i/p space

Model complex relations

o/p space

Model complex relations

Simple linear kernels

Graphical model

Complex graph or first order kernels

First order prob models

$K(O_1, O_2) \in \mathbb{R}$

2 classes: Study ways of generating interpretable "features" for $O_1$ & $O_2$ s.t the clustering / classification / regression problem at hand can be solved effectively.

# Unsupervised feature mining

| Transaction ID | Items |
|---|---|
| 1 | {D,O,N,K,E,Y} |
| 2 | {M,O,N,K,E,Y} |
| 3 | {M,A,K,E} |
| 4 | {M,U,C,K,Y} |
| 5 | {C,O,O,K,I,E} |

$$\Sigma = \{A, C, D, E, I, K, M, N, O, U, Y\}$$

Figure 1: Dataset $\mathcal{D}$

2. Let us say you are provided the data set $\mathcal{D}$ as in Figure 1. Propose an efficient algorithm to find all subsets of $\Sigma$ that are subsets of more than $m$ transactions (for some fixed $m$). (This is the idea behind the classic apriori algorithm).

Eg: Find all $x \subseteq \Sigma$ s.t $x$ covers atleast 2 transactions

$\text{Covers}(x, t) = 1$ iff $x \subseteq t$

$\text{covers}(x, t) = 0$ o/w

$\{C\} = \{4,5\}$ $\{E\} = \{1,2,3,5\}$ $\{M\} = \{2,3,4\}$

$\{C, E\} = \{5\}$ $\{E, M\} = \{2,3\}$

$\{D\} = \{1\}$ $\{Y\} = \{1,2,4\}$

Because $|D|=1$

$\therefore |\{D, Y\}| \leq 1$

$\{D, Y\} = \{1\}$

$S_i$ = set of all subsets of $\Sigma$ of size $i$ s.t. each element of $S_i$ covers at least $m$ transactions

**Algo**

1. Find all $s_1 \in \Sigma$ st $|\{t \mid covers(s_1, t) = 1\}| \geq m$
   $S_1$ = union of all such $s_1$
2. Given $S_1, S_2 \ldots S_{i-1}$ how can I construct $S_i$?

(a) $S_{i-K} \times S_K$ (b) $S_{i-1} \times S_{i-1}$

$\boxed{\text{③ Find subset of } S_i \text{ that is actually frequent \& go back to ②}}$

Different options for step 2

a: $S_i = S_{i-K} \times S_K$ for any $K \in [1, i-1]$

such that if $S_i = S_{i-K} \cup S_K$ then $S_{i-K}$ & $S_K$ are disjoint

choice of k may depend on how small $S_{i-K}$ or $S_K$ can be

eg: $\{N, K, E, Y\} \Rightarrow \{1, 2\}$

$\{N, K\} \cup \{E, Y\}$

$\{N, E\} \cup \{K, Y\}$

$\{N, Y\} \cup \{K, E\}$

$\underbrace{\quad}_{S_2 \cup S_2' \quad (k=2)}$

$S.t.\ S_2 \cap S_2' = \phi$

selectivity (size) of $S_2$ is more

$\{N, K, E\} \cup \{Y\}$

$\{N, K, Y\} \cup \{E\}$

$\{K, E, Y\} \cup \{N\}$

$\{N, E, Y\} \cup \{K\}$

$\underbrace{\quad}_{S_3 \cup S_1}$

$S.t.\ S_3 \cap S_1 = 0$

selectivity (size) of $S_3$ is small

than that of
$S_3$ but less than
$S_1$

Antimonotonicity property:

If $S_n$ covers $t$ (denoted by $S_n \sqsubseteq t$) then $\forall\ k \leq n$, s.t $S_k \subseteq S_n$, $S_k$ covers $t$ ($S_k \sqsubseteq t$)

The covers relation $\sqsubseteq$ is a partial order, which means:

1. Reflexivity: $s \sqsubseteq s$
2. Antisymmetry: if $s \sqsubseteq t$ & $t \sqsubseteq s$ then $s = t$
3. Transitivity: if $s \sqsubseteq t$ & $t \sqsubseteq u$ then $s \sqsubseteq u$

(6) Since selectivity of $S_{i-1}$ is more than that of $S_{i-2}$ . . .

