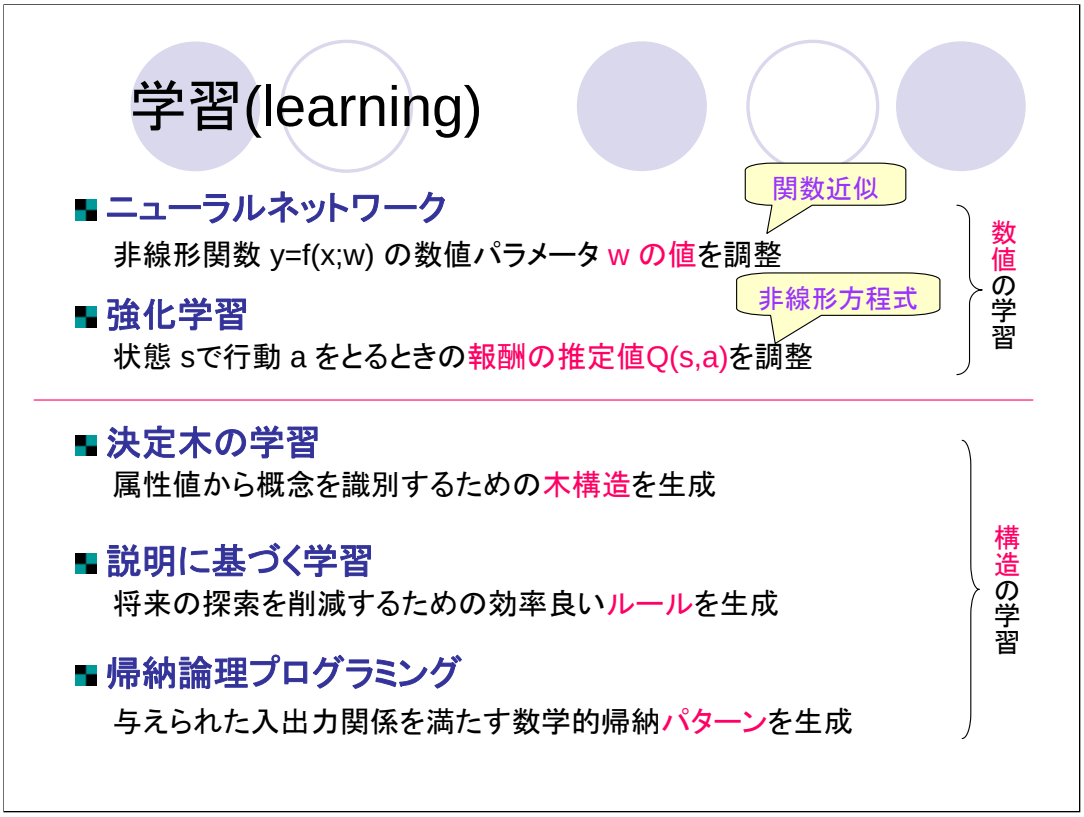
知能情報処理　学習
訓練と経験から知識を獲得するエージェント

決定木の学習

- 学習
- 決定木
- ID3アルゴリズム
- 性能評価と応用



人工知能における「**学習**」とは，エージェントの意思決定や行動に必要なアルゴリズムを事前に詳細設計できないときに，不完全ながらも，とにかく動くように大枠だけを作っておき，後に例題や実経験によってエージェントを**訓練**することによって，詳細部分に必要な知識をエージェント自身に仮説として生成させる**知識獲得**の技術である．この機能がうまく働けば，一部欠損した**不完全な知識**しかなくても，学習で得られた仮説によって知識を補って，だんだん現在よりもうまい意思決定や行動ができるようになってくる．

実際には，どのような状況でどんな情報から何を学習するのかによって，さまざまな学習手法が研究開発されてきている．よく見かけるものから5つ選んでみたのがこのスライドである．

ニューラルネットワークは神経回路網の構造にヒントを得た数理モデルで，ニューロンと呼ばれる基本素子の結合によって，ある種の非線形関数 y=f(x; w)を表現しているものである．その学習の基本的な考え方は，入力 x およびそれに対する望ましい出力 y からなる訓練例が多数与えられたときに，ほぼ y=f(x; w) となるように w というパラメータ（重みとか結合係数と呼ばれる）を決定することである．したがって，コンピュータサイエンスの数値計算の分野で関数近似と呼ばれている技術に関係している．f が w に関して線形なら最小二乗法が思い出されるところである．

強化学習は，基本的には，エージェントの行動がマルコフ決定過程(MDP)と呼ばれる数理モデルで与えられているときの学習方法である．エージェントは各状態 s で何らかの行動 a をとると，ある報酬 r を得る．そこで長期にわたって高い期待報酬を得るために，各 s, a ごとに状態 s で行動 a をとるときの報酬の推定値 Q(s,a) を経験から学習させる．Q(s,a)はいろいろな s, aにからんで再帰的な方程式を満たすので，この技術は数値計算における非線形方程式の反復解法に似ている面がある．

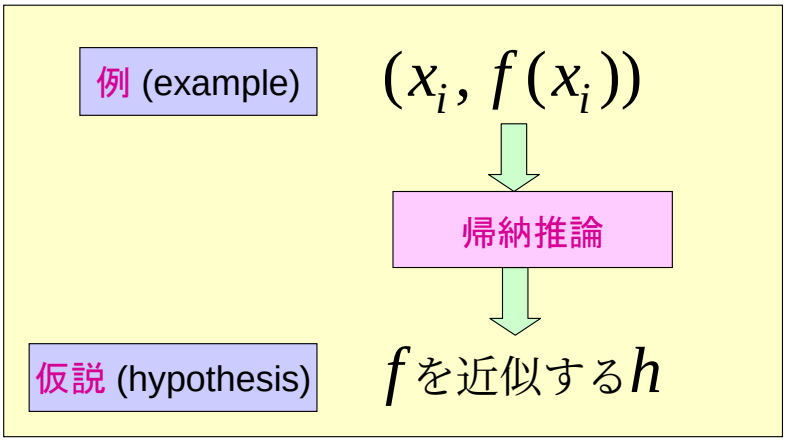
これら2つの学習は，情報処理的には，いずれも簡単な数値計算によって「数値」を学習するものである．

それに対し，より高度なことをねらった研究分野では，「構造」を学習しようとしている．

今回の授業で学ぶ**決定木の学習**では，その構造とは「木」である．

帰納推論(inductive inference)

多くの学習アルゴリズムは帰納推論の形式をとる.

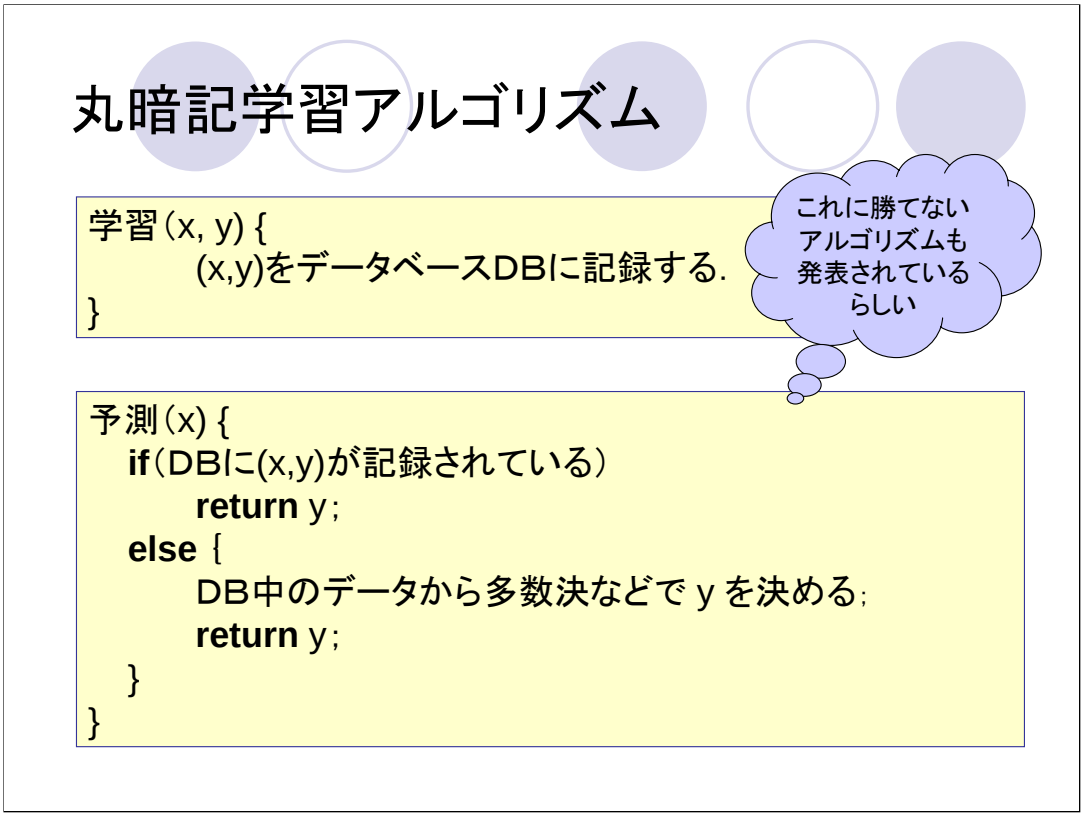


ある仮説を他の仮説より優先して選ぶ基準をバイアス(bias)という.

