Risk Aware Agent for Bidding in Second-Price Sealed-Bid Auctions and variants

This technical report introduces two forms of a risk aware agent for bidding in Simultaneous Second-Price Sealed-Bid Auctions but who's heuristics are general enough to be used in other auction settings. The main idea of the risk aware agent is the computation of a utility score that given predictions of prices over bundles of goods, considers the risk associated with the distribution of prices over each good contained in a given bundle. This utility is a generalization of the classic acquisition problem described in (Boyan & Greenwald, 2001) and used by every agent in (Yoon & Wellman, 2011) .

As a baseline, all agents from (Yoon & Wellman, 2011) were implemented to test the efficacy of the proposed risk-aware agent as well as parallel implementations of both point price predicting and distribution predicting algorithms in order to maximize the number of auction simulations executed in minimal time. The implementation language is Python and all code and documentation can be found in the github repository:

git://github.com/bam593/bmProjects.git

The contents can be viewed with a web browser at the address:

https://github.com/bam593/bmProjects/tree/master/courses/fall2011/csci2951

Motivation

Every strategy profile described in (Yoon & Wellman, 2011) assumes the bidding process can be decomposed into two independent modules:

1) Identify a set of goods that if won would maximize the agent's surplus, defined as valuation less cost (this step is known as the acquisition problem).
2) Compute bids to place on the goods in the bundle that solves that the acquisition problem.

That is we first define a bundle as a collection of goods that we can potentially procure at auction.

If there are m goods available, we define the set of all possible bundles, X, available for purchase as

$$X = \{x : x \in \mathbb{P}^m\}$$

where $\mathbb{P}^m$ represents the power set of m items. For example, lets' assume $m = 4$, we can enumerate every possible bundle using bit vectors of length 4, representing the set of all bundles as:

$$X = \{[0,0,0,0], [0,0,0,1], \dots, [1,1,1,0], [1,1,1,1]\}$$

If the j^{th} entry is 1 in the i^{th} bundle bit vector, this indicates the j^{th} good is purchased in bundle i, otherwise the j^{th} good is not included in the bundle.

The agents' valuation function is then a function that assigns a scalar to each bundle in the powerset and is known to each agent apriori

$$v : \mathbb{P}^m \to \mathbb{R}.$$

The function used in this report are described in (Sodomka & Greenwald).

Given a vector of prices representing the price paid for each good in a bundle, $p \in \mathbb{R}^m$ we can calculate the cost , c, associated with each bundle as the dot product of the i^{th} bundle and the price vector p and thus compute the surplus σ for each bundle.

$$c = x_i \cdot p$$

$$\begin{aligned}\sigma_i(X, p) &= v - c \\ &= v - x_i \cdot p\end{aligned}$$

Therefore the acquisition problem of step (1) is a matter of solving (Boyan & Greenwald, 2001):

$$X^* = ACQ_i(p) = \arg\max_{X \subseteq \mathcal{X}} \sigma_i(X, p)$$

The non-additive valuation function described in (Sodomka & Greenwald) and (Yoon & Wellman, 2011) is used to model complementary goods. Goods A and B are complementary if:

$$v(A\ U\ B) > v(A) + v(B)$$

That is if we are able to purchase only part of the bundle that solves the acquisition problem, the valuation of the bundle may be zero exposing the agent to a possible negative surplus as he will still be obligated to pay for the items he wins. Therefore, given X^*, (Yoon & Wellman, 2011) describe different strategy profiles that bid differently on this bundle, using techniques based on marginal value calculations shade the bids up to make sure the agent obtains the necessary goods in the target bundle. They also describe a profile bidEvaluator, that uses multiple base strategy profiles to generate candidate bids then evaluate the confidence in each bid against a price distribution and choose the bid by evaluating the candidate bids against the probability distribution over prices of goods.

There are two versions of the risk aware strategy profile, riskAware1 and riskAware2.