(eg: $|S_3| = 4$ $|S_2| = 6$ $|S_1| = 4$)

idea behind apriori algorithm for mining freq patterns

$$S_i = \underbrace{S_{i-1} \times S_{i-1}}$$

Consider unioning $S_{i-1}$ & $S'_{i-1}$ s.t

$|S_{i-1} \cap S'_{i-1}| = i-2$

i.e $S_{i-1}$ & $S'_{i-1}$ have exactly 1 element differing

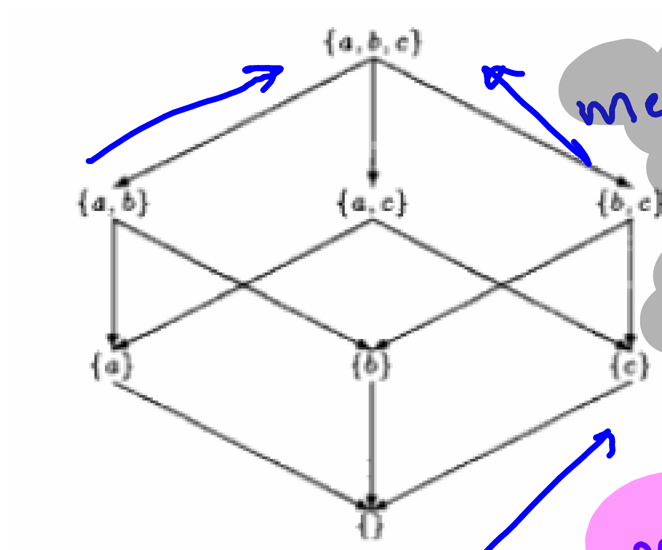
$\Rightarrow |S_{i-1} \cup S'_{i-1}| = i$

Eg: $S_{i-1} = \{N, K, E\}$ $S'_{i-1} = \{K, E, \underline{Y}\}$

$|S_{i-1} \cap S'_{i-1}| = |\{K, E\}| = 2 = i-2$

$S_i = \{K, E, Y, N\}$

These 2 search paradigms span the entire spectrum of methods for constructing "interpretible" features



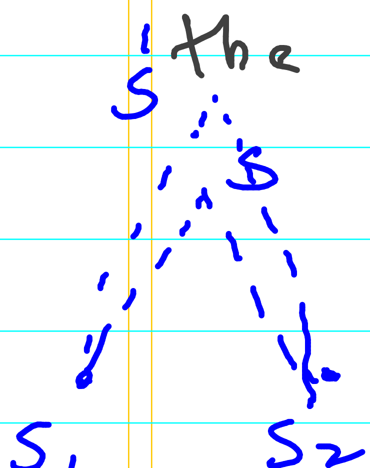

LGG method (b) computes the "least general generalisation of 2 nodes

method (a) traverses edges to compute new nodes (for case k=1)

REFINEMENT

Figure 1.1: The lattice structure of $\langle S, \preceq \rangle$, where $S$ is the power set of $\{a, b, c\}$

Given a p.o $\langle S, \sqsubseteq \rangle$ or $\langle S, \preceq \rangle$ the lgg of $s_1$ & $s_2$ is



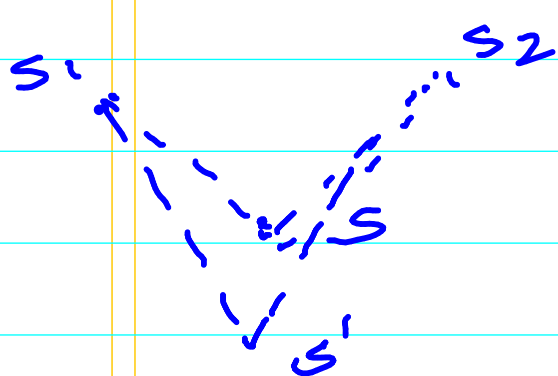

$s_1 \sqcup s_2 = s$ iff ① $s_1 \preceq s$
$s_2 \preceq s$

lgg = least uper bound (lub)