決定木の学習に入る前に，学習という技術全般をやや形式的に特徴付けておく.

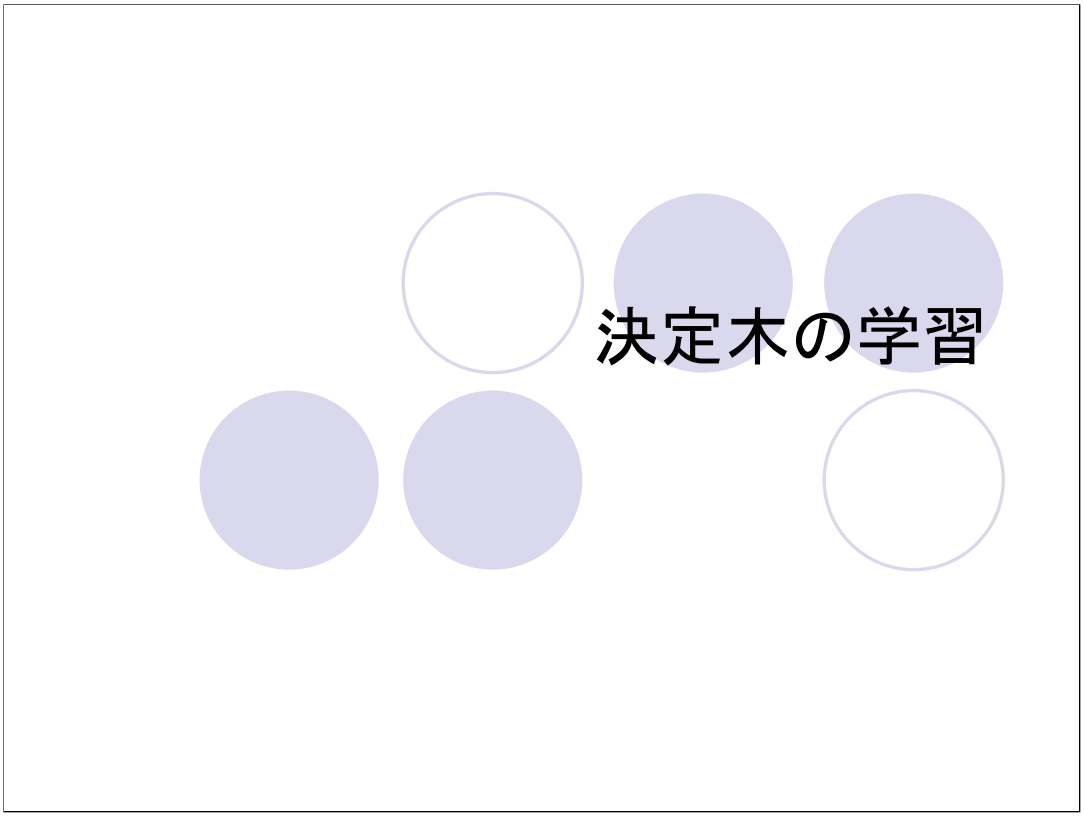
多くの学習アルゴリズムは，論理学的には，**帰納推論**と呼ばれるものの一種となる. すなわち，入力 x と出力 y を関係付けるある未知関数 y=f(x) があるとして，その入出力の**例** (xi, f(xi))が数多く与えられたとしよう. 帰納推論は，それらのデータを一般化して，f を近似すると考えられる関数 h を**仮説**として出力するものである.

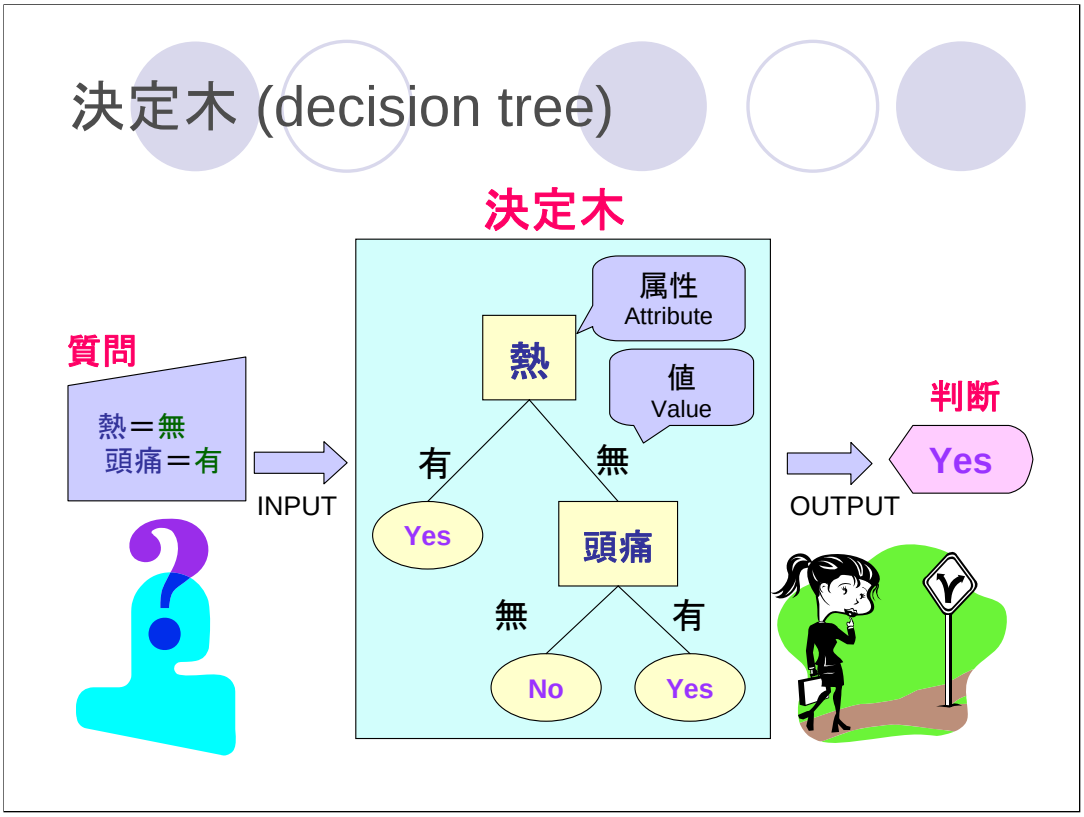
もちろん，そのような仮説 h の候補はたくさんあり得るわけだが，ある仮説を他の仮説より優先して選ぶ基準(**バイアス**)を適当に設定して，適切な h を出力するのである.



学習アルゴリズムの自明なものがこのスライドに書いてあるものである.

このアルゴリズムは**丸暗記学習**ともいえるものである. その問題点は, DBに(x,y)の膨大なデータを記憶する必要があることと, それらのデータを一般的に説明する有益な知識を獲得していないので, 未知のxに対して出力yを予測する目的にはほとんど役に立たないことである. 学習アルゴリズムと名のるものの中には, このアルゴリズムにも勝てないものがあるらしいので注意が必要である.


