2

Prop: Proposal for project of ~ p 1, 79 ff.

This will be partly a more detaild dscn of what approach I expect to take on TM next.

TM62

Index 73 TM 323.20 Has a lot of recent stuff on this
Also ibid 323, 33 (Tug says).

21674-TM62

Perhaps code corpus so that a random code is t. most likely contnn. After t. contnn. is found, t. resultant corpus is recoded, so that again, a random contnn. is t. most likely

Q. : what about initial "overhead cost" of devising large new defnns.?

This mite be treated by using a fairly large effective "radix". T. radix size is one type of restriction on t. kinds of regys. allowd. Radix = ∞ is t. most genl. possl. coding method, in this way of looking at it.

While this is not diflt. to do., t. main Q. is how to do this w. reasonable (& eventually), minimal cost.

v(313)

37 Suppose one has a corpus & one has listed all of t. regys that one has found in it. For each regy there is also gn. its "short codes" (

: Superficially, it would seem that we would only be able to devr. new raggys in which t. "new corpus" had enuf of t. raggy to pay for t. overhead of defining it. — we would not use any "raw data" in t. old corpus to help pay t. overhead of defining new raggys.

— Actually, I'm not sure this is true. If t. defn. of a new raggy is almost paid for by data in t. "new part of t. corpus" then one would, presumably, look at raw data in t. old corpus to see if it helps pay overhead.

This entire concept seems reasonable! I.e. that it must, a pri, be expressable into a correct, usable form! By writing all raggys in t. form 62.37 one has, indeed, written down all that one knows about t. corpus thus far, — including derrys about derrstr. It would seem that given no bits of t. new corpus, then t. info of 62.37 would be all that one could use for predn. of t. next bit or any seq. of next bits.

Hvr., gn. t. data of 62.37 and t. initial 100 bits of t. new corpus, t. problem is rather different. T. Q is "how does one update one's "prediction state" (t. data of 62.37 could be called a "predn. state".)

It would seem that by updating t. "predn. state" every bit, one would get less good predn. per ccost, than if one updated it every 100 bits, say.

Superficially, using a 100 bit "updating quantum" is equivt. to using a radix of 2^{100}.

So I have this "predn. state": consisting of ① Devn. of t. underlying umc. ② list of all raggys ③ T. new Δcorpus : Problem: ⓐ to make optimum predn. for Δcorpus = 0 ⓑ to update t. predn. state for Δcorpus ≠ 0.

CMI should tell how to do this — but in a not nearly v.g. ccost form.

T. main Q, then, is to express it all in a minimal ccost form —

→ ie. Gn. a predn. state, to get t. best possl. answers for ⓐ & ⓑ w. a gn. ccost.

Essentially, t. Porgg. is a formalizn. of what I expect to be doing after t. "final report" on induction.

From t. predn. state, it should be possl. to ⓐ list all of t. past raggys. in order of prost of their "names". ⓑ list all possl. new raggys in order of prost. Hvr., note that ccosts are not considered in ⓑ. We can, hvr., consider ccost in ⓐ because we have had some experience w. them. If we have a TM_2, we can make ccost estimates in ⓑ. This TM_2 need not be very clever. It may even use a fairly simple set of estimation methods.

: "introspection", "meditation" consists of (a) looking at old copys known to be good & trying to find better codes for them (b) Doing the same w. sections of the old corpus. — (Note that both (a) & (b) can use new abbs — & corpus segments that were unknown at the time the old codes were devised.)

TM64

Actually, even w.r.t. "Predn. state" as derbd on 62.37, its not nearly so easy to make an optimum predn. One can, of courses select the "Best" pgm that one has. One can also try linear combs of various pgms (I've written much on this latter).