<u>riskAware1</u>

riskAware1 attempts to solve a more general form of the acquisition problem, thus modifying the first step above. Given this alternative target bundle, we can use any strategy described in (Yoon & Wellman, 2011) to produce a bid.

From economics, mean-variance utility function for returns on investments is defined as (Bodi, Kane, & Marcus, 2010):

$$U = \mathbb{E}_f[r] - \frac{1}{2} * A * var[r]$$

The free parameter A defines how risk-adverse a particular investor is, $\mathbb{E}[r]$ defines the expected rate of return of an investment portfolio given a probability distribution f and $var[r]$ defines the variance associated with the rate of return of the investment and serves as a measure of risk.

Though this function is used in the context of identifying optimal investments for an investor given the investors' level of risk aversion, we can modify this function to conform to our problem.

Define:

$$U = \mathbb{E}_f[\sigma_i] - \frac{1}{2} A \, var[\sigma_i]$$

Using the following property of variance:

$$var[\sigma_i] = var\left[v_i + \sum_{j=1}^{m} x_{ij} * p_j \right]$$
$$= var\left[\sum_{j=1}^{m} x_{ij} * p_j \right]$$

we write the mean-variance utility as:

$$U = \mathbb{E}_f[\sigma_i] - \frac{1}{2} A \, var\left[\sum_{j=1}^{m} x_{ij} \, p_j \right]$$

Let us assume as in (Yoon & Wellman, 2011) that prices are independent over goods j,

$$U = \mathbb{E}_f[\sigma_i] - \frac{1}{2} A \sum_{j=1}^{m} x_{ij}^2 var[p_j]$$
$$= \mathbb{E}_f[\sigma_i] - \frac{1}{2} A \sum_{j=1}^{m} \delta(j \in X_i) var[p_j]$$

Given the function v and a distribution over prices, we easily calculate $\mathbb{E}_f[\sigma_i]$ and $var[p_j]$. The intuition behind this heuristic is that we still want to pick the bundle that maximizes the expected surplus for the agent, however, there may be "safer" bundles, quantifying the risk associate with each bundle by the sum of the variance of the individual goods in the bundle.

We can adapt the variance term to better quantify risks present in auctions. Given an arbitrary distribution over the price of a good, we can see that the risk posed to the agent are the realization of closing prices that are greater than what the agent expects.

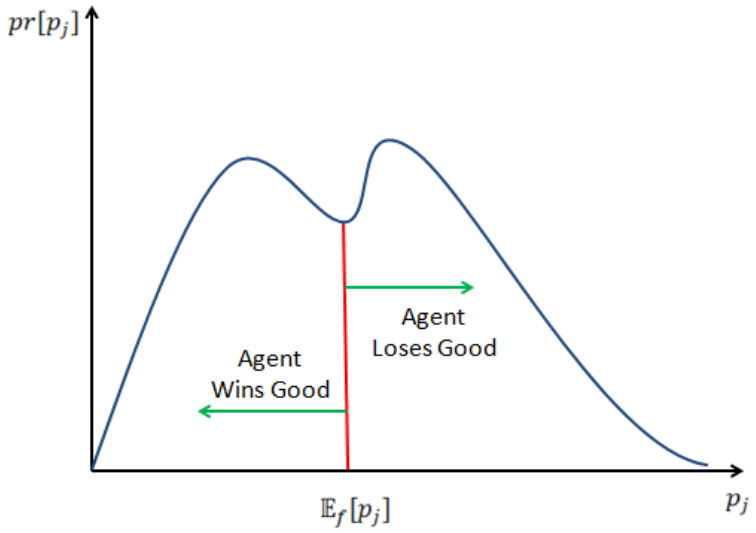


Figure 1: Agent Risk

For an auction with one item for sale, the agent risks losing the item to another agent if the closing price of the good is greater than his/her bid. Therefore, quantifying risk associated with the expected surplus of a good by estimating the variances of closing prices overestimates the risk associated with each good. Idealy an agent would only consider the prices that are above the expected value of a good's closing price when computing a measure of risk.