決定木の学習



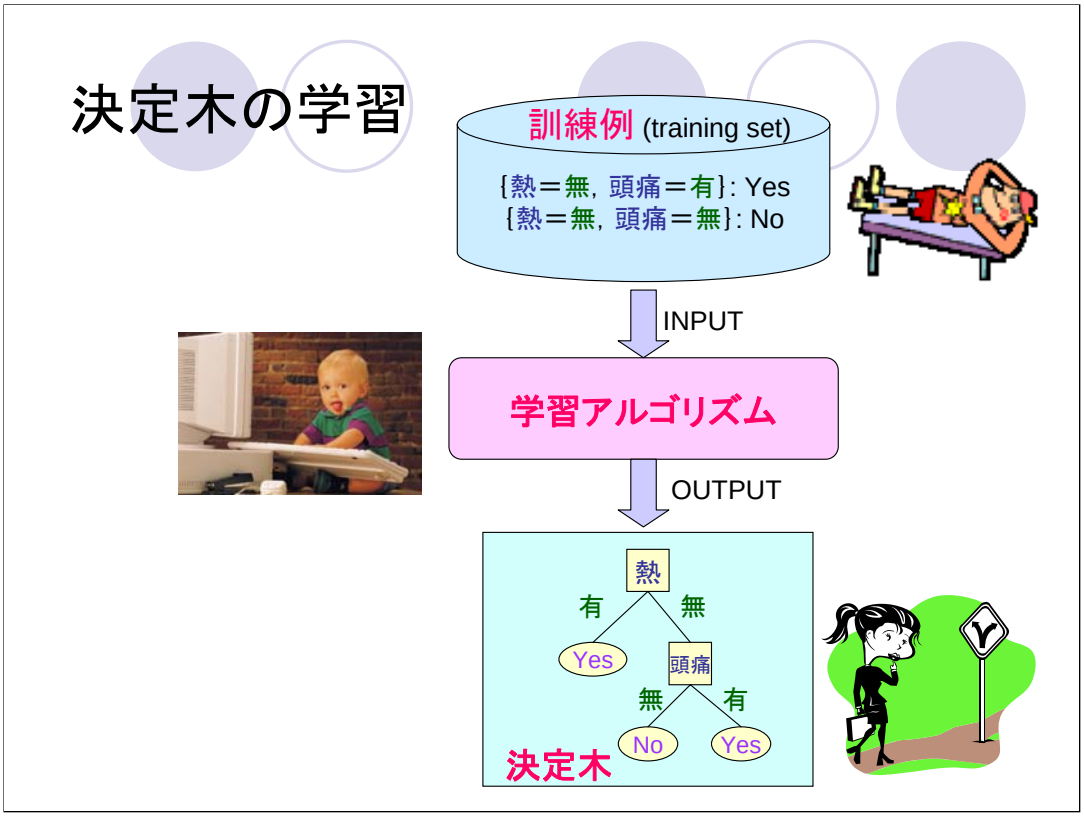
決定木の学習は，構造を学習するアルゴリズムのうちで最も簡単な部類のものではあるが，実用的に良く用いられて，うまくいっている方式の1つである．

決定木の目的は，**属性**とその**値**の組{属性1＝値1，…，属性n＝値n}によって表現されたデータをいくつかの**クラス**と呼ぶものに**分類**することである．たとえば，エージェントが見ている目の前の花を，{花びらの数＝5，花びらの色＝ピンク}などのデータを使って，桜や桃などのクラスのうちの1つに分類したい．また，健康コンサルティングをしているエージェントが，ユーザとの問診から得られた{熱＝有，頭痛＝有}のようなデータから，病院へ行くべき症状かどうかYes/Noに2分類して判断に使いたいというような応用を想定している．

決定木は，そのような判断を行うための木構造である．スライドの例は，病院へ行くべきかどうかの決定木で，熱が有るか，または，頭痛が有るときに，Yesの判断をするものである．

決定木の**非終端ノード**には属性のラベルが付けられ，そこから出ている枝にはその属性の取りうる値が付されている．**終端ノード**には最終的な分類が書かれている．ただし，この授業では，簡単のため，分類は二値（Yes/No，true/false，1/0）に限るとする．

決定木は，ある種の簡単なアルゴリズムを表しているともいえる．決定木にすべての属性値がそろった{属性1＝値1，…，属性n＝値n}の形式のデータを**質問**入力として渡すと，属性の値をテストしながら，木の根からつぎつぎと枝をたどり，最終的に到達した終端ノードの分類を判断結果として出力するからである．



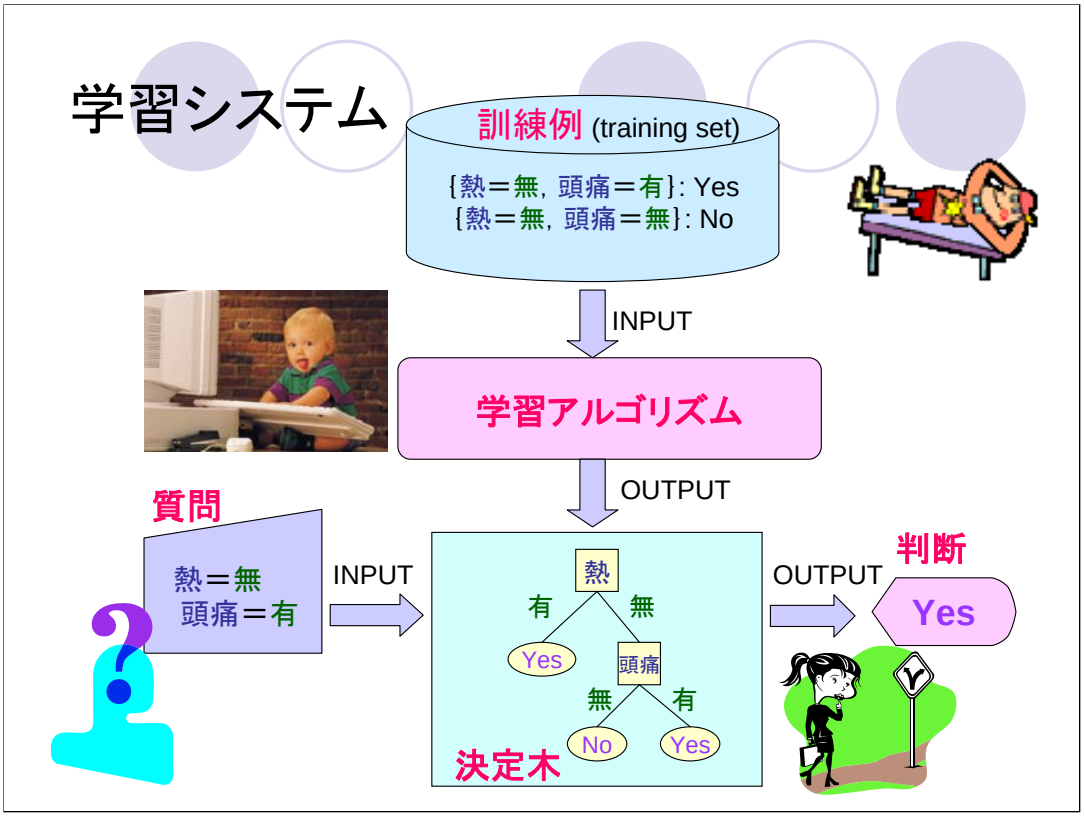
決定木の学習とは，具体的な判断事例から決定木を生成することである．

事例は

入力＝{属性1＝値1，…，属性n＝値n}:出力＝Yes/No

のような形式のデータで，入力の属性値が具体的にどのような値のときに出力がYesなのかNoなのかを指定する．

このようなデータの集まりを，学習アルゴリズムに対する**訓練例**という．学習アルゴリズムは，訓練例を入力として受け取ると，その訓練例をなるべく良く再現できるような決定木を出力するものである．



直前の2枚のスライドを組み合わせると，このスライドのような学習システムの全体図ができる．

システム設計者は，事前に学習アルゴリズムに訓練例を与えて，決定木を生成しておく．システムの運用時には，その決定木はエージェントによって保持されており，質問データ{属性1＝値1，…，属性n＝値n}が与えられると，エージェントはこの決定木を使用して分類結果を出力することができる．

訓練例(training set)

	性別	年齢	薬物	犯罪歴	実刑	
A	男性	未成年	覚醒剤	初犯	No	負例
B	女性	未成年	覚醒剤	初犯	No	
C	女性	老人	麻薬	初犯	No	
D	男性	成年	覚醒剤	初犯	Yes	正例
E	男性	老人	シンナー	累犯	Yes	
F	男性	老人	麻薬	初犯	Yes	
G	女性	成年	シンナー	累犯	Yes	
H	男性	未成年	シンナー	累犯	Yes	
I	男性	老人	麻薬	累犯	Yes	
J	女性	未成年	麻薬	累犯	Yes	

