Online Learning to Rank with Features

**Setup**
- Many items to be ranked
  - Eg. Movies, news articles
  - Online setting, limited modeling assumptions
  - L items, K positions, L \( \gg \) K
- Each round the learner chooses an ordering of the items.
- Observes feedback in the form of "clicks"

**Modeling Assumptions**
- Hopeless to learn the value of all rankings
- Assume items are associated w. features
- \( P(C_i = 1|A_i) = \alpha(A_i) \psi_c(A_i) \)
- \( \alpha(a) = \langle a, \theta \rangle \) Atractiveness
- \( \psi: A \rightarrow [0,1] \) satisfies
  - \( \psi_c (A) = \psi_c (A') \) when \( A(E\neg i) = A'(E\neg i) \)
  - \( \psi_{i+1}(A) \leq \psi_i(A) \) Exam. prob. is decrease
  - \( \psi_i(A) \geq \psi_i(A') \) when \( A^{*} \) orders items by attractiveness.
- Assumptions are satisfied by most standard click models
- Exam. prob. is smallest for \( A' \)

**Optimal Design**
- let \( A \in R^d \)
- Choose \( X_1, \ldots, X_N \in A \)
- Observe \( Y_1, \ldots, Y_N \) s. \( Y_i \sim \text{Normal}(\langle X_i, \theta \rangle, 1) \)
- LSE: \( \hat{\theta} = G^\top \hat{\beta} \), \( \hat{\beta} = \frac{1}{N} \sum_{i=1}^{N} X_i Y_i \)
- \( E[\hat{\theta}] = \theta \), \( E[\langle \theta - \theta, a^2 \rangle] = \Vert a \Vert^2 \)
- Should choose \( (X_i)^{\top} \) in prop.
  - \( \Pi^* = \arg \min_{a \in A} \frac{1}{N} \sum_{i=1}^{N} \langle X_i, a \rangle \)
  - Remaining items are sorted according to empirical estimate of attractiveness.
- Then \( E[\langle \theta - \theta, a^2 \rangle] = O(N) \)

**Theory**
- Regret measured w. respect to action that orders items from most
to least attractive
- \( R_n = \mathbb{E} \left[ \sum_{i=1}^{n} V_i(A^*) - V_i(A_i) \right] \)
- \( = O(KN \sqrt{n \log(N)}) \)

**Algorithm**
- Ranking = Sorting with noise
- Algorithm maintains blocks of incomparable items
- First position in each block is used for exploration
- Remaining items are sorted according to empirical estimate of attractiveness.
- Block is split if learner identifies a partition such that all items in one
  part are more attractive than the other with high probability.
- Summarized by RecurRank

**Previous Work**
- Either do not use features or,
- Depend on strong click model assumptions
- TopRank \( R_n = O(K^{3/2} \sqrt{n \log n}) \)
- New algorithm improves on previous best for "generic" model, even in
tabular case.

**Summary**
- Ranking with guarantees
- Minimal assumptions
- Efficient Algorithm
- Theory improves on prior work with few assumptions
- Limitation: Fixed features