Features and Constraints

May 20, 2014 2:32 PM

Search Strategies

A*
Backtracking
Local Search

Features

Domain

e.g.
$\text{dom}(x_1) = \{a, b, c\}$
$x_1 \leftarrow a$

Boolean Satisfiability (Example)

Variables

$A, B, \ldots, G$

Domains

$\text{dom}(A) = \{\text{true}, \text{false}\}$

Constraints

$(\neg A \vee \neg B) \wedge (\neg B \vee \neg C \vee D)$

Constraint Terminology and Notation

Intentional Constraint Description

Formula to be satisfied

Extensional Constraint Description

A list of valid tuples

Tuples

$t = (1,4)$ with variables x_1, x_2
$t[x_1] = 1$
$t[x_2] = 4$

Vars

C is a constraint
$\text{vars}(C)$ is variables in constraint C

Example: n-Queens

4-Queens as a constraint satisfaction problem (CSP)

Variables

Each grid location, x_{ij}, $i = 1, \ldots, 4$, $j = 1, \ldots, 4$

Domains

$\text{dom}(x_{ij}) = \{0,1\}$
$x_{ij} \leftarrow 1:$ There is a queen on (i, j)

Constraints

$$\forall i \sum_{j=1}^{4} x_{ij} = 1$$

$$\forall j \sum_{i=1}^{4} x_{ij} = 1$$

And each diagonal $\sum x_{ij} \leq 1$

Alternate Formulation

Variables

x_i, $i = 1, \ldots 4$ (one for each column)

Domains

$\text{dom}(x_i) = \{1, 2, 3, 4\}$ (row positions)
$x_i \leftarrow j$ There is a queen in column i, row j

Constraints

$\forall i \forall j \;\; x_i \neq x_j \wedge |x_i - x_j| \neq |i - j|$

Example: Crossword Puzzle

Variables

$x_1, \ldots, x_{23}$

Domains

$\text{dom}(x_i) = \{'a', 'b', 'c', \ldots, 'd'\}$

Constraints

Consecutive grid locations form words in dictionary. All words used exactly once.
Non Binary

Alternate Formulation

Variables

1Across, 1Down, 2Down, ...

Domains

$\text{dom}(\text{1Across}) = \{\text{All 5 letter words}\}$

Constraints (Binary)

1Across and 1Down agree on assignment of the first letter.
... Same for all pairs of intersecting rows/columns.

Alldifferent constraint - 4 Queens Example

Variables

x_1, x_2, x_3, x_4 (each column)

Constraints

$\text{alldifferent}(x_1, x_2, x_3, x_4)$
...

Constraint Propagation

May 22, 2014 2:33 PM

Example

Variables

x, y, z

Domains

$\{1,2,3\}$

Constraints

$c_1: \; x < y$

$c_2: \; y < z$

Check Arc Consistency and Remove Inconsistent Values

x and c_1

$\mathrm{dom}(x) = \{1,2,3\} \Rightarrow \{1,2\}$

y and c_1

$\mathrm{dom}(y) = \{1,2,3\} \Rightarrow \{2,3\}$

y and c_2

$\mathrm{dom}(y) = \{2,3\} \Rightarrow \{2\}$

z and c_2

$\mathrm{dom}(z) = \{1,2,3\} \Rightarrow \{3\}$

x and c_1

$\mathrm{dom}(x) = \{1,2\} \Rightarrow \{1\}$

n variables
m values for each
k constraints per variable

$\frac{nk}{2}$ constraints total

Constraints always satisfied.
Number of constraint checks:

Naïve backtracking

nk checks

MAC

$$\frac{nk}{2}m + nkm$$

Comparison of naïve backtracking vs MAC

n variables.
Each variable has domain size m
There is a single binary constraint for each pair of variables.

$\frac{n(n-1)}{2}$ constraints total.

Each constraint is satisfied with probability p

What is the branching factor of naïve backtracking?

