MIT 6.034 - Artificial Intelligence.

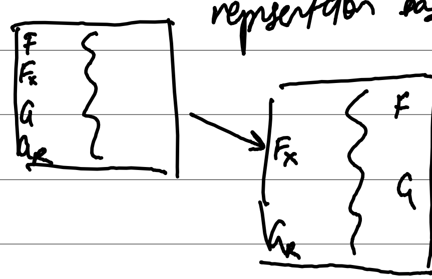
Instructor: Patrick Winston.

Lecture 1: Introduction & Scope.

representations of
Models targeted of:
Thinking, perception, Action.

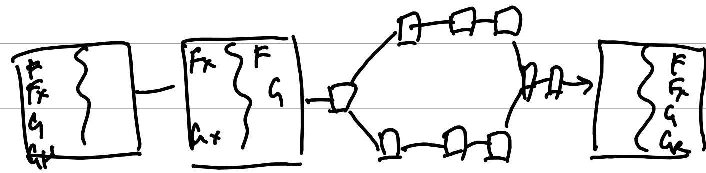
representation of problems:
- Farmer fox goose grain & robot problem.

representation based on location



2^4 possibilities of representations

STATE DIAGRAM.



representations ↗
will expose constraints

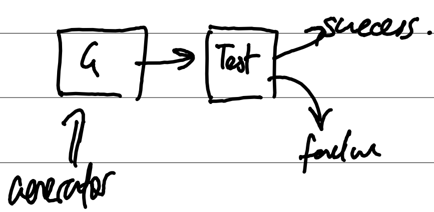
AI: Algorithms enabled by constraints exposed by representations that support models targeted at thinking, perception & actions

Generate & Test.

Artificial intelligence Generated test.

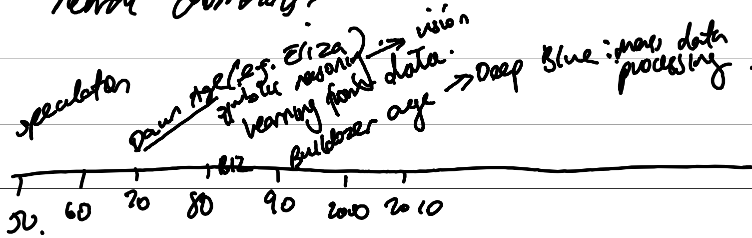
ex: flick through book to identify a leaf.

- Generate AND Test



Rumplestiltskin principal.

Rumplestiltskin principle: Once you know the name of something, you have power over it.

simple vs. trivial.	simple ≠ trivial. → of little worth. ↓ can be very powerful.
	Q: How many countries in Africa does the equator cross?
knowledge translation	→ map ⇒ 6 countries. language → brain → vision → brain → language. → how do we translate knowledge across these neural domains?
History of AI	
	rule based expert systems.
	• 'mycin' system.
	• B1Z's expert system (plane parking).

1. Humans have been around for ≈200,000 years in the current form.

2. 50,000 years ago, some of us developed the ability to separate from one another.

   - small group could create 2 concepts & create a 3rd concept without disturbing the 1st two concepts.

eg. imagine running down the street holding a full bucket of water → human language system allows us to imagine & describe this without explicitly being told.

4 activities in this course:

1. Lectures.
2. Recitations - (..) & expansion of material.
3. Mega-recitations
4. Tutorials.

Lecture 2: Goal Trees & Problem Solving

Agenda:

- ☐ Modelling problem solving
  - ○ Generate & Test
  - ○ problem Reduction
- ☐ Problem Reduction Tree
- ☐ Transforms & example.
- ☐ Reflections.

ex:

$$\int \frac{-5x^4}{(1-x^2)^{5/2}}\, dx.$$

1. Transform & simplify until you can find it in a table of integrals.

Problem Reduction.

SKILL

UNDERSTAND

WITNESS.

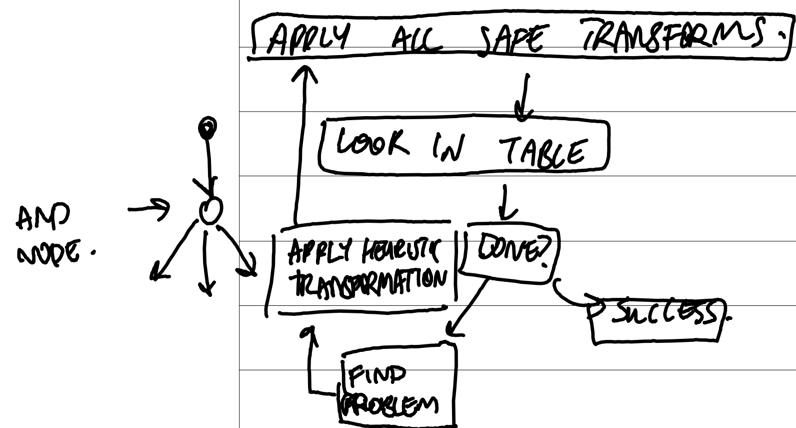
SAFE TRANSFORMATIONS

$$\int c f(x)\,dx = c \int f(x)\,dx$$

$$\int \sum f(x)\,dx = \sum \int f(x)\,dx.$$

$$\int -f(x)\,dx = -\int f(x)\,dx$$

$$\int \frac{P(x)}{Q(x)} \longrightarrow \text{DIVIDE.}$$

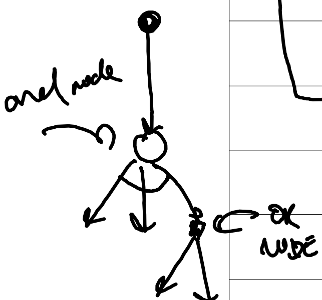


$$\int \frac{-5x^4}{(1-x^2)^{5/2}}\,dx \longrightarrow \int \frac{5x^4}{(1-x^2)^{5/2}}\,dx$$

$$\longrightarrow \int \frac{x^4}{(1-x^2)^{5/2}}\,dx \xrightarrow{x=\sin y} \int \frac{\sin^4 y}{\cos^4 y}\,dy$$

$$\longrightarrow \int \frac{1}{\cot^4 x}\,dx$$

$$\longrightarrow \int \tan^4 x\,dx \xrightarrow{y=\tan(x)} \int \frac{y^4}{1+y^2}\,dy$$



Problem Reduction tree /

And/OR tree /

Goal tree.

$$\int \left(y^2 - 1 + \frac{1}{1+y^2}\right) dy$$

(AND NODE)

$$\int y^2\,dy - \int dy + \int \frac{1}{1+y^2}\,dy$$

$$\xrightarrow{y=\tan z} \int dz$$

HEURISTIC TRANSFORMATIONS.

- sometimes useful, sometimes useless.

A: $f(\sin x, \cos x, \tan x, \cot x, \sec x, \operatorname{cosec} x)$

$= g(\sin x, \cos x)$

$= g_2(\tan x, \operatorname{cosec} x)$

$= g_3(\cot x, \sec x)$

B: $\int f(\tan x)\,dx = \int \frac{f(y)}{1+y^2}\,dy$

C: $1-x^2$ $\quad x = \sin(y)$

$1+x^2$ $\quad x = \tan(y)$

CATECHISM (2):

- what kind of knowledge is required/involved
  - transformation
  - Goal trees
  - looking in tables.
- how is knowledge represented?
  - lists, tables
  - goal trees embedded in procedures.
- how is it used?
  - transformations - simple.
- how much knowledge is required?

Knowledge about knowledge is power.

Lecture 3 - Reasoning: Goal Trees & Rule-Based expert systems

Program answers questions about its own behavior.

Agenda:

- ☐ Blocks
- ☐ Questions

} L1: Goal centered programming

- ☐ Zoos
- ☐ Questions
- ☐ Knowledge Engineering

} L2: Rule based "Expert" systems.

* Simon's Ant (Herb Simon)
* Engineering Yourself
  - specific cases
  - Δ =
  - experiment

Put on method.

↳ Find space. → GetRidOf

↳ Grasp → clear top → GetRidOf

↳ move

↳ ungrasp

Example:

Bx
B1

BY
B2

Puton B1 B2

Find Space Grasp B1 move ungrasp

Cleartop B1

(WHY?) ... → GetRidOF BX

(HOW?) ...

Put Bx on table.

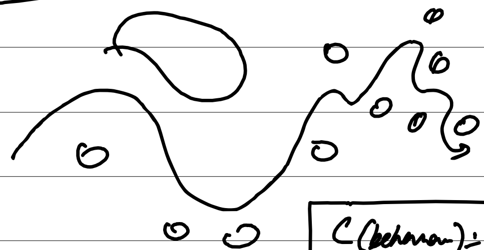
Find Space Grasp move ungrasp.

Look up 1 level and return it

Look down 1 level and return it

GOAL TREE.
AND/OR TREE
⇓
can answer questions of its behaviour.

Simon's ant (metaphor)



movement of an ant on the beach.

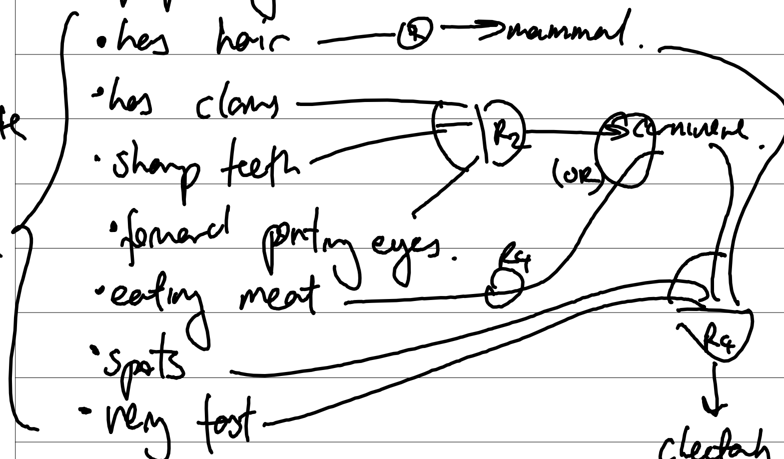
$$C(\text{behavior}) = \max(C(\text{program}),\ C(\text{environment}))$$

Rule based expert systems.

ex. property of animals.

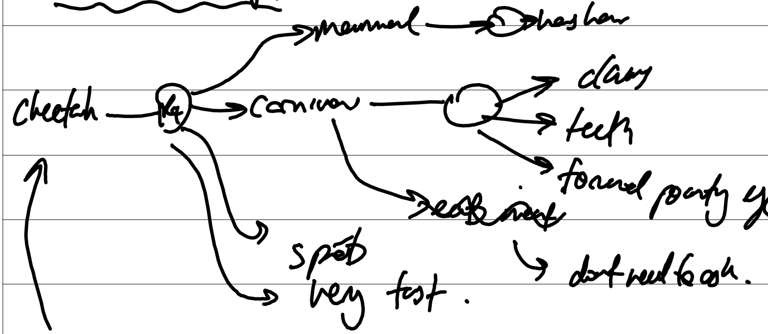
D and gate

forms a goal tree.



Forward Chaining rule-based "expert" system

Deduction System



is it a cheetah? → start w/ hypothesis towards facts & observations

"Backward chaining rule-based expert system"

Principles of knowledge engineering

heuristics

(1) specific cases: canned goods, milk, macaroni, etc.

(2) similar things handled differently:

e.g. canned peas vs. frozen peas

if frozen, place in frozen plastic bag.

(3) Build system & see when it cracks

Are these expert systems intelligence?

↳ no common sense, no understanding of WHY things are done apart from what is told to it.

Megarecitation 1: Rule-Based Systems.

* Check assertions before using a rule.
* Disambiguation key
* Matching / Firing
* Let's talk about ?x

Assertions

A0: Millicent lives in Slytherin Dungeon
A1: Millicent is ambitious
A2 Seamus lives in Gryffindor Tower.
A3 Seamus snogs Millicent

P = proposition
R = rule.

+2,3=5
Prefix notations

Rules

variable waiting to be bound.

P0 $\boxed{\text{IF}}$ (?x is ambitious)
$\boxed{\text{AND}}$ (?x is a squib)
THEN ?x has a bad term.

P1 IF ?x lives in Gryffindor Tower (GT)
THEN ?x is a protagonist.

P2 IF ?x lives in Slytherin Dungeon (SD)
antecedent THEN ?x is a villain
?x is ambitious } 2 consequences.

P3 IF (?x is a protagonist OR ?x is a villain)
AND ?x is ambitious.
THEN ?x studies a lot.

P4: IF ?x studies a lot
AND ?x is a protagonist
THEN ?x becomes Hermione's friend.

P5: IF ?x snogs ?y
AND ?x lives in Gryffindor Tower.
AND ?y lives in Slytherin Tower.
THEN ?x has a bad term.

Backward Chaining

1. Look for matching assertion.
2. Look for rule of consequence.
3. Backward chaining doesn't add assertions to list of assertions.

Q: prove Millicent becomes Hermione's friend.

① No matching Assertion.
② P4 rule:
③ not Assertion.

millicent studies a lot

P4 millicent becomes Hermione's friend.

Millicent Becomes protagonist

P1

M lives in GT

not true.

not true (no assertion)

M lives in G.T.

④ not assertion

M is protagonist

P2

M lives in SD — M is villain

P3

M is ambitious

A0 ⇒ True.

A1 ⇒ True

Q: Determine the min # of assertions we need to add for Millicent to become Hermione's friend.

A: Millicent lives in GT

Q: The assertion causes an issue ⇒ what is the how to fix it?

A: M lives in GT & SD ⇒ Take out SD.

Forward Chaining

Matching

1, 2, 5.

$\boxed{1}$, 2, 5	F_1	S. is a protagonist. → impostor.
1, $\boxed{2}$, 5	F_2	M is a villain.
1, 2, $\boxed{3}$, 5	F_3	M studies a lot
1, 2, $\boxed{5}$	F_5	S has a bad term.

A

↓ P3.

Lecture 4: Search: Depth-First, Hill climbing, Beam

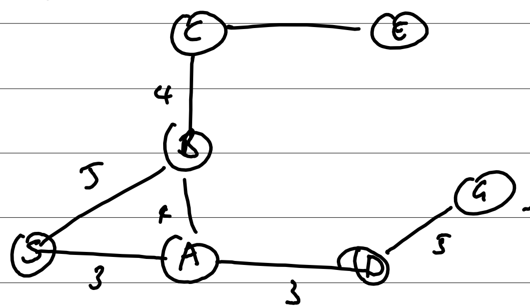
Agenda

Search ≠ maps

	Backtracking	Use Enqueued List	Informed
☐ British museum	✗	✗	✗
☐ depth first	✓	✓	✗
☐ breadth first	✗	✓	✗
☐ Hill climbing	✓	✓	✓
☐ Beam			

* Search is about choice

Starting position (S) to goal position (G)



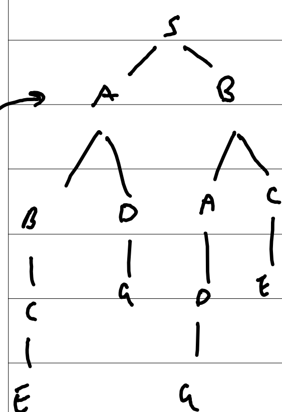
British museum search/expansion.

Finds all possible paths.

