symbolic machine learning

learning = improving with experience at some task

- relevant disciplines
  - AI, psychology (incl animal learning theory), neurobiology, biology, statistics, information theory, control theory, Bayesian methods, complexity theory, philosophy (inductive logic, confirmation theory) .....
- well posed learning problem
  - task T
  - performance measure P
  - (source of) experience E

    eg: **learn to play checkers**

    T: play checkers

    P: % of games won in world tournament

    E: opportunity to play against self

why ml?

- new capabilities for computers
  - data mining
    - medical records --> medical knowledge
    - credit card transactions --> fraud detection
  - self-customizing programs
    - learning newsreader...
  - things we can't program well / at all, i.e. only **learning** programs perform well
    - speech recognition
    - transmembrane protein identification
    - autonomous driving (ALVINN: 70m/h speed for 90m on public highways!)
    - adaptive and alife algorithms
- understand human learning / teaching
- understand adaptiveness

learning checkers

what experience?

what exactly should be learned?

how to represent it?

what algorithm?

unsupervised or not?

intermittent or immediate feedback?

what target function?

eg: **ChooseMove: Board --> Move** or

V: Board --> *Reals*

b is final won board state --> V(b) = 100

b is final lost board state --> V(b) = -100

b is final drawn board state --> V(b) = 0

b is not final --> V(b) = V(b'), where b' is best final state achievable from b, playing optimally till end of game

[not operational.... complete search ahead!]

how to represent target function? (NN, rules, polynomial)

learning checkers, cont.

- non-operational target function

  V: Board --> *Reals*
- V is approximated by **learned function V'**
  - V'(b) = w0+w1*bp(b)+w2*rp(b)+w3*bk(b) +w4*rk(b)+w5*bt(b)-w6*rt(b) where bp, rp = # black, red pieces; bk, rk = # black, red kings; bt, rt = # red (black) pieces threatened by black (red).
  - learning task now reduced to learning w0,w1...
- training examples
  - must assign specific values to board states, including intermediate states
  - {< b, V_{train}(b) >}

    eg: <<bp=3, rp=0, bk=1, rk=bt=rt=0>, +100> i..e black has won…

    training value V_{train}(b) for intermediate b can be assessed in several ways, eg.

  $\mathbf{V_{train}(b) \leftarrow V'(successor(b))}$

    where V' is program's current approximation and successor(b) is board state after program's move and opponent's response (when it is again the program's turn to move). We use estimates of value of successor(b) to estimate value of b!!!

Least Mean Squared Error

- which weights w_i best fit the examples?
- best set of weights minimizes squared error between training examples and values predicted by V'

$$E = \Sigma_{\text{training examples}} (V_{train}(b) - V'(b))^2$$

- E-minimizing weights ≈ most probable hypothesis ({weights}) given the examples
- E-minimizing, incremental weight refinement: LMS weight tuning rule