Consider a node with l free variables.
This node is satisfiable if there is any assignment of the l variables that satisfy all $C = \frac{l(l-1)}{2} + l(k-l)$ constraints.
An assignment is satisfactory with probability p^C
None of the m^l assignments are satisfactory with probablity

$$(1-p^C)^{m^l}$$

So a node at height l is satisfiable with probablity

$$q(l) = 1 - (1-p^C)^{m^l}$$

Given that a node is satisfiable with probability $q(l)$, what is the expected branching factor?
Probability that a node at height l has branching fractor b is

$$P(B_l = b) = \left(1 - q(l)\right)^{b-1} \times q(l)$$

This is a geometric distribution with expected value

$$E(B_l) = \frac{1}{q(l)}$$

What is the expected number of nodes in the search tree starting at height l?

N_l is the expected number of nodes. B_l is the random variable giving the branching factor of nodes at neight l.
Recurrence:

$$N_l = \sum_{b=0}^{m} P(B_l = b)\, N_{l-1}$$

Local Search

May 27, 2014 2:33 PM

Example: 4-Queens

Alternative 1

All constraints into cost function

+1 for each constraint that isn't satisfied

Set of all states: CSP with no constraints.

Set of all solutions: set of all 4-tuples over the set $\{1, 2, 3, 4\}$

Neighbourhood function:

Swap pairs of values? Solution isn't necessarily reachable. e.g. from $(1, 1, 1, 1)$

Alternative 2

Cost function: +1 for each of $|x_i - x_j| \neq |i - j|$ that is not satisfied

Set of all states must satisfy $x_i \neq x_j \; \forall i, j$

More specifically, alldifferent(x_1, x_2, x_3, x_4)

Set of all states: : all permutations of 1, 2, 3, 4.

Neighbourhood function: Swap pairs of values.

Local Search for TSP

Nodes: Permutations of Cities

Cost function: cost of tour

Neighbourhood function:

2-opt: delete two edges from tour to break tour into two pieces and then reconnect

Starting Tour

- Greedy Algorithm
  - Pick lowest cost one next
- Alternative: randomly pick a starting node, run greedy algorithm
- Alternative: pick randomly from lowest few in greedy.

Satisfiability

Set of states: all possible assignments of true or false to Boolean variables.

Cost function: +1 for each unsatisfied constraint

Neighbourhood function: Change/flip k variables

Example: Partition

Set of all states:

x_i: $i = 1, \ldots,$ # of objects

$\text{dom}(x_i) = \{0, 1\}$

All possible assignments where

$x_i = 0$ means x_i is in U

$x_i = 1$ means x_i is in V

Cost function : difference in weights of U and V

e.g. $u = \{a, b, c, d\}$, $v = \{e, f, g, h\}$, $|32 - 58| = 26$

Neighbourhood function

Poor:

Swap two: pick an object from U and one from V and swap them

Better:

Pick an object and move it to the other set.

Set Covering

Cost function:

size/cost of cover setting

penalize for uncovered rows

Genetic Algorithms

May 29, 2014 2:45 PM

Example Representations: 4-queens

What are the x_j?

1. Permutation representation
   e.g. $\langle 2, 1, 4, 3 \rangle$
2. Extended pair representation
   For each pair of queens, which one comes before the other
   $\frac{x_{12}}{0}, \frac{x_{13}}{0}, \frac{x_{14}}{1}, \frac{x_{23}}{1}, \frac{x_{24}}{0}, \frac{x_{34}}{1}$
   $x_{ij} = 1$ if $x_i < x_j$ and 0 otherwise
   Solution: 1 0 1 0 0 1
3. Possible row positions encoded in binary
   $\frac{x_1}{00}, \frac{x_2}{00}, \frac{x_3}{00}, \frac{x_4}{00}$
   01 01 01 01
   10 10 10 10
   11 11 11 11

Fitness Function

\# of constraints satisfied

Genetic Operations

Mutation (Unary)

Flip a bit(s) with some small probability

Crossover (Binary)

Given $a = (a_1, \dots, a_m)$ and $b = (b_1, \dots, b_m)$
child $= (c_1, \dots, c_m)$
$c_i =$ choose between a_i or b_i (not a good description)

Logic & Inference

May 29, 2014 3:16 PM

Holmes Scenario

Variables

w – watson calls
g – gibbon calls
a – alarm
b – buglaring

Knowledge Base

$w \Rightarrow a, \quad g \Rightarrow a, \quad a \Rightarrow b$
But these are not categorically true. Logic insufficeint

