Probability and Asset Updating Using Bayesian Networks for Combinatorial Prediction Markets
Wei Sun†, Robin Hanson‡, Kathryn B. Laskey§, Charles Twardy†
†C4I Center, ‡Economics Dept., §SEOR Dept., George Mason University

Abstract
A market-maker-based prediction market lets forecasters aggregate information by editing a consensus probability distribution either directly or by trading securities that pay off contingent on an event of interest. Combinatorial prediction markets allow trading on any event that can be specified as a combination of a base set of events. However, explicitly representing the full joint distribution is infeasible for markets with more than a few base events. A factored representation such as a Bayesian network (BN) can achieve tractable computation for problems with many related variables. Standard BN inference algorithms, such as the junction tree algorithm, can be used to update a representation of the entire joint distribution given a change to any local conditional probability. However, in order to let traders reuse assets from prior trades while never allowing assets to become negative, a BN based prediction market also needs to update a representation of each user’s assets and find the structural state in which a user has minimum assets. Users also find it useful to see their expected assets given an edit outcome. We show how to generalize the junction tree algorithm to perform all these computations.

Innovations and Contributions
1. First published implementation of BN-based combinatorial prediction markets.
2. Factored representation of a trader’s assets using the junction tree structure that enables significant space and computational efficiency.
3. Applying non-propagation over the assets junction tree to find the guaranteed minimum payoff so as to allow traders to reuse assets for many more trades, i.e., enables more efficient information aggregation.
4. Efficiently finding the largest amount by which the edit could raise or lower the probability of an event of interest, defined as the edit limit, beyond which the edit may result in the trader holding negative assets.
5. Efficient computations of conditional or unconditional expected assets to help a user to know her “long” or “short” position.

Bayesian Networks

Combination of Predictions

Probability decomposition via the junction tree structure

\[
p(X) = \prod_{i \in C} p_i(X_i) \prod_{j \in S} p_j(X_j) \tag{1}\]

Asset Decomposition & Efficient Management

\[
\Delta S_x = b \ln \frac{p_X}{p'_X} \\
S_x = b \ln \left( \frac{p_X}{p'_X} \right) \\
q(X) = \prod_{i \in C} q_i(X_i) \\
q'_X(X_i) = q_i(X_i) \frac{p'_X}{p_X} \tag{2}\]

Reusing Assets

\[
P(T = t | A = a) \approx \frac{1 - P(T = t | A = a)}{m} \tag{3}\]

Probability and Asset Updating Algorithm

Algorithm 1 Update probability and user assets for a BN representing a combinatorial prediction market.

Require: a BN model \( B \) over a set of domain variables \( X \) that represents the combinatorial prediction market joint distribution, the clique tree \( T \) corresponding to \( B \), consisting of cliques \( C \) and separators \( S \).

Require: The current market probability distribution \( p \) represented by a probability junction tree.

Require: The current assets \( a \) represented by an assets junction tree.

for each conditional edit \( p' \) on the target variable \( T = t \) with assumptions \( A = a \) by user \( u \) do

- Tell the user the expected scores \( S(T = t, A = a) \) and \( S(T \neq t, A = a) \) indicating the user’s long/short status.
- Calculate the edit limits for \( p(T = t | A = a) \) using Equation (3).
- Allow user to trade \( p(T = t | A = a) \) within the edit limits. And apply \( p(T = t | A = a) \) as soft evidence to the junction tree.
- Update probability distributions of cliques and separators to be \( p'_c, p'_s, c \in C, s \in S \) by calling the junction tree inference algorithm.
- Find the clique \( c \) containing target variable \( T \) and assumed variables \( A \), and update the assets clique corresponding to \( c \) using Equation (2e).
end for

return User’s expected assets after the edit; user’s min-\( a \) value and its associated min-\( a \) states.

Numerical Examples & Scalability

Experimental setup:
1. Scalability – randomly generated networks with varied number of variables from 30 to 960, and the treewidth from 5 to 20.
2. Potential reject rate under different market environments, modeled by frequency of edits – 100 traders, random questions, random assumed conditions, random probability edits.

Table 1: ALARM Simulation Results for 1,000 Edits

<table>
<thead>
<tr>
<th>Market Intensity</th>
<th>Average Rejects</th>
<th>Average Rejection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 edits/minute</td>
<td>11.3</td>
<td>1.2%</td>
</tr>
<tr>
<td>8 edits/minute</td>
<td>39.2</td>
<td>14.3%</td>
</tr>
<tr>
<td>30 edits/minute</td>
<td>142.6</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

dagre.org

Supported by the Defense Advanced Research Projects Agency (DARPA) via Department of Defense contracts W31P4Q-07-2-0060 and contract number D11PC20062. The U.S. Government is authorized to reproduce and distribute reproductions of Governmental purposes notwithstanding any copyright annotation thereon. Disclaimers: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA, the Army, or the U.S. Government.