repeat

1) select training example <b, $V_{train}(b)$> at random

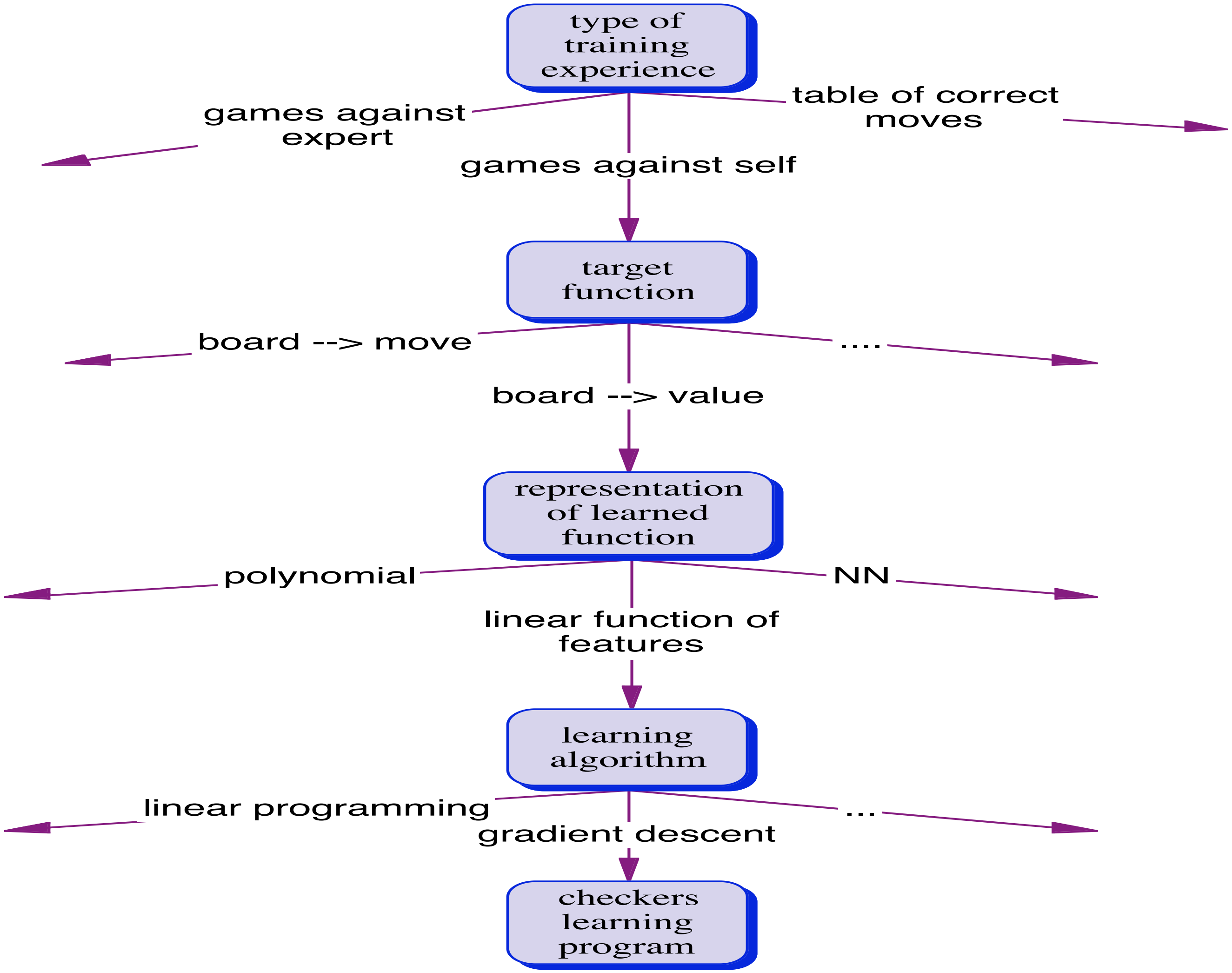
2) $error(b) \leftarrow V_{train}(b) - V'(b)$

3) for each board feature f_i, update w_i:

$w_i \leftarrow w_i + c \bullet f_i \bullet error(b)$

(c is small constant, eg 0.1, to moderate learning rate)