Query

b? Is there are burglary in progress)

Probability

June 3, 2014 2:41 PM

Axioms of Probability

1. All probabilities are between 0 and 1
   $0 \le P(a) \le 1$
2. Necessarily true propositions have probability 1
   Necessarily false propositions have probability 0
   $P(true) = 1, \quad P(false) = 0$
3. The probability of a conjunction is given by,
   $P(A \wedge B) = P(A) + P(B) - P(A \vee B)$

Example - Slides: Holmes Example

$P(B) = p_1 + p_3 + p_5 + p_7$
$P(W \wedge B) = p_5 + p_7$
$P(W \vee B) = p_1 + p_3 + p_4 + p_5 + p_6 + p_7)$
$P(W \vee \neg W) = 1$

$$P(B|W) = \frac{P(B \wedge W)}{P(W)} = \frac{p_5 + p_7}{p_4 + p_5 + p_6 + p_7}$$

$$P(\neg B|W \wedge A) = \frac{P(\neg B \wedge W \wedge A)}{P(W \wedge A)} = \frac{p_6}{p_6 + p_7}$$

Examples of Probabilistic Reasoning

Example: (B)urglary and (A)larm
Suppose the alarm in 95% of cases is accurate. i.e. if there is a burglary, the alarm goes.
In 97% of cases when not burglary, the alarm does not go
We get

False positive
$P(A|B) = 0.95, \quad P(A|\neg B) = 0.03$
$P(\neg A|B) = 0.05, \quad P(\neg A|\neg B) = 0.97$
Probability of burglary $P(B) = 0.0001,\ P(\neg B) = 0.9999$

Suppose alarm goes. What is the probability of a burglary.

$$P(B|A) = \frac{P(B \wedge A)}{P(A)} = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|\neg B)P(\neg B)} = \frac{0.95 \times 0.0001}{0.95 \times 0.0001 + 0.03 \times 0.9999}$$

$$= \frac{0.000095}{0.030092} = 0.00316$$

Belief Network

June 5, 2014 3:05 PM

Edges represent dependencies between variables.
What do the numbers mean?
Frequentist approach / statistics
- objective

Bayesian / subjectivist approach
- degrees of belief

Exact algorithms for finding a particular joint probability given a belief network:
Variable elimination
Cache intermediate results
Factor as much as possible
Exact query answering: #P-Complete (worse than NP-Complete)

Approximate Algorithms

Example

$P(B = \text{false}, G = \text{true}, W = \text{true})$

$$= \sum_{e \in \text{dom}(E)} \sum_{a \in \text{dom}(A)} \sum_{r \in \text{dom}(R)} P(B = \text{false}) P(G = \text{true} | A = a) P(W = \text{true} | A = a) P(A = a | B = \text{false}, E = e) P(E = e) P(R = r | E = e)$$

$$= P(B = \text{false}) \left[\sum_a \sum_e \sum_r P(G = \text{true} | A = a) P(W = \text{true} | A = a) P(A = a | B = \text{false}, E = e) P(E = e) P(R = r | E = e) \right]$$

$$= P(B = \text{false}) \left[\sum_a \sum_e P(G = \text{true} | A = a) P(W = \text{true} | A = a) P(A = a | B = \text{false}, E = e) P(E = e) \underbrace{\left(\sum_r P(R = r | E = e) \right)}_{1} \right]$$

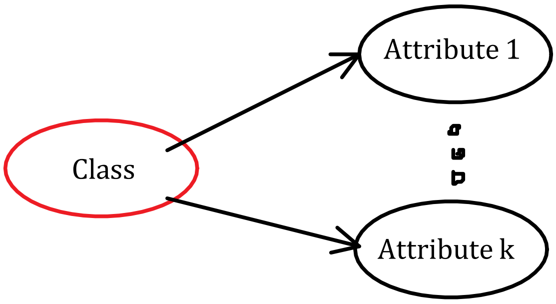
$$= P(B = \text{false}) \left[\sum_a P(G = \text{true} | A = a) P(W = \text{true} | A = a) \left[\sum_e P(A = a | B = \text{false}, E = e) P(E = e) \right] \right]$$