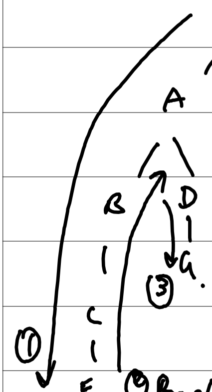
search tree

alphabetical order
"lexical" order.

no tail biting →



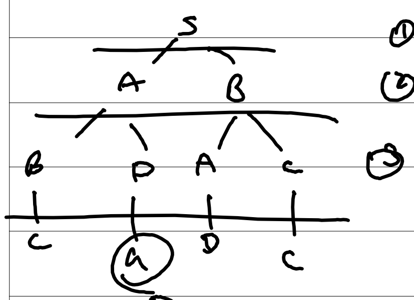
Depth First search.



go down left branch by convention.

② Backup / Backtrack to last decision to be made.

Breadth First search



Build level by level.

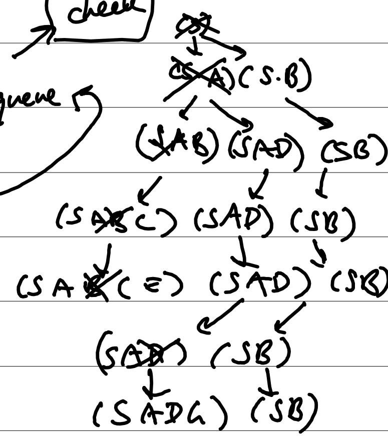
found solution → stop search.

Depth First Algorithm

1. initialise queue.
2. extend first path on queue
3. ENQUEUE

check

DEPTH FIRST: FRONT



Both go off to left of starting in middle.

Breadth First Algorithm.

1. Initiate que.
2. extend first path on que.
3. Enque:

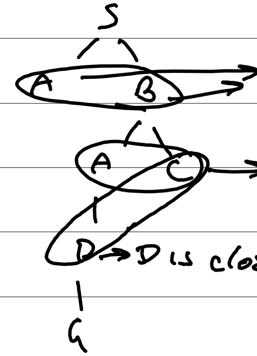
   Breadth first: <u>Back</u>

Issue with Breadth first: extends paths which have previously been extended.

2 (FIX): extend first path on queue: ONLY IF NODE HAS NOT BEEN PREVIOUSLY EXTENDED.

HILL CLIMBING SEARCH (modified depth first)

(no backtracking in this example)



comparing → b is closer to goal.

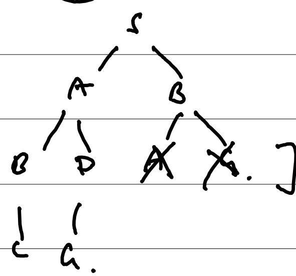
A is closer to goal.

D is closer to goal.

BEAM SEARCH (modified breadth first)

limit # of search states w

eg. w=2



keep the 2 closest to goal.

HILL CLIMB & BEAM ALGORITHM

Initialise que.

↓

extend path on queue if not enqueued previously

↓

enque.

HILL CLIMB: FRONT: SORTED } heuristic information.
BEAM: KEEP w best

Problem w/ Hill Climb

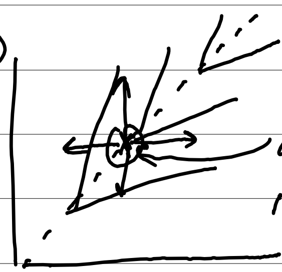
Ⓐ

stuck at local maxima.

Ⓑ

telephone pole problem → can't draw but location of flat.

Ⓒ

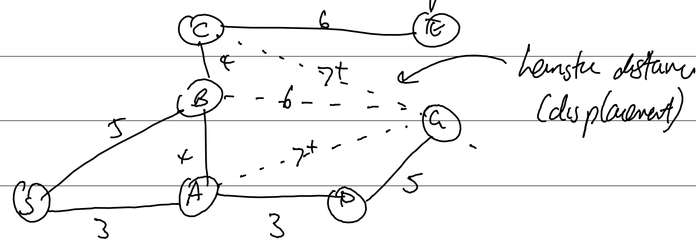


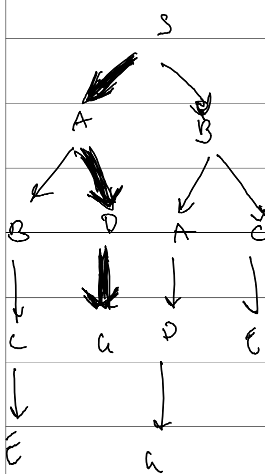
at all directions (except 1), it goes downhill

⇒ might be confused to think you are at the top

Lecture 5: Search: Optimal, Branch and Bound, A*.

how do you find the shortest path?





Agenda

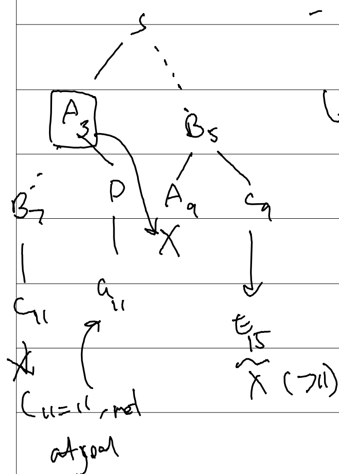
- ☐ Oracle
- ☐ Branch & Bound
- ☐ + extended list / enqueued list
- ☐ + Admissible Heuristic
- ☐ A*

	case 1 extensions at ends.	case 2 middle.
oracle		
branch & bound	835	57
+ ext list ~~enqueued list~~	38	35
+ admissible heuristic "lower bound heuristic"	70	6
A^*	27	6

Oracle

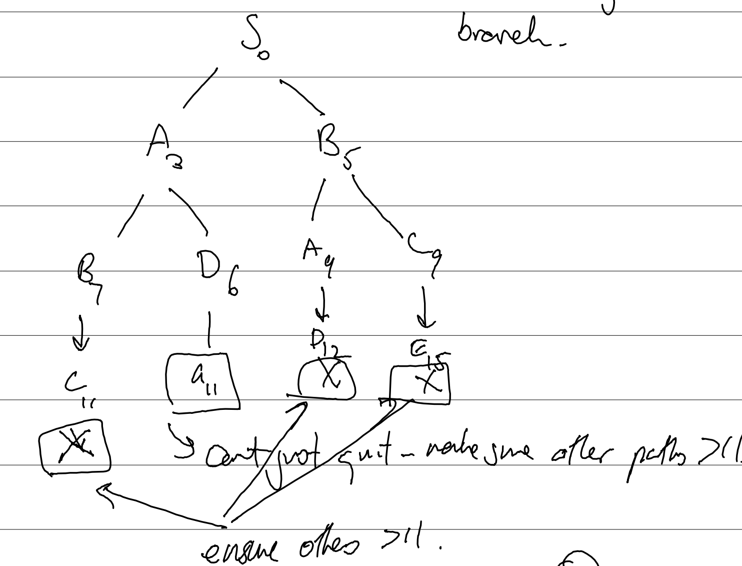
- extend shorter length (cumulated)
- prove your distance is the shortest one!

↳ continuously compare to oracle.



Branch & bound

- extend only shortest branch.



Algorithm A*

(1) Initialize Queue → Test ~~1st~~ shortest path on queue → ☺.

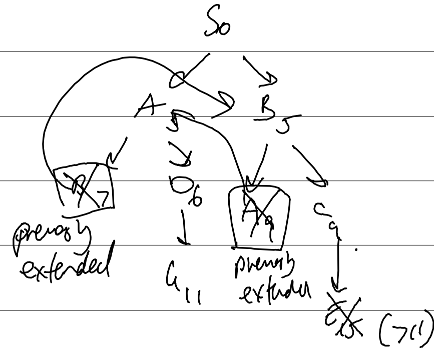
→ extend first path & sort.

By accumulated distance & admissible heuristic.

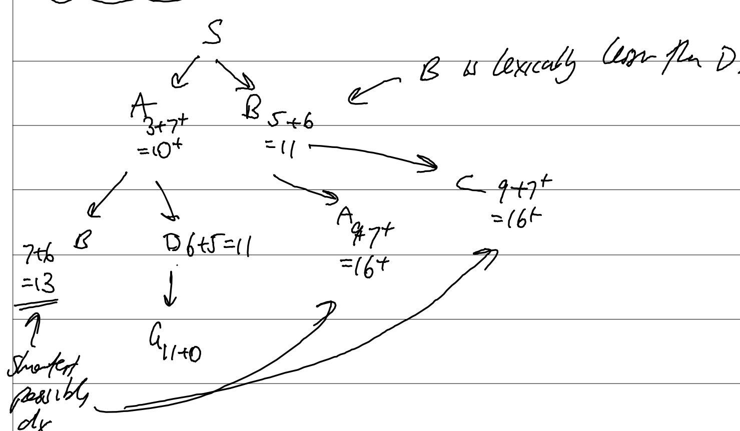
IF NOT ALREADY EXTENDED.

(keep extended list)

Branch & Bound + extended list

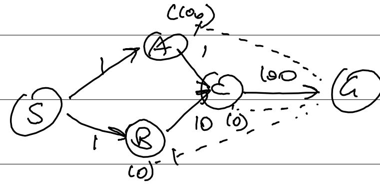


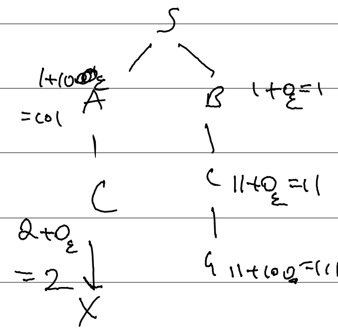
Branch & Bound + Admissible Heuristic.



Admissible heuristic can get you in trouble.

- search not always w/ maps.





no extension
as C has been
extended.

definition:

Admissible $H(x,G) \le D(x,G)$

H → expected distance b/w x & goal

D → actual distance b/w x & Goal.

Consistency $|H(x,G) - H(y,G)| \le D(x,y)$

Lecture 6: Search - Games, Minimax, and Alpha-Beta

Agenda:

- ways to play
- minimax
- $\alpha-\beta$
- progressive deepening
- deep blue.

gives same answer.

(α-β)

- dead horse principle
- marshall arts principle
- anytime algorithms.

insurance policy.

always an answer at any time (progressive deepening)

Ways of playing chess.

1. Analysis, STRATEGY, TACTICS → move. noone knows how to do this.

2. IF-THEN rules: checks to see moves available, ranks them, picks the best move.

3. Look ahead & evaluate: look ahead & determine the best combination of moves.

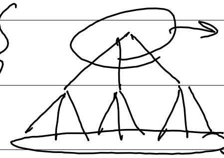
4. British Museum → look at all possible moves for n moves.

5. Look ahead as far as possible.

$$\underset{\text{state value}}{S} = g(f_1, f_2, \dots f_n)$$ ← features of the board.

↳ linear polynomial.

$$= c_1 f_1 + c_2 f_2 + \dots + c_n f_n$$ ← linear scoring polynomial.



using british museum, $\boxed{\approx 10^{120}}$ leaf nodes.

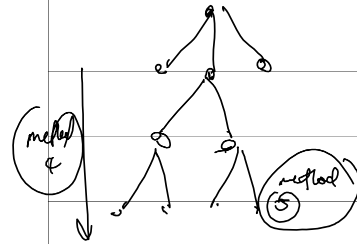
- 10^{80} atoms in universe

$\pi \times 10^7$ seconds a year.

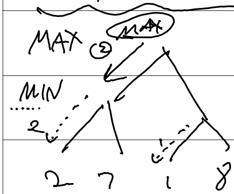
$\times 10^9$ nanoseconds a year.

10^{10} years in the universe.

10^{107} operations if each atom performed nanosecond operations for the life of the universe → impossible!



$d=2,\ b=2$ example of game tree. (MINIMAX)



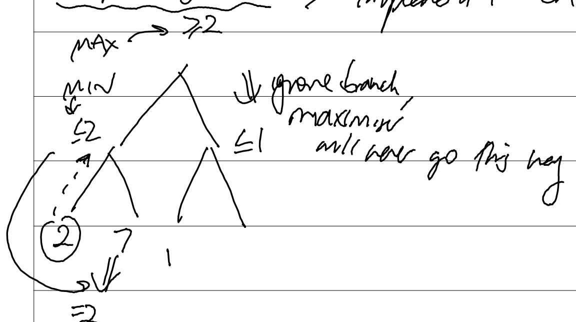
perspective of board from player at top. (MAX).

MIN player wants to minimize the number.

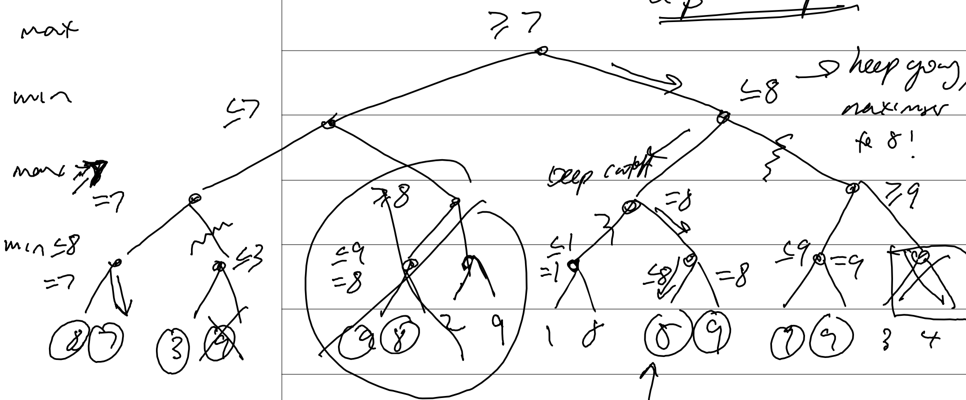
① min player chooses direction.

② max player chooses 1 level up to maximize the directional value.

α-β algorithm → implementation on minimax tree.



α-β example



→ keep going, maximiser could get to 8!

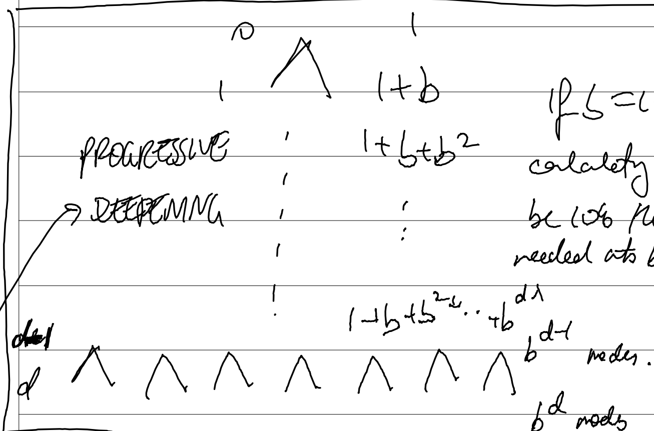
adversarial situation.

$$\boxed{s \simeq 2b^{d/2}}$$ → number of calculations needed.

calculated every one

$\frac{b^{d-1}}{b-1} \simeq b^{d-1}$

insurance policy of every level



If $b = 10$, calculating b^{d-1} will be 10% the computation needed at b^d

$\Delta = \frac{1}{b}$

Deep blue = minimax + α-β + progressive deepening

+ parallel computing + opening book +

end-game + uneven tree development

going in a certain node that

makes big moves

eg. blow out a node that

takes the opponent's queen.

Mega R2 - Basic Search, Optimal Search.

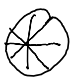
* lexicography / alphabetical order.

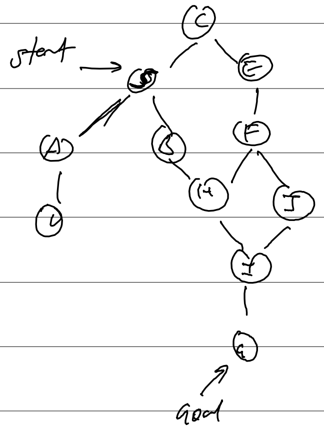
*

do not bite your own tail.