② If $\exists s'$ s.t. $s_1 \leq s'$, $s_2 \leq s'$ then $s \leq s'$

Given a p.o $\langle S, \sqsubseteq \rangle$ or $\langle S, \leq \rangle$ the glb (greatest lower bound) of $s_1$ & $s_2$ is

$s_1 \sqcap s_2 = s$ iff ① $s \leq s_1$
$s \leq s_2$
② If $\exists s'$ s.t $s' \leq s_1$ & $s' \leq s_2$ then $s' \leq s$




**A Partial Order $\langle S, \sqsubseteq \rangle$ or $\langle S, \leq \rangle$ for which a lub and glb exist in $S$ for any $s_1 \in S$ & $s_2 \in S$ is called a LATTICE**

**NOTE:** In the table that follows, definitions of $\leq$ could be reversed to make them definitions for $\geq$. Eg. If $s_1 \leq s_2$ then there exists $\sqsubseteq$ s.t $s_1 \leq s_2$ iff $s_2 \sqsubseteq s_1$, s.t $s_1 \sqcap_{\leq} s_2 = s_1 \sqcup_{\sqsubseteq} s_2$ & $s_1 \sqcup_{\leq} s_2 = s_1 \sqcap_{\sqsubseteq} s_2$

| Kernel | Corresponding S, a possible $\preceq$ and whether a lattice |
|---|---|
| | [S = implicit feature space] |
| 1) Set kernels<br>(each eg is bag of features)<br>say $\Sigma$ = vocab of features | 1) • $S = 2^{\Sigma}$<br>$\preceq$ is $\subseteq$<br>• Yes, as seen above, we have a lattice<br>• "$\sqcup$" ie lub is "$\cup$" (union)<br>• "$\sqcap$" ie glb is "$\cap$"<br>• Downward refinement = deleting element from set<br>• Upward refinement = adding element to set |
| 2) String kernels<br>(eg eg is a sequence of features)<br>Say $\Sigma$ = vocab of features<br><br>glb is longest common subsequence (LCS) | 2) • $S = \bigcup_{K=1}^{\infty} \overbrace{\underbrace{\Sigma \times \Sigma \times \cdots \Sigma}_{k \text{ times}}}^{\text{set of all subsequences}}$<br>$S_1 \preceq S_2$ if characters in $S_1$ occur in $S_2$ in same order<br>• $S_1 \sqcup S_2 = S$ → (verify this exists)<br>• $S_1 \sqcap S_2 = S$<br>• Downward refinement = deleting element from sequence<br>• Upward refinement = adding |

element to sequence

en.wikipedia.org/wiki/Longest_common_subsequence_problem

Algos for computing lcs

3) Parse tree kernel (each eg is a subtree of the parse tree such that all productions are complete)

4) Rational kernel

www.cs.bgu.ac.il/~karyeh/nptl.pdf

5) Graph kernel

en.wikipedia.org/wiki/Subgraph_isomorphism_problem

6) KLOG

www.di.uniba.it/~appice/publications/MLDM09.pdf

3) • S = subset of first order logic language (subset language restricted to that representing parse trees)

• $\leq$ ≡ θ subsumption in definite first order logic (FOL) (sections 1.4.6 & 2.1.1 of

www.cse.iitb.ac.in/~cs717/notes/classNotes/extendedNotes.

• $\sqcap$ & $\sqcup$ are defined in terms of glb & lub in definite FOL (sections 2.1.1 onwards of

www.cse.iitb.ac.in/~cs717/notes/classNotes/extendedNotes.