Supervised Learning

June 12, 2014 2:32 PM

Naïve Bayes

Querying a Naïve Bayes Network



Let domain of class variable be $\{c_1, \dots, c_k\}$
Given values for the attributes

Attribute 1 $= a_1$
$\vdots$
Attribute k $= a_k$

To predict class

$$\underset{c_i}{\operatorname{argmax}}\, P(\text{class} = c_1 | \text{evidence}) = \underset{c_i}{\operatorname{argmax}}\, P(\text{class} = c_i | \text{Attribute 1} = a_1, \dots, \text{Attribute k} = a_k)$$

$$= \underset{c_i}{\operatorname{argmax}}\, P(\text{class} = c_i | a_1, \dots, a_k) = \underset{c_i}{\operatorname{argmax}} \frac{P(\text{class} = c_i \wedge a_1, \dots, a_k)}{P(a_1, \dots, a_k)}$$

$$= \underset{c_i}{\operatorname{argmax}} \frac{P(\text{class} = c_i) \prod_{j=1}^{k} P(a_j | \text{class} = a_j)}{P(a_1, \dots, a_k)}$$

$$= \underset{c_i}{\operatorname{argmax}}\, P(\text{class} = c_i) \prod_{j=1}^{k} P(a_j | \text{class} = a_j)$$

Learning arcs and probabilities

Each attribute/features becomes a node in the network.

Steps

1. For each attribute and each possible set of parents, calculate a score

   $a_i, \quad a_1 \to a_i, \dots, a_k \to a_i, \dots, \quad (a_1, a_2) \to a_i, \dots, (a_{k-1}, a_k) \to a_i, \dots$

   Many possible scores. Scores capture goodness of fit and penalty term for complexity.
   Two popular scores: BIC & BDeu

   BIC = Bayesian Information Criterion
   Bdeu = Bayesian Dirichlet (likelihood equivalence) (uniform joint distribution)
2. For each attribute/class variable pair pick a parent set such that
   a. there are no cycles, and
   b. the sum of scores is minimized

   Pruning rule: Two parent sets p, p' for some attribute

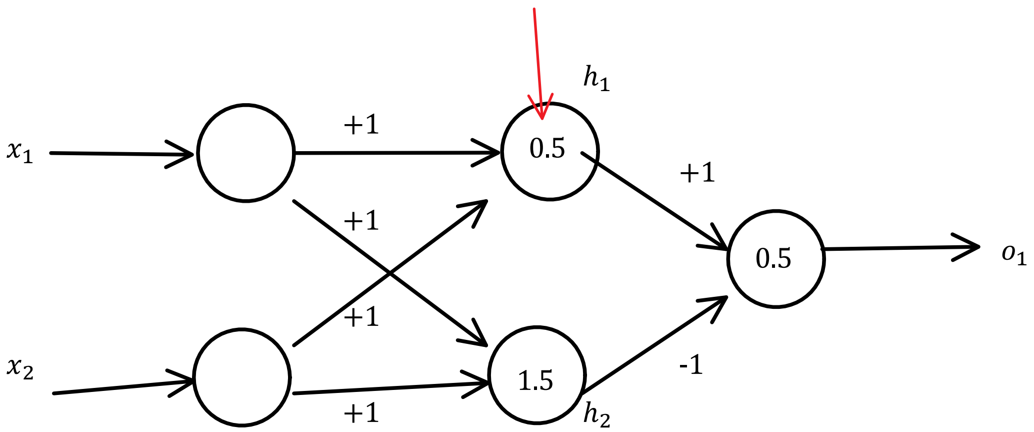
   $p \subset p'$ and $\text{cost}(p) \leq \text{cost}(p')$ then prune (remove) p'

Neural Networks

June 26, 2014 3:28 PM

Example: XOR Function

Input	Output
x_1 x_2	y_1
0 0	0
0 1	1
1 0	1
1 1	0



Step function $f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$

$h_1 = f(x_1 + x_2 - 0.5)$
$h_2 = f(x_1 + x_2 - 1.5)$
$o_1 = f(h_1 - h_2 - 0.5)$

x_1	x_2	h_1	h_2	o_1
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Backpropagation Learning Algorithm

