Communication amid Uncertainty

Madhu Sudan
Microsoft, Cambridge, USA

Based on:

- Universal Semantic Communication – Juba & S. (STOC 2008)
- Goal-Oriented Communication – Goldreich, Juba & S. (JACM 2012)
- Compression without a common prior … – Kalai, Khanna, Juba & S. (ICS 2011)
- Efficient Semantic Communication with Compatible Beliefs – Juba & S. (ICS 2011)

Uncertainty in Communication?

- Always has been a central problem:
  - But usually focusses on uncertainty introduced by the channel
  - Standard Solution:
    - Use error-correcting codes
    - Significantly:
      - Design Encoder/Decoder jointly
      - Deploy Encoder at Sender, Decoder at Receiver

New Era, New Challenges:

- Interacting entities not jointly designed.
  - Can't design encoder+decoder jointly.
  - Can they be build independently?
  - Can we have a theory about such?
    - Where we prove that they will work?
    - Hopefully:
      - YES
      - And the world of practice will adopt principles.

Example 1

- Intersystem communication?
  - Google+ ↔ Facebook friendship ?
  - Skype ↔ Facetime chat?

- Problem:
  - When designing one system, it is uncertain what the other's design is (or will be in the future)!

Example 2

- Heterogenous data?
  - Amazon-marketplace spends N programmer hours converting data from mom-n-pop store catalogs to uniform searchable format.
  - Healthcare analysts spend enormous #hours unifying data from multiple sources.
- Problem: Interface of software with data:
  - Challenge:
    - Software designer uncertain of data format.
    - Data designer uncertain of software.