ここからは，学習アルゴリズムを具体的に説明するために，**法律に関する推論**の分野を簡単化した例題を，

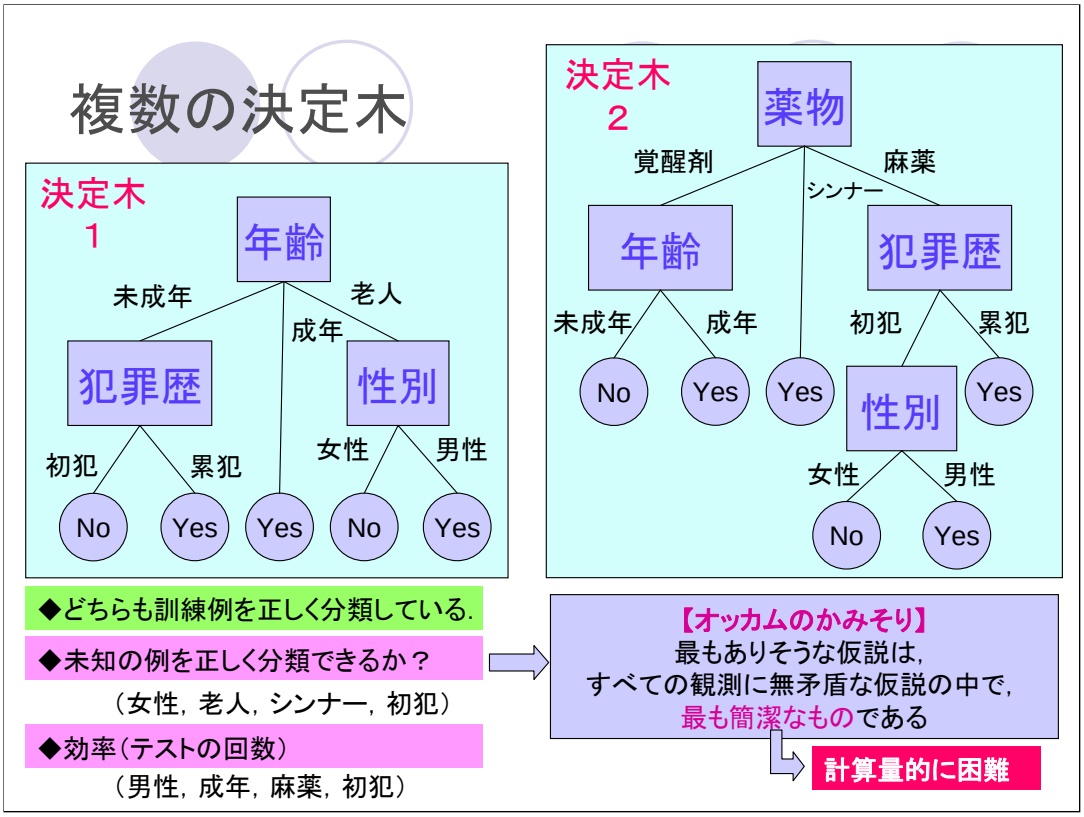
新田克己著，知識と推論，サイエンス社(2002)

の5.1節から引用する．

薬物に関する刑法事件の犯人の属性として，性別，年齢，薬物の種類，および犯罪歴が与えられたとして，判決（の予想）を実刑か執行猶予かに分類する決定木を考えよう．

このスライドの10の事例を訓練例として，いかにして学習アルゴリズムが決定木を生成するかを見ていく．

この訓練例には，出力がYesおよびNoである例がそれぞれ7個および3個含まれている．それらをそれぞれ，**正例**および**負例**という．



この例題の場合，属性をどのような順番でテストするかによって，生成される決定木が異なる．

このスライドにある2つの決定木は，いずれも訓練例をすべて再現できることを確認してほしい．

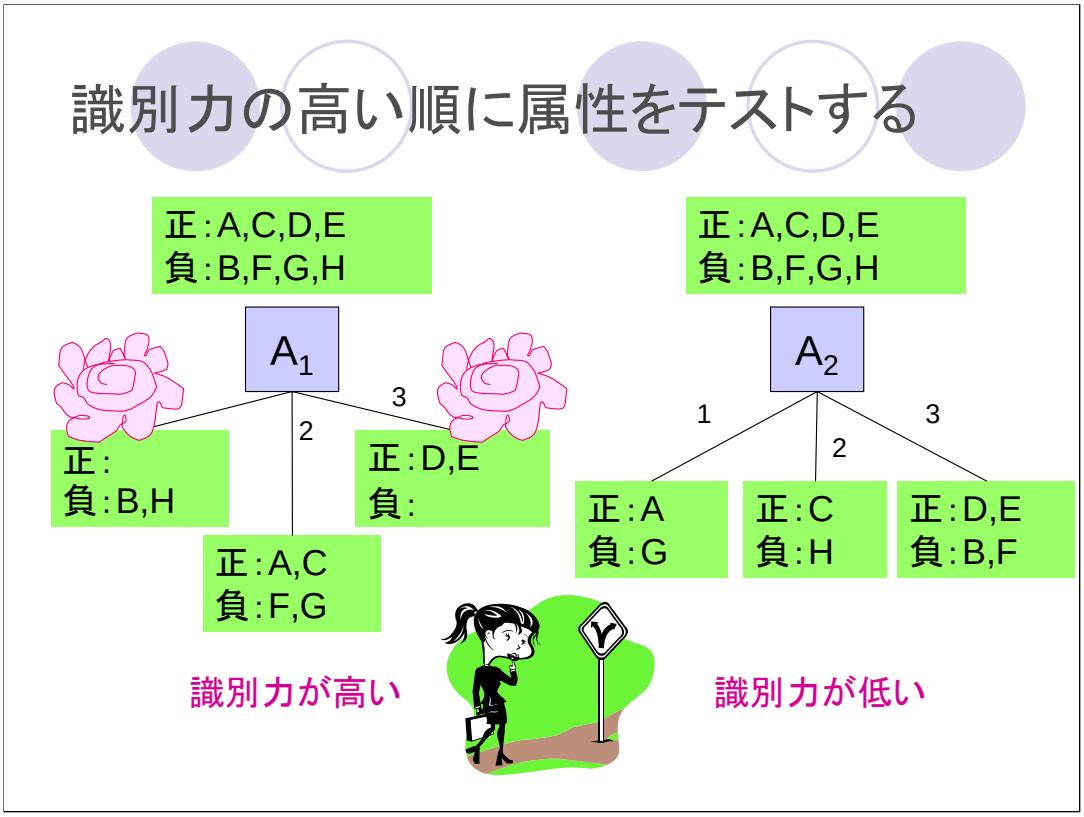
しかし，問題は訓練例に含まれていない未知の例を正しく分類できるかどうかである．たとえば，（女性，老人，シンナー，初犯）という質問に対して，決定木1ではNo，決定木2ではYesという出力が得られる．どちらが正しいかは，訓練例からはわからない．

出力が同じでも，効率が異なる場合がある．たとえば，（男性，成年，麻薬，初犯）という質問に対する出力は，どちらの決定木でもYesとなるが，決定木1では1回のテストで済んでいるのに対し，決定木2では3回のテストを必要としている．

このような場合の判断基準として，**オッカムのかみそり**と呼ばれる一般原則がある．それは，最もありそうな仮説は，すべての観測に無矛盾な仮説の中で最も簡潔なものであるという主張である．

決定木の場合，「簡潔」かどうかは，木のサイズ（ノード数）で表すことができる．したがって，すべての訓練例に無矛盾な決定木の中で最もサイズの小さなものを求めればよい．しかし，残念ながら，それは計算量的に困難であることが知られている．求めるのに膨大な時間がかかるのである．

次のスライド以降では，ある単純なヒューリスティックにより，最小ではなくても，かなり小さな決定木を得ることができる方法を調べる．



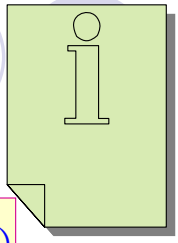
アイデアのポイントは，何らかの意味で「**識別力**」の高い属性を1つだけ選択し，それを先にテストすることに決め打ちしてしまうことである．バックトラック法などの探索アルゴリズムでそれ以外の選択肢を考慮してしまうと計算量が膨大になるので，その可能性は考えない．一手先の目先の評価しか考慮しないという意味で，このアルゴリズムは一般に，**グリーディ**（欲張り）という言葉で形容されるものとなっている．

たとえば，いま，4つの正例と4つの負例からなる訓練例を，属性A1と属性A2で識別すると，8個の事例が図のように分割されたとしよう．このとき，直観的に，A1の方が識別力が高い．なぜなら，A1=1のときはすべてが負例なので，そこでNoと決定でき，A1=3のときにはすべてが正例なので，そこでYesと決定できるからである．A2=2のときだけ，さらに他の属性を用いた識別が必要となる．

しかし，他方のA2による識別では，A2=1,2,3のいずれによっても，まだYes/Noが定まらず，さらに他の属性を必要とする．

このような直観を反映するように，適切に識別力を定義するにはどうしたらよいだろうか．

情報理論を利用する



平均情報量（エントロピー）

$$I(P_1,\ldots,P_n)=\sum_{i=1}^{n}P_i\log_2\frac{1}{P_i}=-\sum_{i=1}^{n}P_i\log_2 P_i \quad (\text{ビット})$$