* Read the instructions

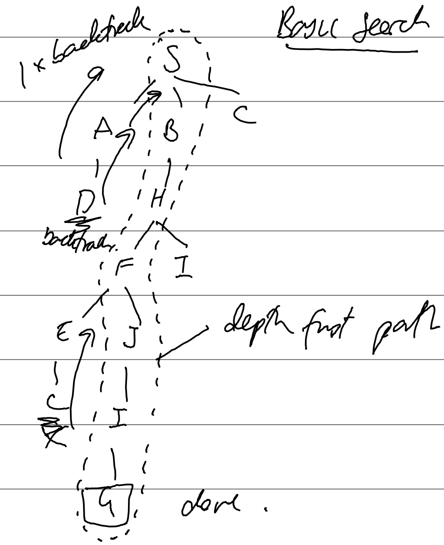
*





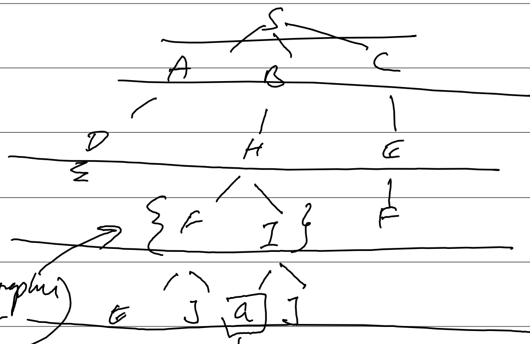
Map

Basic search (depth first)



basic ~~depth~~ breadth first search.

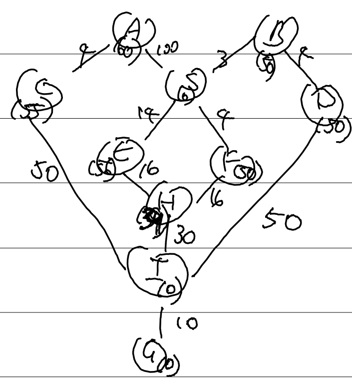
→ finds shortest # jumps.



(lexicographic order)

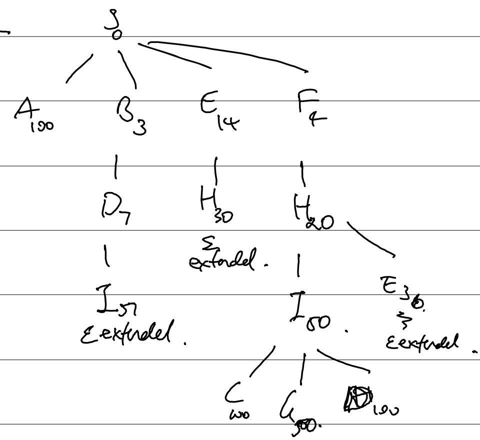
done in implementation.

Deal with heuristic distances.



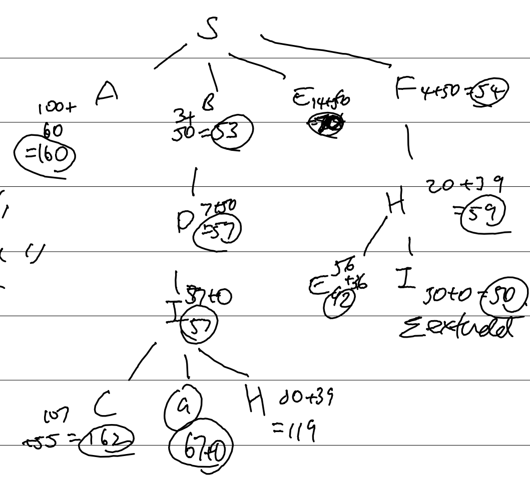
Branch & Bound

extended list: SBFDEHIG.



A* example

S BFDI H



not the right answer, as the heuristic is not consistent

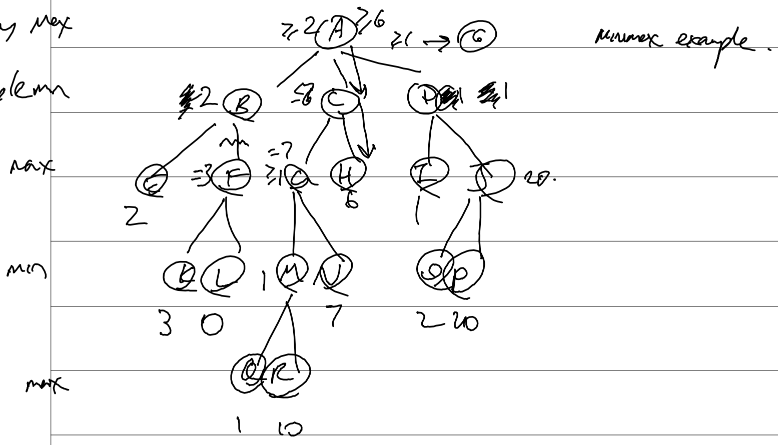
↳ heuristic needs to be an underestimate for it to work

e.g. Branch & Bound is heuristic value = 0.

Mega R3 – Games, Minimax, α–β

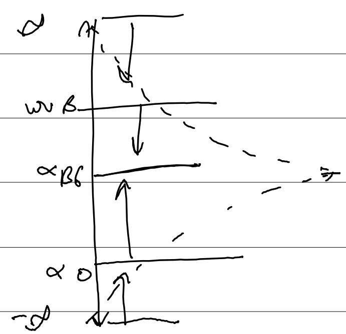
* Nuclear options max

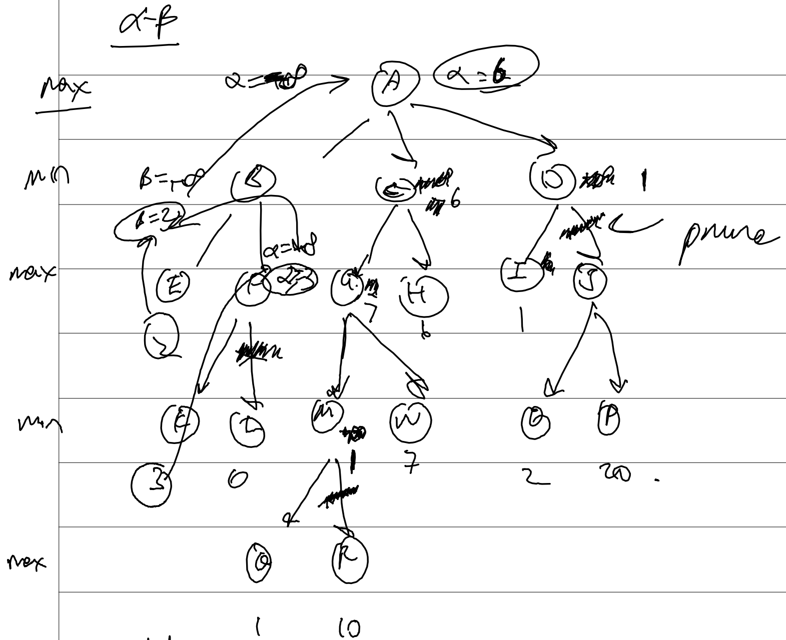
✓ Snow White preplemn



nuclear option! α = maximizing option ($-\infty$)

β = minimizing option. ($+\infty$)





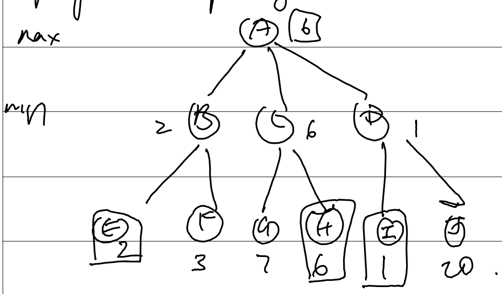
E, K, Q, N, H, I evaluated.

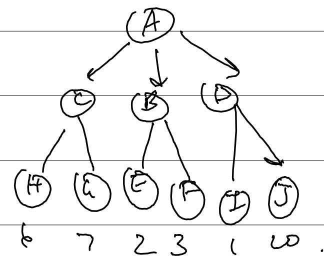
Sanity check if you can skip: if the node contains $-\infty$, will it change anything? if it contained $+\infty$, will it change anything?

If $\alpha \ge \beta$, skip γ

α, β, γ

Progressive deepening



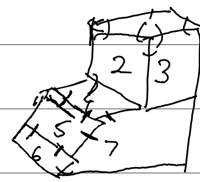


re-order

Lecture 7: Constraints - Interpreting Line drawings.

Agenda

- Let C be a constraint ..
- Guzman (experimentalist) → Huffman (mathematician) → Waltz

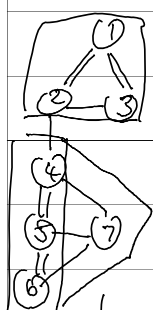


Guzman

how many objects in this line drawing

→ arrow junction ⇒ same object

fork junction ⇒ 3 pairs of objects seemed to belong to same object



1. 1 link = same object
   ⇒ too liberal
2. 2 link theory = 2 links to same object
   ⇒ too conservative.
3. 2 links repeated

→ super-regions are joined as single object.

THREE FACE VERTEX (VERTICES)

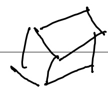
⇒  or  "abduction".

arrow fork.

Huffman's approach

Presumptions.

1. General position (no screw cases)

 rather than  for a cube.

2. Trihedral. - all vertexes formed from 3 planes/faces
3. 4 kinds of lines/labels.

① concave $-$ ② convex $+$

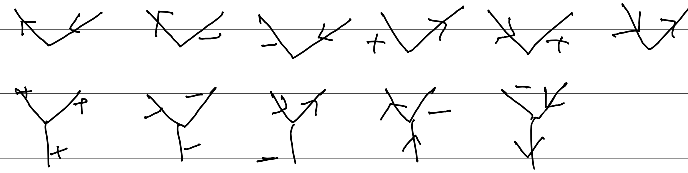
③ ④ Boundary $\rightarrow$ $\leftarrow$ object.

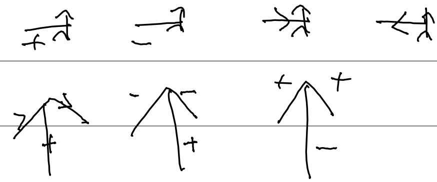
(cracks & shadows (left out.)

Vocabulary

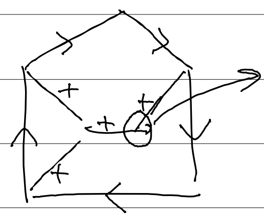
Vertex	Junction
Edge	Lines.

Line label arrangement on junctions.





18 ways lines can come together around junctions

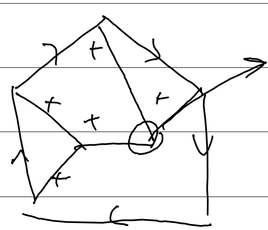


L-style jn w/ +s on 2 sides.

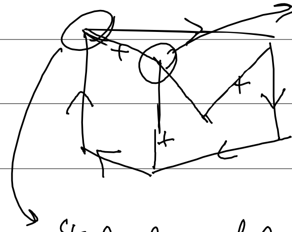
⇓

doesn't pass a condition for construction!

not a possible L-style junction.



satisfies constraints, but cannot be constructed in our world.



can't label this

4 faced junction ⇒ cannot be labelled.

⇓

but can be made!

Waltz

(+) cracks, shadows, non-trihedral vertices, light.

4 labels ⟶ 50+ labels.

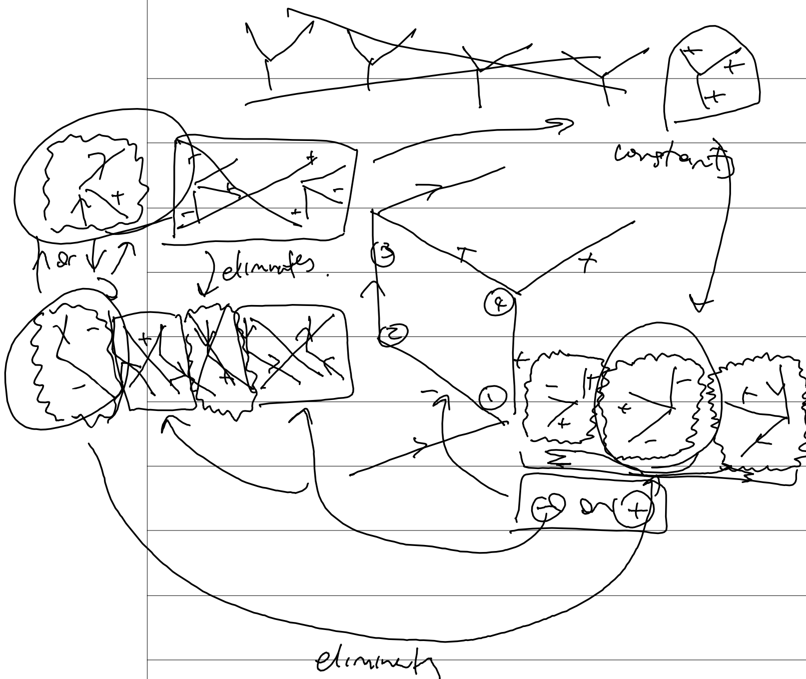
⇒ thousands of possible junctions.

↳ how to label?

↳ depth-first search program.

↳ cannot be done ⇒ too computationally expensive!

need a different search method!



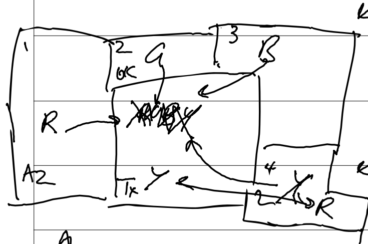
junctions place constraints on one another.

Lecture 8: Constraints: Search, Domain Reduction

any reason this can't be R? B? G? Y?

↳ no constraints propagated.

RGBY	RGBY
RGBY	RGBY



← colouring squares

no adjacent states w/ same color.

← chose Y because don't want to overuse colours.

↑ marshal Arts principles

↳ local constraints as we go to prevent downstream issues.

Vocabulary

states → variable V: something that can have assignment.

color → value x: something that can be an assignment.

remaining colors → domain D: bag of values

no states along borders have same colour → constraint C: limit on variable values domain reduction algorithm

FOR EACH DEPTH FIRST SEARCH ASSIGNMENT:

FOR EACH VARIABLE V_i CONSIDERED

FOR EACH x_i in D_i

FOR EACH CONSTRAINT $C(x_i, x_j)$ where $x_j \in D_j$

IF $\nexists\, x_j \ni C(x_i, x_j)$ SATISFIED

REMOVE x_i FROM D_i

IF D_i EMPTY:

BACK UP

CONSIDER:

① Nothing → doesn't work (no satisfaction of constraints).

② Assignment → take forever.

③ - neighbors → 9/39 dead ends, but faster.

⑤ - propagate checking through V with reduced D (0 dead ends)

④ D reduced to 1 value. - least constraints checked.