We can do this by calculating an upper-partial variance (Bodi, Kane, & Marcus, 2010):

$$\widehat{UPV_j} = \sum_{t=1}^{T} \delta(p_{jt} > \mathbb{E}_f[p_j])(\mathbb{E}_f[p_j] - p_{jt})^2$$

This estimator is assuming a discrete (histogram) distribution over prices with bins indexed by t. We add a price only to the variance calculation if the realized price is greater than the expected value.

Thus the final form of our utility function is:

$$U = \mathbb{E}_f[\sigma_i] - \frac{1}{2}A\sum_{j=1}^{m} \delta(j \in X_i)\widehat{UPV_j}$$

Where

$$\mathbb{E}_f[\sigma_i] = v_i - X_i\mathbb{E}_f[p]$$

We can then define the acquisition problem as:

$$X^* = \arg\max_{X \subseteq \mathcal{X}} U(X, p)$$

Given the optimal bundle X^* we can then calculate a bid based on any strategy outlined in (Yoon & Wellman, 2011), e.g. bidding target prices or marginal values etc.

Note that A is a free parameter of our system and can be learned or chosen experimentally. We can provide some intuitive explanations for different values of A:

- $A > 0$
  - Agent is risk-adverse
  - The agent's utility is reduced proportionally to the quantified risk
- $A < 0$
  - Agent is a risk-lover or gambler
  - The agent gains utility not only through increased expected surplus but by the action of taking more risky positions
- $A = 0$
  - The agent is risk-neutral
  - The agent does not factor risk into the overall utility of a bundle
  - This reduces to the classical acquisition formulation and shows that the proposed utility is a more general form of the original acquisition problem in (Boyan & Greenwald, 2001)

riskAware2

In riskAware1 we assumed partitioning the bidding process into two separate modules, one identifying an optimal bundle on which to bid and another defining what to bid given an optimal bundle, would give reasonable results.

It may be advantageous to consider the bidding problem as one global problem. If we are given a probability distribution over closing prices of goods, we can compute the probability of winning goods given a deterministic bid. We can then calculate the expected value, cost and surplus of bundles given a bid then discount the surplus by risk and optimize the resulting utility with respect to the bid.

Let us define a utility as in riskAware1 with some modifications. Let us consider the same mean upper-partial variance utility function as riskAware1 but change our perspective. Let us assume we are searching for an optimal bid instead of an optimal bundle. Given deterministic bids, can we maximize the utility or expected utility?

$$U(b) = \sigma\left(X^{final}\right) - \frac{1}{2}A\left(\widehat{UPV} \cdot X^{final}\right)$$

This equation is the same as before only we have explicitly made the utility a function of the deterministic bid and the random vector X^{final}, the bundle we would win if we bid b.

Taking the expectation of this utility:

$$\begin{aligned}
\mathbb{E}_f[U(b)] &= \mathbb{E}_f\left[\sigma\left(X^{final}\right) - \frac{1}{2}A\left(\widehat{UPV}\cdot X^{final}\right)\right] \\
\mathbb{E}_f[U(b)] &= \mathbb{E}_f\left[\sigma(X^{final})\right] - \frac{1}{2}A\,\mathbb{E}_f\left[\widehat{UPV}\cdot X^{final}\right] \\
&= \mathbb{E}_f\left[\sigma\left(X^{final}\right)\right] - \frac{1}{2}A\mathbb{E}_f\left[\sum_{i=1}^{m}\widehat{UPV}_i X_i^{final}\right] \\
&= \mathbb{E}_f\left[\sigma(X^{final})\right] - \frac{1}{2}A\sum_{i=1}^{m}\widehat{UPV}_i\mathbb{E}_f\left[X_i^{final}\right]
\end{aligned}$$

Where $\mathbb{E}_f\left[X_i^{final}\right]$ is the expected value of the i^{th} binary entry of X^{final}. Notice we have not achieved the reduction from bundle to individual goods through independence assumptions but arose naturally from the linearity of expectation and the definition of the dot product of two vectors.

