прогноза*	*Харесва*
1	Слънце	Топъл	Нормална	Силен	Студена	Промяна	да
2	Облаци	Топъл	Нормална	Силен	Студена	Промяна	да
3	Дъжд	Топъл	Нормална	Силен	Студена	Промяна	не

Действително, най-специфичната хипотеза, съвместима с първите два примера е:

S_2: <?, Топъл, Нормална, Силен, Студена, Промяна>

Обаче, макар че хипотезата е най-специфичната в *H*, тя все пак е прекалено обща, тъй като неправилно покрива третия (отрицателен) пример. Проблемът е в това, че

нашия алгоритъм има *пристрастие* към научаване само на конюнктивни хипотези, а дадения случай изисква едно по-изразително пространство от хипотези.

2.7.2. Безпристрастен алгоритъм за научаване на понятия

За да осигурим наличието в пространството на хипотези на търсеното целево понятие, едно очевидно решение е да снабдим пространството от хипотези с възможността да представя *всички научаеми понятия*, т.е. то да бъде в състояние да представя всяко подмножество от множеството на примери. Терминологически, множеството от всички подмножества на множество X се нарича *множество-степен* на X. В общия случай броят на различни подмножества на множество X, т.е. размерът на множеството-степен на X е $2^{|X|}$. В случая на понятие *Харесва*, описвано със 6 атрибута, $|X| = 96$ и размерът на множеството-степен е 2^{96} или приблизително 10^{28} – това са брой на различни целеви понятия, които могат да се дефинират в това пространство и които могат потенциално да бъдат научени. Да напомня, че размер на пристрастното (конюнктивното) пространство от хипотези за тази задача е само *973*!

Нека да преформулираме задачата за научаване на понятие *Харесва* по "безпристрастен" начин, определяйки новото пространство от хипотези H', което може да представя всяко подмножество от примерите X, т.е. H' е множеството-степен от X. Един способ да дефинираме подобното пространство от хипотези е да позволим всякакви конюнкции, дизюнкции и отрицания при конструиране на хипотези. Например, целевото понятие *"Небе = Слънце" или "Небе = Облаци*" сега може да се представи като <*Слънце,?,?,?,?,?*> v <*Облаци,?,?,?,?,?*>.

В това пространство на хипотези ние можем да използваме алгоритъма за елиминиране на кандидати без да се притесняваме, че целевото понятие може и да не бъде изразено с избрания език за представяне на хипотези. Обаче, премахвайки проблема с изразителността на езика на хипотезите, този подход води до другия, не по-лек проблем: нашият алгоритъм сега е напълно лишен от своята възможност да обобщава извън наблюдаваните примери! Да предположим, че ние научихме 3 положителни (x_1, x_2, x_3) и 2 отрицателни (x_4, x_5) примера. В този момент границата S на пространството на версии ще съдържа само една хипотеза, която е просто дизюнкцията от положителните примери: $S : \{< x_1 \text{ v } x_2 \text{ v } x_3>\}$, тъй като това е най-специфичната хипотеза, покриващата всички 3 положителни примера. Аналогично, границата G ще съдържа само една хипотеза, която отхвърля 2 известни отрицателни примери: $G: \{\neg< x_4 \text{ v } x_5>\}$. Проблемът с това много изразително представяне е, че S *винаги* ще бъде дизюнкцията на всички наблюдавани до момента положителни примери, а G – отрицанието на дизюнкцията на всички отрицателни примери. По тази причина, единствени примери, които могат да бъдат класифицирани с абсолютна сигурност, е само тези, които *съвпадат с вече известни обучаващи примери!* Освен това, за да осигурим сходимостта на подобен алгоритъм към едно единствено целево понятие ние трябва да му представим а*бсолютно всички примери* от X като обучаващи примери!

2.7.3. Безполезност на безпристрастното обучение

И така, фундаменталната характеристика на индуктивния извод е, *че алгоритмът за самообучение, който не прави никакви предварителни предположения за същността на целевото понятие, няма никакво рационално основание за класификация на непознати примери.* Тъй като индуктивното самообучение изисква предварителни предположения за характера на научаваното понятие, т.е. *индуктивното пристрастие*, полезно е да опитаме да характеризираме различни алгоритми за научаване на понятия чрез тяхното индуктивно пристрастие.