例：コイン投げ

表 $P_1=1/2$

裏 $P_2=1/2$

$$I(\frac{1}{2},\frac{1}{2})=\frac{1}{2}\log_2 2+\frac{1}{2}\log_2 2=1$$

◆正例が p 個，負例が n 個のときの平均情報量

$$I(\frac{p}{p+n},\frac{n}{p+n})=\frac{p}{p+n}\log_2\frac{p+n}{p}+\frac{n}{p+n}\log_2\frac{p+n}{n}$$

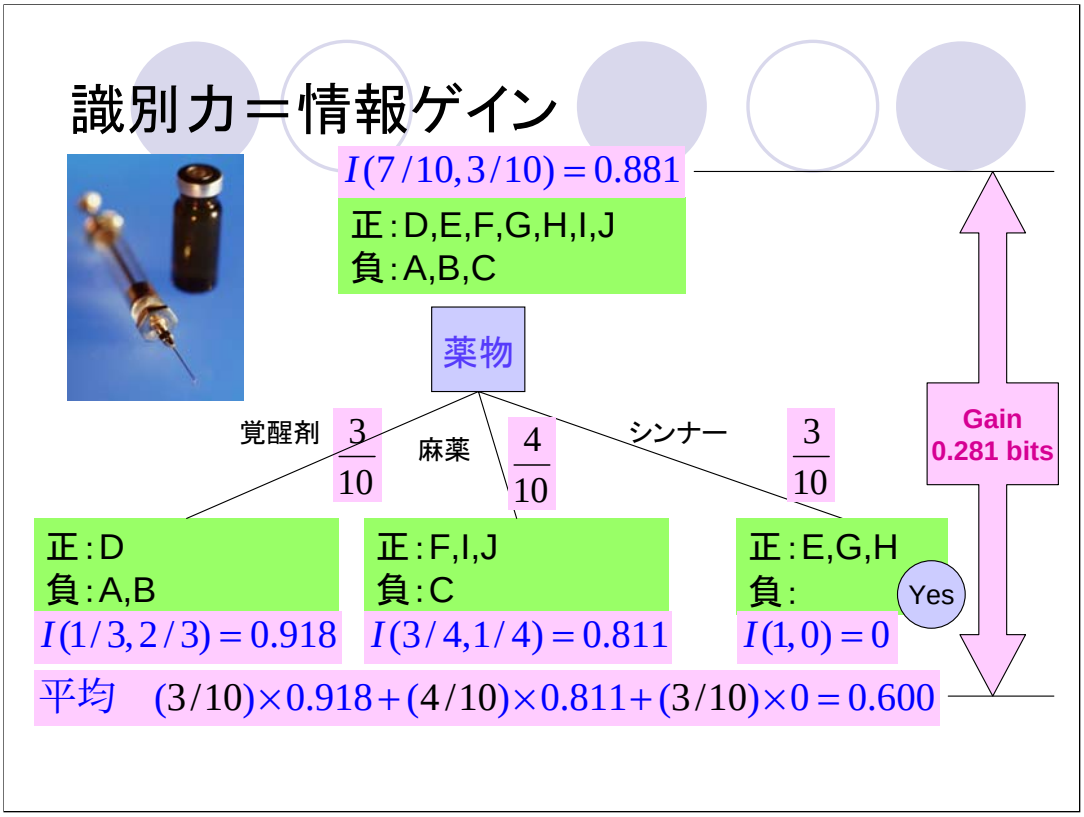
識別力を定義する際のポイントは**情報理論**にあるので，ここで少し復習しておく．

データがn個の**クラス**のどれか1つに分類できるとし，クラス i に属する確率を Pi と表す．

情報理論での定義によると，あるデータがクラス i に属することを知るには，最低でも log(1/Pi)ビットの**情報量**が必要である．（logの底は2である．）したがって，そのデータをクラス1～nのいずれかに分類するのに必要な情報量は，その期待値を計算して，スライドの式で求められる．これを**平均情報量**（**エントロピー**）といい，I(P1,...,Pn)で表す．

例として，コインの表裏の分類には，1ビットの平均情報量が必要となる．

決定木では，正例（Yes）と負例（No）の2クラスしか扱わないので，この特殊ケースはスライドの式のようになる．



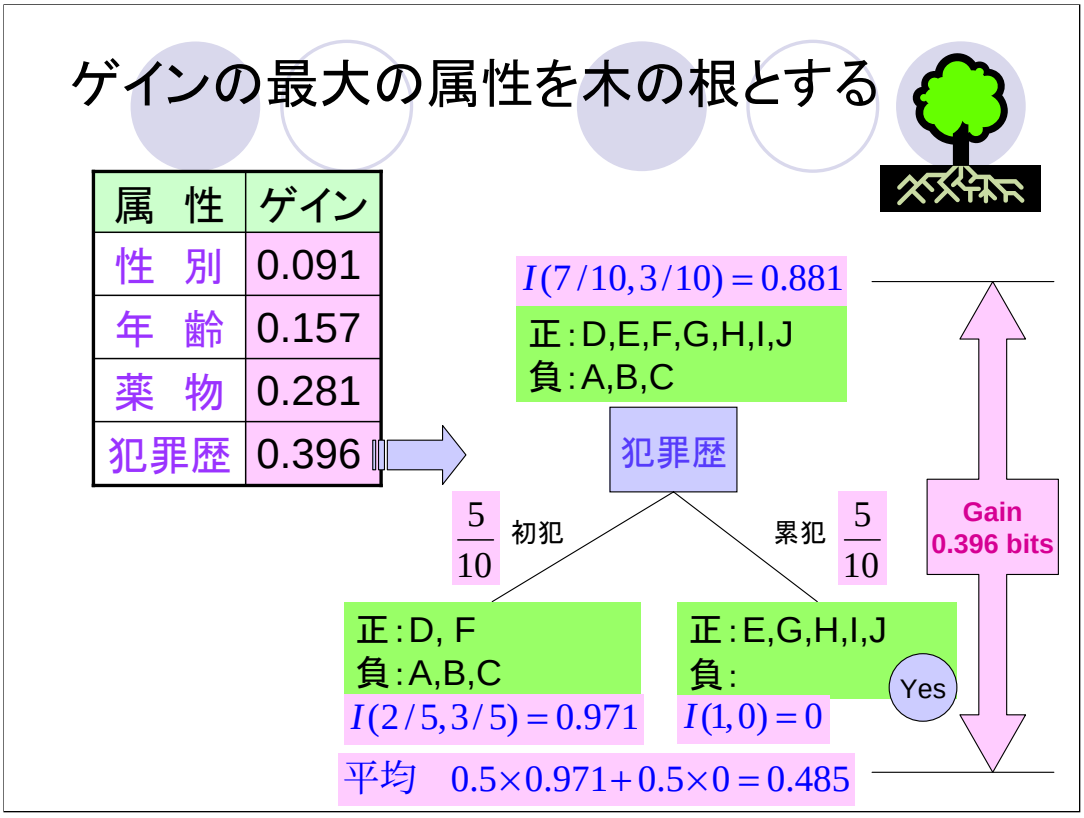
さて，もとに戻って，属性の識別力を情報のゲインとして定義することにしよう．つまり，その属性をテストすることによって得られる情報量を識別力と考えるのである．

法律の例に沿って，具体的に考えてみよう．前のスライドで見たように，訓練例の正例の数と負例の数から，平均情報量が求まる．その値は，今後最終的にYes/Noの区別を付けるために必要な情報量の期待値である．法律の例では，最初，訓練例は正例が7個，負例が3個なので，平均情報量＝0.881ビットとなっている．この全訓練例を表すノードを生成して，それを決定木の根ノードとする．

そこで，ある属性（この例ではまず「薬物」）をテストして，その結果に応じて，訓練例を3つに分割し，それに応じて，3つの子ノードを生成して，決定木の一部を作り出す．それぞれの平均情報量を求めると，それぞれ0.918, 0.811, 0 となる．

任意に質問が与えられたときには，訓練例は，この3分割のうちのいずれか1つに縮小される．その確率は，それぞれ，薬物＝覚醒剤，麻薬，シンナーである確率であるが，これを訓練例から推定すると 0.3, 0.4, 0.3 である．したがって，薬物テスト後の平均情報量は，スライドのような計算で，0.600ビットとなる．識別には，平均してあと0.6ビット必要ということだ．

