Center of Excellence in C4I
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://c4i.gmu.edu/ 703-993-5538

Dynamic Asset Cluster Model, a Trade-Based Asset Model using Junction Tree Structure, with interval questions, for Combinatorial Prediction Markets

Wei Sun
wsun@c4i.gmu.edu

Technical Report (March, 2014)

Abstract

CONFIDENTIAL: This is the internal documentation of implementing DAC (Dynamic Asset Cluster) model with interval questions, for combinatorial prediction markets. The implementation is coded in Matlab.

1 Prerequisites

In our context, basically we partition the continuous values state space of the variable into dynamically changing set of intervals, so we call an interval question value-partitioning node (VP-node). It is not real continuous variables in our current application yet. We assume that there are a set of atomic units as the lowest level elements (sometimes called leaf in hierarchy) for any VP-node. When we partition values for a VP-node, we mean to group corresponding atomic units into an interval or range.

In general, a trade involving a VP-node has to define its reference range (*rr*) and target range (*tr*). By default, *rr* is the entire value set of the VP-node.

We maintain a data structure for representing a VP-node, in which we have attributes called 'min', 'max' for representing its whole value set, 'cutpoints' to save all cutting points in between 'min' and 'max' but not including them. 'cutpoints' needs to be updated after any interval trades. Consequently, number of states for the VP-node, a.k.a. node size, can be derived by counting cutting points + 1.

We maintain a *DAC* asset structure for any user who has at least one trade. *DAC* has following attributes: (1)*base*–base asset, which is initial asset + manna + realized gain/loss; (2)*Obj_list*–Object list; (3)*Q_list*–question list; (4)*ns*–node sizes; (5)*abc*–asset blocks collection.

Whenever there is a trade involving a VP-node, we need to update *DAC* model by updating all asset blocks containing the corresponding VP-node. Arguments needed for the function are:

1. current *DAC*;
2. involving VP-node;
3. asset change *delta*, which is always a vector of two values;
4. *tr*, target range;
5. *rr*, reference range (optional);
6. regular assumption questions (optional);

7. given states of regular assumption questions (optional);
8. node sizes of regular assumption questions (optional);
9. conditioning VP-node (optional);
10. given range of conditioning VP-node (optional).

2 Test Cases & Implementations

2.1 Case 1: a trade on the target range only for a VP-node $p(tr)$

Suppose a VP-node called *RainVP* has its atomic units represented by integers from 1 to 19 as its entire value set. *Rain* has uniform prior $p0$. The trade is, without specifying any reference range (by default, rr is the entire value set), to decrease the current probability over the range of $[3, 10]$ from about 0.421053 by 15%, namely, to lower down to about 0.357895. We then proportionally assign updated probabilities to individual atomic units. Particularly, updated probability distribution for *RainVP* after the trade is $p1 = [0.0584, 0.0584, 0.0447, 0.0447, 0.0447, 0.0447, 0.0447, 0.0447, 0.0447, 0.0447, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584]$. Accordingly, *RainVP* has cutpoints $\{3, 10\}$ partitioning its state set into 3 bins, namely, $[1, 3]$, $[3, 10]$, and $[10, 19]$. Correspondingly, asset changes in those bins are shown in Figure 1.

