Setting up the Learning Problem

- **Features:**
  - Current hotel and flight prices
  - Current time in game
  - Hotel closing times
  - Agents in the game (when known)
  - Variations of the above

- Built **8 separate predictors**:
  - Good Hotel vs. Bad (2)
  - Day 2,3 vs. day 1,4 — exploit symmetry (2)
  - 1st min. (pre-hotel-quotes) vs. later (2)

- **Data**: seeding round games
  - Features $\mapsto$ actual clear prices

The Learning Algorithm

- X = feature vector
- Given X, estimate distribution of $Y = ($ hotel closing price $) - ($ current bid $)$
- Break Y into $k \approx 50$ cut points $b_1 \leq \cdots \leq b_k$
- For each b_i, estimate probability $Y \geq b_i$, given X
  - Can convert to estimated distribution of $Y|X$
- Say X belongs to class C_i if $Y \geq b_i$
  - k-class problem: each example in many classes
- Use **BoosTexter** (boosting "decision stumps")
  - Can convert output to class prob. estimates

Department of Computer Sciences
The University of Texas at Austin

The Learning Algorithm (cont.)

- Use logistic regression form of boosting
- Given X, get estimates p_j of $\Pr[Y \geq b_j]$
- So can estimate $\Pr[b_j \leq Y < b_{j+1}]$ by $Q(j) = p_j - p_{j+1}$
- Gives distribution Q over bins $[b_j, b_{j+1})$
- To sample:
  - Choose one bin at random according to Q
  - Choose point within bin uniformly at random
- Piecewise constant density function

Department of Computer Sciences
The University of Texas at Austin