4) • S = All weighted finite state transducers

4) (same as above, ie defined in terms of subset of FOL just as in the case of parse trees) Additionally, "$\leq$" could correspond to SUBGRAPH ISOMORPHISM discussed in earlier lectures

5] S = definite clause FOL. Entire chapter 2 of www.cse.iitb.ac.in/~cs717/notes/classNotes/extendedNotes.pdf

For all above cases, existence of partial order implies the existence of a simple "antimonotonic quality criterion" (ie $Q(S) = True$ iff $S \subseteq$ example)

$Q$ is called an anti-monotonic quality criterion iff

$$Q(s) = True \Rightarrow Q(s') = True \text{ for all } s' \leq s$$

Consider the following Q's

(A) $D = \begin{array}{|c|c|} e_1^+ & e_1^- \\ e_2^+ & e_2^- \\ \vdots & \vdots \\ e_{n_1}^+ & e_{n_2}^- \end{array}$ $n_1$ +ve & $n_2$ -ve egs.

Let $S$ be the space of "features" (like one of the above) & let

S could be any lattice with some "$\leq$"

each $e_i^+ \in S$

& each $e_j^- \in S$

$Q_i(s) = \text{true}$ iff $|\{e_i^+ \mid s \leq e_i^+\}| \geq k$

& = false o/w

Is $Q(s)$ antimonotonic?

$\{\bar{e}_i^+ \mid s' \leq \bar{e}_i^+\} \supseteq \{e_i^+ \mid s \leq e_i^+\}$

if $s' \leq s$ & $s \leq e_i^+ \Rightarrow s' \leq e_i^+$

(since $\leq$ is transitive)

$\therefore$ if $s' \leq s$

$|\{\bar{e}_i^+ \mid s' \leq \bar{e}_i^+\}| \geq |\{e_i^+ \mid s \leq e_i^+\}|$

$\Rightarrow$ Q will be antimonotonic

(B) D is same as before

$Q(s) = \text{true}$ iff $\overbrace{|\{e_i^+ \mid s \leq e_i^+\}|}^{\text{pos}} - \overbrace{|\{e_j^- \mid s \leq e_j^-\}|}^{\text{neg}} \geq k$

= false o/w

i.e (pos-neg) loss should $\geq k$

Ans: Q is NOT antimonotonic

Proof: By counter example [H/W]

(c) D is same as before

Q(S) = True iff max margin loss $\geq k$

Ans: Q is NOT antimonotonic

Proof: By ________________ [H/W]

Q: For general quality criteria Q(S) that are antimonotonic, you could use the frequent pattern mining algo (using subroutines (a) or (b) discussed earlier) to search for all the "good" patterns in $\langle S, \leq \rangle$

- But if $Q(S)$ is not antimonotonic, what can you do? → Data Mining problem
- Instead of asking for → Machine Learning problem

$$S^* = \{ s \in S \mid Q(s) = \text{True} \}$$

if I asked for

$$S^* = \underset{S' \subseteq S}{\text{argmax}} \; \boxed{\text{objective}(s')}$$

eg:

pos-neg

likelihood of data $D$ based on $s'$

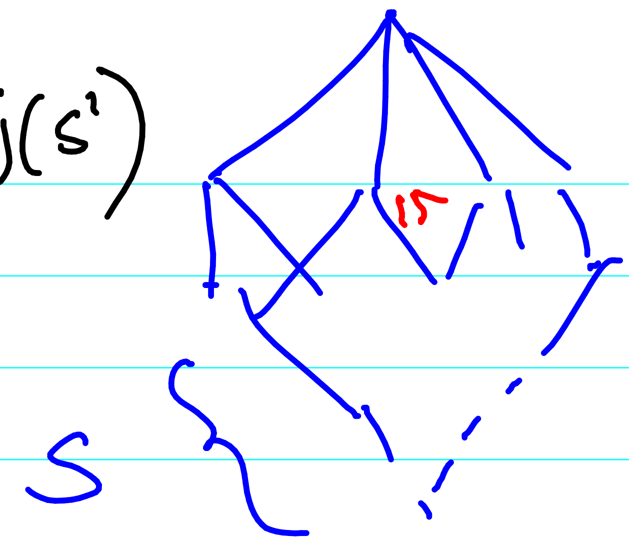
hinge loss with regulariser based on $D$

how do you search for $S^*$ effectively.