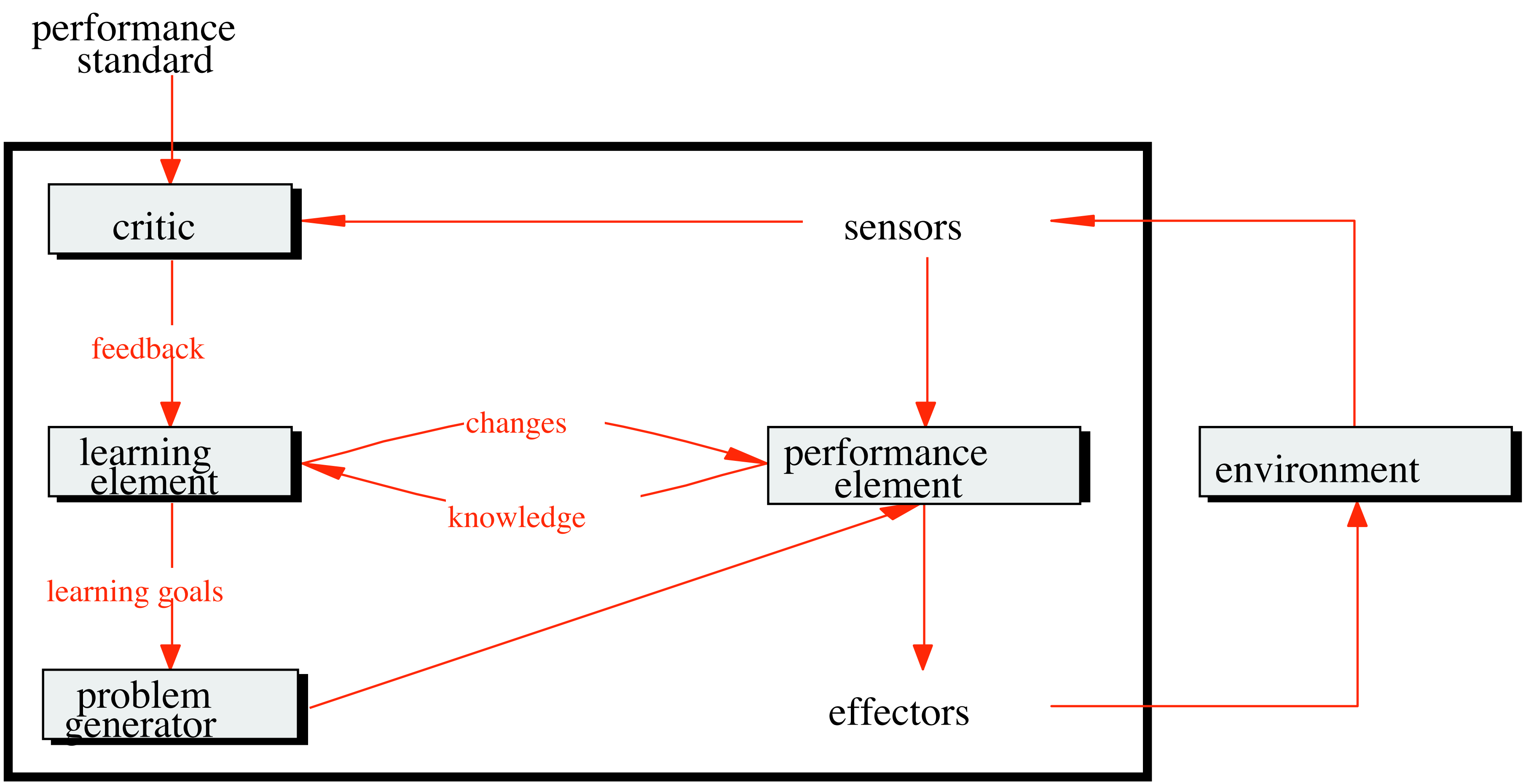
design choices



ml issues

- which algorithms for learning general functions from specific examples are best for which problems / representations?
- for a given hypothesis space, how much training data is sufficient?
- how does (approx. correct) prior knowledge help in guiding generalization? (**inductive bias**)
- what are useful strategies for choosing helpful next training examples?
- **can the reduction of learning tasks to function approximations be automated?**
- **can the learner automatically change its representation?**
- **∃ completely unsupervised learning?**

symbolic machine learning



design issues for learning elements

- which components of pe to improve?
  - direct mapping from conditions on current state to actions
  - means to infer properties of the world from percept sequence
  - information about results of possible actions
  - utility information about desirability of states
  - information about desirability of particular actions in particular states
  - goals (classes of states) whose achievement maximizes agent's utility
- available feedback
  - correct outputs provided: **supervised learning**
  - actions are evaluated: **reinforcement learning**
  - no hints about correct outputs: **unsupervised**
- prior knowledge helps
  - to learn something, it helps if you already know a lot....