したがって，識別に必要な平均情報量は，薬物テスト前の0.881からテスト後の0.600へ，0.281ビット減少したことがわかる．これが，属性「薬物」テストによって得られる情報の獲得量（**ゲイン**）である．そして，それが「薬物」属性の「識別力」であると考えるのである．



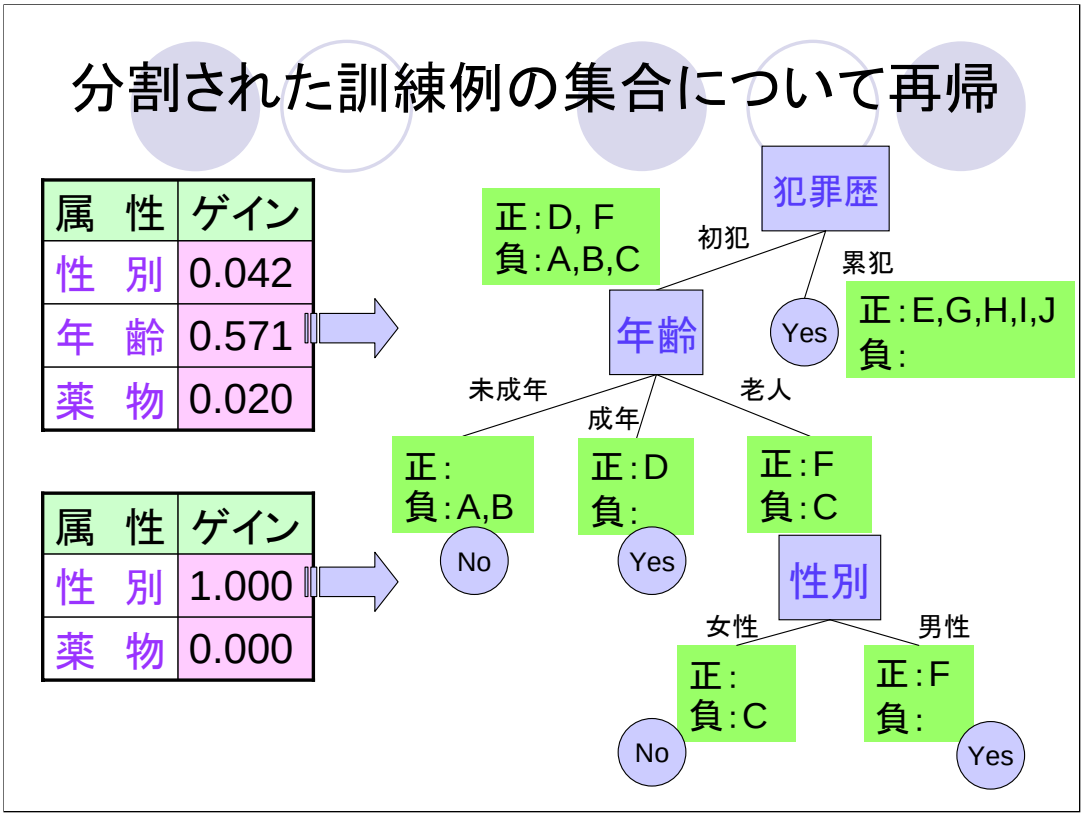
同様に，薬物以外の3つの属性について，ゲインを計算すると，スライドの表のようになる．

そこで，これらの中で，最大のゲインが得られる属性を決定木の根として採用する．その結果は「犯罪歴」を根とする図のような部分的な決定木となる．

犯罪歴によって，訓練例が2つに分割されている．そのそれぞれについて，これまでの考え方を再帰的に適用して，ここから下の部分の決定木を生成していけばよい．もちろん，「犯罪歴」はもう考える必要がないので，他の3つの属性のそれぞれについて，またゲインを計算するのである．

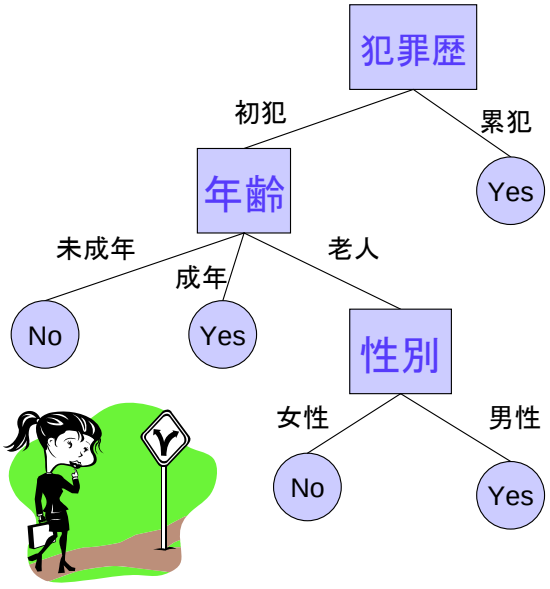
実際には，犯罪歴＝累犯のケースは，正例しか残らないので，ここをYesノードとし，それ以上のテストを行わない．

したがって，犯罪歴＝初版のケースだけ，再帰的に処理を続行する．



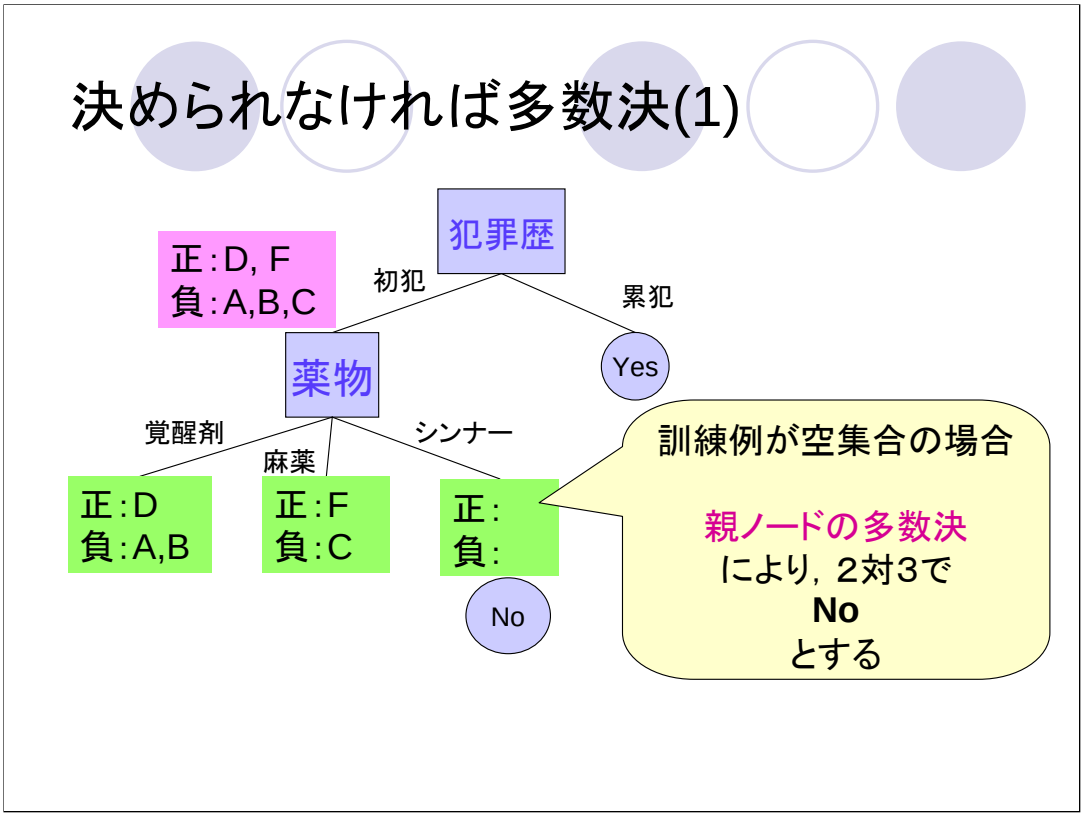
その結果がこの図である.