The Machine Learning problem can be more interesting & generic:-

Approaches to solve the search problem (say machine learning setting)

Find $s^* = \underset{s' \subseteq S}{\arg\max}\ obj(s')$




option 1

DFS (exhaustive search) ✗

Traverse all subsets of nodes using DFS & find best

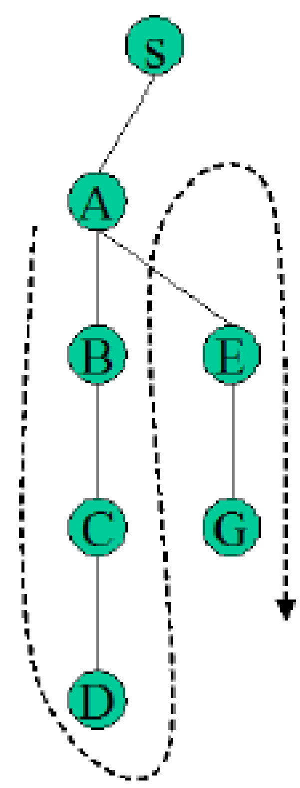



Figure 6.5: The DFS tree from $S$ to $G$ through the map in Figure 6.3.

Read about all these search algos in chapter 6 of

http://www.cse.iitb.ac.in/~cs717/notes/classNotes/extendedNotes.pdf

Option 2 Greedy hill climbing = Greedy DFS

(Greedy for DFS)

At A, find most "promising path" and choose between B & E

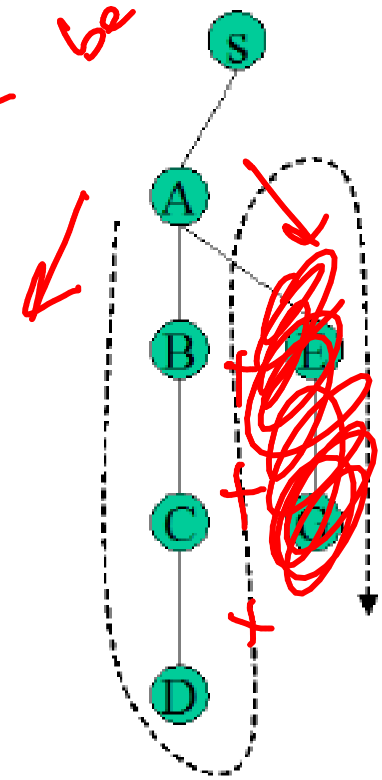



Figure 6.5: The DFS tree from $S$ to $G$ through the map

To choose next node at A

$N^* = \arg\max \; obj(\{S, A, N\})$

$N = B$ or

$N = E$

i.e

$N^* = \arg\max \; obj\left[\{S, A, B\}, \{S, A, E\}\right]$

Examples of algos / approaches that use this idea

ⓐ MLRules: Greedy search over simple conjunctions of features

objective used = Likelihood of

naive bayes model over the set S'

(b) kFOIL: Greedy search over space of first order features

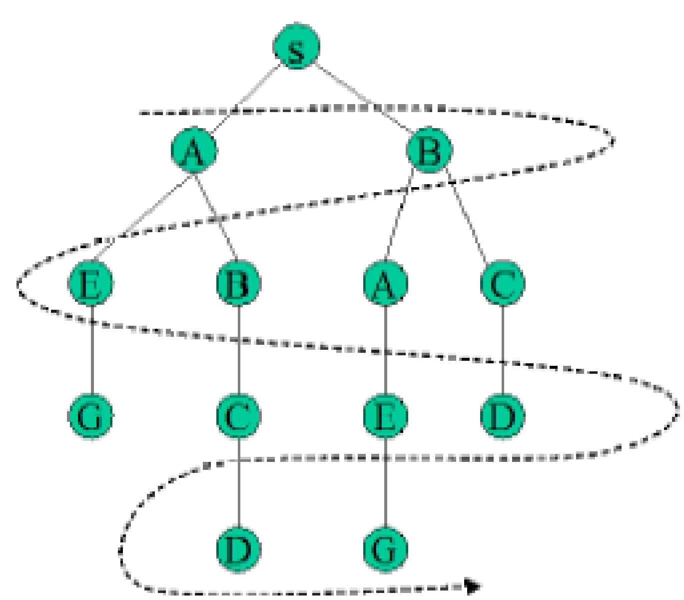
objective used = SVM objective / SVM accuracy