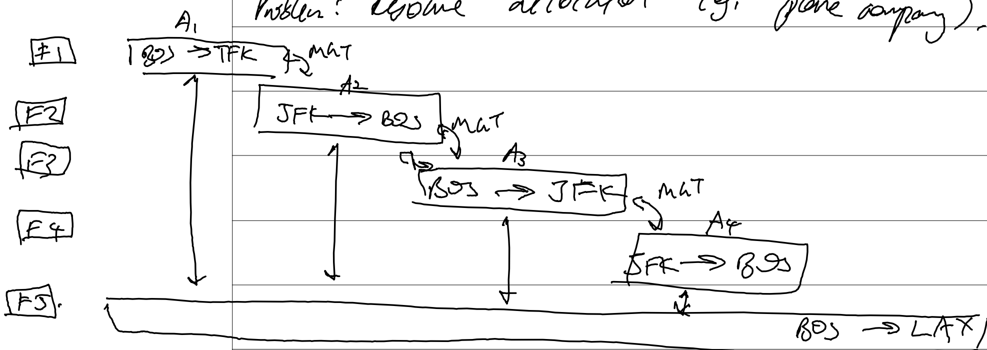
⑥ - everything → checks all other states → too slow

• Arrangement of filling the variables with values

→ start w/ MOST constrained first

▷ faster (vs. starting w/ least constraints)

Problem: Resource allocation (e.g. plane company).



C: no physical aircraft (i) can fly on 2 flights at same time.

no back-to-back flights w/ same plane due to minimum ground time.

not same city → plane taking time to fly back.

* To figure out minimum # of resources, do under/over approach to find termination/completion and work to reduce the boundaries → this is the ~~A~~ANYTIME ALGORITHM approach

Rumplestiltskin principle

Goldilocks principle

power of correlation

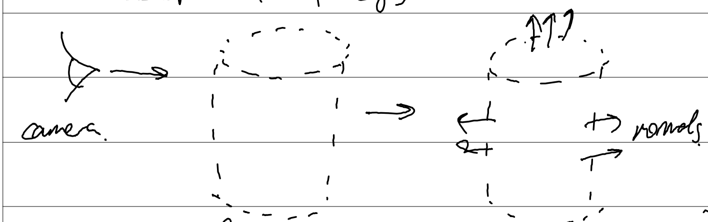
Lecture 9: Constraints: Visual Object Recognition

Agenda

- ☐ Concept according to Marr
- ☐ Ullman's Alignment Theory
- ☐ Ullman's intermediate features theory
- ☐ Correlation

304

(1) Marr's Concept — look for edges

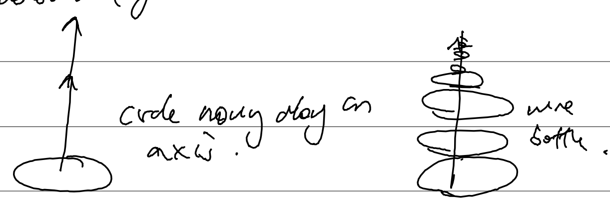


(1) primal sketch

$2\frac{1}{2}$ D sketch

(2) (camera-centre)

(3) Generalised cylinders



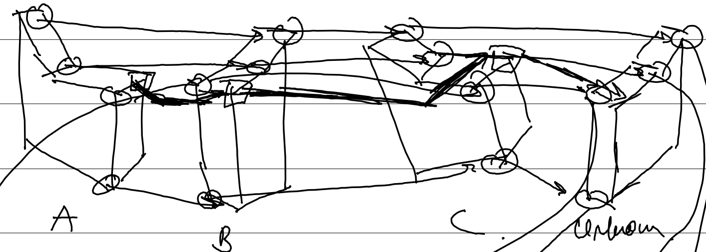
⟹ Recognition.

None could make this work.

Ullman's Alignment Theory

* take 3 pictures of object, can reproduce any object view.
    * not true ⇒ only possible to produce a view in orthographic projection.

eg.



is Unknown from the same object as A, B & C?

image/object recognition application!

use fixed pt to predict point on unknown.
⇒ if non-concordant, cannot recognise object.

$x_i = \alpha x_A + \beta x_B + \gamma x_C + \tau$

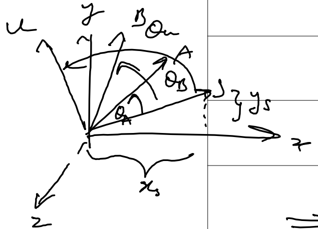
$x_i = \alpha x_A + \beta x_B + \gamma x_C + \tau$

$x_i = \alpha x_A + \beta x_B + \gamma x_C + \tau$

$x_i = \alpha x_A + \beta x_B + \gamma x_C + \tau$

$\alpha, \beta, \gamma, \tau$ same for all

relationships b/w 4th object & library are linearly related.



constants

$$x_A = x_s \cos\theta_A - y_s \sin\theta_A$$
$$x_B = x_s \cos\theta_B - y_s \sin\theta_B$$
$$x_u = x_s \cos\theta_u - y_s \sin\theta_u$$

$$\Rightarrow x_u = \alpha x_A + \beta x_B$$

linear combination of other positions.

doesn't work on natural world objects

$\Downarrow$

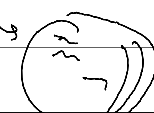
no 'exact points' in real life to choose from.

objects aren't consistent in real life.

<u>Ullman's Intermediate Features Theory</u>

in "face" library

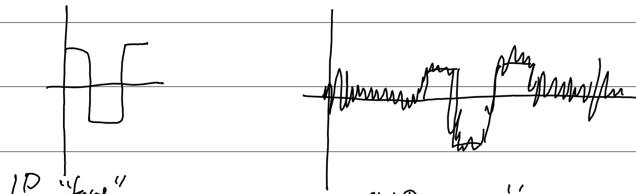
"new" face



look for <u>features</u>

not too big, but not too small

$\Rightarrow$ "Goldilocks principle"



1D "face" "1D image"

how to find?

$$\max_{X,Y} \int_x f(x,y)\, g(x-X, y-Y)$$

face

integral largest when the X, Y offset overlaps the image.

signal at offset

works through the noise.

Issues

turning upside down → lose correlation of eyes → nose!

→ hard to do alignment

stretching can affect correlation

↳ in vertical dimension more than horizontal.

Lecture 10: Introduction to learning, Nearest neighbors.

Agenda:

I The lay of the learning land.

II Nearest neighbours

- cones
- cells
- information
- arm control.

III Problems.

Learning —

HUMAN-LIKE

BULLDOZER COMPUTING.

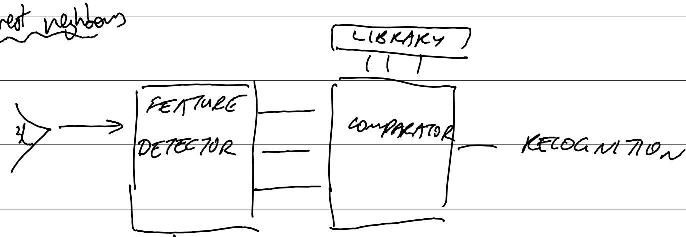
1 CONSTRAINTS

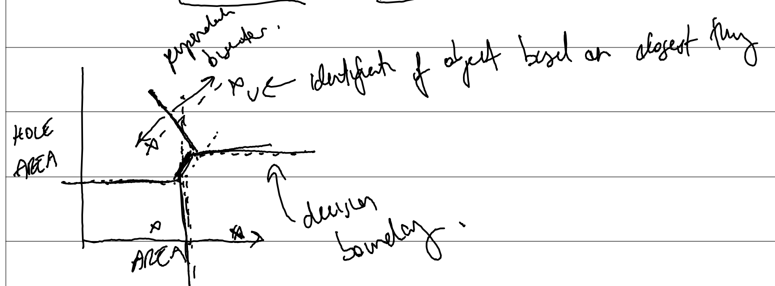
- ONE SHOT
- EXPLANATION BASED LEARNING.

2 Regularity

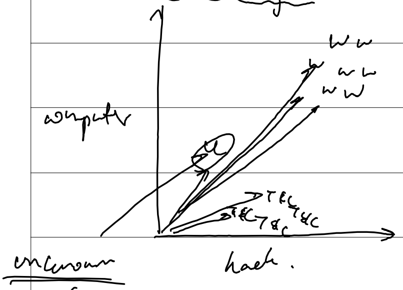
- PATTERN RECOGNITION
  - nearest neighbors.
- MIMIC BIOLOGY
  - neural nets
- THEORY
  - Boosting

Nearest neighbors





Articles from magazine - Classification of magazines based on its articles

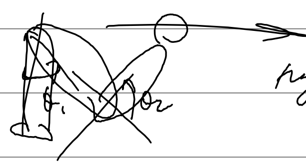


measure cosine of angle b/w vectors.

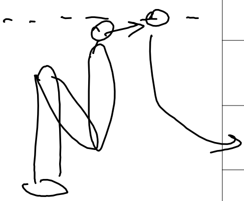
$$\cos\theta = \frac{\sum U_i \times A_i}{\|U\| \, \|A\|}$$ = dot product.

$U_i = A_i$ (angles) $\Rightarrow \cos\theta = \cos 0 = 1$

Robotic arm control:



move ball at speed, velocity of trajectory.



Arm parameters.

θ_1	θ_2	$\dot{\theta}_1$	$\dot{\theta}_2$	$\ddot{\theta}_1$	$\ddot{\theta}_2$	τ_1	τ_2
		velocity		acceleration			

Lookup it up.

Find τ for movement.

⇒ originally computers too slow for computers.

↳ neglected when computers became fast enough.

- each practice/attempt gets recorded into its memory in table

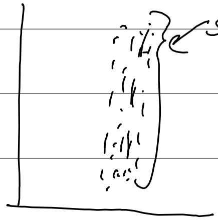
---

100 bytes/joint
100 joints
100 segments for a pitch.
100 pitches thrown a day
100 days pitching/year.
100 years a career.

$\}$ 10^{12} bytes.

10^{10} neurons in brain
10^{11} neurons in cerebellum.
10^{5} synapses/neuron
$\Downarrow$
☆ 10^{16} synapses in cerebellum.

SPREAD (NON UNIFORMITY PROBLEM.

① 

spare of samples

→ x vals useful, y vals not

⇓

can normalise data.

$\boxed{x} \parallel \sigma_x^2 = \frac{1}{n} \sum (x-\bar{x})^2$

$x' = \frac{x}{\sigma_x}$ ⇝ variance of this $= 1$.

② WHAT MATTERS?

→ e.g. y value doesn't matter at all to outcome.

③ CAKE BUT NO FLOUR.

→ need data that answers the question!

Problem solving $\propto$ sleep?

36 h sleep deprivation

↳ loss of problem solving capacity.

72 hrs ⇒ 30% of original capacity

⇒ sleep loss accumulates.

↳ naps help

↳ caffeine helps.

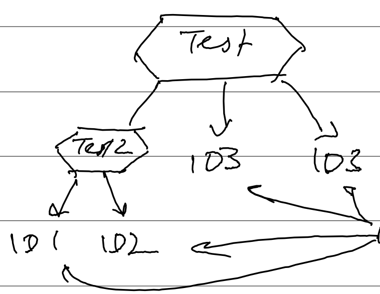
24h no sleep → problem solving equivalent to being drunk

Lecture 11 - Learning: Identification trees, Disorder.

Agenda.

- ☐ Identification trees.
- ☐ measure disorder
- ☐ Rules.
- ☐ Simplification of rules.

- categorical data (not numeric) cannot be used in nearest neighbours.
- some characteristics of data do matter (only sometimes) & some don't matter.
- cost of retrieving data (expensive relative to other tests).



Test tree

How to arrange test in trees?

⇓

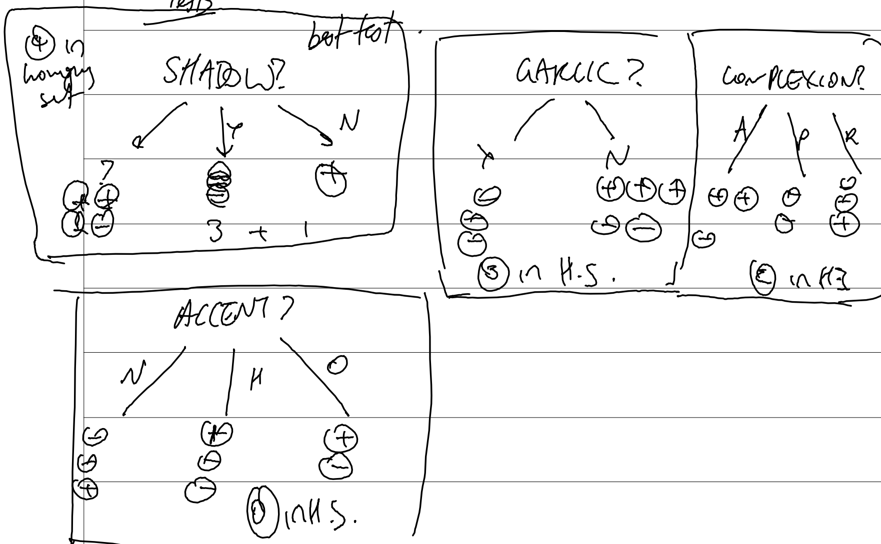
identification tree

NOT decision tree

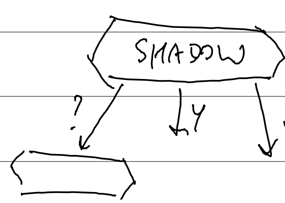
Characteristics of a good identification tree.

- small tree (less tests (lays)) w/ homogeneous leafs.
  1. cost
  2. Occam's razor → small tree = simple explanation

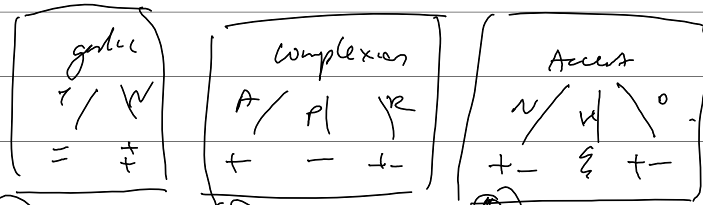
Tests



① look for test & make homogeneous sets.

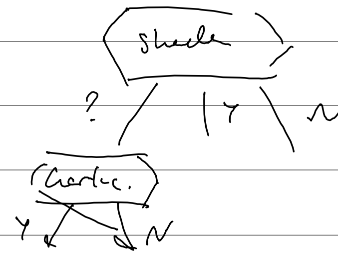


Step 2: repeat scoring of tests.



(4) → best set. (2) (0)

⇓



<u>However</u> – large datasets do not produce homogeneous sets

↓

ask someone who measures disorder in sets.

↳ Physics ⇒ <u>entropy</u>

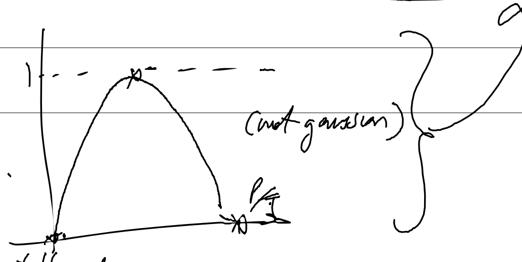
↳ information theory

$$\boxed{D(\text{set}) = -\frac{P}{T}\log_2\left(\frac{P}{T}\right) - \frac{N}{T}\log_2\left(\frac{N}{T}\right)}$$

$\frac{P}{T} = \frac{1}{2} \Rightarrow D_{set} = -\frac{1}{2}\log\left(\frac{1}{2}\right) \times 2$

$= \log 2 = 1$

$\frac{P}{T} = 1 \Rightarrow -\log_2(1) - \underbrace{0\log_2(0)}_{\text{l'Hopital's rule}}^{=0}$



How do we measure the QUALITY of a test?

$$Q_{TEST} = \sum_{\text{sets produced}} D_{set} \times \underbrace{\left\{ \frac{\text{\# of samples in set}}{\text{\# of samples handled by test}} \right\}}_{\text{normalizes by \# of samples classified}}$$

measurement of disorder in large datasets

SHADOW?