Let b_i denote the bid for the i^{th} good and p_{ci} denote the closing price of the same item.

$$\begin{aligned}
\mathbb{E}_f\left[X_i^{final}\right] &= \Pr[b_i = p_{ci}]\cdot 1 + (1 - \Pr[b_i = p_{ci}])\cdot 0 \\
&= \Pr[b_i = p_{ci}]
\end{aligned}$$

So we have reduced our utility to:

$$\mathbb{E}_f[U(b)] = \mathbb{E}_f\left[\sigma\left(X^{final}\right)\right] - \frac{1}{2}A\sum_{i=1}^{m}\widehat{UPV}_i\Pr[b_i = p_{ci}]$$

and can now consider the expected surplus calculation.

$$\begin{aligned}
\mathbb{E}_f\left[\sigma\left(X^{final}\right)\right] &= \mathbb{E}_f\left[v\left(X^{final}\right) - c\left(X^{final}\right)\right] \\
&= \mathbb{E}_f\left[v\left(X^{final}\right)\right] - \mathbb{E}_f\left[c\left(X^{final}\right)\right]
\end{aligned}$$

Firstly considering expected cost:

$$\begin{aligned}
c\left(X^{final}\right) &= b\cdot X^{final} \\
\mathbb{E}_f\left[c\left(X^{final}\right)\right] &= \mathbb{E}_f\left[b\cdot X^{final}\right] \\
&= \mathbb{E}_f\left[\sum_{i=1}^{m} b_i X_i^{final}\right] \\
&= \sum_{i=1}^{m} b_i\mathbb{E}_f\left[X_i^{final}\right] \\
&= \sum_{i=1}^{m} b_i\Pr[b_i = p_{ci}]
\end{aligned}$$

And the harder problem of expected valuation:

$$v(X^{final}) = \sum_{i=1}^{\mathbb{P}^{\mathcal{X}}} v_i \delta(X_i = X^{final})$$

$$\mathbb{E}_f[v(X^{final})] = \sum_{i=1}^{\mathbb{P}^{\mathcal{X}}} v_i \mathbb{E}_f[\delta(X_i = X^{final})]$$

$$\mathbb{E}_f[v(X^{final})] = \sum_{i=1}^{\mathbb{P}^{\mathcal{X}}} v_i \Pr[X_i = X^{final}]$$

This last result is a property of the expectation of indicator function over sets in a probability space: http://en.wikipedia.org/wiki/Indicator_function

$\Pr[X_i = X^{final}]$ denotes the probability that a given bundle is the final bundle. The probability of winning a specific bundle is the probability of winning the bundle's constituents and losing all other goods. To facilitate a reasonable calculation, we can assume independence between distributions over closing prices with respect to individual goods.

Therefore, the probability of winning the goods contained in a bundle is the product of the probability of winning those individual elements. The probability of losing the goods not contained in a bundle is likewise the product of losing those non-member elements.

Let γ denote the set of all indices of goods in the i^{th} bundle and $\bar{\gamma}$ denote the indices of items not contained in the bundle. That is:

$$\gamma \coloneqq \{i \mid x_i \in X_i\}$$

$$\bar{\gamma} \coloneqq \{i \mid x_i \notin X_i\}$$

Then using our independence assumptions:

$$\Pr[X_i = X^{final}] = \left(\prod_{\gamma} \Pr[b_\gamma = p_{c\gamma}]\right) \cdot \left(\prod_{\bar{\gamma}} (1 - \Pr[b_{\bar{\gamma}} = p_{c\gamma}])\right)$$