(c) nFOIL: Like kFOIL, searches over space of first order features

objective used = likelihood of Naive Bayes (MLRules)

3 Breadth first search

Exhaustive search (X not feasible)



④ BEAM SEARCH = greedy BFS

Breadth first search but retain & expand only top few promising nodes at each level

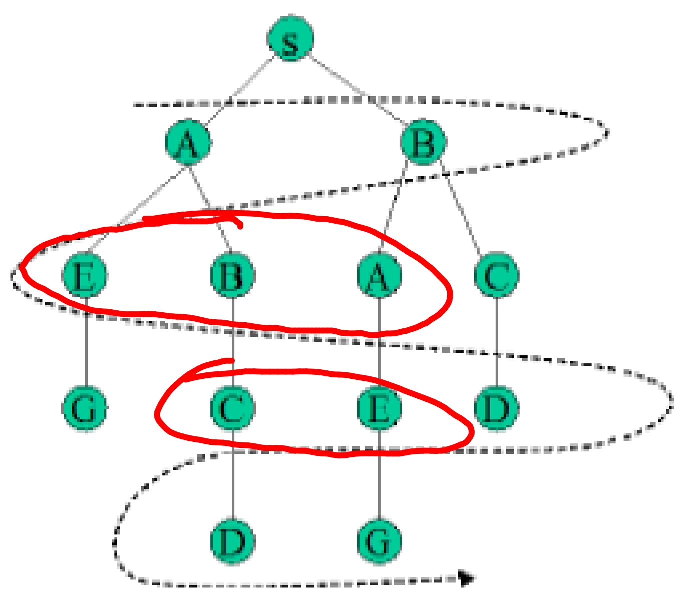
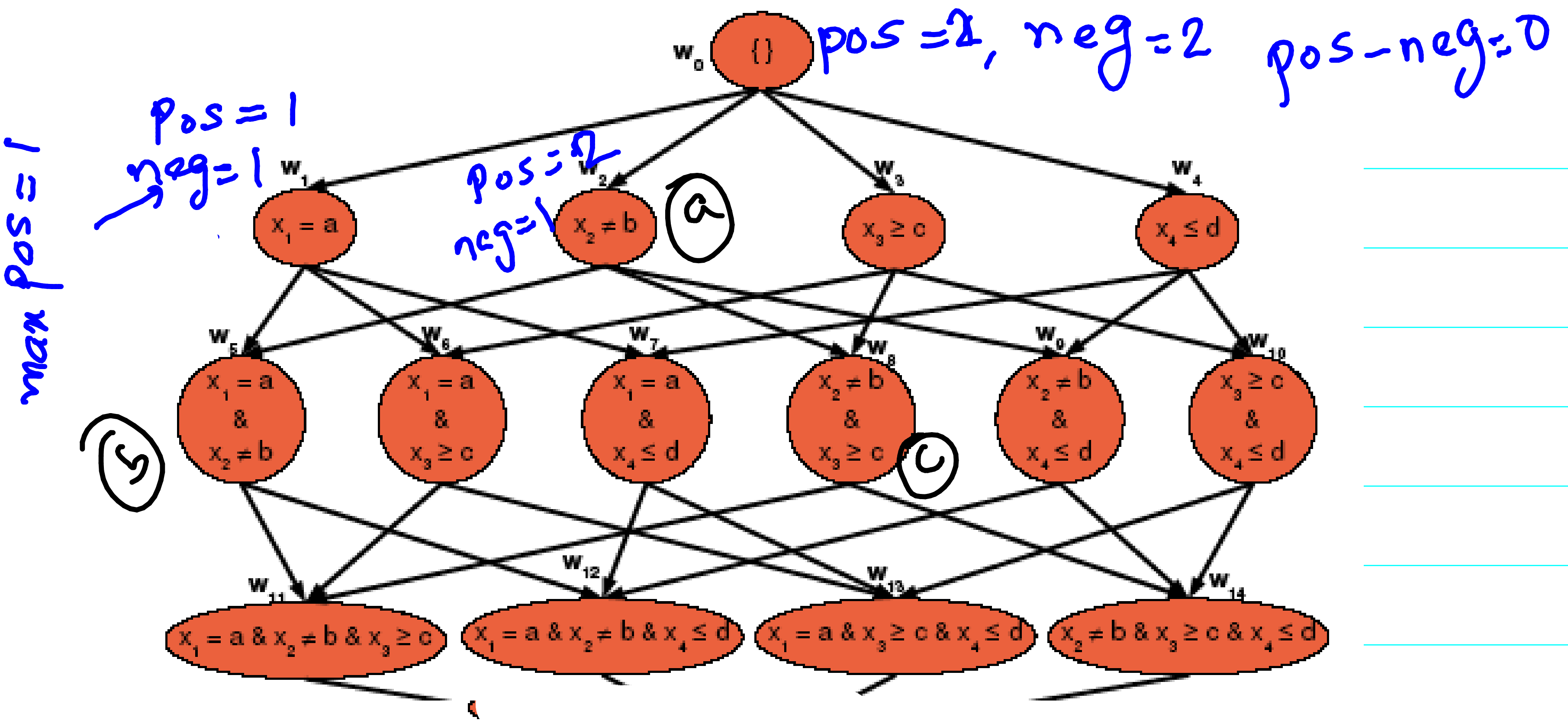



