Protocol

Algo selects encoder

Adv selects channel (this is not known to algorithm)

Alg receives one bit information if the packet was decoded or not.

Assumption on the adversary:

Type 1 (Utsav)

- Stochastic Adversary selects channel i with prob gamma_i

Type 2: (Subhash)

- Adversary has a Markov chain over channels.
- P(channel(t+1) = j/Channel(t) = i) = P_ij

Assume we are given a **error tolerance** \theta (ex \theta = 0.3)
- Empirical error of the algorithm should be within \theta (with high prob)

For the moment, assume we have type-1 adversary and we know \gamma

E_ij = Prob of error of using encoder i on channel j (given)

Expected error of Encoder i = \sum_j E_ij*\gamma_j = J_i
R_i = Rate of using encoder i (k_i/n)

Goal: What is the distribution \lambda Algo should use over the encoders?

max_{\lambda is in simplex} \sum_i lambda_i * R_i (average rate)

s.t \sum_i J_i*\lambda_i <= tolerance

[linear program]

Goal:

We want to come up with algorithms such that

"If the algortihm is run for T(epsilon, delta) rounds, then with high probability (>= 1 – delta),
(1) empirical error <= tolerance,
(2) empirical_rate <= best_possible_rate - epsilon"

Algorithm

- start with \lambda(1,i) = 1/h for all encoders
- for t = 1,….. T
    - Play encoder e(t) by sampling encoder_i with prob lambda(t,i)
    - Adversary picks c(t) = i w.p \gamma_i
    - receive b(t) \in {0,1} with Bernoulli with prob E_{e(t),c(t)}
    - **Get estimate for gamma using Maximum Likelihood - gamma(t)**
    - Get lambda(t+1) by solving the linear program using gamma(t)

  end

- Get estimate for gamma using Maximum Likelihood - gamma(t)

Treat observations from encoder i as from Ber(J_i)

$J_i = \sum_j E_{ij}*\gamma_j$

$L(\text{data}; \text{gamma}) = \prod_{i=1}^{h} \quad (J_i)^{(N(1,i))} (1-J_i)^{N(0,i)}$

max L(data; gamma) such that gamma is in simplex.

Algorithm: Projected Gradient Descent.

Assume reasonable values for h, E, γ

Plots:

- Empirical error as a function of t
- Empirical rate as a function of t