? $\left(\frac{4}{8}\right)$ Y $\left(\frac{3}{8}\right)$ N $\left(\frac{2}{8}\right)$

$\begin{matrix} + + \\ - - \end{matrix}$ $\begin{matrix} - \\ - \\ - \end{matrix}$ $\begin{matrix} + \\ + \end{matrix}$

$\left(1 \times \frac{4}{8} = \frac{1}{2}\right) \left(\frac{3}{8} \times 0\right) \left(\frac{2}{8} \times 0\right)$

$\Rightarrow \sum = \frac{1}{2} + 0 + 0 = \frac{1}{2} \Rightarrow$ good test

GARLIC

Y $\frac{3}{8}$ $\frac{5}{8}$ N

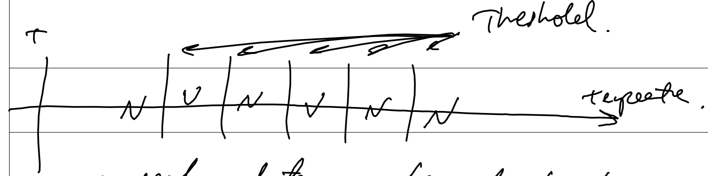
$\begin{matrix} - \\ - \\ - \end{matrix}$ $\begin{matrix} + + + \\ - - \end{matrix}$

$\frac{3}{8} \times 0$ $\frac{5}{8} \times \approx 0.9$

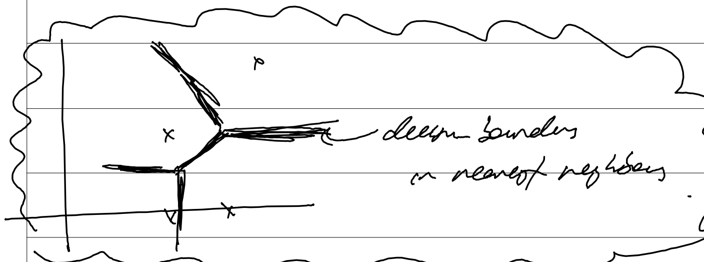
$\Rightarrow \sum \doteq 0.6$

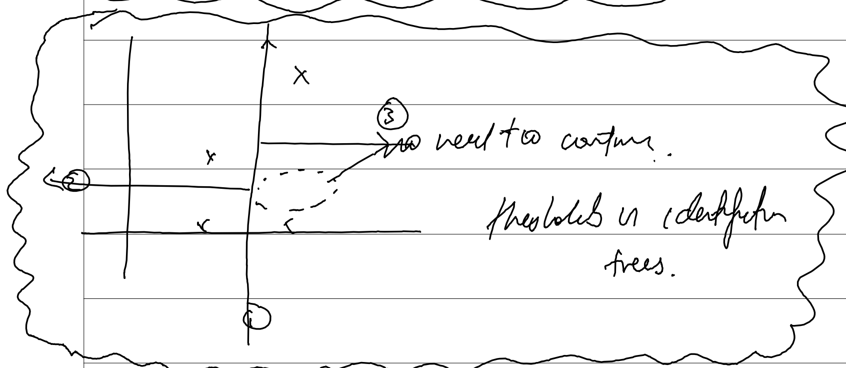
Complexion $\rightarrow \approx 0.7$

Accent $\rightarrow \approx 0.95 \Rightarrow$ bad test



numeral data ⇒ how do I place the threshold to classify N(normal) & V(vampire)?

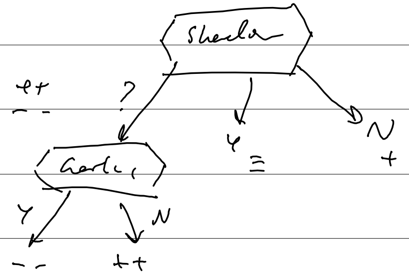




Identification trees can be converted into a set of rules.

→ Go down branch til you hit a leaf.

eg If shadow=? & Garlic=Y → then ⊖



Simplify rules ⇒ Garlic = Y ⇒ not vampire under all conditions, irrespective of shadow state.

simple rule

Lecture 12a - Neural Nets

Agenda

2 models of problem solving

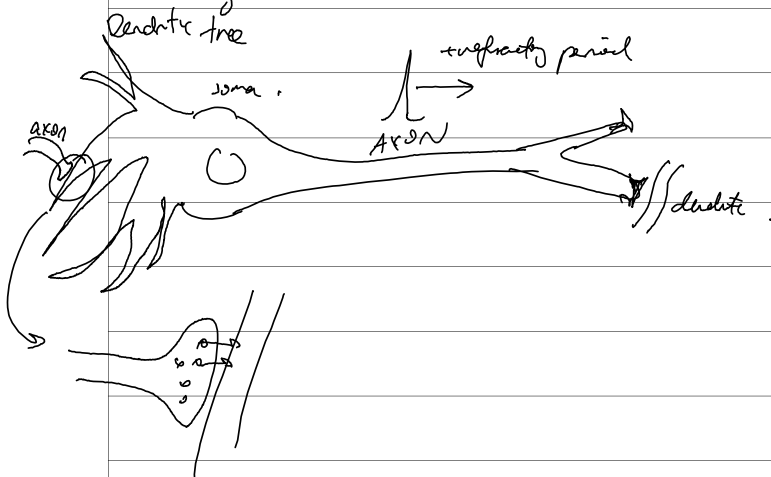
2 models of learning

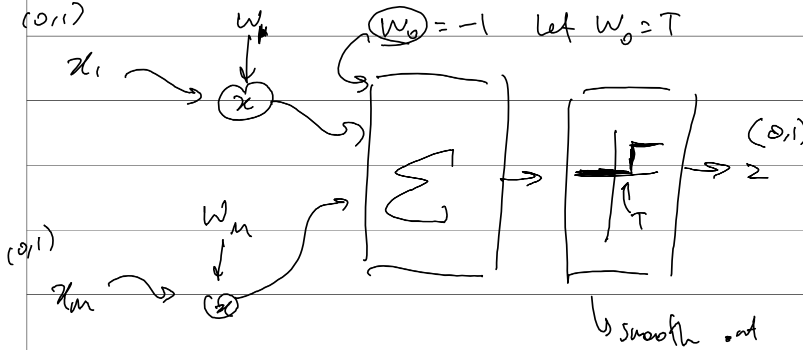
Basic ... biologically inspired.

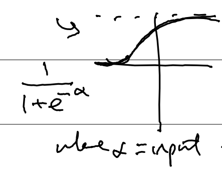
12 neural nets.

- naive neurobiology
- hill climbing
- threshold func
- sigmoid func.

* All Great ideas are simple
* 2 tricks → observation → great
* Recycle principle





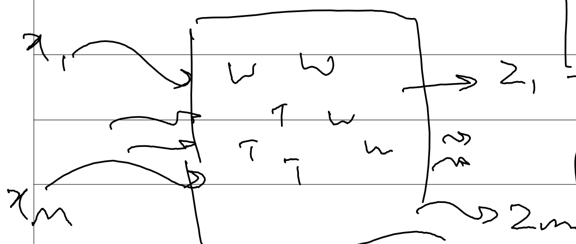


1. All or none.
2. Cumulative influence.
3. Synaptic weight

Refractory period.

axonal bifurcation.

time patterns



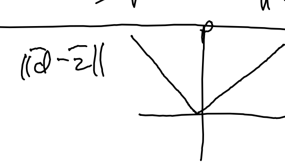
$\boxed{\bar{z} = f(\bar{x}, \bar{w}, \bar{T})}$

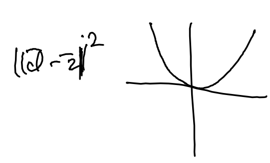
adjust

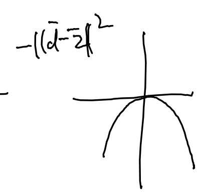
$\boxed{\bar{d} = g(\bar{x})}$

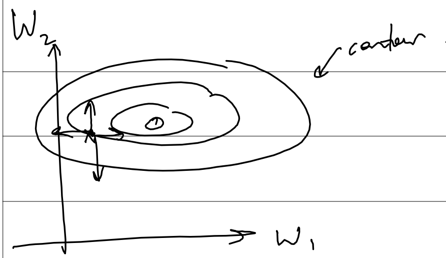
desired.

$\boxed{\text{Performance} \rightarrow P = -\|\bar{d} - \bar{z}\|^2.}$









$\frac{\partial P}{\partial w_1}$ $\frac{\partial P}{\partial w_2}$

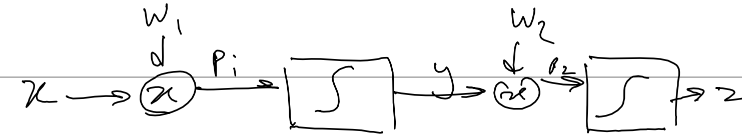
improvement w/ moving in directions w_1 & w_2.

$$\Delta \bar{w} = \Big(\underbrace{\frac{\partial P}{\partial w_1} i + \frac{\partial P}{\partial w_2} j}_{\text{move up the gradient.}}\Big) \underbrace{r}_{\text{rate constant}}$$

why can't we use gradient ascent/descent?

↳ this is not a continuous/linear function.

$\Sigma' = f'(\bar{x}, \bar{w})$ when you use a $w_0 = -1$ weight



$$p = \frac{1}{2}(d - z)^2$$

desired actual.

$$\frac{\partial p}{\partial w_2} = \frac{\partial p}{\partial z} \times \frac{\partial z}{\partial w_2} \rightarrow \frac{\partial z}{\partial p_2} \times \frac{\partial p_2}{\partial w_2}$$

$$\frac{\partial p}{\partial w_1} = \frac{\partial p}{\partial z} \times \frac{\partial z}{\partial p_2} \times \frac{\partial p_2}{\partial y} \times \frac{\partial y}{\partial p_1} \times \frac{\partial p_1}{\partial w_1}$$

$p_2 = y \times w_2$

$$\underbrace{\frac{\partial p_2}{\partial w_2}}_{y} \; \underbrace{\frac{\partial z}{\partial p_2}}_{z(1-z)} \; \underbrace{\frac{\partial p}{\partial z}}_{(d-z)} \qquad z = \frac{1}{1+e^{-\alpha}} \Rightarrow$$

same

$$\frac{\partial p_1}{\partial w_1} \times \frac{\partial y}{\partial p_1} \times \frac{\partial p_2}{\partial y} \times \boxed{\frac{\partial z}{\partial p_2} \times \frac{\partial p}{\partial z}}$$

re-use computation to reduce exponential blow-up.

1. Linear in depth
2. WRT width $= w^2$

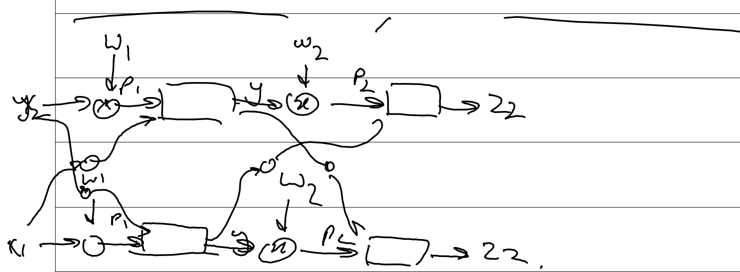
$$z = \frac{1}{1+e^{-\alpha}} \quad \frac{d}{d\alpha}(1+e^{-\alpha})^{-1}$$

$$= -(1+e^{-\alpha})^{-2} \; e^{-\alpha} \times -1$$

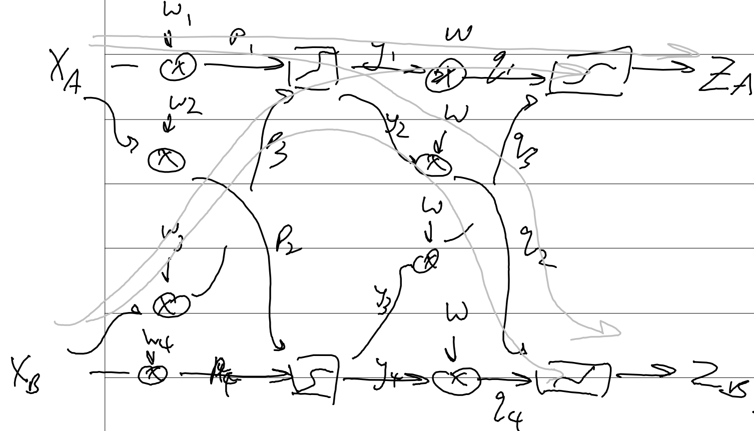
$$= \frac{1+e^{-\alpha} - 1}{1+e^{-\alpha}} \times \frac{1}{1+e^{-\alpha}} = \left(\frac{1+e^{-\alpha}}{1+e^{-\alpha}} - \overbrace{\frac{1}{1+e^{-\alpha}}}^{\beta}\right) \times \overbrace{\frac{1}{1+e^{-\alpha}}}^{\beta}$$

$$= (1-\beta)(\beta)$$

rate constant should vary depending on how close you are getting.



Lecture 12b: Deep Neural Nets



$$\frac{\partial P}{\partial w_1} = \frac{\partial P}{\partial Z_A}\frac{\partial Z_A}{\partial q_1}\frac{\partial q_1}{\partial y_1}\frac{\partial y_1}{\partial p_1}\frac{\partial p_1}{\partial w_1} + \frac{\partial P}{\partial Z_B}\frac{\partial Z_B}{\partial q_2}\frac{\partial q_2}{\partial y_2}\frac{\partial y_2}{\partial p_1}\frac{\partial p_1}{\partial w_1}$$

what's computed is computed and need not be recomputed!

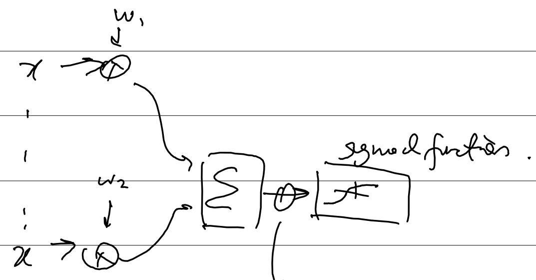
$$\frac{\partial P}{\partial w_3} = \frac{\partial P}{\partial Z_A}\frac{\partial Z_A}{\partial q_1}\frac{\partial q_1}{\partial y_1}\frac{\partial y_1}{\partial p_3}\frac{\partial p_3}{\partial w_3} + \frac{\partial P}{\partial Z_B}\frac{\partial Z_B}{\partial q_2}\frac{\partial q_2}{\partial y_2}\frac{\partial y_2}{\partial p_3}\frac{\partial p_3}{\partial w_3}$$

Agenda

- ☐ models of problem solving
- ☐ models of learning
- ☐ "Deep" neural nets.