Нека да формулираме това понятие по-формално, като основната идея е да "хванем" какъв метод използва дадения алгоритъм при обобщение извън наблюдаваните обучаващи примери.

Да предположим, че имаме някакъв алгоритъм за самообучение L, на който е представено множеството от обучаващите примери $D_c = \{<x, c(x)>\}$ на понятието c. След обучението L трябва да класифицира новия пример x_i. Нека $L(x_i, D_c)$ означава класификацията на x_i, която алгоритмът присвоява на примера след обучението с D_c. Тази индуктивна стъпка може да се представи по следния начин:

$$(D_c \wedge x_i) \succ L(x_i, D_c)$$

където $y \succ z$ означава, че z е индуктивно изведен от y. В случая на нашата задача за научаване на понятието *Харесва*, нека L е алгоритмът за елиминиране на кандидати, D_c са обучаващите примери от Таблица 1.1, а x_i е първия пример от Таблица 1.3. Тогава индуктивният извод, изпълняван в този случай, води до заключение, че $L(x_i, D_c) = (Харесва = да)$.

Тъй като L е индуктивният алгоритъм, то резултатът от неговата класификация не винаги е верен, т.е. тази класификация не следва дедуктивно от обучаващите данни и описание на примера, подлежащ на класифициране. Обаче, възниква въпросът – какво може да се добави като знания, за да може класификацията на примера да се извежда дедуктивно? Точно това множество от допълнителни предположения B, достатъчни за да оправдаят индуктивният извод като дедуктивния, ще наричаме *индуктивното пристрастие на алгоритъма L*:

Определение: Нека L е алгоритъм за научаване на понятия, а X е множество от примери. Нека c е някое понятие, определено на X и $D_c = \{<x, c(x)>\}$ е някое множество от обучаващи примери на c. Нека $L(x_i, D_c)$ е класификация, назначавана на пример x_i от L след обучение върху D_c. Тогава индуктивното пристрастие на L е *минималното множество от предположения B* такива, че за всяко целево понятие c и съответните обучаващите примери D_c

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \Rightarrow L(x_i, D_c)$$

където $y \Rightarrow z$ означава, че z следва дедуктивно (т.е. се извежда) от y.

Индуктивна и дедуктивна система

Индуктивното пристрастие на алгоритъма за елиминиране на кандидати е предположението, че целевото понятие се съдържа в изследваното пространство от хипотези.

Резюме

- Научаването на понятия може да се разглежда като задача за търсене в голямо, предварително дефинирано пространство от потенциални хипотези.
- Частичната наредба на хипотези *от-общото-към-частното*, която може да бъде дефинирана за всяка задача за научаване на понятия, осигурява една полезна структура за организиране на търсене в пространството на хипотези.
- Алгоритмът FIND-S използва наредбата *от-общото-към-частното,* изпълнявайки в пространството на хипотези търсене в посока от частното към общото само по един клон на тази частична наредба, за да намери най-специфичната хипотеза, съвместима с обучаващите данни.

- Алгоритмът за елиминиране на кандидати използва наредбата *от-общото-към-частното* за изчисляване на пространството на версии (множество от всички хипотези, съвместими с обучаващите данни) чрез инкременталното изчисляване на множествата от максимално специфични (*S*) и максимално общи (*G*) хипотези.
- Тъй като *S* и *G* ограничават цялото множество от хипотези, съвместими с обучаващите данни, те осигуряват обучаемия със степента на сигурност относно точното описание на целевото понятие. Пространството на версии може да бъде проверявано, за да се определи, дали алгоритмът има сходимост към целевото понятие, дали обучаващите данни не са противоречиви. Алгоритмът позволява и да генерира информационни заявки за по-нататъшно уточняване на пространството на версии, както и да определя, кои нови примери могат да бъдат класифицирани с абсолютна сигурност.
- Индуктивните алгоритми за обучение са способни да класифицират нови примери само заради това, че имат неявно индуктивно пристрастие, позволяващо да бъдат предпочитани едни съвместими хипотези пред други. Пристрастието на алгоритъма за елиминиране на кандидати е предположението, че целевото понятие се намира в пространството на хипотези ($c \in H$). Изходните хипотези и класификацията на новите примери следват дедуктивно от това предположение и обучаващите данни.