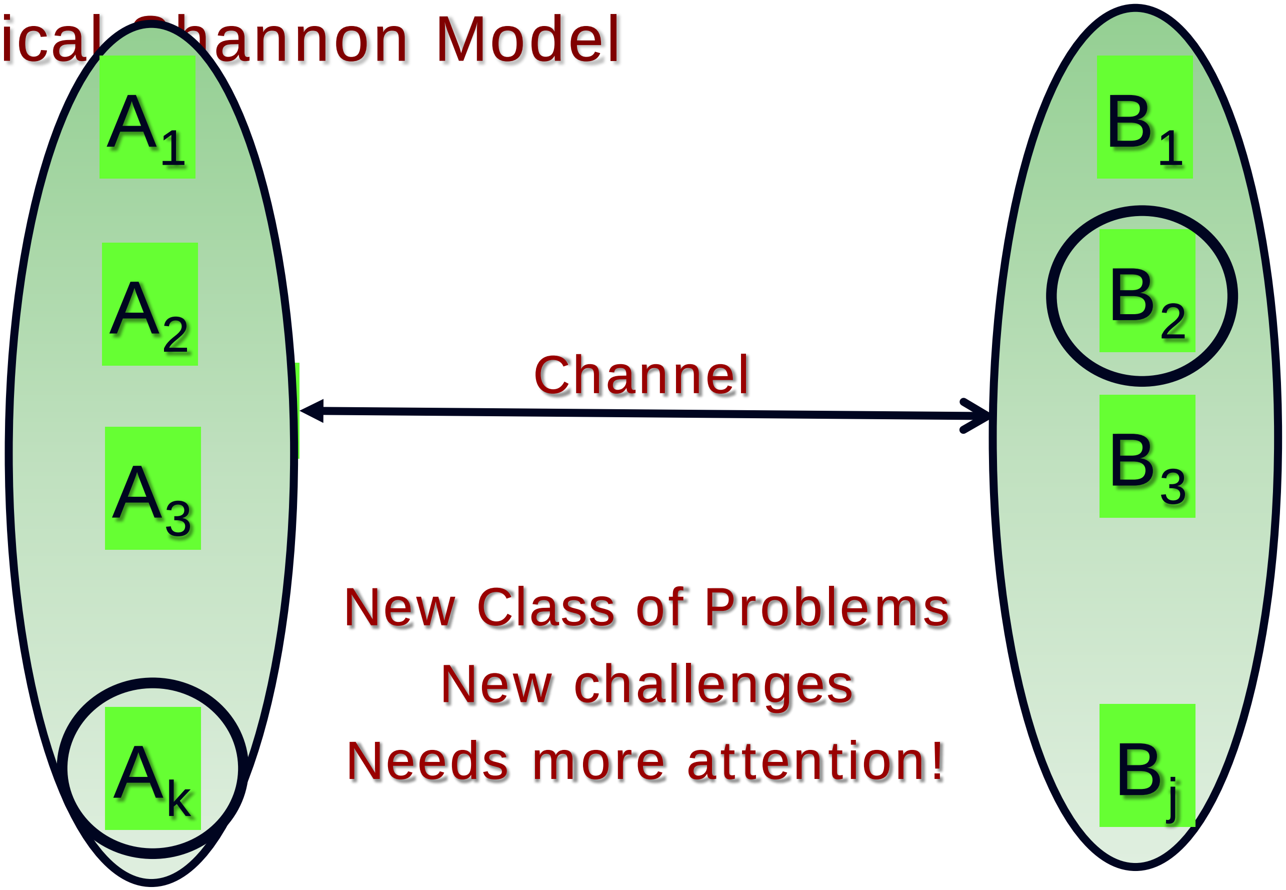
Example 3

- Archiving data
  - Physical libraries have survived for 100s of years.
  - Digital books have survived for five years.
  - Can we be sure they will survive for the next five hundred?

- Problem: Uncertainty of the future.
  - What systems will prevail?
  - Why aren't software systems ever constant?

Modelling uncertainty

Semantic Communication Model
Classical Shannon Model



Nature of uncertainty

- A_i's, B_j's differ in beliefs, but can be centrally programmed/designed.
  - [Juba,Kalai,Khanna,S.'11] : Compression in this context has graceful degradation as beliefs diverge.
- A_i's, B_j's differ in behavior:
  - Nothing to design any more.
  - Best hope: Can highlight certain A_i's (universalists) that can interact successfully with many B_j's
  - [Juba,S'08; Goldreich,J,S'12; J,S'11]: "All is not lost, if we keep goal of communication in mind"
  - Details don't fit in margin …

II: Compression under uncertain beliefs/priors

Motivation: Human Communication

- Human communication (dictated by languages, grammars) very different.
  - Grammar: Rules, often violated.
  - Dictionary: Often multiple meanings to a word.
  - Redundant: But not as in any predefined way (not an error-correcting code).
  - Our thesis: Emerges from uncertainty:
    - Sender of message uncertain about receiver's background/context/prior.
    - Will try to explain in the context of Redundancy

Behavioral aspects of natural communication

- (Vast) Implicit context.
- Sender sends increasingly long messages to receiver till receiver "gets" (the meaning of) the message.
- Sender may use feedback from receiver if available; or estimates receiver's knowledge if not.
- Language provides sequence of (increasingly) long ways to represent a message.
- Question: What is the benefit of choosing short/long messages?

Some reasoning

- Reason to choose short messages: Compression.
  - Channel is still a scarce resource; still want to use optimally.

- Reason to choose long messages (when short ones are available): Reducing ambiguity.
  - Sender unsure of receiver's prior (context).

    ("uncertainty")
  - Sender wishes to ensure receiver gets the message, no matter what its prior (within reason).
  - But doesn't want to abandon prior either.

A teaser:

- Suppose you and I have a ranking of N players.
  - Rankings $\pi, \sigma : [N] \to [N]$
- Further suppose we know the rankings are close.
  - $\forall\, i \in [N]: |\pi(i) - \sigma(i)| \leq 2$.
- You want to know: Is $\pi^{-1}(1) = \sigma^{-1}(1)$
- How many bits do I need to send to you (non-interactively).
  - $O(1)$?
  - $O(\log N)$?
  - $O(\log\log\log N)$?

Model

- Wish to design encoding/decoding schemes (E/D) to be used as follows:
  - Sender has distribution P on M = {1,2,...,N}
  - Receiver has distribution Q on M = {1,2,...,N}
  - Sender gets $X \in M$
  - Sends E(P,X) to receiver.
  - Receiver receives Y = E(P,X)
  - Decodes to $\hat{X}$ = D(Q,Y)

  - Want: X = $\hat{X}$ (provided P,Q close),
    - While minimizing $Exp_{X \leftarrow P} |E(P,X)|$

Contrast with some previous models

- Universal compression?
  - Doesn't apply: P,Q are not finitely specified.
  - Don't have a sequence of samples from P; just one!
- K-L divergence?
  - Measures inefficiency of compressing for Q if real distribution is P.
  - But assumes encoding/decoding according to same distribution Q.
- Semantic Communication:
  - Uncertainty of sender/receiver; but no special goal.

Closeness of distributions:

- P is α-close to Q if for all $X \in M$,

$$\frac{1}{\alpha} \leq \frac{P(X)}{Q(X)} \leq \alpha$$

- P α-close to Q $\Rightarrow$ $D(P||Q), D(Q||P) \leq \log \alpha$.

Dictionary = Shared Randomness?

- Modelling the dictionary: What should it be?