types of learning

- classification
  - supervised, preclassified
- association
  - look for any 'interesting' association of features, not just those that predict a *class* value
  - since a rule can 'predict' several attribute values, there are **many** rules; impose **accuracy** and **coverage** restrictions

    coverage = **support** and accuracy = **confidence**
- clustering
  - find groups of examples that 'belong together'
  - e.g. iris data without iris types
  - given clusters/classes, use classification to find simple rules for categorization
- numeric prediction
- instance-based learning … … …
- learning by analogy …

contact lenses data

- **contact-lenses**
  - attributes

    age {young, pre-presbyopic, presbyopic}

    spectacle-prescrip {myope, hypermetrope}

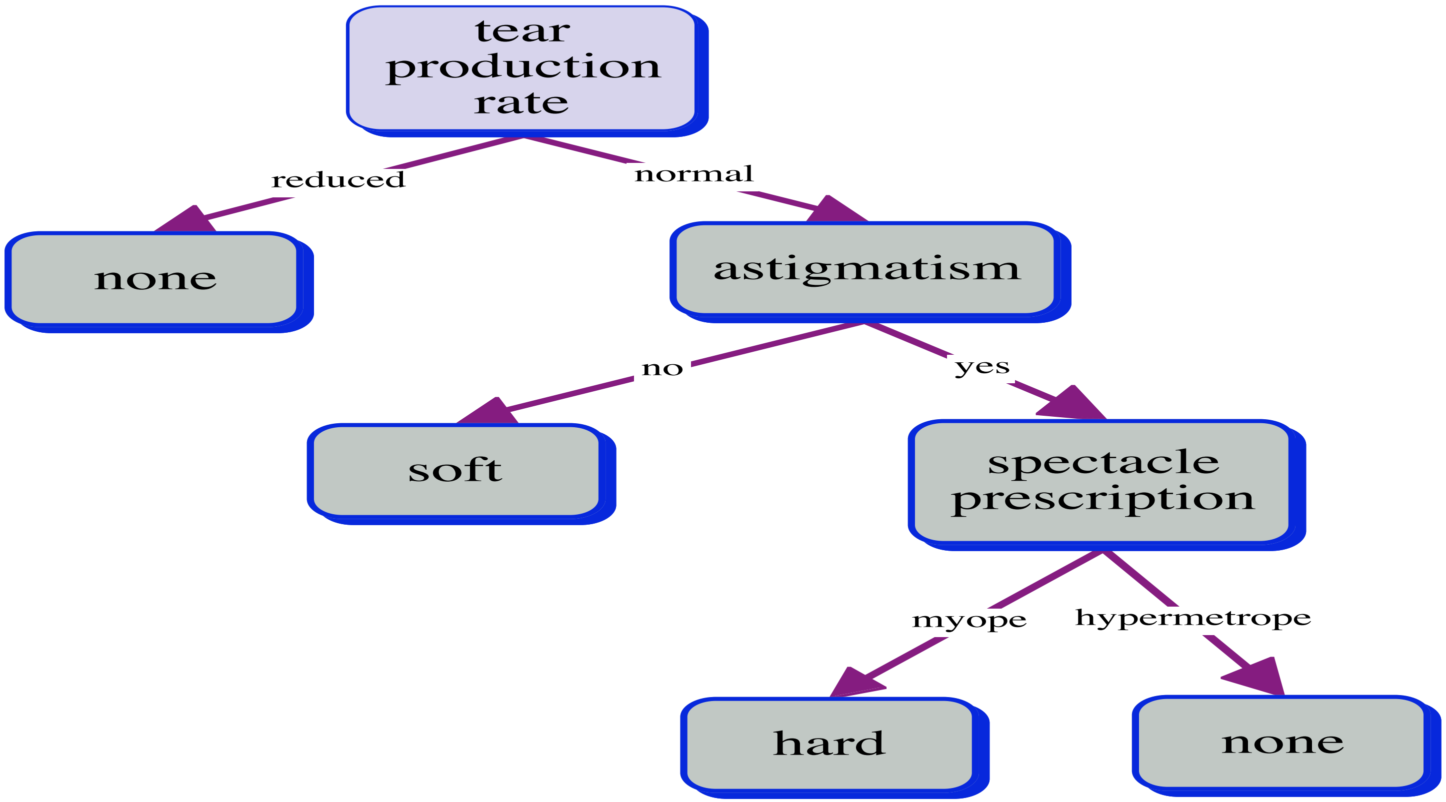
    astigmatism {no, yes}

    tear-prod-rate {reduced, normal}

    contact-lenses {soft, hard, none}
  - data

    young,myope,no,reduced,none
    young,myope,no,normal,soft
    young,myope,yes,reduced,none
    young,myope,yes,normal,hard
    young,hypermetrope,no,reduced,none
    young,hypermetrope,no,normal,soft
    young,hypermetrope,yes,reduced,none
    young,hypermetrope,yes,normal,hard
    pre-presbyopic,myope,no,reduced,none
    pre-presbyopic,myope,no,normal,soft
    pre-presbyopic,myope,yes,reduced,none
    pre-presbyopic,myope,yes,normal,hard
    pre-presbyopic,hypermetrope,no,reduced,none … etc

contact lenses, rules



iris data

- **iris** (famous numeric dataset due to R.A. Fisher)
  - attributes

    sepallength REAL
    sepalwidth REAL
    petallength REAL
    petalwidth REAL
    class {Iris-setosa,Iris-versicolor,Iris-virginica}

  - data

    5.1,3.5,1.4,0.2,Iris-setosa
    4.7,3.2,1.3,0.2,Iris-setosa ... etc
    7.0,3.2,4.7,1.4,Iris-versicolor
    6.4,3.2,4.5,1.5,Iris-versicolor ... etc
    6.3,3.3,6.0,2.5,Iris-virginica
    5.8,2.7,5.1,1.9,Iris-virginica … etc