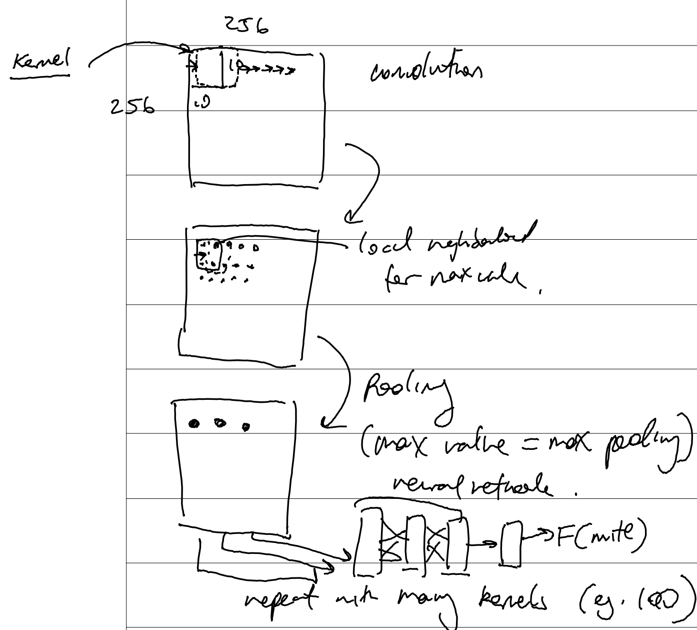
old dogs   pooling   auto coding   fooling

convolution   sigmoid/logistic output   dropout   SOFTMAX

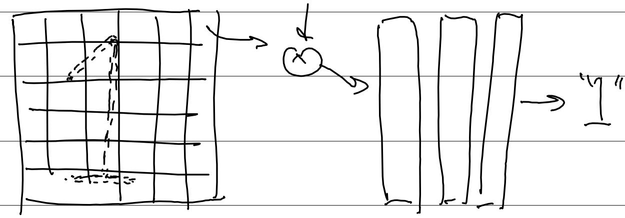


$\sum x_i w_i \Rightarrow$ dot product.

Convolutional neural net

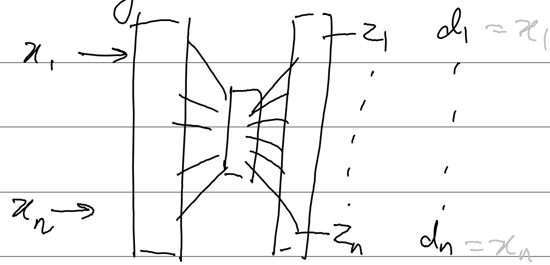


old neural network methods.



Autocoding

do it before training on own data/. expected output

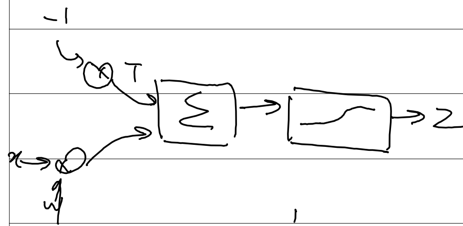


informs you of what happens inside the neural net.

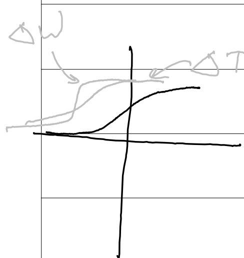
cheetah, zebra, giraffe } shadow → encoding → output
↳ compare to desired.

middle layer / hidden layers
dont generalise the inputs,
they encode generalise.

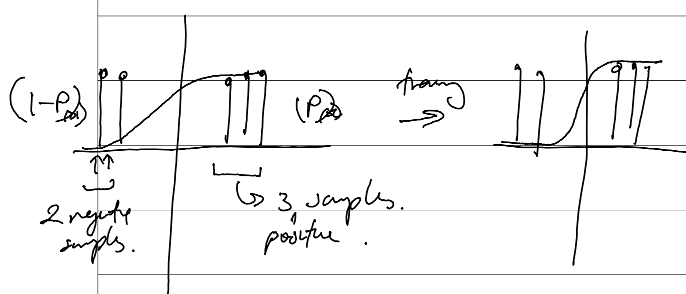
mystery as unclear how this encoding works.



$$Z = \frac{1}{1+e^{-wx+T}}$$



w & T adjusted based on samples.



output: $\rightarrow F(c_1)$

$\rightarrow F(c_2)$

$\rightarrow F(c_m)$

value for likelihood of class 1.

$$\boxed{P(c_1) = \frac{F(c_1)}{\sum_i F(c_i)}}$$

can produce probability of each class by dividing by a normalizing function.

$\Downarrow$

SOFTMAX

Dropout

→ can get stuck on local maxima.

→ on any iteration, flip a coin for each neuron,

if tails → neuron dies ("drop out")

↳ repeat for every set

Wide neural nets

↳ more neurons → local maxima turned into saddle points

↳ network crawls itself to find a maximum

!!

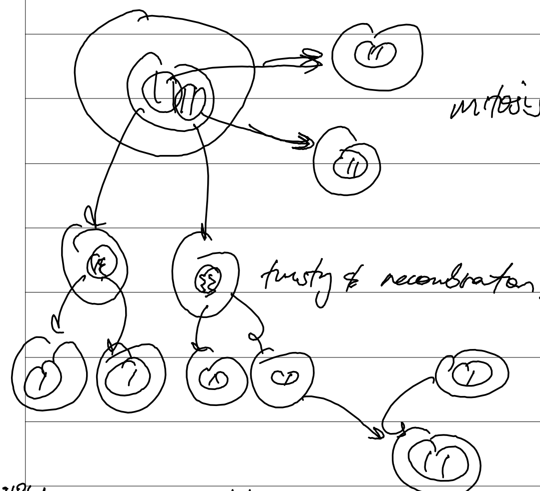
we can train a wide network then knock out neurons, it will still work,

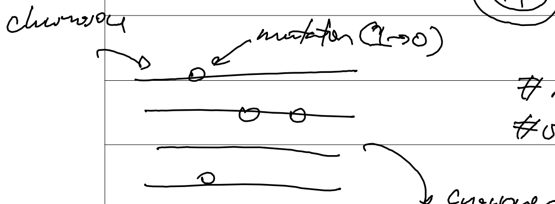
But if we train with few neurons initially, we may never reach predictive capacity.

Lecture 13 - Genetic Algorithms

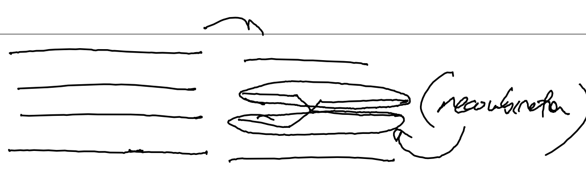
Agenda

- naive evolution
- ? mimicking.
- Choices about
- Fitness space
- examples.

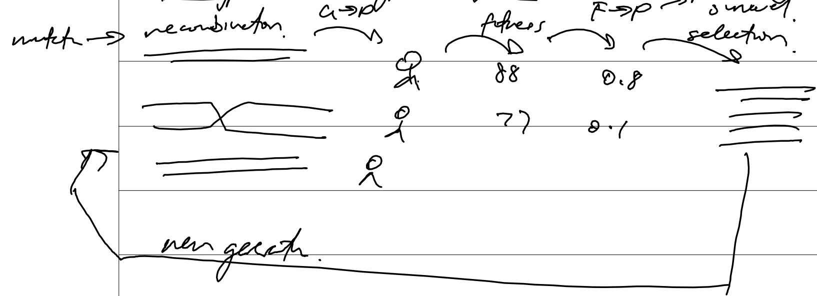




mutates
slip through w/o mutate.



Genotype → Phenotype mapping

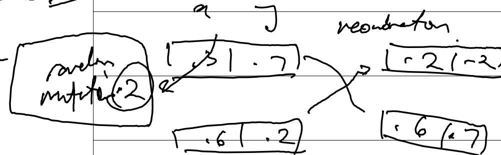


#1

$$P_i = \frac{f_i}{\sum_i f_i}$$

$f(x,y) = \sin^2(wx) + \sin^2(wy) + e^{+\frac{(x+y)}{\sigma}}$

SIZE OF MUTATION = STEP SIZE



#2

Rank-Space method

don't care about fitness, but rank probability of fitness

$P_1 = P_c$

$P_2 = (1-P_c)P_c$

$P_3 = (1-P_c)^2 P_c$

$P_{n-1} = (1-P_c)^{n-2} P_c$

$P_n = (1-P_c)^{n-1}$

large step size to skip local max.

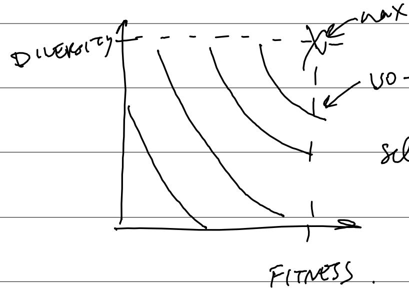
⇓

clamp down

⇓

simulated annealing = big step size, then drop the step size.

#3 select based on ① Fitness & ② Diversity from already selected population.



select based on highest Fitness / Diversity rank.

↳ high diversity causes spread, but cannot centre in on best result.

Practical applications of genetic algorithm,

#1 S_1 S_2 S_3 . . .

S_1' S_2' S_3' . . .

⇓

S_1 S_2 S_3' S_x' . . .

chromosome = steps of plan.

#2 IF X Y Then ~ IF L M Then ~ IF X' M then ~

mutation & crossover.

Genetic algorithms have a rich solution space, but are not necessarily the most useful.

Lecture 14: Learning - Sparse spaces, Phonology

Basic methods → $\boxed{\text{naive mimickry}}$ → $\boxed{\text{focus on problem}}$ → focus on theory → FLOS.

- ☐ what if god were an engineer.
- ☐ Yip-Sussman machine / learner
- ☐ reflections

• neural nets
• genetic algo

phonology

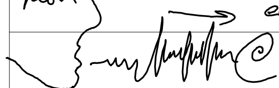
dog → dogs "z"
cat → cats "s"
} phonological rules.

McGurk effect (vision's effect on interpretation)

person

acoustic wave.

ear



→ sequence of distinctive feature vectors.] 14 features.

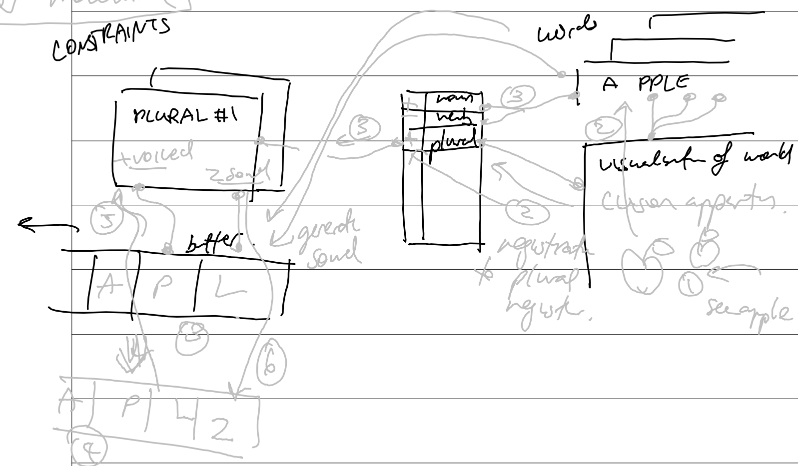
5 pieces of meat

≈ 2^{14} possible sounds → meaning

	AE	P	L	Z
SYLLABIC → can it form a core of a syllable.	+	−	−	−
VOICED	+	−	+	+
CONTINUENT – open vocal apparatus.	+	−	+	+
STRIDENT – tongue to form jet of air	−	−	−	+

MACHINE: for word generator using phonological rules

$\boxed{\text{Propagator machine}}$



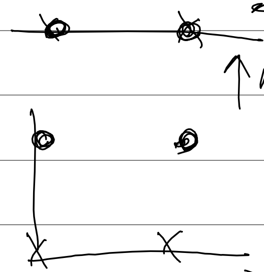
	K	A	T	Z		D	O	G	Z
SYLABIC	−	+	+	−				+	−
VOICED	−	+	−	−				+	+
CONTINUENT	−	+	−	+				−	+
STRIDENT	−	−	−	+		−	−	−	+

$\boxed{\text{14 features}}$

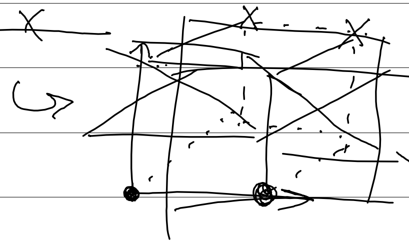
when do we use z sound?

1. collect the +ve & −ve examples
2. Pick + seed (example)
3. Generalise until we admit/match a negative example → Quit

← cannot separate ⊚ from X.



↑ Projection of 2D space.



3 axis of purity in hyperplanes.

Higher dimension hyperspace makes it easier to separate. ←
$$\left\{\begin{array}{l}(\frac{1}{2}, 0, 0)\\ (0, \frac{1}{2}, 0)\\ (\frac{1}{2}, \frac{1}{2}, 0)\end{array}\right.$$

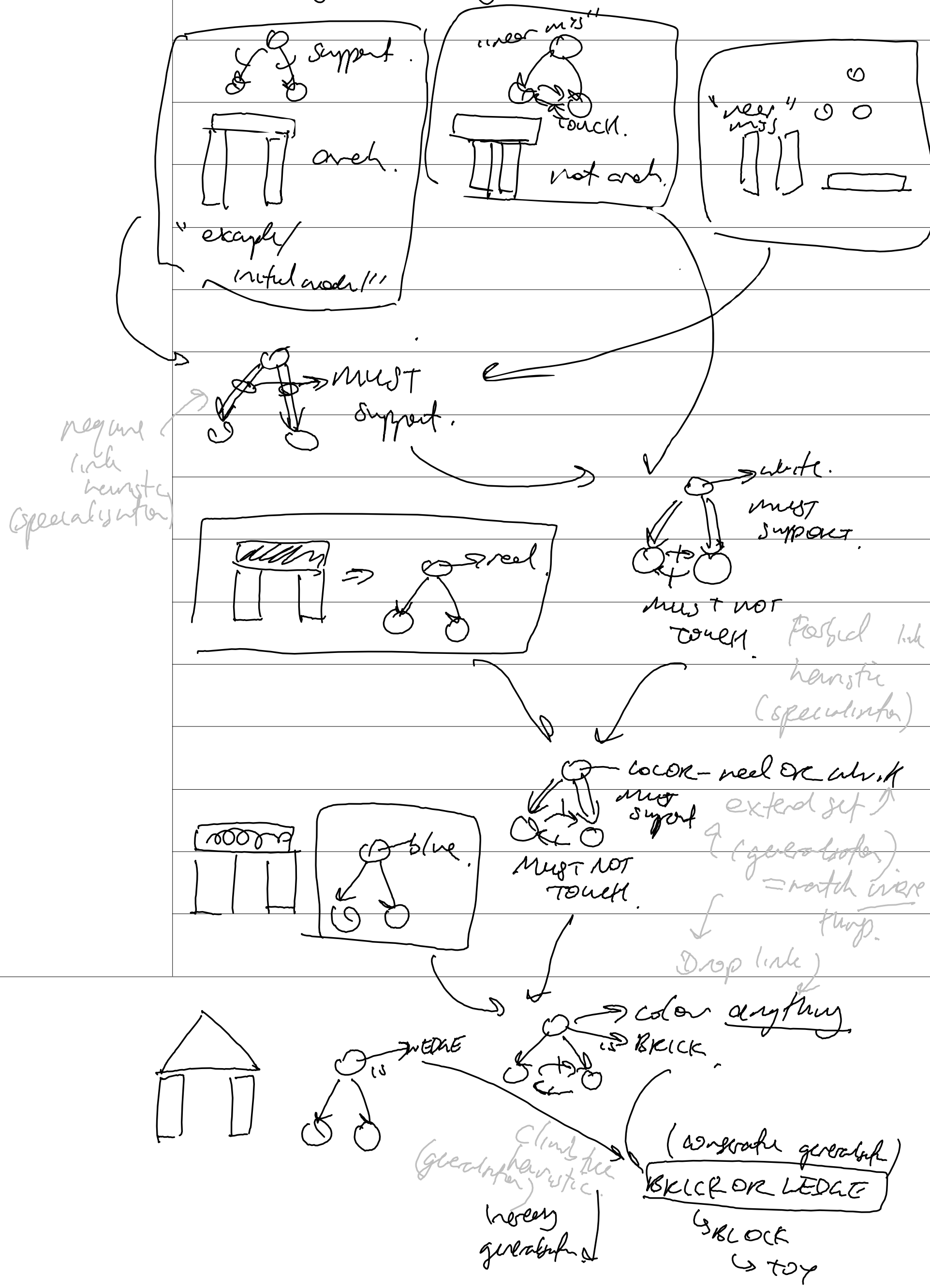
<u>Marr's catechism.</u>

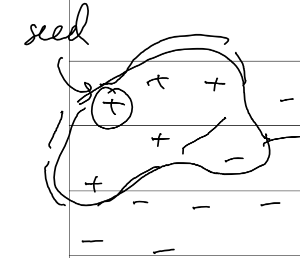
1. problem
2. representation
3. Approach, method
4. Pick mechanism, algorithm — "mechanism" envy
5. experiment

} explicit. expose constraint; locality criteria

Lecture 15: Learning: Near Misses, Felicity Conditions

Generating & updating models.