- Simplifying assumption – it is shared randomness, so …

- Assume sender and receiver have some shared randomness R and X is independent of R.
  - Y = E(P,X,R)
  - $\hat{X}$ = D(Q,Y,R)

- Want $\forall X,\ \Pr_R[\hat{X} = X] \geq 1 - \epsilon$

Solution (variant of Arith. Coding)

- Use R to define sequences
  - $R_1\,[1], R_1\,[2], R_1\,[3], \ldots$
  - $R_2\,[1], R_2\,[2], R_2\,[3], \ldots$
  - …
  - $R_N\,[1], R_N\,[2], R_N\,[3], \ldots.$
- $E_\alpha(P, x, R) = R_x[1 \ldots L]$, where L chosen s.t. $\forall z \neq x$

  Either $R_z[1 \ldots L] \neq R_x[1 \ldots L]$

  Or $P(z) < \frac{P(x)}{\alpha^2}$
- $D_\alpha(Q, y, R) = \mathrm{argmax}_{\hat{x}}\ \{Q(\hat{x})\}$ among $\hat{x} \in \{\, z \mid R_z[1 \ldots L] = y\}$

Performance

- Obviously decoding always correct.
- Easy exercise:
  - $\mathrm{Exp}_X \left[E(P,X)\right] = H(P) + 2\log\alpha$
- Limits:
  - No scheme can achieve $(1-\epsilon)\cdot[H(P)+\log\alpha]$
  - Can reduce randomness needed.

Implications

- Reflects the tension between ambiguity resolution and compression.
  - Larger the α ((estimated) gap in context), larger the encoding length.
- Coding scheme reflects the nature of human process (extend messages till they feel unambiguous).
- The “shared randomness” is a convenient starting point for discussion
  - Dictionaries do have more structure.
  - But have plenty of entropy too.
  - Still … should try to do without it.

III: Uncertainty on Action: Goal-Oriented Communication

Back to meaning

- What if sender is sending instructions?
  - Sender and receiver are uncertain about each other's "instruction ↔ bits" association?
  - Can we ensure receiver decodes the right instructions?
- Translation of bits to instructions?
  - Well studied in language/computer science.
  - (Many) "Complete" languages/codebooks exist.
    - Each translates bits to meaning.
    - All equivalent (upto "Kolmogorov constant")
    - But not same.

Goal of communication

- Easy negative result:
  - (Due to plethora of languages/codebooks): In finite time, **can't** guarantee "receiver understands instructions."
  - Is this bad?
    - If receiver can not distinguish correct instructions from incorrect ones, why should it try to do so?
- Goals of communication:
  - Communication is not an end in itself, it a means to achieving some end.
  - Hopefully receiver wishes to achieve a goal and using information from sender to achieve this goal.
  - Semantic communication:
    - Help communication achieve its goal.
    - Use progress towards goal to understand meaning.

Utility of Communication?

- The lens of computational complexity:
  - To prove some resource is useful:
    - Step 1: Identify hardest problems one can solve without the resource.
    - Step 2: Show presence of resource can help solve even harder problems.
- Classical resources:
  - CPU speed, Memory, Non-determinism, Randomness …
- In our case:
  - Communication in presence of understanding.
  - Communication w/o understanding.

Computation as a goal [Juba & S. '08]

- Model: Simple user talking to powerful server.
  - Class of problems user can solve on its own:

    ~ probabilistic polynomial time (P).
  - Class of problems user can solve with perfect understanding of server:

    ~ Any problem. (Even uncomputable!)
  - Class of problems user can solve without understanding of server:

    ~ Polynomial space.
- Roughly: If you are solving problems and can verify solutions, then this helps. If you have a solution, you are done. If not, you've found some error in communication.
- Moral: Communication helps, even with misunderstanding, but misunderstanding introduces limits.

Summarizing results of [GJS 2012]

- But not all goals are computational.
  - We use communication mostly for (remote) control.
  - Intellectual/informational goals are rare(r).
- Modelling general goals, in the presence of misunderstanding:
  - Non-trivial, but can be done.
  - Results extend those from computational setting:
    - Goals can be achieved if user can sense progress towards goal, servers are "forgiving" and "helpful"

Useful lessons

- User/Server can be designed separately.
- Each should attempt to model its "uncertainty" about the other.
- Each should plan for uncertainty:
  - Server: By assuming some short "interrupt" sequence.
  - User: By always checking its progress.

Future goals

- Broadly:
  - Information-theoretic study of human communication, with uncertainty as an ingredient.
    - Should exploit natural restrictions of humans:
      - Limited ability to learn/infer/decode.
      - Limited bandwidth.
  - Conversely, use human interactions to create alternate paradigms for “designed communications.
    - Place semantics on solid foundations.

Future?

- Understand human communication?
  - How does it evolve
  - What are influencing factors?
    - (My guesses): Compression, Computation, Survival of fittest.

- Extend to other “distributed design” settings.

- Architecture/Program for preserving Data?
  - Blend safe assumptions, with “likely-to-be-fast” performance.

Thank You!