Figure 6.6: The BFS tree from $S$ to $G$ through the map in Figure 6.3.

Non-exhaustive

eg: Decision Tree construction in propositional as well as first order logic [TILDE]

⑤ Branch & Bound search




$D =$

| | |
|---|---|
| $e_1^+ = \{x_1 = a, x_2 \neq b\}$ | $e_1^- = \{x_1 \neq a, x_2 \neq b\}$ |
| $e_2^+ = \{x_2 \neq b, x_3 \geq c\}$ | $e_2^- = \{x_1 = a, x_3 \geq c\}$ |

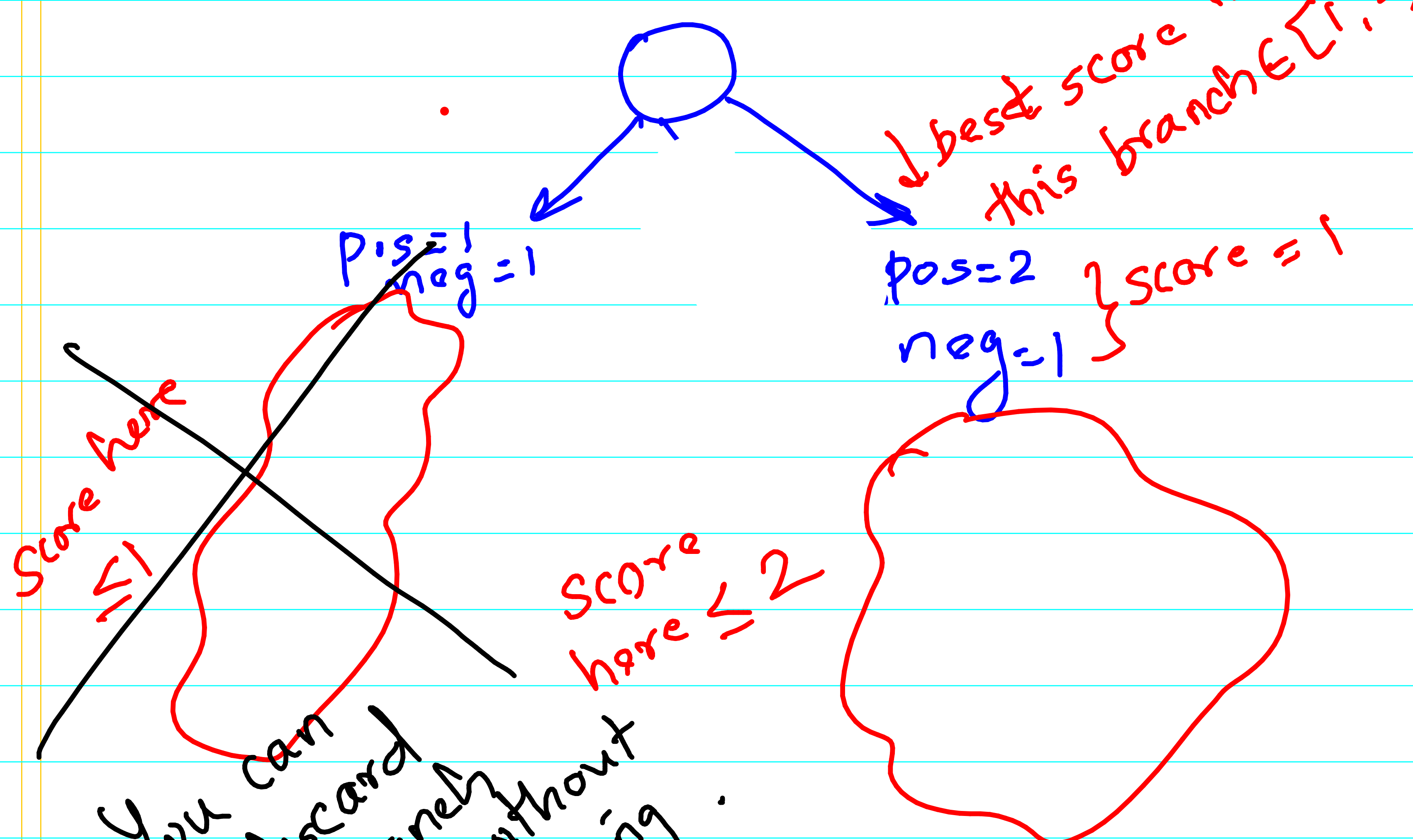
Find $s^*$ s.t. $s^* = \underset{s \in S}{\arg\max}\ pos(s) - neg(s)$

If $pos(s)$ & $neg(s)$ are # of positive & negative examples covered by $s$, then $\forall\ s' \leq s,\quad pos(s') \leq pos(s)$

$\leq\ \equiv\ \supseteq \qquad 0 \leq neg(s') \leq neg(s)$

$$\Rightarrow\ pos(s') - neg(s') \leq pos(s)$$

pos(s) is an upper bound on score of all children of s.




You can discard this branch even without exploring.

In fact, we can show that

$$\text{node } (a) = \underset{s \in S}{\operatorname{argmax}} \; pos(s) - neg(s) \quad (*)$$

→ our branch & bound works for this

$$\{ (b), (c) \} = \underset{S' \subseteq S}{\operatorname{argmax}} \sum_{s \in S'} pos(s) - neg(s) \quad (**)$$

How abt branch & bound for this?

While for pos-neg, a bound for (**) is not known, we have bounds found for this obj

**Generalized HKL**

$$\min_{\mathbf{w}, b} \frac{1}{2} \left( \sum_{v \in \mathcal{V}} d_v \|\mathbf{w}_{D(v)}\|_\rho \right)^2 + C \sum_{i=1}^{m} L\left( y^i, \sum_{v \in \mathcal{V}} w_v R_v(\mathbf{x}^i) - b \right) = obj(\mathcal{V}' \subseteq \mathcal{V})$$

where $1 < \rho \leq 2$.

Hierarchical norm regulariser

$S = \mathcal{V}$

We solve $\underset{\mathcal{V}' \subseteq \mathcal{V}}{\operatorname{argmax}} \; obj(\mathcal{V}')$

using branch & bound

bound is just a sufficiency condition

We have also extended this branch and bound approach to a modified StructSVM formulation (Regulariser modified as above)

All this and more at

http://www.cse.iitb.ac.in/~cs717/notes/classNotes/InterpretibleFeatures/featureInductionTalk.pdf