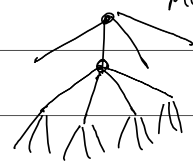


pick heuristic to begin the seed definition

→ initial class

↳ may contain a negative.

Pick seed.

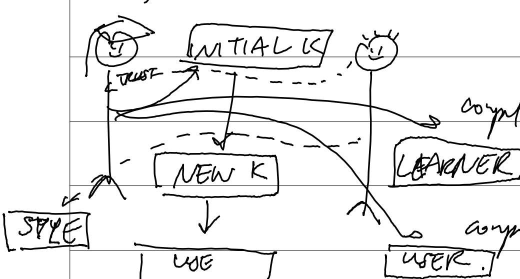


Beam search.

examples → result in generalisation.

near miss → result in specialisation.

Felicity conditions



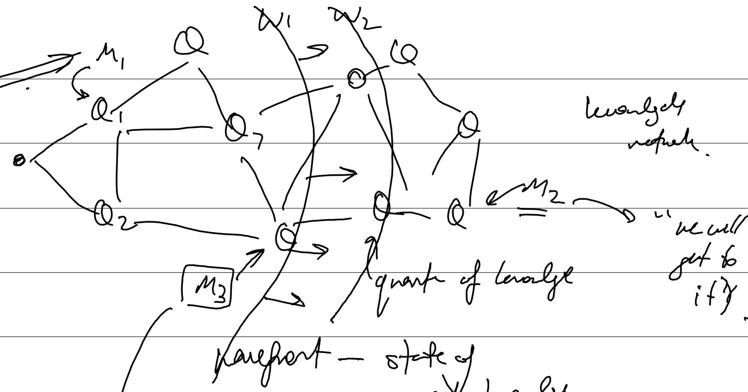
computational capacity of learner.

computational capacity of the user

⇓

teach stuff the student cant use.

remind b/c student knows M_1.



knowledge network.

"we will get to it".

quanta of knowledge

wavefront — state of current knowledge.

teaching moment to push the wavefront out.

* Talk to yourself

packaging of ideas.

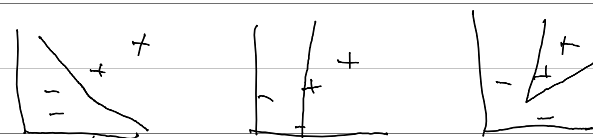
1. Symbol / visual handle & mark.
2. Slogan / verbal handle "near miss"
3. Surprise - one shot learning (definite learning from single example).
4. Salient → one shot learning via near miss.
5. story.

Lecture 16: Learning - Support Vector Machines

Agenda

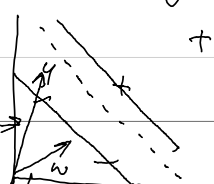
Support vector machines.

- decision boundaries.
- widest street approach.
- Kernel functions
- History lesson



nearest neighbor — ID trees. — neural nets.

SVM Algorithm



vector points to unknown

widest street to separate +ves from -ves.

↳ vector perpendicular to gutter/median line of street.
↳ unknown length,

↳ is unknown on left or right side of street.

$$\vec{w} \cdot \vec{u} \geq c \quad (\text{if } c = -b \ldots)$$

(1) $$\boxed{\begin{array}{c}\vec{w} \cdot \vec{u} + b \geq 0 \\ \text{Decision rule}\end{array}}$$ Then the sample

↳ no $\vec{w}$ & no value of b known.

positive sample.

$\bar{w} \cdot x_+ + b \geq 1$

$w \cdot x_- + b \leq 1$

negative sample.

y_i such that $y_i = +1$ for + samples.

$= -1$ for - samples.

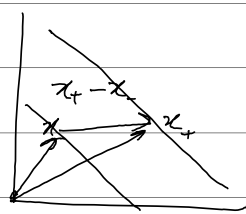
$y_i(\bar{w} + x_i + b) \geq 1$

$y_i(\bar{w} + \bar{x}_i + b) \geq 1$

$y_i(\bar{x}_i \cdot \bar{w} + b) - 1 \geq 0$

(2) $\boxed{y_i(\bar{x}_i w + b) - 1 = 0}$

for x_i in/on gutter.



$$\text{width} = (x_+ - x_-) \cdot \frac{\bar{w}}{\|w\|}$$

$1-b \quad 1+b$

$\bar{w}$, the normal.

(3) $$\boxed{\text{width} = \frac{2}{\|w\|}}$$

$$\max\left(\frac{2}{\|w\|}\right) \rightarrow \max \frac{1}{\|w\|} \rightarrow \min \|w\|$$

$$\boxed{\min\left(\frac{1}{2}\|w\|^2\right)}$$

Lagrange multiplier: $$L = \frac{1}{2}\|\bar{w}\|^2 - \sum \alpha_i \underbrace{[y_i(\bar{w} \cdot \bar{x}_i + b) - 1]}_{\text{constraint}}$$

$$\frac{\partial L}{\partial \bar{w}} = \bar{w} - \sum \alpha_i y_i x_i = 0$$

(4) $\Rightarrow \boxed{\bar{w} = \sum_i \alpha_i y_i x_i}$ (linear sum of samples)

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i = 0.$$

$$\Rightarrow \boxed{\sum \alpha_i y_i = 0}$$

$$L = \frac{1}{2}\left(\sum \alpha_i y_i x_i\right)\cdot\left(\sum \alpha_j y_j x_j\right) - \left(\sum \alpha_i y_i\right) x_i \cdot \left(\sum \alpha_j y_j x_j\right)$$

$$- \underbrace{\sum \alpha_i y_i b}_{=0} + \sum \alpha_i$$

$$\boxed{L = \sum \alpha_i + \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{x_i \cdot x_j}}$$

optimization depends on pairs of samples.

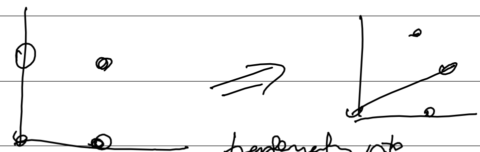
$$\sum \alpha_i y_i \boxed{\bar{x}_i \cdot \bar{u}} + b \geq 0$$

THEN +.

decision only depends on dot product.

convex space $\Rightarrow$ never stuck in local maxima.

not linearly separable?



transformation into
a different space $\phi(\bar{x})$

$\phi(\bar{x}_i) \circ \phi(x_j)$ to maximise.

$\phi(\bar{x}) \circ \phi(\bar{u})$

$$\boxed{K(x_i, x_j) = \phi(\bar{x}_i) \circ \phi(\bar{x}_j)}$$

↳ kernel function provides dot product in alternative space.

① $(\bar{u} \circ \bar{v} + 1)^n$ (linear kernel)

② $e^{-\frac{\|x_i - x_j\|}{\sigma}}$ (radial basis function)

↳ σ too small → can cause overfitting

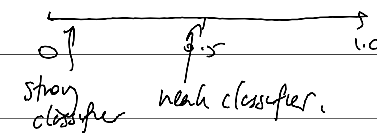
Lecture 17: Learning: Boosting

- weak → strong
- AdaBoost
- Tricks
- overfitting

binary classification

h [−1, +1]

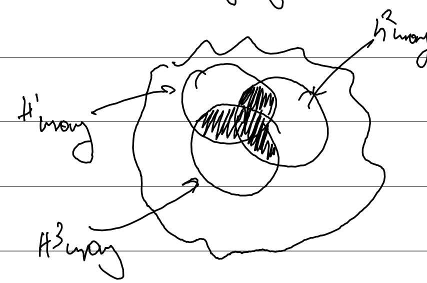
ERROR



$H(x) = \text{sgn}(h^1(x) + h^2(x) + h^3(x))$

↳ sample

↳ takes majority vote.



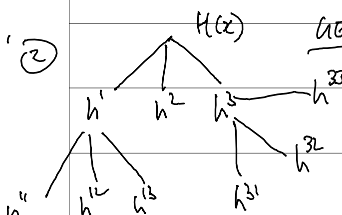
DATA $\longrightarrow h^1$

DATA
EXAGGERATION OF
H_1 errors
(different weighting) $\longrightarrow h^2$

(1)

DATA
exaggeration of
h_1 different from h_2 $\longrightarrow h^3$

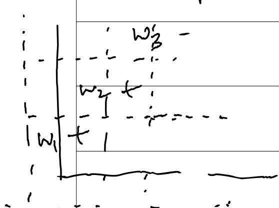
add recursion: (2)



GET OUT THE VOTE idea.

(3) Decision tree stumps

12 possible tests



$$\underset{\text{error rate}}{\epsilon} = \sum_{\text{wrong}} \frac{1}{N}$$

#samples

$$\underset{\text{weight}}{w_i^1} = \frac{1}{N} \qquad \epsilon = \sum_{\text{wrong}} w_i$$

constraint on weights: $\boxed{\sum_i w_i = 1}$

across whole space.

enforces distribution.

(4) $H(x) = \text{sgn}\left(\alpha^1 h^1(x) + \alpha^2 h^2(x) + \alpha^3 h^3(x) + \dots\right.$

(5)

← time 1

Let $W_i^1 = \frac{1}{N}$ (all equal in beginning).

N → # samples.

Pick h^t that minimizes error^t ← time t → pick α^t

Calculate W_i^{t+1}

(6) Suppose $W_i^{t+1} = \frac{W_i^t}{Z} \times e^{-\alpha^t h^t(x) y(x)}$

Z: normalizer to make $\sum W_i = 1$

i.e. $Z = \sum_i W_i^t \, e^{-\alpha^t h^t(x) y(x)}$

minimum error BOUND for whole thing of H → error rate bounded by exponentially decreasing function.

(7) $\alpha = \frac{1}{2} \ln\left(\frac{1-\varepsilon^t}{\varepsilon^t}\right)$ ← error at time t

correct case i.e. $h^t(x)\, y(x) = 1$,

(7) → (6)

(8)

$$W_i^{t+1} = \frac{W_i^t}{2} \times \begin{cases} \sqrt{\frac{\varepsilon^t}{1-\varepsilon^t}} & \text{correct} \\ \sqrt{\frac{1-\varepsilon^t}{\varepsilon^t}} & \text{wrong} \end{cases}$$

checking constants.

$$Z = \sqrt{\frac{\varepsilon^t}{1-\varepsilon^t}} \underbrace{\sum_{\text{correct}} W_i^t}_{1-\varepsilon} + \sqrt{\frac{1-\varepsilon^t}{\varepsilon^t}} \underbrace{\sum_{\text{wrong}} W_i^t}_{\varepsilon}$$

$$= 2 \times \sqrt{\varepsilon^t (1-\varepsilon)}$$

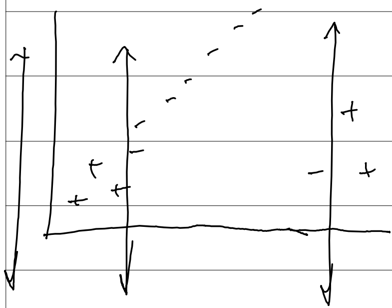
$$W_i^{t+1} = \frac{W^t}{2} \cdot \frac{1}{1-\varepsilon} \quad \text{correct} \quad \frac{1}{2} \frac{1}{1-\varepsilon} \underbrace{\sum_{\text{correct}} W^t}_{1-\varepsilon}$$

$$\sum_{\text{correct}} W_i^{t+1} = \frac{1}{2}$$

$$W_i^{t+1} = \frac{W^t}{2} \frac{1}{\varepsilon} \quad \text{incorrect}.$$

$$\sum_{\text{incorrect}} W_i^t = \frac{1}{2}$$

all you have to do is scale the weights



no test between

any classifying tests correctly

is good

↳ ∴ 3 tests instead of 11.

very few tests matter.

boosting doesn't seem to overfit.

possibly because boosting will fight wrong around outliers so much that

no other point can fit in it.

Lecture 18: Representations: Classes, trajectories, transitions.

- SVM, boosting dont give a model of human intelligence.
-

Agenda.

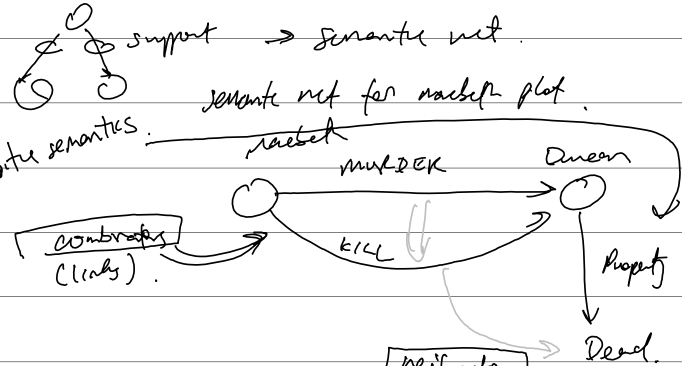
- The symbolic species.

- Strong story hypothesis. → humanity's symbolic ability to tell stories is what allowed intelligence.
- Dont use pronouns → places syntactic burden on the reader.
- Dont use Former or Latter. → people stop & go back
- dont call a shovel a spade → dont change the word as it may change meaning (introduces ambiguity)

support → semantic net.

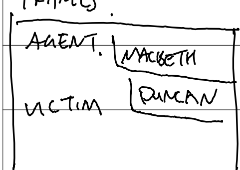
semantic net for macbeth plot.

Problem: primitive semantics.



reification
(treaty links as
objects linkable)

FRAMES



← Localisation

sequence

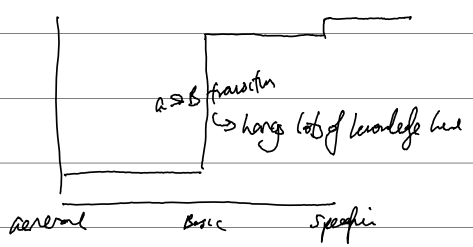
useful uses of semantic nets.

Classification

① INSTRUMENT TOOL FRUIT

Piano Hammer. Apple.

Bosendorfer. Ball peen hammer. Mac.



② Transition of state

car crashing into wall.

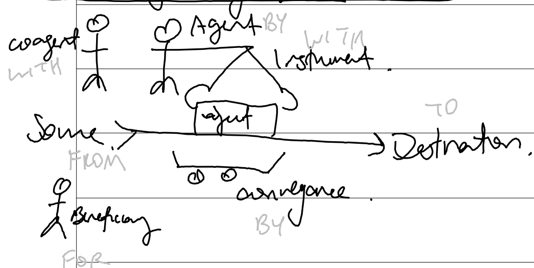
	before	during	after
speed of car	$\cancel{\Delta}$	$\to 0$	$\cancel{A}$
distance to wall	$\downarrow$	$\to 0$	$\cancel{D}$
condition	$\cancel{\Delta}$	$\Downarrow$	$\cancel{\Delta}$

↳ 10 things to describe change.