$x_1 - h_1 \ - o_1$
X X
$x_2 - h_2 \ - o_2$
$\vdots$
X X
$x_A - h_B - o_C$

Each hidden and output unit uses signal function $f(x) = \frac{1}{1+e^{-kx}}$

Output is a value between 0 and 1

Handling Thresholds

$$h_j = f\left(\left(\sum_{i=1}^{A} \mathrm{w1}_{ij} \cdot x_i\right) - \beta_j\right), \qquad j = 1, \dots, B$$

Rewrite as,

$$h_j = f\left(\sum_{i=0}^{A} \mathrm{w1}_{ij} \cdot x_i\right), \qquad j = 1, \dots, B$$

where $x_0 = -1$ and $\mathrm{w1}_{0j}$ is the threshold for h_j

Same for output layer:

$$o_k = f\left(\sum_{j=0}^{B} \mathrm{w2}_{jk} \cdot h_j\right), \qquad k = 1, \dots, C$$

where $h_0 = -1$

Error Term

$$\mathrm{error} = \frac{1}{2}\sum_{k=1}^{C} (y_k - o_k)^2$$

Algorithm

1. Initialize weights & thresholds to small random values

   $\mathrm{w1}_{ij} = \mathrm{random}(-0.5, 0.5), \qquad i = 1, \dots, A, \qquad j = 1, \dots, B$

   $\mathrm{w2}_{jk} = \mathrm{random}(-0.5, 0.5), \qquad j = 1, \dots, B, \qquad k = 1, \dots, C$

   $x_0 = -1,\ h_0 = -1$ These never change
2. Choose an input-output pair ffrom the training set. Call it $\bar{x}, \bar{y}$

   where $\bar{x} = (x_1, \dots, x_A),\ \bar{y} = (y_1, \dots, y_C)$

   Assign activation levels of $x_1, \dots, x_A$ (input units)
3. Determine activation levels of hidden units.

   $$h_j = f\left(\sum_{i=0}^{A} \mathrm{w1}_{ij} \cdot x_i\right), \qquad j = 1, \dots, B$$
4. Determine activation levels of output units

   $$o_k = f\left(\sum_{j=0}^{B} \mathrm{w2}_{jk} \cdot h_j\right), \qquad k = 1, \dots, C$$
5. Determine how to adjust weights between hidden and output layer for this example.

   $$\mathrm{E2}_j = \underbrace{k}_{\text{from sigmoid}} \cdot o_j(1 - o_j)(y_j - o_j), \qquad j = 1, \dots, C$$

   Sigmoid: $f(x) = \frac{1}{1+e^{-kX}}$
6. Determine how to adjust weights between input and hidden layer for this example.

   $$\mathrm{E1}_j = k \cdot h_j \cdot (1 - h_j) \sum_{i=1}^{C} \mathrm{E2}_i \cdot \mathrm{w2}_{ji}\,, \qquad j = 1, \dots, B$$
7. Adjust weights between hidden and output layer.

   $\mathrm{w2}_{ij} = \mathrm{w2}_{ij} + \mathrm{LearningRate} \cdot \mathrm{E2}_j \cdot h_i, \qquad i = 0, \dots, B, j = 1, \dots, C$
8. Adjust weights between input and hidden layer

   $\mathrm{w1}_{ij} = \mathrm{w1}_{ij} = \mathrm{LearningRate} \cdot \mathrm{E1}_j \cdot x_i, \qquad i = 0, \dots, A, j = 1, \dots, B$
9. Repeat steps 2-8 until done.

Parameters

Learning rate:

range: 0.05 to 0.35

Sigmoid constant k

$$f(x) = \frac{1}{1 + e^{-kx}}, \qquad k \geq 0$$

Number of hidden units

Stopping Criteria

Maximum number of epochs. epoch = once through training set.
Error is acceptably small

- discrete/classification error
- total number of bit-errors
- continuous error $\sum(y_j - o_j)^2$