The easiest way to do bargaining under incomplete information is a TIOLIO by the uninformed party.

Cremer, Pratt, Garicano

Language is a way of coordinating on communication

This is a paper about bounded rationality - people can only communicate so much

- Team theoretic
- Engineer - salesman
  - clients with events "that demand solutions."
  - events: $x \in X$ (finite)
  - prob $f_x > 0$
- Bounded rationality: Language/code C is a <u>partition</u> of X. e.g $X = \{x_1, x_2, x_3\}$, and language has at most 2 words. Each element of the partition is a word.

$$\{C\} = \{\{\underbrace{\{x_1, x_2\}}_{W_1}, \underbrace{\{x_3\}}_{W_2}\}, \{\underbrace{\{x_1\}}_{W_1}, \underbrace{\{x_2, x_3\}}_{W_2}\}\}$$

more generally $C = \{W_1, \dots, W_K\}$, breadth of word k, n_k, is the number of elements in W_k. The frequency of word k is

$$P_k = \sum_{j \in W_k} f_j$$

- Engineer needs to diagnose the problem. The cost of doing so is increasing in n_k. $d(n_k) \uparrow$ in n_k
- Expected diagnosis cost $D(C, f) = \sum_{k=1}^{K} P_k d(n_k)$
  - There is no notion of "closeness" of words.
- For a given K, what is the best partition C?

Prop: In an optimal code, $n_k \geq n_{k'} \Rightarrow f_x \leq f_{x'}$ for all $x \in W_k$, $x' \in W_{k'}$.

- less precise words must relate to less frequent events.
- If not, switch two events, and gain is
  $[d(n_k) - d(n_{k'})][f_x - f_{x'}] > 0$
- Frequent events must be identified more precisely

Prop: If d is convex in n, (ie $d(2)-d(1) > d(1)-d(0)$)
then $n_k - n_{k'} \geq 2 \Rightarrow p_{k'} \geq p_k$ $\quad n_k - 1 \geq n_{k'} + 1$

- precise words should be more frequent.
- If not, switch one event and $D(C) - D(\tilde{C})$

$$= d(n_k)p_k + d(n_{k'})p_{k'} - d(n_k - 1)(p_k - f_j) - d(n_{k'}+1)(p_{k'} + f_j)$$

where $j \in W_k$.

$$= [d(n_k) - d(n_k - 1)]p_k - \overbrace{[d(n_{k'}+1) - d(n_{k'})]}^{\leq d(n_k) - d(n_k - 1)} \cdot p_{k'} + f_j \underbrace{[d(n_k - 1) - d(n_{k'}+1)]}_{0}$$

$$> \underbrace{[d(n_k) - d(n_k - 1)]}_{\geq 0}(\underbrace{p_k - p_{k'}}_{\leq 0}) \geq 0 \text{ by convexity}$$

e.g:

Suppose $\underline{X} = \{x_1, x_2, x_3\}$, $f(x_1) = f(x_2) = \frac{1-p}{2}$, $f(x_3) = p > 1/3$

Then, optimal $C = \{\{x_1, x_2\}, \{x_3\}\}$

$D(C) = p + 2(1-p)$ if $d(1) = 1, d(2) = 2$
$= 2 - p$

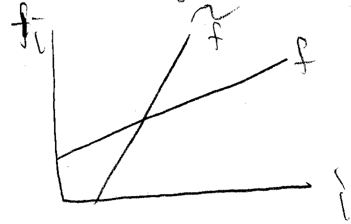
- environment is more complex if p is smaller

Rank events s.t. $f_1 \le f_2 \le \dots$

- $F_i \equiv Pr[j \le i] = \sum_{j=1}^{i} f_j$

<u>Definition:</u> $\tilde{f}$ is less complex than f if $F_i \ge \tilde{F}_i \; \forall i$,

where $\tilde{F}_i$ uses the "indexing" of F_i.

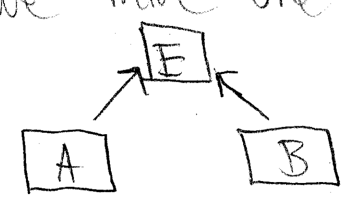


$\tilde{f}$ less complex than f

<u>Proposition:</u> f more complex than $\tilde{f}$ $\Rightarrow$ best code under f will be more costly than the best code under $\tilde{f}$: $\min D(C, f) \ge \min D(C, \tilde{f})$

Moreover, an increase in the number of words from 1 to $K > 1$ lowers C more for less complex environments but going to very large numbers of words decreases C more for more complex environments

What if we have one engineer and two salesmen?



- should A and E have "different dialects" than B and E?

<u>Prop:</u> Optimally, $C_A = C_B$

<u>Pf:</u> - Take two codes C_A, C_B s.t. $C_A \cup C_B$ contains K words.

- If $C_A \ne C_B$, then both have at most $K-1$ words
- Take <u>narrowest</u> (smallest n_k) noncommon word. If it belongs to C_A, add it to C_B, while taking the underlying

events away from C_B words.

e.g. $C_A = \{ \{1,4\}, \{2,5\}, \{3,6\} \}$

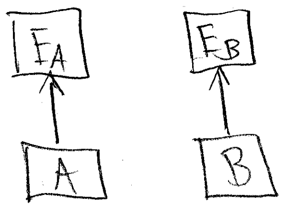
$C_B = \{\{1,2,3\}, \{4,5,6\}\}$

$\Rightarrow C_{B'} = \{ \{1,4\}, \{2,3\}, \{5,6\} \} \Rightarrow C_{B''} = \{ \{1,4\}, \{2,5\}, \{3\}, \{6\} \}$

Then, $\forall x$ length of word in new $C_{B'}$ containing x is not larger than in C_B. Moreover, $C_{B'}$ has at most K words.

Some x's are in strictly narrower words.

What about this structure?



why work together?

Integration makes more sense with more homogeneity in underlying populations.

- pooling can help deal with more customers.