$\uparrow$ $\downarrow$ Δ ← change A ← appear D ← disappear.

$\cancel{\uparrow}$ $\cancel{\downarrow}$ $\cancel{\Delta}$ $\cancel{A}$ $\cancel{D}$ NOT events.

③ Trajectory frame/role.



adverbs to define the comport of the frame.

· in 100 sentences

↳ 25 transitions or trajectories.

putting it together → 4 story sequences.

ex:

Pat comforted Chris

terrorized

role frame:

Agent	pat
Action	??
object	chris
result	

Transition frame

object	chris.
mood	↑ ↓

Pat kissed Chris.

stabbed.

role

Agent	pat
action	?
object	Chris
result	
hallucination	

Transition

object	Chris
mood	↑ ↓
health	↓

Trajectory

object	pat's ~~lips~~ knife
destination	chris' ~~lips~~. body

Story libraries

- event — Time & place.
  - disaster — Fatalities $
    - earthquake
      - magnitude
      - Fault
    - hurricane
      - category
      - name.
  - party
    - Birthday
    - wedding
      - Bride
      - Groom

a frame in of itself to understand the situation.

Lecture 19: Architectures: GPS, SOAR, Subsumption, Society of Mind.

Agenda:

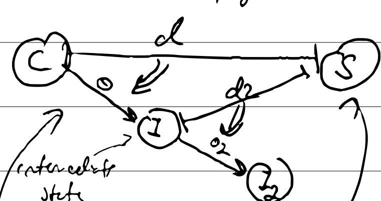
Architectures

- GPS: Newell & Simon
- SOAR: Newell
- Emotion machine: Minsky
- Subsumption: Brooks.
- Genesis: Winston et al.

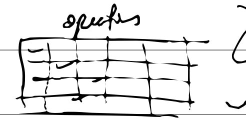
General Problem Solver (GPS)

symbolic distance

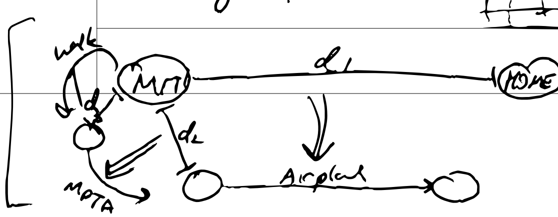
(problem solving hypothesis)



"means-ends analysis"



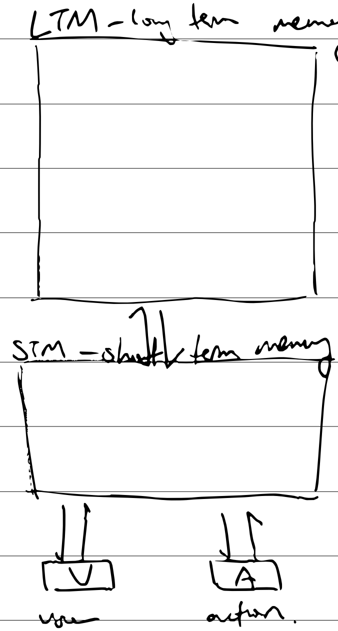
recursive



need to build a table.
&
table is hard to build

SOAR - state operator and result (built on GPS)

(Symbol system hypothesis)



architectural components

1. STM & LTM   cognitive science
2. LTM:
   AI [Assertions & rules. a.k.a productions
3. Preference system.
4. Problem spaces.
   to solve problem, need to produce space, then search through space.
   GPS
5. Universal subgoaling

Emotion Machine

self conscious thinking
self reflecting layer
reflective thinking
Deliberative thinking ...> SOAR & GPS.
learned reactions.
instinctive reaction ...> subsumption.

Common sense hypothesis
→ to do all layers, you need to have common sense.

Subsumption

Creature Hypothesis

V | R | A

~~vision~~ | ~~reasoning~~ | ~~action~~

update individual systems
breaks the system

→

seek.	V/A/R
explore	V/A/R
wander	V/A/A
Avoid	V/A/R

layers of abstraction

↳ basis behind the Roomba

Features:

1. no representation
   (no models)
2. use world instead of model.
   everything you do is reactive
3. mechanisms in purest form are finite state machines.

Genesis

→ stories → culture

Description of events

macro
- country, religion.

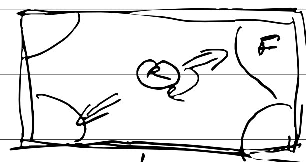
micro
- family, personal experien

Language

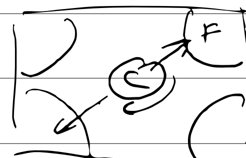
Perception

Strong story hypothesis.

⇒ this model is all there is.



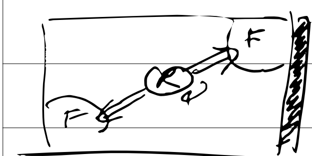
rat



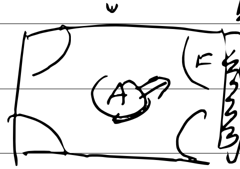
child/adult.

until.

when does a child become an adult



rat, child



adult.

* Code - listen - draw - talk.

* Beware of fast talkers ⇒ jam language processes
⇒ can't think.

Lecture 21: Probabilistic Inference I

Agenda

- Probabilistic inference
  &
- Belief nets
- naive Bayes classification
- Bayesian model discovery

$n = 2^{\# \text{possibilities}}$

status S	hacks H	art show A	Tally	P
F	F	F	405	0.2
F	F	T		⋮
F	T	F		
F	T	T		
T	F	F		
T	F	T	225	
T	T	F	40	
T	T	T	45	

Joint probability table.

↳ issue with recording all numbers & guessing these numbers w/ more features.

How to deal with probabilities without the table?

Joint probability Table

① Basic probability

② conditional probability

③ chain rule

④ independence

⑤ conditional independence

⑥ Belief nets

⑦ Joint probability table.

① Basic probability.

Axioms

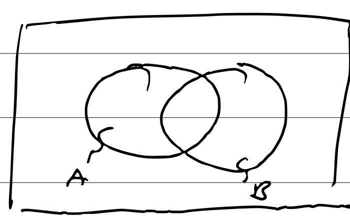
① $0 \le P(A) \le 1$

② $P(\text{TRUE}) = 1$

$P(\text{FALSE}) = 0$

③ $P(A) + P(B) - P(A \cap B) = P(A \cup B)$

Intuitions

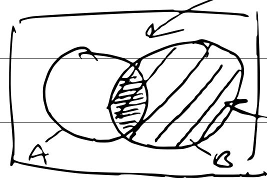


② Conditional probability

Definition

$$P(A|B) = \frac{P(A,B)}{P(B)} \quad (= P(A \cap B))$$

Intuition



probability of a AND B

probability of B.

or $P(a,b) = P(a|b)P(b)$

$P(a,b,c)$

let $y = P(b,c)$

$$\begin{aligned} P(a,b,c) &= P(a,y) \\ &= P(a|y)P(y) \\ &= P(a|b,c) \cdot P(b,c) \\ &= P(a|b,c) \cdot \underbrace{P(b|c)}_{\text{b depends on 1 thing}} \underbrace{P(c)}_{\text{c depends on nothing.}} \end{aligned}$$

a depends on 2 things

$\Downarrow$

chain rule

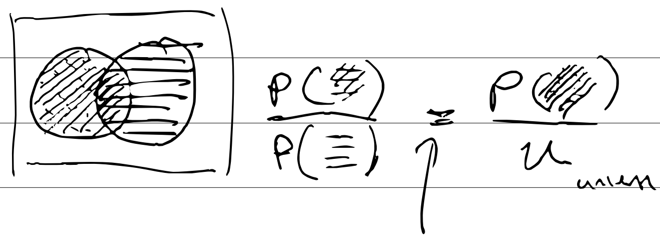
$$\boxed{P(x_1, \ldots, x_n) = \prod_{i=n} P(x_i \mid \underbrace{x_{i-1}, \ldots, x_1}_{\text{index are smaller than left.}})}$$

③ Independence.

Definition:

$P(A|B) = P(A)$ if A independent of B.

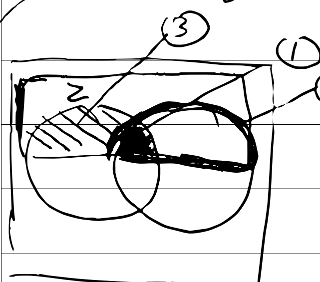
Intuition.



They are the same if independent

⑤ conditional independence definition:

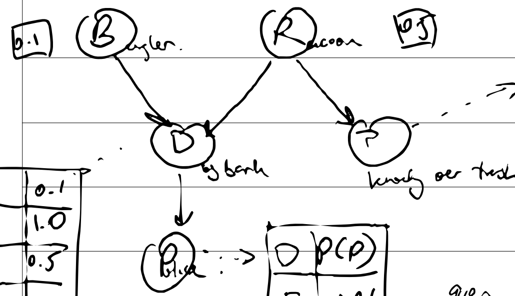
$P(A|BZ) = P(A|Z)$



$$\frac{P(3)}{P(2)} = \frac{P(1)}{P(2)}$$

$P(A,B|Z) = P(A|Z) \times P(B|Z)$

6 Belief networks.



B	R	
F	F	0.1
T	F	1.0
F	T	0.5
T	T	1.0

R	T
F	0.001
T	0.8

D	P(P)
F	0.01
T	0.1

given a node, its probability is not related to anything except its direct parents.

specified 10 numbers.

vs. joint probability table: $2^5 = 32$

⇒ considerable saving.

conditional independence.

$$P(p,d,b,t,r) = P(p|d\cancel{b}\cancel{t}\cancel{r})\,P(d|b\cancel{t}r)\,P(b|\cancel{t}\cancel{r})$$
$$P(t|r)\,P(r)$$

↳ get this from the belief network

Lecture 22: Probabilistic inference II.

Agenda: From probabilistic inference to structure

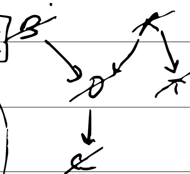
▷ Chain rule +

$$P(x_1,\ldots,x_n) = P(x_n \mid x_{n-1},\ldots,x_1)\,P(x_{n-1} \mid x_{n-2},\ldots,x_1)\cdots P(x_1)$$

Chain rule

List construction

$\boxed{C, D B T R}$



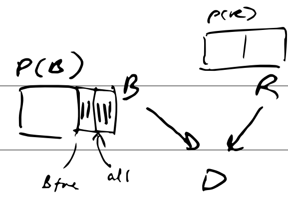
↳ no loops in these nets

↳ always a bottom

for any element, no descendent lies to its left.

Independent of any non descendents given its parents.

$$P(C,D,B,T,R) = P(C \mid DBTR)\,P(D \mid BTR)\,P(B \mid TR)\,P(T \mid R)\,P(R)$$



B	R	P(D)	D is true	All
T	T		1	11
T	F			
F	T			
F	F			1

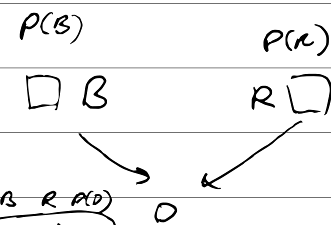
results to fill in

(1) TTT.

(2) FFF

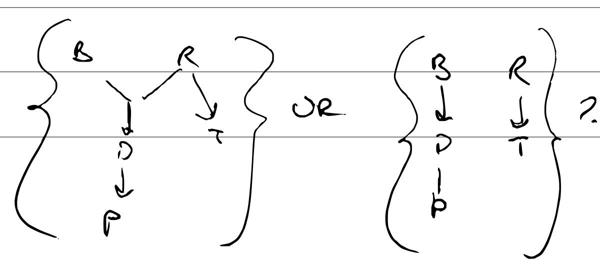
(3) TTF

. . . . etc.



B	R	P(D)
T	T	
T	F	
F	T	
F	F	

↳ but how do we know of the model ??



↳ Bayesian Inference

Bayesian Inference

$$P(a|b) = \frac{P(a,b)}{P(b)}$$

$$P(a|b)P(b) = P(a,b) = b(b|a)P(a)$$

Bayes' rule

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

$a \equiv$ class

$b \equiv$ evidence (observed $5x$)

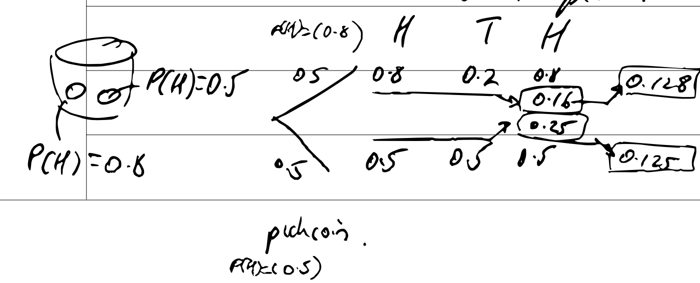
$\Downarrow$

$$P(c_i|e) = \frac{P(e|c_i)P(c_i)}{P(e)} = \frac{P(e_1,\ldots,e_n|c_i)P(c_i)}{d}$$

$$\frac{P(e_1|c_i)P(e_2|c_i)\ldots P(e_n|c_i)P(c_i)}{d}$$

example: coin A 50/50 & coin B 80/20 $\Rightarrow$ flip coin thrice, get H&T

which coin did I pick?



2 models w/ probability assigned.
↳ take data;
calculate P(left model) & P(R model) given data.
× a-priori probabilities (assumed equal)

Bayesian rules can be used for <u>structure discovery</u> (selecting between 2 models)

useful in diagnosis problems
↳ lie detection.
↳ medical diagnosis.
↳ material quiz diagnosis
↳ spacecraft diagnosis (equipment diagnosis)
..

* the right thing to do when you don't know anything ↳ bayesian analysis.

* You often don't know anything

	D	D	D	D	D
$R_{part} = 0.6$	0.2	0.2	0.2	0.2	0.2
$D_{part} = 0.4$	0.6	0.6	0.8	0.8	0.8

$P(R \mid 5 \times D) = 0.000192$

$P(D \mid 5 \times D) = 0.13$

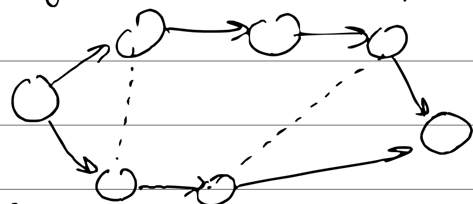
Lecture 23: model merging, cross-modal coupling

Agenda:

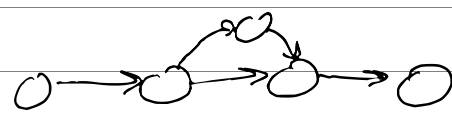
- Bayesian story merging
- cross modal coupling & the zebra finch.
- Big issues.

Bayesian story merging

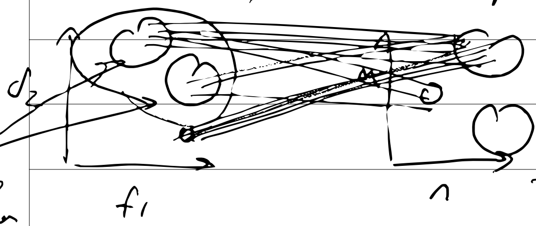
story 1 finite state graph.



story 2



cross-modal coupling.



projects → projection to each other
↓
same vector,
↓
closest clusters.

→ can we merge clusters to form bigger cluster w/ its own meaning.