CIS 700/4: Machine Learning and Econometrics **January 24, 2017**

Lecture 3: Rankings from Pairwise Comparisons

Professor Shivani Agarwal *Zach Schutzman*

NB: *These notes are from CIS700 at Penn.*

Rankings

Last time: we talked about permutation data and some extension to partially-ranked data. Today, we'll talk about rankings from pairwise comparisons.

Example: We have chess matches, which are pairwise comparisons with some noise. We have a budget to allow for 3 more games. Which games should be played to gain maximal information?

These things show up a lot, in terms of performance, preference, and outcomes. There are a lot of different approaches for dealing with them, from a wide range of disciplines.

Consider a set of n items, and we note $i \succ j$ if i is preferred to j, pairwise.

What might we want?

> A learning algorithm for a pairwise probability distribution.
>
> Aggregation for a consensus ranking
>
> Voting rule to choose a favorite

Pairwise comparison estimates are the fraction of times that i beats j over the number of time the two are compared. $\hat{P}_{ij} = \frac{W_{ij}}{N_{ij}}$. Assign $0 = \frac{0}{0}$.

Copeland Pairwise Voting Rule assigns score $C(i) = \sum\limits_{i \neq j} 1(\hat{P}_{ij} > \frac{1}{2})$

This looks like the Kemeny ranking and is also NP-Hard in general

Other rankings include spectral ranking, least squares, statistical learning methods, etc.

Spectral ranking looks kind of like PageRank. Construct transition matrix $\hat{Q}_{ij} = \frac{\hat{P}_{ij}}{\sum\limits_{k=1}^{n} \hat{P}_{ki}}$.

This puts higher rank on the elements more likely to beat others. You can find a stationary distribution of $\hat{Q}$ like a Markov chain, then sorting the outcomes.

Least Squares ranking finds scores $\hat{s}_i$ satisfying $\hat{s} \in \arg\min \sum (s_i - s_j - \ln \frac{\hat{P}_{ij}}{\hat{P}_{ji}})$.

Again, sort and output.

Probabilistic Models

A probabilistic model is a preference matrix $P \in [0,1]^{n\times n}$

The Bradley-Terry-Luce Model is a set of parameters $w \in \mathbb{R}^n$ where $P_{ij} = \frac{w_i}{w_i+w_j}$. These parameters can be found by a MLE procedure using the log-likelihood.

The Thurstone Model is a set of parameters $s \in \mathbb{R}^n$ such that $P_{ij} = \Phi(s_i - s_j)$, where Φ is the CDF of the standard normal distribution. This implies that $P_{ij} > \frac{1}{2} \iff s_i > s_j$. This is the method used in the Elo system.

Parameters can be estimated with a least-squares method, where we find $\hat{s} \in \arg\min \sum((s_i - s_j) - \Phi^{-1}(\hat{P}_{ij}))^2$

Score-based Models are parameters generated by a transformation function such that P_{ij} increases with how strongly i beats j. The BTL model is a special class of score-based models.

Random Utility Models use random variables $X_1 \dots X_n$, and we set $P_{ij} = \Pr(X_i > X_j)$.

If we say the X_i are Gumbel random variables, then we get the logistic case of the score-based models.

If we say the X_i are normal random variables, then $P_{ij} = \Phi(\frac{\mu_i - \mu_j}{\sqrt{2}})$

Properties in comparison models

Does the model belong to a particular family?

Does the model satisfy some conditions:

Noisy Permutation Property: there exists some $\sigma \in S_n$ and a $p < .5$, then P_{ij} is $1-p$ if $\sigma(i) < \sigma(j)$ and p if $\sigma(i) > \sigma(j)$. That is, if there is a probability p of seeing the wrong outcome, whenever we compare two items, we see the correct outcome with probability $1-p$.

Stochastic Transitivity Property: If i beats j with probability greater than .5 and j beats k with probability greater than .5, then i beats k with probability greater than .5. Equivalently, there is a complete ordering on the outcomes.

Condorcet Winner Property: the model admits a Condorcet winner, i.e. $\exists i^* \in [n]$ such that $P_{i^*j} > \frac{1}{2}\ \forall i^* \neq j$

Binary Choice Probability Property: there is a distribution p on permutations such that $P_{ij} = P_{\sigma\sim p}(\sigma(i) < \sigma(j))\ \forall i \neq j$. That is, the model can be described by choosing comparisons randomly from some distribution.

Sorting

Standard sorting requires/assumes outcomes consistent with a total ordering. We can sort with $n\log(n)$ comparisons if we can choose our comparisons. We can model this as an algorithm choosing a pair and receiving an outcome from an oracle. A noisy version of this assumes the oracle gives the wrong answer sometimes.