Hence,

$$\mathbb{E}_f[v(X^{final})] = \sum_{i=1}^{\mathbb{P}^{\mathcal{X}}} v_i \left(\prod_{\gamma} \Pr[b_\gamma = p_{c\gamma}]\right) \cdot \left(\prod_{\bar{\gamma}} (1 - \Pr[b_{\bar{\gamma}} = p_{c\gamma}])\right)$$

and we can calculate the expected mean upper-partial utility function:

$$\mathbb{E}_f[U(b)] = \mathbb{E}_f\left[v\left(X^{final}\right)\right] - \mathbb{E}_f\left[c\left(X^{final}\right)\right] - \frac{1}{2}A\sum_{i=1}^{m}\widehat{UPV}_i \Pr[b_i = p_{ci}]$$
$$= \sum_{i=1}^{\mathbb{P}^{\mathcal{X}}} v_i \left(\prod_{\gamma}\Pr\left[b_\gamma = p_{c\gamma}\right]\right)\cdot\left(\prod_{\overline{\gamma}}\left(1 - \Pr\left[b_{\overline{\gamma}} = p_{c\gamma}\right]\right)\right) - \sum_{i=1}^{m} b_i \Pr[b_i = p_{ci}]$$
$$-\frac{1}{2}A\sum_{i=1}^{m}\widehat{UPV}_i \Pr[b_i = p_{ci}]$$

All quantities are known apriori and can be computed given a bid candidate b.

Intuitively, this utility describes the expected value we will gain from placing a bid less the cost we are expected to incur given the bid discounted by the risk of every good at auction in proportion to the probability of winning the item (incurring the risk). Again A is a free parameter of the utility gauging the importance (and direction via its sign) of risk in our bid consideration.

We can use any method to optimize the utility with respect to a bid vector p. We can sample from $f_{p_c}(b)$ and evaluate $\mathbb{E}_f[U(b)]$ or perform deterministic optimization (Levenberg-Marquardt, Gauss-Newton, Nelder-Mead, etc..) but despite the method, we are searching for the solution to:

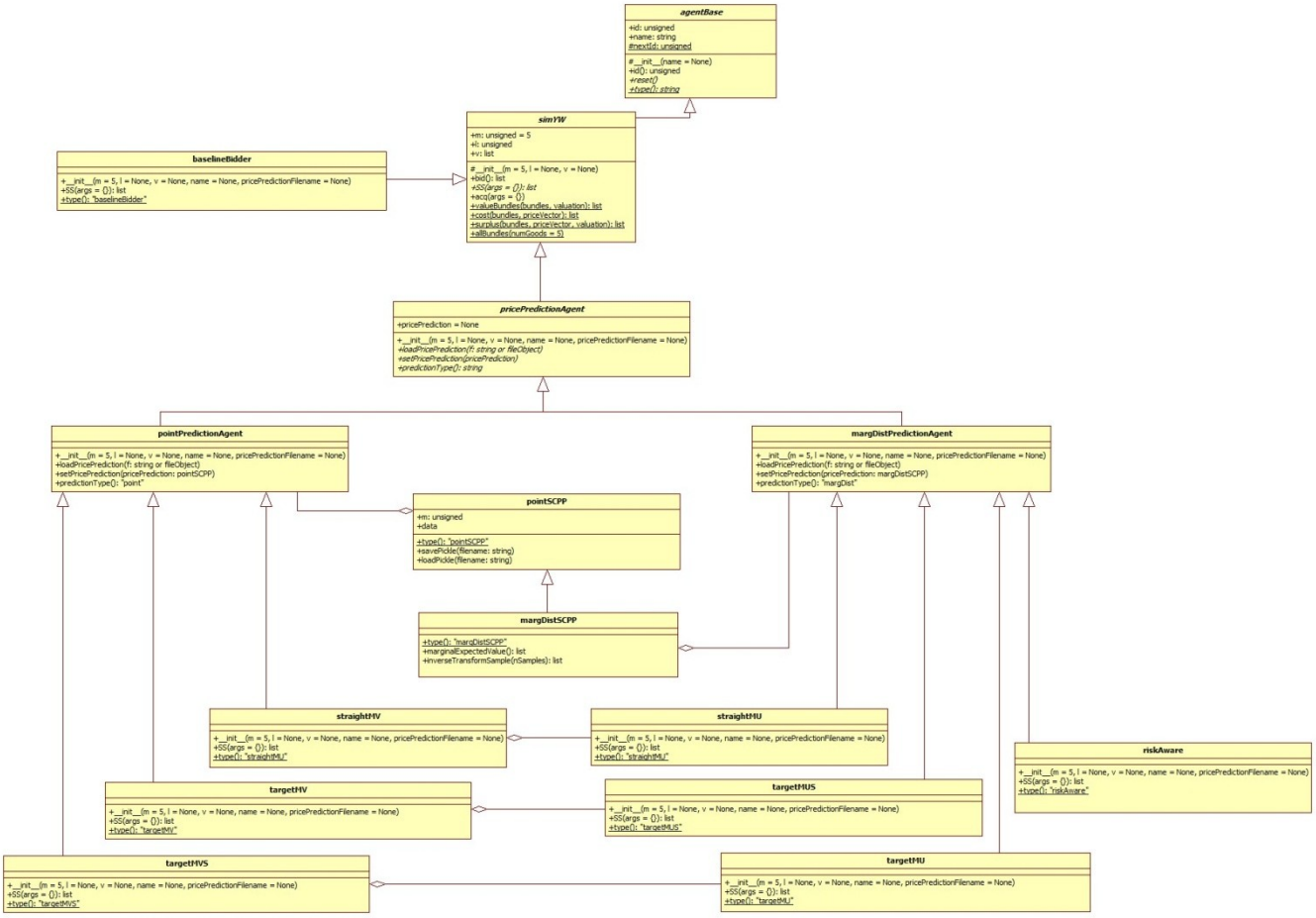
$$b^* = \arg\max_b\left\{\mathbb{E}_f[U(b)]\right\}$$
$$= \arg\max_b\left\{\sum_{i=1}^{\mathbb{P}^{\mathcal{X}}} v_i \left(\prod_{\gamma}\Pr\left[b_\gamma = p_{c\gamma}\right]\right)\cdot\left(\prod_{\overline{\gamma}}\left(1 - \Pr\left[b_{\overline{\gamma}} = p_{c\gamma}\right]\right)\right) - \sum_{i=1}^{m} b_i \Pr[b_i = p_{ci}]\right.$$
$$\left.-\frac{1}{2}A\sum_{i=1}^{m}\widehat{UPV}_i \Pr[b_i = p_{ci}]\right\}$$

Implementation

I relied heavily on an object oriented approach to keep the implementation light and extremely modular. Modularity was essential as the distribution based strategies described in (Yoon & Wellman, 2011) make use of the point distribution strategies and bidEvaluator uses all the strategies. I accomplished this by defining a static method SS on the base simYW class and which each strategy profile would overload. However, each agent maintains an inherited bid() function that will fill the parameters of the correct SS function and return the appropriate bid given the strategy profile and object status. Using static functions also eliminated the need to constantly instantiate worker objects of specific concrete strategy profiles. The following UML diagram outlines the relationship between classes. Note that the inheritance and aggregation relationships are somewhat ambiguous at this high level and in a dynamic language like python. For instance the aggregation relationship between margDistPredictionAgent and margDistSCPP indicates that margDistPredicitonAgent owns a pointer to an instance of margDistSCPP. However, the aggregation indicated between straightMU and straightMV only indicates that straightMU is aware of straightMV's existence as it only needs to call the static SS

function implemented by straightMV when necessary. Context needs to be considered when consulting the diagram.

The figure appears small in this report. For a larger version see the github repository.



I also implemented the price prediction simulators using python's multiprocessing module. Though I feel there are more efficient was to implement these algorithms in parallel, most required making operating system specific choices. Creating separate processes to run individual games yielded the most platform independent solution.