What one should optimumly do, is take all of the raggys — express each as a set of codes for the corpus (& alternative contents of the corpus), — Then, to combine them, just use all codes as they have — the Boolian Union. Is there any reasonable way to do this mathially? It would seem that there should be some method by which one could take 2 or more arby pgms, & find the Boolian Union of their respective codes

---

Well, if one uses the method of W's thms 5 to xfm Pgm dcrns into FORs & then into codes for a UMC, it certainly seems that if one has 2 pgms that differ in all, there is no overlap in their dcrns of the corpus (& alternative corpi), they have no codes in common. — ~~The codes consist~~

These codes for the first UMC consist of 3 parts: $\alpha \Delta \beta \Delta \gamma$:
β is a dcrn. of the pgm itself; α is a fixed routine derbd in W's thm 5, that tells how to go from β to a FOR; γ is a sequence that derbs the particular corpus of interest.
All pgms result in codes that have $\alpha\Delta$ in common. — However, they all differ in the "β" part, so they can have no completely codes in common.

.30 .31 would be true if the only codes ~~for a pgm~~ w.r.t. a gn. pgm were of this form $\alpha\Delta\beta\Delta\gamma$. It is very likely that there are other codes & so the possy of code overlap for diffnt pgms still does exist. Hrr, no!

Trouble is: while α can be very long, so $\alpha\Delta\beta$ may not be a min dcrn of a pgm., β can very well be a min dcrn of a pgm.

A possl. soln: that one can combine pgms [redacted] by linearly utily them by their possls — but only if the set of pgms being combined have been obtained in an "unbiased" way. A concrete illustration

of this is in linear regression. T. codes are all indep. One can select any subset of codes one likes & get a hyper total cost of the corpus — TM65
But to actually get better predn., one must either use all of the pcms. (w. = wts, for the same no. of coeffs) or select a sample of them in a suitably unbiased way. — see Tudor 323.15 (In PW; 384-399)

In my recent (late 1974) discn. of optimized linear coeffs for various pcms., I also worked out some conjectures on the expected "accuracy" (Ad-hochness) of the method. Around that time, I considered the problem of how to combine pcms, to be a major problem.

The problem of how to devise by a priori new pcms, involves optimum "factoring" of the old pcms — in the sense of expressing them in terms of a minimal set of maximal prosit abss. In the work I expect to start again in the next few months on devising the sups. & TM's to "track" them, I will know how certain induction problems were solved, & I will try to find out what the sup. could give out of the rite "factors" so that the "observed" or "necessary" induction could take place.

One of the values (if any!) of these last few pp. is that it may make it possl. to work on the problem of optimizing the operation of the machine wrt. min cost.

So, one has this bunch of pcms & abstractions (≡ factors). One is suddenly giv. a chunk of new corpus. The problem is then how to devise new by prosit pcms & test them rapidly, for many new (& old) trial pcms, one could quickly reject them on the basis of only part of the new corpus (i.e. small sample). [There is, unfortly, the possy that a pcm rejected on one part of the new corpus would work o.k. on some other part of the new corpus].

76.01
spec

66 is Friday

01:75.40: 65.40 spac → On TM_1 & TM_2 : At each point in time TM_1 [redacted]

does the best search it can possibly do to find regys in t. new Δ corpus. This "best" is w.r.t. regys (& factorizations of them) that it has found in t. corpus up to now. This is t. work of TM_1.

The actual search technique devised by TM_2 is the best possl., in view of t. techniques that have proved useful on t. past corpus up to now. At first, I will be TM_2. Later, of course, [redacted] when TM_1 has had enuf successful experience, it can be TM_2 as well as TM_1.

To start out, with a minimal TM_2, TM_1 would have t. set of regys & factorizns & regy names from t. previous corpus. To try to code t. new Δ corpus, random (or rote sequential) combnns. of factors would be usd, with a reasonable C.B. for each trial — with perhaps some small sample decisions usd to reject proposd routines if they do very badly on initial segments of t. new Δ corpus. So this rudimentary set of search techniques constitutes TM_2's initial state.

As more regys are found in t. (growing) corpus, [redacted] & I, as TM_2, devise new techniques to find them, [redacted] & with the large list of corpus regys found, it is possl. to devise a reasonable ordering of regy trials, so they could have been found sooner. Anyway — with this large list of successful regys found, & t. list of successful (& unsuccessful) [redacted] search [redacted] techniques, (& factored & named), it becomes possl. to devise an optimum set of [redacted] new trial search [redacted] techniques for TM_2.