labor negotiation

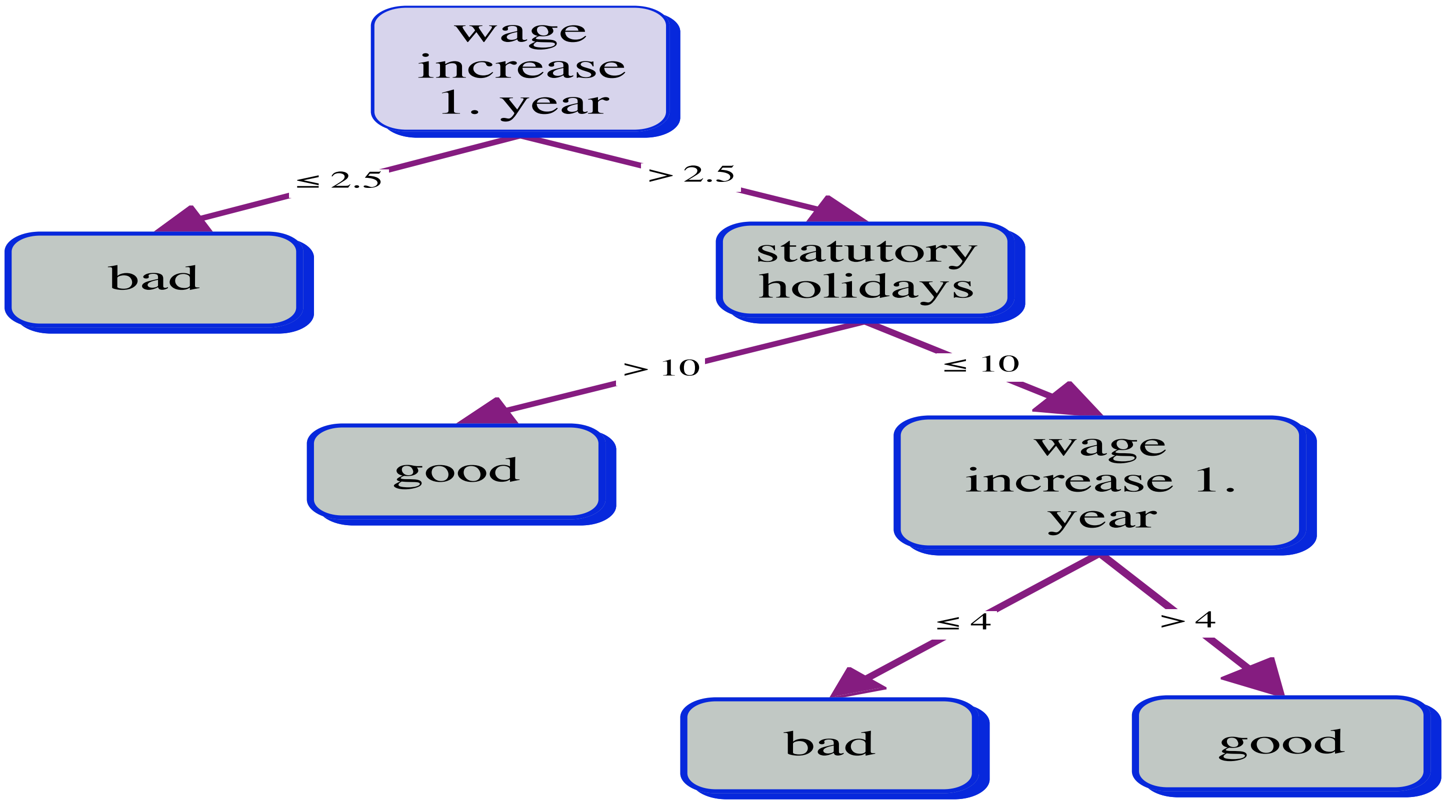
- contract deemed 'good' when acceptable by both labor and management
  - **attributes**
  - 'duration' real
  - 'wage-increase-first-year' real
  - 'wage-increase-second-year' real
  - 'wage-increase-third-year' real
  - 'cost-of-living-adjustment' {'none','tcf','tc'}
  - 'working-hours' real
  - 'pension' {'none','ret_allw','empl_contr'}
  - 'standby-pay' real
  - 'shift-differential' real
  - 'education-allowance' {'yes','no'}
  - 'statutory-holidays' real
  - 'vacation' {'below_average','average','generous'}
  - 'longterm-disability-assistance' {'yes','no'}
  - 'contribution-to-dental-plan' {'none','half','full'}
  - 'bereavement-assistance' {'yes','no'}
  - 'contribution-to-health-plan' {'none','half','full'}
  - 'class' {'bad','good'}

labor negotiation, data

- 1,5,?,?,?,40,?,?,2,?,11,'average',?,
  ?,'yes',?,'good'
- 2,4.5,5.8,?,?,35,'ret_allw',?,?,
  'yes',11,'below_average',?,'full',?,
  'full','good'
- ?,?,?,?,?,38,'empl_contr',?,5,?,11,
  'generous','yes','half','yes','half',
  'good'
  etc etc
- 1,2.1,?,?,'tc',40,'ret_allw',2,3,
  'no',9,'below_average','yes','half',
  ?,'none','bad'
- 1,2,?,?,'none',38,'none',?,?,'yes',
  11,'average','no','none','no','none',
  'bad'
- 2,2.5,3,?,?,40,'none',?,?,?,11,
  'below_average',?,?,?,?,'bad'etc etc

note missing data; numeric and nonnumeric data mixed

labor negotiation, rules



simplified / pruned to avoid overfitting!

the ubiquitous weather data, **nominal**

attributes

outlook {sunny, overcast, rainy}
temperature {hot, mild, cool}
humidity {high, normal}
windy {TRUE, FALSE}
play {yes, no}

data

sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no

when should I play?