学習結果の決定木



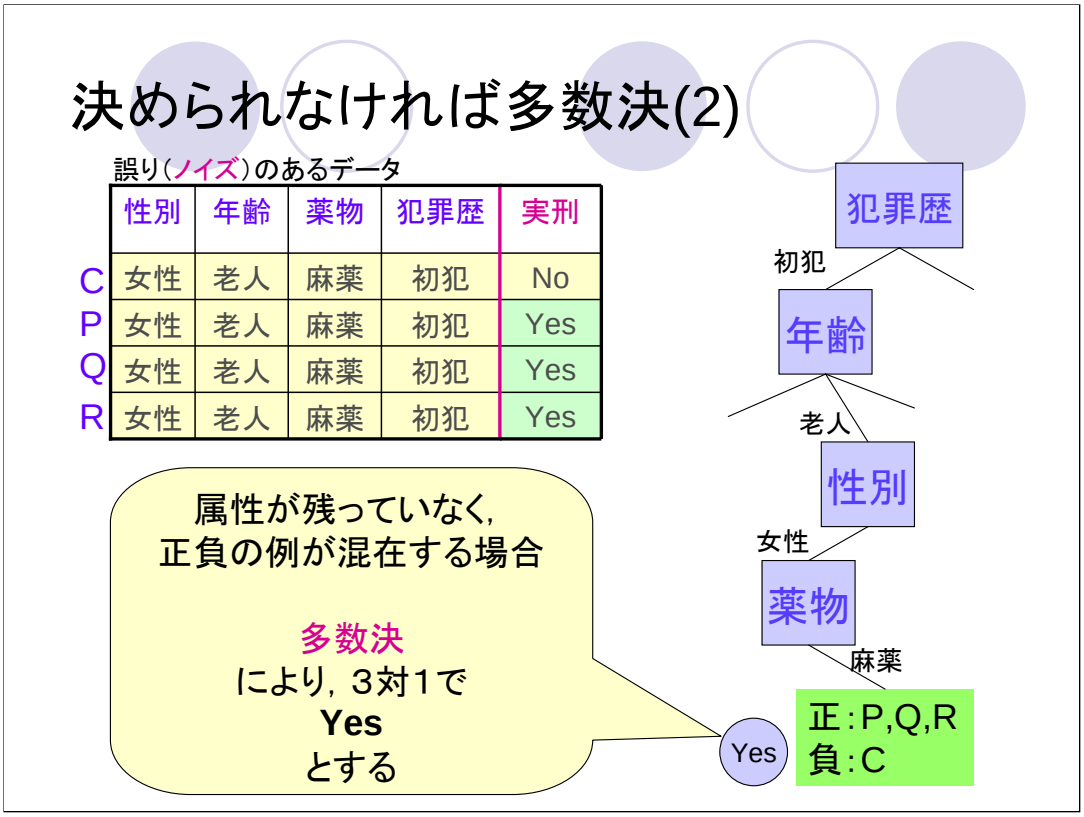
これが得られる決定木である.

休憩

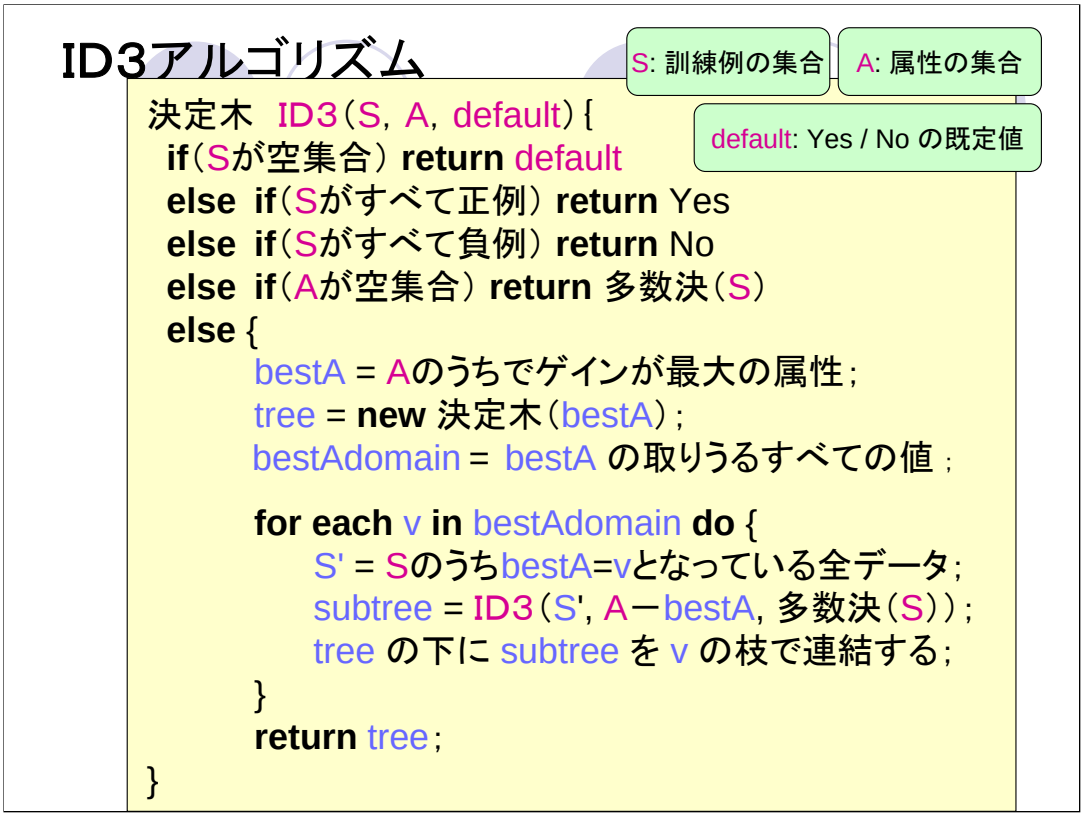


いま述べた処理で，例外的なケースが2つある．

1つは，このスライドの「薬物＝シンナー」のテスト結果のように，訓練例の分割の結果，空集合となった場合である．このときには，空集合からは何も判定できないので，その親ノード（この例では「薬物」）における訓練例を用いた**多数決**でYesまたはNoに決める．この例では，正例2：負例3の多数決の結果，負例が多いので，Noとする．



もう1つの例外的なケースは，このスライドのように，訓練例に誤りがある場合である．この例では，(女性，老人，麻薬，初犯)のすべての属性を調べても，実刑かどうかは訓練例からは確定できない．これは，誤りというより，調べるべき属性が足りないことによる情報不足で，**ノイズ**とも呼ばれている．このように，すべての属性をテストしても正負の例が混在する場合には，残っている訓練例の多数決でYesかNoに決める．

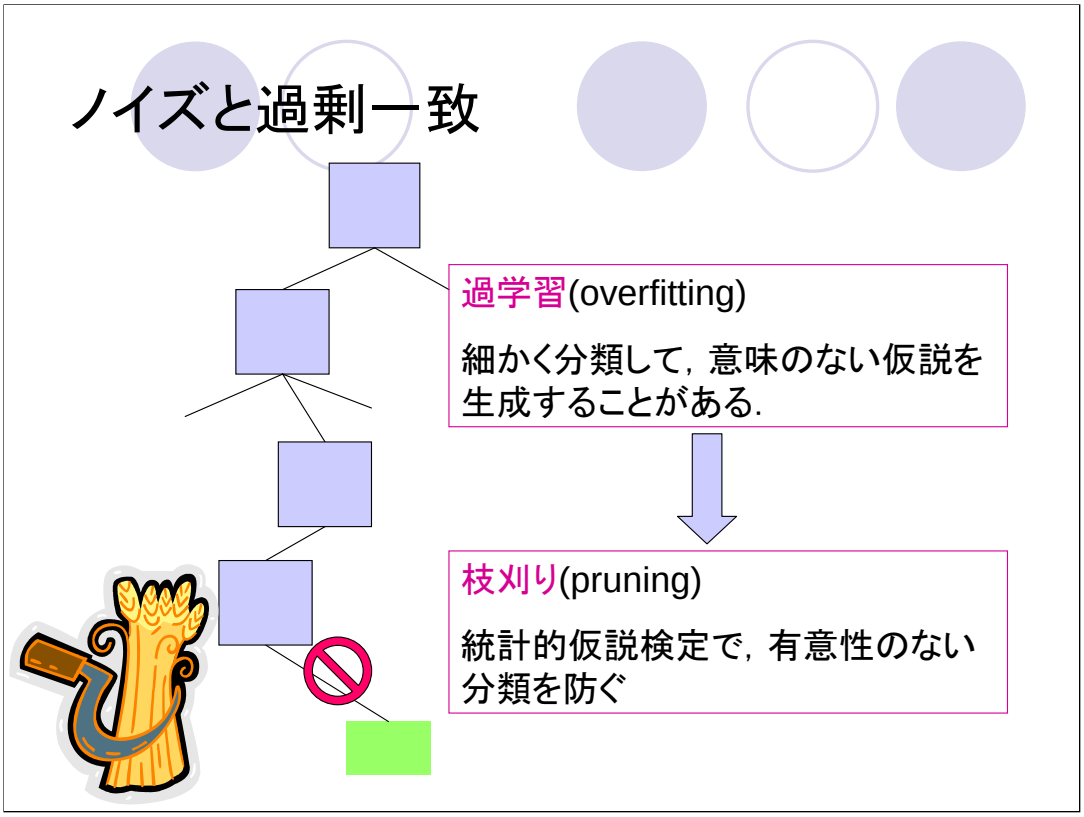


これまでの考え方を総合的にまとめたのが，このスライドの **ID3** と呼ばれるアルゴリズムである．

このアルゴリズムは，入力として，訓練例の集合S，属性の集合A，およびYes/Noのデフォルト値を受け取り，それを説明するための決定木を生成して出力する．

平均情報量のゲインが最大の属性 bestA を選び，その属性値 v ごとに，ID3を再帰的に呼出しながら決定木を成長させている．

再帰呼出しの引数は，Sのかわりに bestA=v を満たす訓練例の部分集合S' に縮小される．属性の集合Aからは bestA が削除される．また，デフォルト値は，親ノードSに残っている訓練例の多数決により決定される YesまたはNoである．