(SN) Incidently, there is [redacted] a Goal for TM_2 — The maximum ↑ in prost of t. corpus per cost.

22474 Pro

TM77

01:76.40 : On to idea of ≈ 62.37 : T. problem w. Δcorpus = 0 is not/too diflt. — or even if Δcorpus = a few bits (i.e. I can use t. "best pam" or a linear (or some other) combination of pams). For larger Δcorpus, t. problem becomes impt. — in t. sense of new tech neques need be usd to find vagy in ⟨corpus + Δcorpus⟩ ~~…~~ w. min cost.

Hvr., note that even if Δcorpus are fairly large chunks, we are still not nearly getting t. best predn.. To get t. best predn, we must try to code ⟨corpus + Δcorpus⟩ as an entire object. This can be done by 1 form of "meditation" — i.e. periodicly reviewing & attempting to recode t. ⟨entire corpus up to now⟩

T. original 62.37 method is essentially, "incremental block" coding. It would seem to be best (i.e. closest to true optimum) when blocks were largest.

Note hvr., that for any block size, no matter what it is, there will be impt vagys that will be undcvrable, because their overhead cost is too large for t. amt of bcost savings available in a block that size.

Breaking outside t. method (by ≈ meditation) one mite, for a certain kind of vagy, suspect that ∮ in ssz mite make it a useful vagy — so looking at t. past corpus mite be reasonable. E.g. Say we have 1k symbols in t. Δcorpus, that our pam costs 500 bits & we save .4 bits per symbol using this pam, & it "looks" like this pam would work just as well on t. past. So one would be justified in ∮ t. ssz & looking at t. past 1k symbols, to see if it will "break even" at least.

1-20 74

~~Phone Kochen! — He's leaving soon — perhaps already!~~

~~Put Mike Arbibon Mailing list~~

~~" Kochen on " "~~

~~Mail Fred~~ Willis paper + my report

There are books listing research being done in ~~E.~~ Cos., & Labs, Univs, all over country; so perhaps

① Sponsors are listed

② Look in ~~A~~ JACM, CACM, ~~IEE~~ IEEE computers etc.
— look for sponsors: AFOSR always makes them put it in — perhaps NIH as well

---

Look at defn. of MSM & UMSM: see if they are really well defined & if t. proofs involving them (existence of UMSM, MSM corresps to any limit fcn, ~~UU~~ UOMS are subset of MSM's, "Provability" of MSMS ~~?~~ — & perhaps others] are all true ~~?~~.

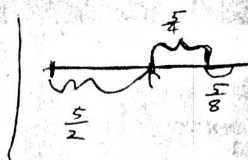
A possl. ~~M~~ proposal! Use of ~ 3 people.

CMI project: 2 aspects: ~ 50 ~~m~~ problems

a) Theoretical Induction: ① Relation of Kcst to W's "info".
Are UMSM's are better than UOM's?

② Are there any practical probs for which UMSM's are better than UOM's?

③ Are there approaches to Universality that involve less computation than direct use of t. Willis method?



ⓑ Applicns of CMI to practical probs.

ⓐ Tng seqs. in Arith, Alg

ⓑ " " for present learning devices

e.g. Sussman's, Goldstein's debuggers ~~[illegible]~~ Winston's Learner,

$\frac{7}{4}$

$\frac{10}{4}$ 1"

$1\frac{1}{2}$ +
$6\frac{1}{4}$ 2508
$\frac{2}{9\frac{1}{4}}$ say 10"

---

① Office on 8th floor

② My appointment expires in ~ 2 Mo.! — I'd like to stay longer.

③ Money problems. ~~[illegible]~~ ⓐ MIT Fnd Liaison Office.
ⓑ Lists of sponsors in Journals.
& ARPA

Discuss w. Pat. Winston.

Just phoning Licklider for advice on who to ask for funds, would be a good approach.