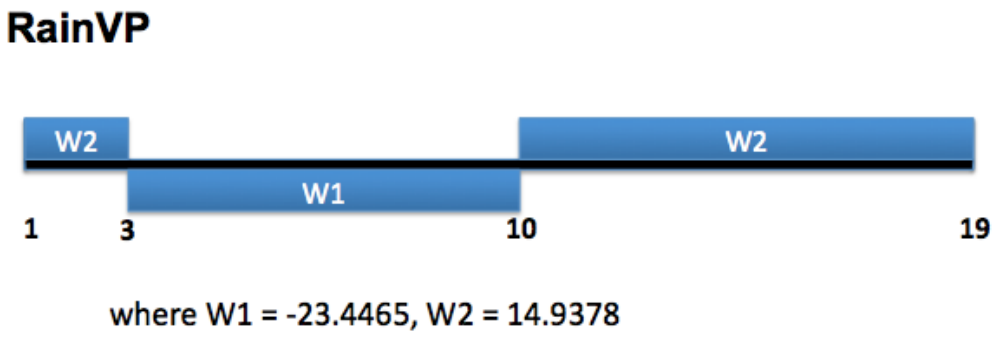

Figure 1: Asset changes distribution on *RainVP*

2.2 Case 2: a trade on the target range given a reference range for a VP-node $p(tr|rr)$

We conduct the second trade on *RainVP* again. This time we specify a reference range as $[5, 13]$, then trade on the target range of $[8, 9]$. Specifically, the trade is to raise up by 25% on the base of the current probability of the interval of $[8, 9]$ given $[5, 13]$. After the trade, the updated probability distribution for *RainVP* becomes $p2 = [0.0584, 0.0584, 0.0447, 0.0447, 0.0419, 0.0419, 0.0419, 0.0559, 0.0559, 0.0419, 0.0547, 0.0547, 0.0547, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584, 0.0584]$. And *RainVP* has new partition sets as $[1, 3]$, $[3, 5]$, $[5, 8]$, $[8, 9]$, $[9, 10]$, $[10, 13]$, and $[13, 19]$. The new trade resulted in new asset changes over new generated bins of $[5, 8]$, $[8, 9]$, $[9, 13]$, illustrated in Figure 2 as $W3$, $W4$:

Please note the old asset gain/loss generated by the first trade have not been combined yet. After aggregation, we have new set of bins and corresponding asset changes shown in Figure 3:

2.3 Case 3: trade on the target range of a VP-node given its reference range and another regular assumption question $p(tr|rr, A)$

Now let consider the case when trading on the target range of a VP-node given its reference range and another regular assumption question $p(tr|rr, A)$.

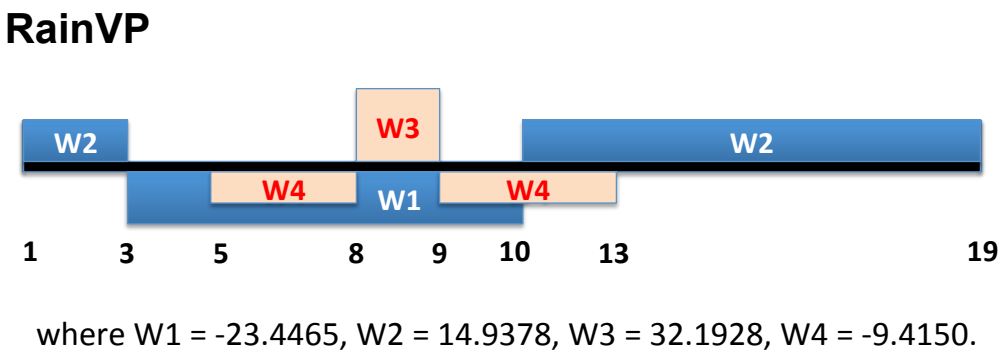

Figure 2: Asset changes over newly generated bins of *RainVP*

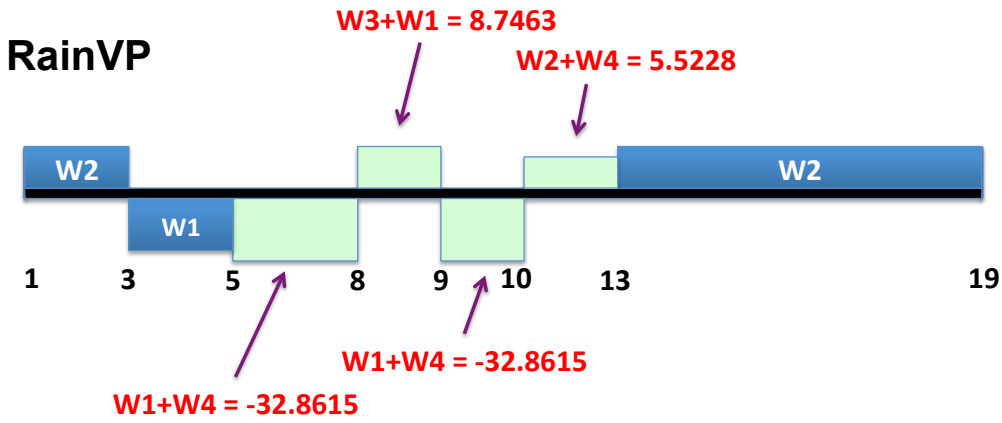

Figure 3: New asset distribution of *RainVP*

2.4 Case 4: $p(tr|rr, VP\_A)$

I made a Netica BN model consisting of one VP-node called *sand_VP*(integers from 1 to 10 as its atomic units), and a regular two-state question A as one of parent of *sand_VP*, and a VP assumption question, called VP_A (integers from 1 to 4 as its atomic units), as another parent of *sand_VP*.