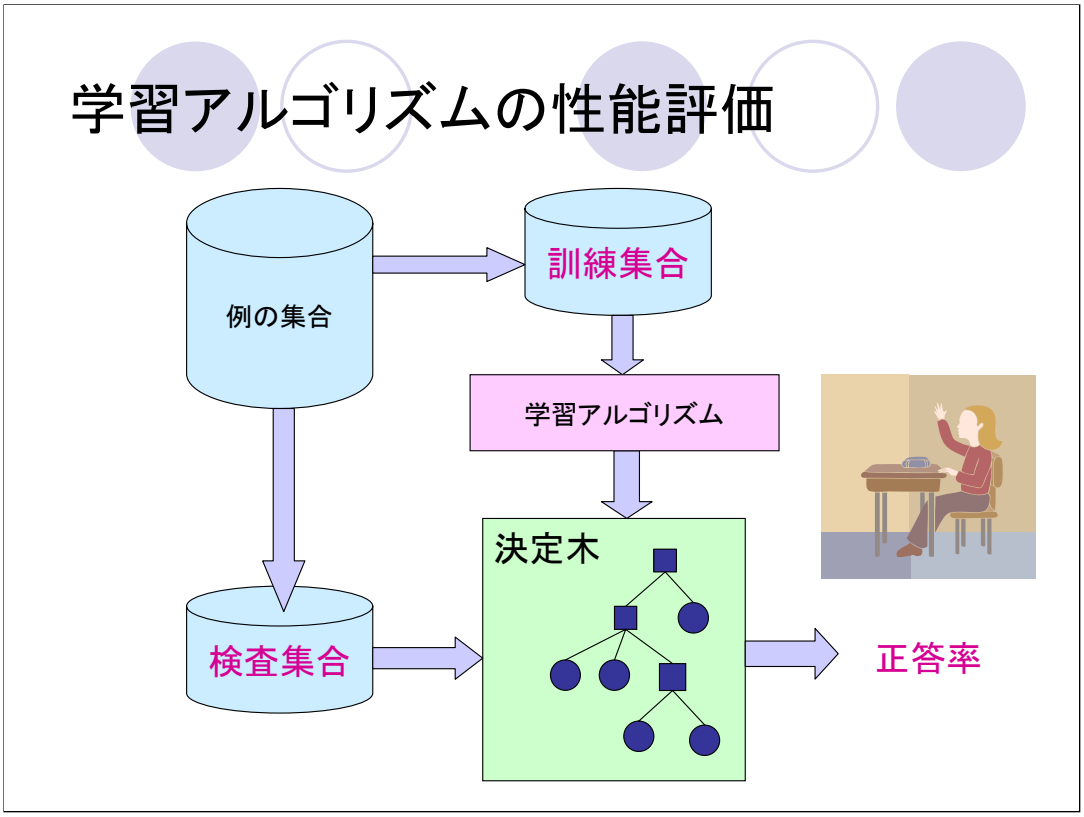
このような学習アルゴリズム全般に言える注意事項が，ノイズと過剰一致についてである．

すでに見たように，一般に，訓練例には**ノイズ**と呼ばれる誤りの一種が含まれている．ノイズまでも再現するような細かな分類は，不適切な決定木を生み出すことが多い．これを**過学習**という．

このような問題を避けるための基本的な考え方は，平均情報量のゲインが十分小さいときには，もうそれ以上の分類をやめて，そこを終端ノードとすることである．これを，決定木の**枝刈り**という．

では，ゲインがどの程度小さければ枝刈りをすればよいだろうか？

この問題は，**統計的仮説検定**の考え方によって解決できる．詳細は省略するが，危険率5%などの設定のもとで，統計的に有意性が認められない分類をしている枝を刈るのである．

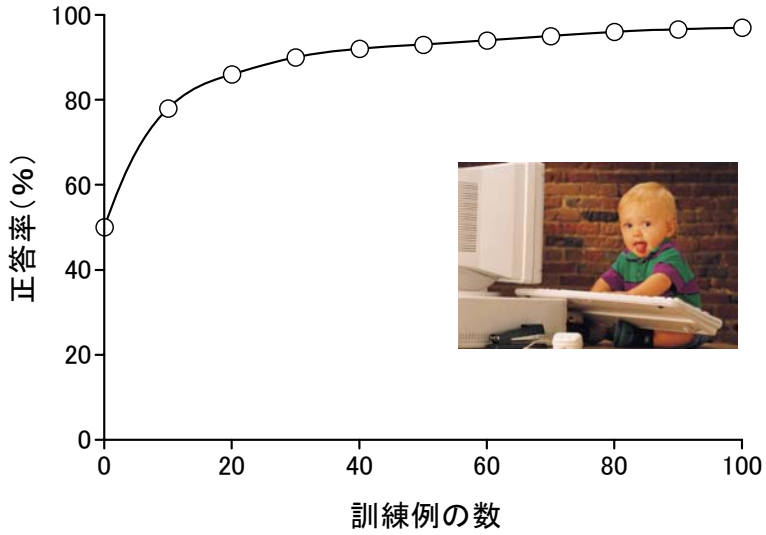


ここで，学習アルゴリズムの性能評価の方法を知っておこう．これは決定木の学習に限らず，どのような学習アルゴリズムの評価にも通じる考え方である．それは，つぎのステップからなる．

1．例をたくさん集める．
2．それを**訓練集合**と**検査集合**の互いに素な集合に分ける．
3．訓練集合を学習アルゴリズムに入力して決定木を生成する．
4．検査集合を決定木に入力して，**正答率**を求める．

訓練集合と検査集合に共通部分がないように分離している点が重要である．分離しなければ，学校で前もって練習していた数学の例題が，本番のテストでそのまま出題されるようなもので，正しい実力を評価しているとはいえない．

学習曲線(learning curve)



いま述べた評価ステップ2～4は，訓練集合のサイズを変えて，各サイズごとに訓練集合をランダムに何通りか選んで実施する．

その結果をグラフにプロットすると，典型的には，このスライドのようなものが得られる．これは，訓練例のサイズを大きくするにしたがって，正答率が増加している様子を示しており，このアルゴリズムが確かに「学習」していることの間接的な証拠となる．このような曲線を**学習曲線**という．

決定木学習の現実的応用(1)

- 油田基地の機器設計（英国石油，1986）
  - 海上油田基地のガス-石油分離システムを設計するエキスパートシステム
  - ガス・石油・水の混合比，流速，圧力，密度，...
  - 人手の設計：10人年→100人日
  - 性能は専門家をしのぐ
  - 100万ドル節約

決定木の学習が現実の世界で応用された事例を2つ紹介する．

これは英国石油が，ガスと石油と水を分離するシステムを設計する**エキスパートシステム**で用いられた応用例である．ここでは，過去の設計のデータベースから得られる多くの事例に基づいて，混合比，流速，圧力，密度などの属性に応じて適切に種々のパラメータを決定するための決定木を学習させ，そこからエキスパートシステムで用いられるルールを抽出した．

決定木学習の現実的応用(2)

- 飛行学習（1992）
  - フライトシミュレータでセスナ機の操縦を学習
  - 3人の熟練パイロットの30回の繰り返しから学習
  - 性能は先生をしのぐ

これはセスナ機の**自動操縦**システムを設計するために決定木学習を応用したものである．セスナ機がどういう状態のときには，どう運転したらよいかを，3人の熟練パイロットの30回の繰り返しのデータから学習させた．

その結果は驚くべきものである．学習アルゴリズムは訓練例のノイズを除去することができるため，得られた決定木は人間パイロットがときどき犯す誤りを取り除いたものとなり，結果として，先生である熟練パイロットよりも上手に飛べるようになったのである．めでたし，めでたし．