Table 1: Customized min-propagation protocol for asset junction tree

1. Let $\phi(\mathbf{S}_{ij})' = \min_{\mathbf{C}_i \setminus \mathbf{S}_{ij}} \phi(\mathbf{C}_i)$, — minimalizng the potential $\phi(\mathbf{C}_i)$ onto the domain of separator $\phi(\mathbf{S}_{ij})$, i.e., finding the minimum values over the dimensions of minimalized variables and then projecting them to the domain of separator.
2. Let $\mathcal{L}(\mathbf{S}_{ij}) = \phi(\mathbf{S}_{ij})' - \phi(\mathbf{S}_{ij})$, — subtracting the old asset in the separator $\mathbf{S}_{ij}$ from its new values. The resulting value is called the separator gain, because it filters out the overlapping information between the sending clique and the separator.
3. Let $\phi(\mathbf{C}_j) = \phi(\mathbf{C}_j) + \mathcal{L}(\mathbf{S}_{ij})$, — summing up with the separator gain to update the asset of $\phi(\mathbf{C}_j)$.

The rest of paper from this page are just the backup for LaTeX formulas. Please ignore contents from this page.

$$\begin{aligned} p'(D,E,F) &= \frac{p'(D,E) \times p'(D,F)}{p'(D)} \\ &= p'(D,E) \times p'(F|D) \\ &= p'(D,E) \times p(F|D) \\ &= p'(D,E) \times \frac{p(D,F)}{p(D)} \\ &= \frac{p'(D,E) \times p(D,F)}{p(D)} \end{aligned}$$

$$\left[p(T=t|\mathbf{A}=\mathbf{a}) exp(\frac{-m_t}{b}), 1-(1-p(T=t|\mathbf{A}=\mathbf{a}))exp(\frac{-m_{\neg t}}{b}) \right]. \tag{1}$$

Asset update - Once we have defined the joint asset over disconnected networks as shown above, we can do the following after making edit on probabilities:

$$\begin{aligned} S'(\mathcal{A},\mathcal{B},\mathcal{C}) &= S(\mathcal{A},\mathcal{B},\mathcal{C}) + b * log(\frac{p'(\mathcal{A}) * p'(\mathcal{B}) * p'(\mathcal{C})}{p(\mathcal{A}) * p(\mathcal{B}) * p(\mathcal{C})}) \\ &= S(\mathcal{A}) + S(\mathcal{B}) + S(\mathcal{C}) - S(\mathcal{A}\mathcal{B}_0) - S(\mathcal{B}\mathcal{C}_0) + b * log(\frac{p'(\mathcal{A})}{p(\mathcal{A})} + b * log(\frac{p'(\mathcal{B})}{p(\mathcal{B})} + b * log(\frac{p'(\mathcal{C})}{p(\mathcal{C})} \\ &= S'(\mathcal{A}) + S'(\mathcal{B}) + S'(\mathcal{C}) - S(\mathcal{A}\mathcal{B}_0) - S(\mathcal{B}\mathcal{C}_0) \end{aligned} \tag{2}$$

Soft Evidential Update: Jeffurey rule. Suppose we have a joint distribution over four variables T,W,X, and Y. Suppose we have current joint probability over these four variables as $P(T,W,X,Y)$. No matter what edit the user likes to do, all of other conditional probabilities should remain same. Essentially, the edit is just a partial change over the original probability space. Let us have an example to show how to do soft evidential update over the joint space. Assuming we have an edit on T given X, $Q(T|X)$, then the new joint probability will be

$$\begin{aligned} P'(T,W,X,Y) &= P'(W|T,X,Y) * P'(Y|T,X) * P'(T|X) * P'(X) \\ &= P(W|T,X,Y) * P(Y|T,X) * Q(T|X) * P(X) \\ &= P(T,W,X,Y) * \frac{Q(T|X)}{P(T|X)} \end{aligned}$$

The key is that we can shuffle the order of variables in the chain rule to always make the one being edited separated out as one term, and all of other terms remain same. And of course, we can always calculate the curren probability of the coming edit from the joint, which is $P(T|X)$ in the example above.

$$v(x) = \sum_{i=1}^{N} (K(x_i))/N \tag{3}$$